Performance of Real-Time Wireless Communication for Railway Environments with IEEE 802.11p

Daniel Richter   Jossekin Beilharz   Lukas Pirl   Christian Werling   Andreas Polze

Operating Systems and Middleware Group
Hasso Plattner Institute for Digital Engineering at University of Potsdam, Germany
P.O.Box 90 04 60, D-14440 Potsdam, Germany
{firstname.lastname}@hpi.uni-potsdam.de

Abstract

IEEE 802.11p complements the widespread 802.11 standard for use in vehicular environments. Designed for communication between wireless devices in rapidly changing environments, it handles situations where connection and communication must be completed in very short periods of time. Even though this is supposed to be a substantial improvement and essential for real-time applications, latencies have been rarely investigated in existing studies. Based on practical experiments, we evaluate how beneficial 802.11p's changes in comparison to regular 802.11n are and whether the usage of IEEE 802.11p is suitable within environments with real-time constraints. We compare latencies of networks in OCB mode to both networks in IBSS (ad-hoc) and BSS/AP (access point) mode by measuring the initial connection speed and the latency of ICMP packets' round-trip times. Furthermore, the response of the latter to disturbances is measured. The results show OCB to be superior to both BSS/AP and IBSS modes in average latency, maximum latency, and standard deviation under all tested circumstances.

1. Introduction

Vehicular ad-hoc networks describe the vision of vehicles that communicate directly with each other and infrastructure nodes. Once a network of interconnected vehicles is established, this network can be used for many applications that increase safety, efficiency, and comfort in road traffic. Therefore, public, industrial, and scientific efforts in the area of vehicular communication systems intensified over the last few years.

But there is more: Standardized car-to-car components can also be used in the railway context with use cases such as the transfer of maintenance data of critical infrastructure elements (e.g. for largely self-sufficient units such as level crossings with
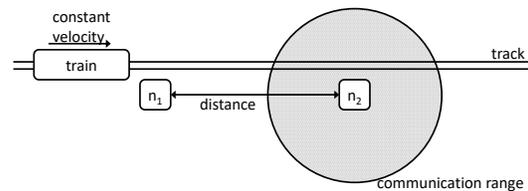

Figure 1. A typical Rail-to-X use case: n_1 and n_2 resemble points of information (e.g. maintenance data of a switch point). Trains passing by are collecting data over the air via dedicated short-range communications. Due to the limited communication range and the high speed of the train, the communication is time-critical.

monitoring signals, diagnostic systems of switches, or weather stations close to the tracks). In this scenario, data transfer via passing trains is an attractive alternative to cost-intensive cable-connected solutions (see Figure 1). Other gainful use cases include the communication between cars and barriers at level crossings (e.g. to request the barriers to open), and the communication needed for request stops (e.g. where a button on the platform can be used for sending a request to stop to the train).

In 2010, the IEEE passed the *Amendment 6: Wireless Access in Vehicular Environments* to the 802.11 standard, creating the 802.11p standard [1, 2]. 802.11p complements the widespread 802.11 standard on the *Physical* (PHY) and *Medium Access Control* (MAC) layer for use in vehicular environments.

Since the regular IEEE 802.11a standard is optimized for local networks with little device mobility, dealing with the problems arising from the inherently high mobility of the devices in networks between vehicles is a challenge. The changes introduced in IEEE 802.11p

HICSS

aim to ensure communication between wireless devices in rapidly changing environments and handle situations where operations must be completed in a short period of time.

Due to potentially higher vehicular speeds in Rail-to-X scenarios (compared to Car-to-X), time windows for data transmissions are smaller and require low latencies and round-trip times (RTTs) – including both establishing a connection and communication itself. While existing work evaluating the performance of 802.11p in different environments (such as [3, 4, 5]) usually expect a connection to be present at least some seconds, we focus on an extremely short timespan for connection and communication (approximately 100 ms).

