IEEE 802.11p in Fast Fading Scenarios: From Traces to Comparative Studies of Receive Algorithms

Practice Paper

Bastian Bloessl*†, Mario Gerla† and Falko Dressler*
* Heinz Nixdorf Institute and Dept. of Computer Science, Paderborn University, Germany
† Dept. of Computer Science, University of California, Los Angeles, CA
{bloessl,dressler}@ccs-labs.org, gerla@cs.ucla.edu

ABSTRACT

We present an approach for using signal traces from Field Operational Tests (FOTs) for later evaluations and comparative studies of receive algorithms for the IEEE 802.11p PHY. In particular, we use Software Defined Radios (SDRs) to record the raw signal, i.e., complex baseband samples, from IEEE 802.11p transmissions during an experiment on the road. These samples are later used with our GNU Radio-based IEEE 802.11p implementation for studying different receive algorithms – allowing for optimal comparability and repeatability. We exemplarily evaluate four typical algorithms ranging from simple ones currently used in commodity WLAN chips to more sophisticated ones proposed specifically for vehicular applications. We can show that fast fading scenarios lead to a significant packet error rate when using the standard algorithms. For better comparability, we make both our GNU Radio implementation as well as the collected traces publicly available.

CCS Concepts

•Networks → Mobile networks; Mobile ad hoc networks; Network measurement;

Keywords

Vehicular Networks; IEEE 802.11p; Software Defined Radio; Fast Fading; Experiments; Trace

1. INTRODUCTION

Future vehicles are envisioned to communicate directly with each other and potentially infrastructure nodes, forming a Vehicular Ad Hoc Network (VANET). Once established, these networks provide the base for many applications including safety, efficiency, and multimedia applications [13]. Given their potential, VANETs received great attention from both industry and academia. An important milestone was the standardization of IEEE 802.11p in 2010. IEEE 802.11p, which is by now part of IEEE 802.11 [7], belongs to the Wireless LAN (WLAN) family of standards and defines a PHY and MAC layer for use in vehicular environments. In Europe and the U.S., ETSI and FCC allocated frequencies in the 5.9 GHz band exclusively for IEEE 802.11p Dedicated Short Range Communications (DSRC).

The IEEE 802.11p PHY is based on Orthogonal Frequency Division Multiplexing (OFDM) and is a slight modification of the IEEE 802.11a standard with all physical layer timings doubled, resulting in 10 MHz channels (as opposed to 20 MHz channels in IEEE 802.11a). The rationale for this decision is simple: WLAN is a well accepted technology, chips are cheap, and by cutting the clock rate in half, standard WLAN chips can, in theory, also be used in VANETs. Stretching the frame in time domain leads to larger guard intervals and, thus, avoids inter-symbol interference even if the channel induces large delay spreads. This, however, may cause new problems given the short coherence time of VANET channels [1].

An important question is, therefore, whether a PHY that was designed for slowly varying indoor environments is able to cope with the dynamics of vehicular networks. To better understand the challenges that vehicular networks pose on receivers, measurements have been conducted to characterize the wireless channel [1]. Some of the data has been used to derive models that allow simulation of VANET channels and study them in a reproducible manner. One observation is that simple receive algorithms are not able to provide adequate performance in VANETs [6].

Driven by those results, advanced receiver were designed that are, at least in parts, able to deal with highly dynamic channels. The proposed solutions include the use of advanced receive algorithms [8, 6], use of additional pilot symbols [11], employing differential encoding [15], and exploiting diversity via relaying or multiple antennas [12].

Following this line of research, we present a measurement based study of selected IEEE 802.11p PHY receive algorithms. Using a USRP N210 Software Defined Radio (SDR), we record complex baseband samples that include a large number of frame transmissions from four different environments (city, rural, highway, and freeway), using different packet sizes and modulations. Our main goal is to study the impact of receive algorithms on the performance of IEEE 802.11p in a reproducible manner and under realistic conditions. We, therefore, implemented four state-of-the-art receive algorithms for our GNU Radio-based IEEE 802.11p transceiver and used them to decode the signal trace. The results clearly confirm the need to go beyond standard WLAN chipsets and to rely on more advanced architectures.