In the following sections, we evaluate based on practical experiments how beneficial *802.11p*'s changes in comparison to regular 802.11n really are and whether the usage of IEEE 802.11p is suitable within environments with real-time constraints. Despite the large number of existing work concerning the performance of 802.11p, latencies have been rarely investigated – even though they are supposed to be a substantial improvement and essential for real-time applications.

2. Background

For many years, 802.11 has been predominantly working on a *2.4 GHz* band on which 14 channels of 5 MHz bandwidth were being used. Nowadays many wireless networks switched to the less crowded 5 GHz band, using different channel layouts based on their regulatory domain (e.g. FCC in Northern America and ETSI in Europe). 802.11p supplements the 5 GHz Band at its upper border and therefore facilitates the *5.9 GHz* band. It works on half- and quarter-rate channels (i.e. 5/10 MHz) and strictly defines one *Control Channel* (CCH) and several *Service Channels* (SCH). These differences from regular 802.11 radio properties are meant to improve resistance to physical disturbances in vehicular environments (e.g. Doppler shift in the carrier frequency due to high mobility).[6]

To satisfy varying infrastructure types and topologies, 802.11 knows several different modes of operation. What most of them have in common is a so-called *Basic Service Set* (BSS) in which they are acting. A BSS is comprised of specific networking parameters (radio frequency, modulation scheme etc.) and can be shared among its members either by a central redistribution point (e.g. an *access point* (AP)) or decentrally by independent stations. They are identified by their *Basic Service Set Identifier* (BSSID).

- **AccessPoint (AP) infrastructure mode** is used by a "master" in a managed network. Its tasks include keeping track of clients and embracing security policies. In addition to its BSSID it is also identified by a natural language network name (*SSID*).

- **Managed mode** works in conjunction with the aforementioned BSS mode and resembles a "slave" in a managed network. In order to join a managed network, it authenticates and associates against an access point.

- In **Ad-Hoc (IBSS) mode** each client manages the *Independent BSS* so it can work without an access point.

- **Mesh mode** works in a similar decentralized fashion as IBSS does, but adds indirect communication through intelligent dynamic routing by its members. It therefore greatly increases the range of the network.

- **Monitor mode** is a passive mode of operation, which enables unfiltered inspection of all wireless packets that are physically receivable.

- **ESS (WDS) mode** enables access points to interconnect with each other as an alternative to a wired backbone. For requirements exceeding the range limitations of an independent BSS, 802.11 defines an *Extended Service Set*.

IEEE 802.11p complements these modes of operation by a novel mode dispensing with a BSS: **OCB mode** (*Outside the context of a BSS*) aims for minimizing the overhead induced by association and authentication. Therefore, these kinds of functionality must then be provided by higher network layers.

3. Related Work

The performance of 802.11p under various conditions has been studied analytically,[7] in simulations,[8, 9, 10, 3] and experimentally [11, 12, 4, 5] over several years. Despite the large number of examinations, most studies only look at throughput, packet loss, and signal range; latencies (especially including the time for establishing a connection) have been rarely investigated – even though they are supposed to be a substantial improvement brought about by the amendment.

Simulations done by Hameed Mir and Filali [9] compared LTE and 802.11p in terms of delay, reliability, and scalability and came to the conclusion that 802.11p offers lower latencies and a higher throughput with fewer than 50 participants and allows end-to-end delays less than 100 ms and throughput equivalent to 10 kbps.

Bilstrup et al. [10] examined real-time requirements on the MAC protocol when used in ad-hoc vehicle communication. With a simulation of periodic broadcasts of heart-beat messages in a vehicle-to-vehicle situation they worked out that carrier sense multiple access (CSMA) is undesirable for time-critical communication and propose self-organizing time division multiple access (STDMA) for real-time data traffic between vehicles.