2. RELATED WORK

Today, as VANETs are about to be deployed on large scales, an in-depth understanding of the performance of IEEE 802.11p receivers in realistic environments is extremely important. Field Operational Tests (FOTs) are often the best way to gain such understanding, as physical layer effects – which, on the one hand, are hard to model but, on the other hand, greatly impact the results – are an inherent part of the method.

When conducting FOTs, researchers can rely on hardware prototypes like the Cohda Wireless MK5 or the NEC Linkbird IEEE 802.11p system [14]. Another option is to use certain commercial WLAN cards that allow tuning to the DSRC band and support 10 MHz channels. Here, Atheros chipsets compatible with the Linux *ath5k* and *ath9k* drivers are well-known examples. While easy to use, the drawback of these solutions is that the PHY is implemented in hardware and, thus, not open for experimentation with algorithms. To make things worse, the implementation might be undisclosed and considered intellectual property of the vendor, making the architecture less attractive for PHY layer research.

SDRs allow overcoming such limitations by replacing the transceiver with freely programmable general purpose hardware that can be used to send and receive arbitrary signals. There are generally two SDR architectures available that differ in where the signal processing is implemented. First, everything can be implemented on a Field-Programmable Gate Array (FPGA), like the well-known WARP platform [9]. Secondly, one can implement the signal processing on General Purpose Processors (GPPs) of a normal PC. This approach is, for example, adopted by many USRPs of Ettus Research, which are often used with the GNU Radio signal processing framework. In terms of software, there are several physical layer implementations for IEEE 802.11p available. We presented an Open Source transceiver in [2, 3] that runs on a normal PC, making it well suited for rapid prototyping.

Even though SDRs provide full access to the PHY, comparability and reproducibility can still be an issue. We, therefore, propose to record raw signal traces from FOTs and post-process this data using different receive algorithms.

3. SDR-BASED IEEE 802.11P PROTOTYPE

In prior works, we implemented an SDR-based IEEE 802.11a/g/p receiver [2], later extending it to a complete transceiver [3]. It is based on GNU Radio, a real-time signal processing framework for use in GPP-based SDR systems, where signal processing is implemented in software on a normal PC. Compared to FPGA-based SDRs, this architecture enables rapid prototyping and lends itself well to study the physical layer of IEEE 802.11. Another important advantage of using a GPP-based approach is that the same transceiver implementation can be used to perform simulation studies and measurements, closing the gap between theory and practice. Our transceiver has been verified by extensive tests against off the shelf WLAN cards as well as IEEE 802.11p prototypes from Cohda Wireless [2, 3]. Its PHY implementation is complete as it supports all modulation and coding schemes defined in the standard.

Given the interest of the research community in studying the impact of fast fading channels on the performance of IEEE 802.11p, we extended the transceiver with an interface to plugin different channel estimation algorithms; even during runtime. This extension was not straightforward, as we wanted to support state-of-the-art decision-direct algorithms. These algorithms adapt the channel estimate during reception also by comparing the data symbols with the ideal constellation points. Using the deviation from the ideal constellations, the channel estimates are updated during reception, which allows tracking of time-varying channels.

Since GNU Radio does not support feedback loops in transceivers, we had to refactor the functionality for channel estimation and decoding of the received constellation points into one functional block. To demonstrate the capabilities of the architecture, we implemented several algorithms, ranging from simple base line to state-of-the-art algorithms. Testing all implementations on a normal desktop PC with an Intel i7-3770 CPU showed that also the complex algorithms can be run in real-time; even with 20 MHz channels.

The Least Squares (LS) equalizer is a simple algorithm, often used as a baseline and cited to be a candidate for typical hardware implementations [10, 5]. In a nutshell, it uses the long training sequence of IEEE 802.11p as block pilots to estimate the channel. Denoting the estimate of a value X as $\hat{X}$, we calculate the channel H at subcarrier k as

$$\hat{H}(k) = \frac{Y_1(k) + Y_2(k)}{2X_{\mathrm{LT}}(k)} \; , \tag{1}$$

where $Y_{1,2}$ are the two received copies of the long training sequence and X_{LT} its known value. With the LS equalizer, this initial estimate is kept during the whole frame. While computationally very efficient, it is well known that this algorithm suffers as frames get longer or the coherence time of the channel gets shorter [10, 5].

The Least Mean Squares (LMS) algorithm overcomes this limitation by adapting the channel estimates during reception. Starting with the same initial estimate as the LS equalizer, it updates the channel after the i-th OFDM symbol using the constellation point $\hat{X}_i$ that the received symbol Y_i was demapped to as

$$\hat{H}_i(k) = (1-\alpha)\,\hat{H}_{i-1}(k) + \alpha\frac{Y_i(k)}{\hat{X}_i(k)} \; . \tag{2}$$

With α, we apply a low-pass filter to average the channel coefficients in time domain. Neither the LS nor the LMS algorithm average in frequency domain, but consider each subcarrier independently.

The Comb equalizer, in turn, interpolates linearly in frequency domain using the four comb pilots that are sent interleaved with the data symbols. Following Fernandez et al. [6], we use the mean value of the pilots at the border of the spectrum and interpolate with the vector $[m_p, P_1, P_2, P_3, P_4, m_p]$, where $P_{1..4}$ are the four comb pilots and m_p their mean. This interpolation is done for every OFDM symbol. Afterwards, a low-pass filter similar to Equation (2) can be applied to also filter in time domain.

The Spectral Temporal Averaging (STA) equalizer is a state-of-the-art algorithm designed to cope with the high dynamics of VANETs [6]. The core idea is to filter in both time and frequency domain by updating the channel estimates in two steps. First, the current symbol is decoded and the current channel estimate is calculated as

$$\hat{H}_{\mathrm{i,\ curr}}(k) = \frac{Y_i(k)}{\hat{X}_i(k)} \; . \tag{3}$$

These estimates are used to average in frequency domain by calculating a moving average over β adjacent subcarriers as

$$\hat{H}_{\mathrm{i,update}}(k) = \frac{1}{2\beta+1} \sum_{n=k-\beta}^{k+\beta} \hat{H}_{\mathrm{i,curr}}(n) \ . \tag{4}$$

In the second step, $\hat{H}_{\mathrm{i,update}}$ is used to average in time domain with a similar low-pass filter as in Equation (2).

The LMS and the STA equalizer are decision-directed, using the decoded data symbols to adapt channel estimates. This also implies that wrong decoding decisions will lead to feedback errors that possibly degrade the receive performance. In this paper, we stick to [6] and select $\alpha = 0.5$ and $\beta = 2$ as parameters for the STA algorithm. For better comparability, we used the same value of α also with the LMS equalizer. Furthermore, we wanted to isolate the effects of time and frequency selectivity and, therefore, did not apply any averaging in time domain with the Comb algorithm.

4. FIELD TESTS AND SIGNAL TRACE

To show the feasibility of the approach and to study the performance of IEEE 802.11p in realistic environments, we conducted a FOT near Paderborn, Germany. The hardware and the most important parameters of the FOT are summarized in Table 1. As transmitter, we used a WLE200NX[1] mini-PCIe WLAN card, which is based on an Atheros AR9280 chipset, supported by the *ath9k* Linux driver. Using a recent Linux kernel, this chip supports IEEE 802.11p, i.e., 10 MHz transmissions on the DSRC band at 6 GHz. Since the WLE200NX is not sold as an IEEE 802.11p device, we conducted preliminary experiments to assert that the DSRC band is not attenuated or distorted by hardware filters. In these experiments, we set the bandwidth to 10 MHz and sent frames on the DSRC band as well as on the regular IEEE 802.11a band at 5.3 GHz and 5.5 GHz. Using the SDR as a spectrum analyzer, we compared the spectrum at the different bands, which showed no differences, i.e., the output power level and the shape of the spectrum are the same on either band. A limitation of the commercial IEEE 802.11a card is its maximum transmit power of 18 dBm, which is below the allowed power level in VANETs. ETSI ITS-G5, for example, allows up to 23 dBm and 33 dBm on service and control channels, respectively [4]. In our experiments, this did not constitute a