Similar to our approach, Demmel et al. [11] conducted on-track experiments with commercial-of-the-shelf hardware and a patched, open-source ath5k Wi-Fi driver in order to enable 802.11p channels. They measured the maximum range, frame loss, and average latency for various relative speeds. Notably, OCB was disabled so that normal 802.11 ad-hoc behavior was used and 802.11p latency is evaluated in a more "classical" 802.11 architecture. Demmel et al. conclude that the average latency is centered around 1.5 ms, whereas latency remains under 4 ms in almost all circumstances, regardless of range and relative speed. The direction of movement does not have any influence on the latencies.

Since we use commercial-of-the-shelf hard- and software for our test setup, a major part in enabling the work done in this paper relies on previous work: Most importantly, the work done by Lisový et al. [6] is responsible for implementing 802.11p's amendments into different Linux subsystems. Their technical report serves well as first-hand practical info on the Linux support and also gives a concise summary of 802.11p's changes and their motivation. Regarding compatible hardware, Fernández' work done in his Master thesis [13] is of great value, as it not only states positively tested cards with 802.11p support, but also gives insights into the physical requirements to such cards in general.

4. Testbed Setup

Since no off-the-shelf hardware is sold with 802.11p support, we had to patch drivers and adjust the operating system configuration.

4.1. Hardware

Wireless cards officially supporting 802.11p can be scarcely found online, especially those sold individually. However, the work done in [13] points out several 5 GHz wireless cards which are physically able to support the 5.9 GHz band as well. To make 802.11p usable for applications, some minor patches to the drivers are needed. In case of Atheros wireless cards previous work exists that patches the driver accordingly [14].

Testbed setup A consists of the following hardware:
- Wireless card
  - *Qualcomm Atheros AR5B22 Mini PCI-e*
  - *Chipset: AR9462*
- Mini PCI-e to PCI-e adapter: *adaptare 49006*
- Dell Workstation
  - *Intel Core i5-3470 CPU*
  - *8 GB RAM*

Testbed setup B consists of the following hardware:
- Wireless card
  - *Qualcomm Atheros AR9462*
  - *2.4/5 Ghz WLAN + Bluetooth*
- HPE GL20 IoT Gateway
  - *Intel I5-4300U CPU*
  - *8 GB RAM*

In both setups the distance between the communicating nodes is 0.5 meters due to laboratory conditions.

4.2. Software

As an operating system *Ubuntu 16.04 LTS* with a *Linux kernel 4.13.0-31-generic x86_64* was used. The authors chose Linux because of the aforementioned implementation of 802.11p's changes into its 802.11 subsystems (Figure 2).

Figure 2. IEEE 802.11 Linux kernel implementation architecture. The support for 802.11p has to be present on all layers. [6]

More specifically, previous work [6] made changes to the *cfg80211*, *mac80211* and *nl80211* layers and thus making the kernel aware of the new OCB mode of operation, e.g. setting the BSSID to a wildcard value in *mac80211*, and new commands in *cfg80211* and *nl80211* to virtually join and leave an OCB network on a particular frequency, i.e. sending and receiving on that particular frequency.

Furthermore, in order to manipulate wireless devices and their configuration through the Linux tool *iw* it has to be aware of the new band and mode as well. Support for 802.11p is available in *iw 4.0* and later.[1]

---

[1] https://git.kernel.org/cgit/linux/kernel/git/jberg/iw.git/commit/?id=3955e5247806b94261ed2fc6d34c54e6cdee6676

wireless-regdb is the regulatory database used by Linux. This, too, needs to be patched in order to work with the new bands. Instructions on how to do so can be found in the authors' repository for benchmarking. [2]

It should be noted that the work done in this paper does not deal with any user-space specification regarding vehicular use.

5. Benchmarks

For a meaningful benchmark, we compare the performance of a *802.11p 5.9 GHz OCB network* (**OCB**) to – staying in the 5 GHz band – both a *802.11n 5 GHz open ad-hoc network* (**IBSS**) and a *802.11n 5 GHz open access point network* (**BSS/AP**).