[1]http://pcengines.ch/wle200nx.htm

	Parameter	Value
Measurement	Distance	56 km
	Duration	56.5 min
	Frames	> 25 000
	Encoding	BPSK 1/2, QPSK 1/2
	Frame Size	200 B, 500 B, 800 B
	Channel	178 (5.89 GHz)
Hardware	Sender	based on Atheros AR9280
	TX Power	18 dBm
	SDR	N210 w/ CBX daugtherboard
	RX Gain	29 dB
	Antennas	ECOM9-5500 (9 dBi dipole)
	GPS Receiver	u-blox NEO-7N

Table 1: Most relevant information on the measurement setup and the hardware.

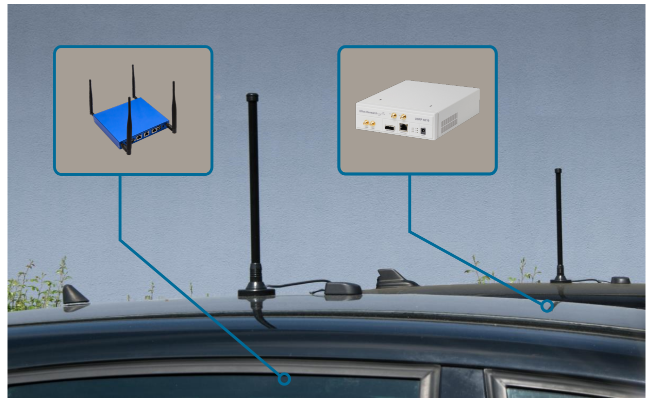

Figure 1: Picture of the setup and the devices used in the experiments.

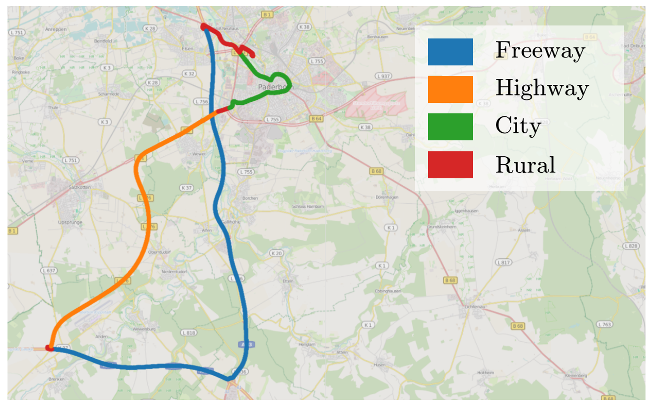

Figure 2: GPS trace of the measurements, showing the various environments. The map is © OpenStreetMap contributors.

problem since the distance between sender and receiver was never larger than 140 m and could easily be covered.

During the experiment, we sent frames with random payloads, cycling through the six combinations of BPSK 1/2 and QPSK 1/2 frames with sizes of 200 B, 500 B, and 800 B. The average frame rate was about eight frames per second, allowing us to generate a large data set with over 25 000 transmissions during the 56.5 min experiment. The frames were sent on channel 178 at 5.89 GHz.

On receive side, we used a USRP N210 from Ettus Research with a CBX daugtherboard that covers the frequencies from 1.2 GHz to 6 GHz. The SDR was used to record the raw signal, dumping the samples directly to an SSD drive without any signal processing. Using 4 B floats for the real and imaginary parts of the complex baseband samples, the 10 Msps stream from the SDR resulted in 80 MB/s that had to be stored. During the 56.5 min FOT, we captured over 270 GB sample data. The gain of the SDR was set constant at 29 dB, corresponding to 92 % of its maximum.