Having in mind a scenario of two fast moving stations, the authors chose to measure the initial **connection speed** of the candidate connections and their **latency**. The latter is measured by timing plain ICMP "echo request/response" packet's *round-trip times* (RTT). For both establishing a connection and exchanging data we require a time frame of around 100 ms, just as [6] does.

5.1. Connection Speed

Regarding connection speed, the following cases were measured (with *testbed setup A*):

- **IBSS**: Joining via `iw ibss join` an existing station in *IBSS (ad-hoc) mode* on a given channel. As this command itself is non-blocking, a script was used to poll the connection status until connected.

- **BSS/AP**: Connecting in *managed mode* to an existing station in open *BSS (AP) mode*.

- **OCB**: Joining via `iw ocb join` an OCB band on a given frequency.

Those actions were performed 100 times and their average duration in milliseconds can be seen in Figure 3. Joining an open ad-hoc network takes about 5 times as long as connecting to an open managed network. Slightly faster is joining an OCB band.

5.2. Latency

Regarding connection speed, the following ICMP packet round trips were measured (with *testbed setup A*):

- **IBSS**: between two stations in an open ad-hoc network on a 5 GHz band
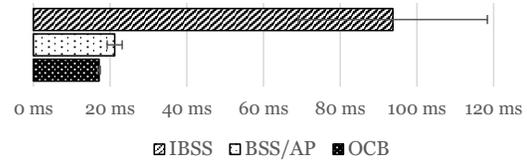
Figure 3. Average connection speeds (100 iterations, testbed setup A) in IBSS mode (93.72 ms), BSS/AP mode (21.2 ms), and OCB mode (17.12 ms).

Table 1. Best, worst, and average case Ping RTTs (250 iterations). OCB is best at average RTT and worst case RTT.

Mode	Minimum	Maximum	Median	Average	Std.dev.
IBSS	0.93 ms	3,449.00 ms	1.45 ms	78.06 ms	422.91
BSS/AP	0.96 ms	115.00 ms	1.43 ms	3.13 ms	11.52
OCB	1.55 ms	1.89 ms	1.60 ms	1.62 ms	0.05

- **BSS/AP**: *ap open*: between two stations in a managed network (one in AP mode, one in managed mode) on a 5 GHz band

- **OCB**: between two stations in an OCB network on a 5.9 GHz band

With the exception of a few extreme outliers in the Ping RTT tests summarized in Table 1 and Figure 4, the 802.11n IBSS and BSS/AP modes seem to deliver performance comparable to the 802.11p OCB implementation; Although, when considering the median RTT both the 802.11n modes outperform 802.11p, for real-time applications only worst case boundaries are of interest.

With an average RTT of 78.06 ms the **IBSS** network performs worst. This may be caused by the management overhead due to the BSS and the ad-hoc mode's decentral fashion. Furthermore, outliers with up to 3,449 ms exist, which far exceeds our required 100 ms timeframe and makes this setup an unreliable candidate for our use case.

With an average RTT of 3.13 ms and a maximum of 115 ms (Figure 4), the **BSS/AP** network is about 25 times faster than IBSS. Having a central point of management, i.e. the access point, one can assume that this makes up the great performance difference compared to the ad-hoc measurements.

Nevertheless, with an average of 1.62 ms the **OCB** network is even twice as fast as its managed counterpart and outperforms an ad-hoc connection (Figure 4). It further scores with small deviations, having RTTs ranging between 1.55 ms and 1.89 ms. Here one can clearly see the benefits of omitting the BSS to the overall latency.

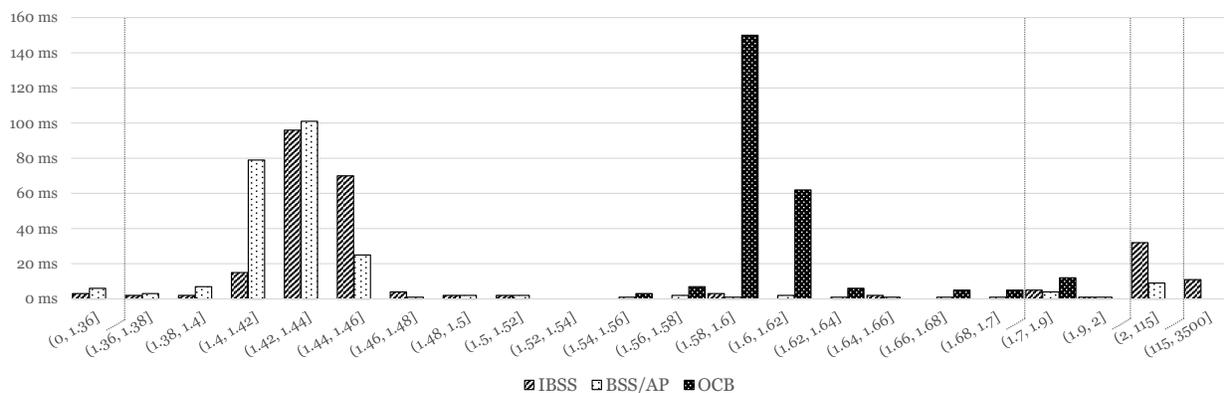The scripts used for benchmarking and their

Figure 4. Histogram of ping RTTs in ms (250 iterations, testbed setup A) between two stations in an open ad-hoc network on a 5 GHz band, (IBSS), between two stations in a managed network on a 5 GHz band (BSS/AP), and between two stations in an OCB network on a 5.9 GHz band (OCB). Considering the median RTT, both the 802.11n modes outperform 802.11p, but for real-time applications only worst case RTT is important (IBSS worst case: 3,449 ms, BSS/AP worst case: 115 ms, OCB worst case: 2 ms)

respective results can be found in the authors' repository for benchmarking.[2]

5.3. Interference Sensitivity

In addition to the comparing the plain latencies in different modes, the sensitivity to disturbances of OCB (802.11p) in comparison to the BSS/AP and IBSS modes (802.11n) was evaluated. To test this, the duration of a full round trip between two nodes was measured, while a third node tries to disturb this connection by sending arbitrary data packets with different intensities on the same channel as the first two nodes (with *testbed setup B*).

Table 2 and Figure 5 show the latencies in 2,000 iterations with varying disturbance intensity for each mode. This intensity is denoted as the size (in bytes) of the packets that are sent as disturbance by a third node. While the latencies for **IBSS** mode in the undisturbed case are very low, as shown in the previous measurements, the performance drops with increasing disturbance intensity. While the median (1.38 ms) and average (1.45 ms) latencies are the best of all tests, the worst case (19.7 ms) is the worst undisturbed latency. The disturbance response is shown to be unpredictable with a standard deviation of up to 34.37 ms.

The usage of the **BSS/AP** modus improves this picture already. Now, the maximum and average latencies are lower than those in IBSS mode for any disturbance intensity. The same holds true for the standard deviation.

OCB is the most predictable mode overall and least prone to disturbances. While also here the disturbances do have an effect especially on the worst case round-trip time (going from 1.9 ms to 45.3 ms), the standard

deviation remains within 5 ms at all times. Especially the good predictability is valuable in a real-time context.

6. Conclusion and Future Work

We showed 802.11p's OCB to be suitable within the time constrains of the railway use cases we envision. The measurements of the latency have shown OCB to be superior to both BSS/AP and IBSS modes in average latency, maximum latency, and standard deviation. These findings still hold true in saturated wireless environments.

We plan to further explore the performance of 802.11p and its applicability to the railway sector in multiple directions. There are two areas where we see the most immediate future work. First, the applicability of 802.11p to vehicles moving at railway speeds has been evaluated in terms of the resulting shorter communication window. In the future, this evaluation should also include the Doppler shifts by using fast moving nodes. Second, the work presented in this paper should be extended to validate the simulations of a railway communication scenario that have been done previously.