Both cars were equipped with 9 dBi ECOM9-5500 dipole antennas mounted on the roof of the cars as shown in Figure 1. To rule out any interactions with the cars' FM antennas, we took them off during the experiments. Furthermore, we equipped the cars with NEO-7N,[2] high precision GPS re-

[2]https://www.u-blox.com/en/product/neo-7-series

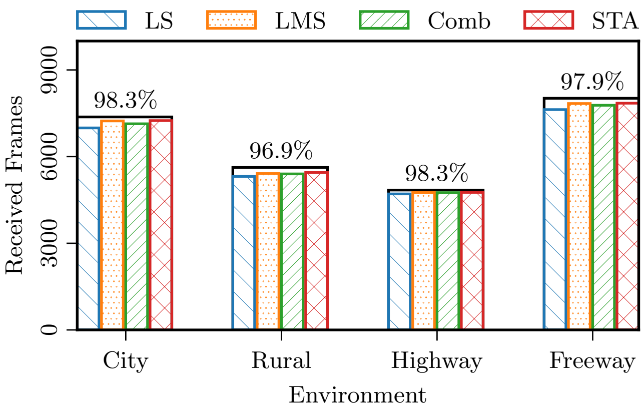

Figure 3: Performance comparison of receive algorithms in different environments.

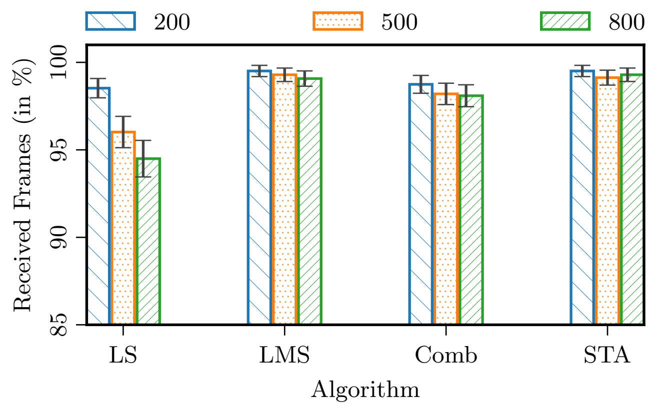

Figure 4: Impact of frame size on the receive performance of the selected algorithms.

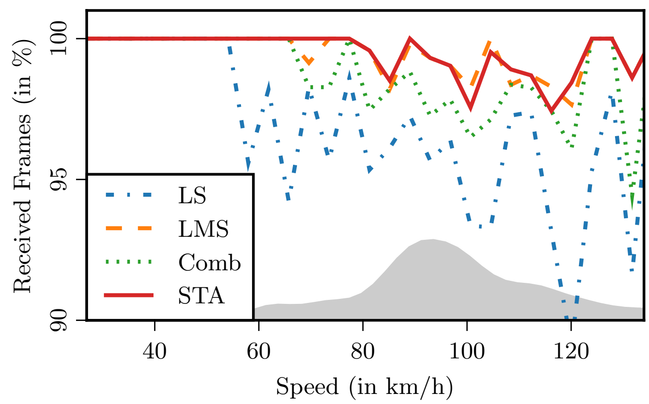

Figure 5: Impact of speed on the receive performance of the selected algorithms.

ceivers from u-blox and logged their position and speed every 0.5 s. The values at the time instances of frame transmissions were later interpolated linearly based on the GPS time series. Figure 2 plots the GPS trace of our 56 km drive, color-coding the surroundings, which range from freeway, to highway, to rural, to city environments. All data is recorded in one take so that the whole setup, i.e., cars, devices, and positions of the antennas remain constant, allowing for a direct comparison between the environments.

On the freeway, we tried to resemble realistic situations by letting one car fall back to later accelerate and overtake. This way, we do not only capture situations with different absolute speeds, but also with different relative speeds. In the other environments, it was not easily possible to overtake each other. We, therefore, stuck to varying the distance and letting other cars and trucks get in between sender and receiver. We spent considerable time in each environment, sending 7373 frames in the city, 5624 frames in rural areas, 4842 frames on the highway, and 8019 frames on the freeway.

5. EVALUATION OF THE SIGNAL TRACE

While our emphasis is mainly on the methodology and not primarily on the algorithms, we conduct an exemplary evaluation of the signal trace to highlight what insights we can gain from the data. For these evaluations, we decoded the trace offline using the algorithms described in Section 3.

Figure 3 shows the number of received frames for each algorithm and environment. On top of the bars, we annotated the percentage of frames that could be decoded by the best algorithm, which was in either case STA. We can see that the difference between the algorithms is rather small, with each algorithm decoding over 94 % of the frames. On the one hand, this shows that our SDR implementation works well also realistic environments, on the other hand it shows that even the simple LS algorithm is able to decode most frames, at least in that particular scenario. This suggests that even WLAN chips that are not specially designed for IEEE 802.11p could provide reasonable performance.

We, however, expect these differences to be more pronounced with higher relative speeds. If the cars approach each other, for example, the channel changes faster, leading to shorter coherence times. Such conditions would degrade the performance of the LS equalizer, as the initial channel estimate becomes outdated during the frame, causing bit errors and, ultimately, dropped frames.

An indicator of the detrimental effect of the frame sizes on the LS algorithm is shown in Figure 4 where we plot the percentage of QPSK 1/2 frames received at speeds above 80 km/h. While the absolute speed does not immediately imply a time-variable channel, it is, at least, correlated, since it determines the relative speed to static reflectors like street signs or the guardrail. We split the data based on the frame sizes and annotate the 95 % confidence intervals. The plot shows that the LMS, Comb, and STA equalizers are able to cope also with larger frame sizes. There is, in fact, no significant difference in our scenario. The LS algorithm, in turn, shows lower performance with larger frames, highlighting the impact of outdated channel estimates.

A different perspective on the data is given in Figure 5. The graph plots the percentage of received QPSK 1/2 frames in the freeway environment against the transmitter's speed. For the plot, we split the data in 30 bins and calculated the frame error rate per bin. The data basis, i.e., number of frames per bin varies significantly, as there are, for example, only few frames at low speeds on the freeway. To indicate the distribution of the in total over 4000 frames, we added a density plot to the figure. Shaded in gray, it shows the distribution of transmitter speeds at which frames were sent.

As expected, the trend is that more frames are lost at higher speeds. The plot also highlights the impact of fast fading on the LS algorithm. While the graph does not look like in a textbook, it clearly shows the LS equalizer's sensitivity to speed. Compared to the other algorithms, the LS algorithm shows a higher variance and lower reception rate. Summarizing the results of our FOT, we can conclude that most transmission did not pose great challenges on the receive algorithms, so that the overall differences between the algorithms were small. If, however, we focus on larger frames or higher speeds, the differences become larger and the limitations of simple algorithms become more prevalent.

6. DISCUSSION AND CONCLUSIONS

We believe that recording the raw signal with an SDR is an interesting option to study VANETs. Potentially complemented with interactive experiments, our approach provides the major advantage that receiver designs can be directly compared, which is not easily possible otherwise. Even when using multiple receivers in parallel, each device, cable, and antenna mounting position will exhibit unique characteristics, leading to systematic differences. A rule of thumb often used with MIMO is that fast fading effects are independent when the antennas are spaced more than 0.4 times the wave length. For IEEE 802.11p, this implies that antennas as close as 2 cm experience independent multipath effects. Such problems can be completely avoided with the proposed method.

The possibility to compare receive algorithms directly provides several benefits: It, for example, allows balancing the tradeoff between performance and complexity, relevant when designing IEEE 802.11p receivers. In particular, it allows estimating the potential performance gains of sophisticated VANET receivers over devices that are based on simpler WLAN chipsets. Moreover, direct comparison of algorithms helps in identifying the major causes of packet loss (e.g. if short coherence times or short coherence bandwidths are more severe) and, thus, deepen the understanding of IEEE 802.11p in general. While storing all complex baseband samples produces large amounts of data, it also provides the possibility to debug a FOT in retrospective. Using the raw sample stream it is possible to investigate the cause of unexpected results, which is hardly possible if the data is not available.

Finally, we think that comparison and reproducibility should not be limited to within one project only. In fact, the approach could allow reproduction even by fellow researchers. While we already took the first step and released our SDR-based IEEE 802.11p implementation as Open Source software, we believe that it could be beneficial for the community to also share traces of FOTs. Data from various environments and with different hardware configurations would allow evaluation of algorithms on the same data and even to use past FOTs to evaluate future ideas. To foster the adoption of the approach and to allow reproduction of the results we will release the trace together with the paper.