References

[1] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an international standard for wireless access in vehicular environments," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE.* IEEE, 2008, pp. 2036–2040.

[2] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, Amendment 6: Wireless Access in Vehicular Environments*, IEEE Standards Association Std. 802.11p, 2010.

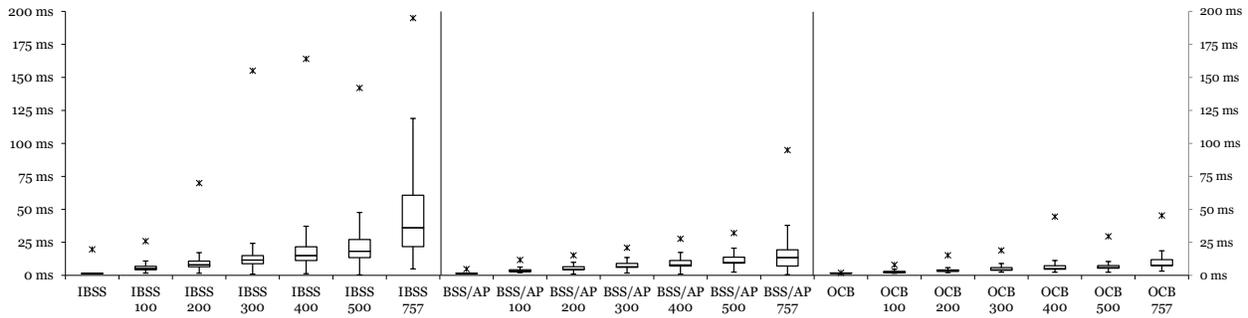[3] V. Vukadinovic, K. Bakowski, P. Marsch, I. D. Garcia, H. Xu, M. Sybis, P. Sroka, K. Wesolowski, D. Lister,

Figure 5. Ping RTT in ms (2,000 iterations, testbed setup B) in IBSS mode, BSS/AP mode, and OCB mode. x denotes the maximum (worst case) RTT measured. The trailing number indicates the disturbance intensity (size in bytes of the packets that are sent at the same time by a third node); no trailing number means undisturbed.

Table 2. Best, worst, and average case Ping RTTs (2,000 iterations, testbed setup B). The trailing number indicates the disturbance intensity (size in bytes of the packets that are sent at the same time by a third node); no trailing number means undisturbed. OCB is best at average RTT and worst case RTT.

Mode	Min	Max	Median	Avg	Std.dev.
IBSS	0.84 ms	19.70 ms	1.38 ms	1.45 ms	0.66 ms
IBSS 100	1.65 ms	25.90 ms	5.11 ms	5.69 ms	2.75 ms
IBSS 200	1.56 ms	69.90 ms	7.91 ms	9.27 ms	5.95 ms
IBSS 300	0.93 ms	155.00 ms	11.60 ms	13.40 ms	10.43 ms
IBSS 400	1.28 ms	164.00 ms	15.00 ms	18.94 ms	16.24 ms
IBSS 500	0.33 ms	142.00 ms	18.30 ms	21.62 ms	14.64 ms
IBSS 757	4.85 ms	195.00 ms	36.00 ms	47.12 ms	34.37 ms
BSS/AP	0.78 ms	4.74 ms	1.42 ms	1.45 ms	0.25 ms
BSS/AP 100	1.97 ms	11.70 ms	3.11 ms	3.79 ms	1.29 ms
BSS/AP 200	0.82 ms	15.20 ms	4.60 ms	4.95 ms	2.45 ms
BSS/AP 300	1.55 ms	20.90 ms	6.40 ms	7.89 ms	2.61 ms
BSS/AP 400	0.88 ms	27.70 ms	7.82 ms	8.02 ms	4.35 ms
BSS/AP 500	0.62 ms	32.10 ms	9.58 ms	11.37 ms	4.37 ms
BSS/AP 757	0.50 ms	94.90 ms	13.40 ms	13.02 ms	9.20 ms
OCB	1.45 ms	1.89 ms	1.64 ms	1.63 ms	0.03 ms
OCB 100	1.58 ms	7.98 ms	2.42 ms	2.62 ms	0.88 ms
OCB 200	2.05 ms	15.20 ms	3.31 ms	3.75 ms	1.38 ms
OCB 300	2.37 ms	18.90 ms	4.11 ms	4.75 ms	1.70 ms
OCB 400	2.32 ms	44.50 ms	4.94 ms	5.88 ms	2.54 ms
OCB 500	1.98 ms	29.60 ms	5.70 ms	6.68 ms	2.56 ms
OCB 757	3.17 ms	45.30 ms	7.76 ms	9.54 ms	4.50 ms