Acknowledgement

This work was supported by a German Academic Exchange Service (DAAD) fellowship within the FITweltweit program.

7. REFERENCES

[1] G. Acosta-Marum and M. Ingram. Six Time- and Frequency-Selective Empirical Channel Models for Vehicular Wireless LANs. *IEEE Vehicular Technology Magazine*, 2(4):4–11, Dec. 2007.

[2] B. Bloessl, M. Segata, C. Sommer, and F. Dressler. An IEEE 802.11a/g/p OFDM Receiver for GNU Radio. In *ACM SIGCOMM 2013, SRIF Workshop*, pages 9–16, Hong Kong, China, Aug. 2013. ACM.

[3] B. Bloessl, M. Segata, C. Sommer, and F. Dressler. Towards an Open Source IEEE 802.11p Stack: A Full SDR-based Transceiver in GNURadio. In *IEEE VNC 2013*, pages 143–149, Boston, MA, Dec. 2013. IEEE.

[4] ETSI. Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band. EN 302 663 V1.2.1, ETSI, July 2013.

[5] J. A. Fernandez, K. Borries, L. Cheng, B. V. K. Vijaya Kumar, D. D. Stancil, and F. Bai. Performance of the 802.11p Physical Layer in Vehicle-to-Vehicle Environments. *IEEE Transactions on Vehicular Technology*, 61(1):3–14, Jan. 2012.

[6] J. A. Fernandez, D. D. Stancil, and F. Bai. Dynamic Channel Equalization for IEEE 802.11p Waveforms in the Vehicle-to-Vehicle Channel. In *Allerton Conference on Communication, Control, and Computing*, pages 542–551, Monticello, IL, Sept. 2010. IEEE.

[7] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. Std 802.11-2012, IEEE, 2012.

[8] T. Kella. Decision-Directed Channel Estimation for Supporting Higher Terminal Velocities in OFDM Based WLANs. In *IEEE GLOBECOM 2003*, pages 1306–1310, San Francisco, CA, Dec. 2003. IEEE.

[9] A. Khattab, J. Camp, C. Hunter, P. Murphy, A. Sabharwal, and E. W. Knightly. WARP: A Flexible Platform for Clean-Slate Wireless Medium Access Protocol Design. *ACM SIGMOBILE Mobile Computing and Communications Review (MC2R)*, 12(1):56–58, Jan. 2008.

[10] C. F. Mecklenbräuker, A. F. Molisch, J. Karedal, F. Tufvesson, A. Paier, L. Bernadó, T. Zemen, O. Klemp, and N. Czink. Vehicular Channel Characterization and its Implications for Wireless System Design and Performance. *Proceedings of the IEEE*, 99(7):1189–1212, July 2011.

[11] K. K. Nagalapur, F. Brännström, and E. G. Ström. On Channel Estimation for 802.11p in Highly Time-Varying Vehicular Channels. In *IEEE ICC 2014*, pages 5659–5664, Sydney, Australia, June 2014. IEEE.

[12] J. Nuckelt and T. Kürner. MRC Performance Benefit in V2V Communication Systems in Urban Traffic Scenarios. In *EUCAP 2012*, pages 2311–2315, Prague, Czech Republic, Mar. 2012. IEEE.

[13] C. Sommer and F. Dressler. *Vehicular Networking.* Cambridge University Press, Nov. 2014.

[14] H. Stübing, M. Bechler, D. Heussner, T. May, I. Radusch, H. Rechner, and P. Vogel. simTD: A Car-to-X System Architecture for Field Operational Tests. *IEEE Communications Magazine*, 48(5):148–154, May 2010.

[15] Y. Zhang, I. L. Tan, C. Chun, K. Laberteaux, and A. Bahai. A Differential OFDM Approach to Coherence Time Mitigation in DSRC. In *ACM VANET 2008*, pages 1–6, San Francisco, California, Sept. 2008. ACM.