and I. Thibault, "3gpp C-V2x and IEEE 802.11 p for Vehicle-to-Vehicle communications in highway platooning scenarios," *Ad Hoc Networks*, vol. 74, pp. 17–29, 2018.

[4] M. Kloc, R. Weigel, and A. Koelpin, "SDR implementation of an adaptive low-latency IEEE 802.11p transmitter system for real-time wireless applications," in *2017 IEEE Radio and Wireless Symposium (RWS)*, Jan. 2017, pp. 207–210.

[5] B. Bloessl, M. Segata, C. Sommer, and F. Dressler, "Performance Assessment of IEEE 802.11 p with an Open Source SDR-based Prototype," *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 1162–1175, 2018.

[6] R. Lisový, M. Sojka, and Z. Hanzálek, "IEEE 802.11p Linux Kernel Implementation," Czech Technical University in Prague, Tech. Rep., 2014.

[7] A. A. A. Almohammedi, N. K. Noordin, A. Sali, F. Hashim, and S. Saeed, "A comprehensive performance analysis of IEEE 802.11p based MAC for vehicular communications under non-saturated conditions," *Journal of ICT Research and Applications*, vol. 11, no. 1, pp. 91–112, 2017.

[8] S. Gräfling, P. Mähönen, and J. Riihijärvi, "Performance evaluation of IEEE 1609 WAVE and IEEE 802.11p for vehicular communications," in *Ubiquitous and Future Networks (ICUFN), 2010 Second International Conference on*. IEEE, 2010, pp. 344–348.

[9] Z. Hameed Mir and F. Filali, "LTE and IEEE 802.11p for vehicular networking: a performance evaluation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, p. 89, May 2014.

[10] K. Bilstrup, E. Uhlemann, E. G. Strom, and U. Bilstrup, "Evaluation of the IEEE 802.11p MAC method for vehicle-to-vehicle communication," in *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*. IEEE, 2008, pp. 1–5.

[11] S. Demmel, A. Lambert, D. Gruyer, A. Rakotonirainy, and E. Monacelli, "Empirical IEEE 802.11p performance evaluation on test tracks," in *Intelligent vehicles symposium (IV), 2012 IEEE*. IEEE, 2012, pp. 837–842.

[12] V. Shivaldova, G. Maier, D. Smely, N. Czink, A. Alonso, A. Winkelbauer, A. Paier, and C. F. Mecklenbräuker, "Performance evaluation of IEEE 802.11p infrastructure-to-vehicle tunnel measurements," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*. IEEE, 2011, pp. 848–852.

[13] J. Fernández Pastrana, "802.11p standard and v2x applications on commercial wi-fi cards," Master's thesis, Universidad de Valladolid. Escuela Tcnica Superior de Ingenieros de Telecomunicacin, 2017.

[14] R. Lisový, "ath9k: Add OCB mode support," Linux driver patch, 2014. [Online]. Available: https://github.com/CTU-IIG/802.11p-linux/commit/bf45e0160af428dac8893e48d506ac428fed16b2