Ad Hoc Networks 000 (2015) 1–11

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Bluetooth Low Energy performance and robustness analysis for Inter-Vehicular Communications

Walter Bronzi*, Raphael Frank, German Castignani, Thomas Engel

Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, L-2721, Luxembourg

ARTICLE INFO

Article history:
Received 2 March 2015
Revised 26 June 2015
Accepted 14 August 2015
Available online xxx

Keywords:
Bluetooth Low Energy
Inter-Vehicular Communications
Mobile computing
Experiments
IEEE 802.11
Interference

ABSTRACT

Bluetooth Low Energy (BLE) is quickly and steadily gaining importance for a wide range of applications. In this paper we investigate the potential of BLE in a vehicular context. By means of experiments, we first evaluate the characteristics of the wireless channel, then we define a set of driving scenarios to analyse how BLE is affected by varying speed, distance and traffic conditions. We found that the maximum communication range between two devices can go beyond 100 m and that a robust connection, capable of handling sudden signal losses or interferences, can be achieved up to a distance of 50 m even for varying traffic and driving conditions. We then present a proof-of-concept mobile application for off-the-shelf smartphones that can be used to transmit data over multiple hops. Next, we analyse how BLE handles other interferences on the same frequency band by building and validating an interference testbed based on the IEEE 802.11 technology. Finally we discuss the advantages and limitations of BLE for Inter-Vehicular Communications (IVC) and propose potential applications.

1. Introduction

Smartphones are increasingly equipped with sensors and communication interfaces. One of the latest additions to the communication technologies is *Bluetooth Low Energy* (BLE), also called *Bluetooth Smart*. Although similar in some regards, BLE is not backwards compatible with previous Bluetooth versions as it uses a different controller (i.e. physical and link layer). However, most devices that support BLE implement both protocol stacks in dual-mode.

This low energy and low latency communication protocol has been developed to facilitate communication between mobile devices and other peripherals (e.g. smartwatches [1]). Common application areas include fitness, healthcare and smart homes, among others. The protocol defines several upper layer functionalities that allow fast and easy message exchange between devices.

In this paper we investigate the potential of BLE for *Inter-Vehicular Communications* (IVC). This work is motivated by the fact that the deployment of specifically designed IVC technologies such as *Dedicated Short Range Communications* (DSRC) based on IEEE 802.11p, is taking longer than initially expected [2]. It is our belief that the ubiquity of BLE enabled mobile devices would allow a fast deployment of new *Intelligent Transportation Systems* (ITS) in a near future. This is especially true as more and more car manufacturers provide interfaces to tightly integrate mobile devices within new vehicles (e.g. Apple CarPlay [3]). It is expected that by 2018, 90% of mobile devices will support the low energy standard [4]. Another advantage is that due to the low energy requirements, BLE services can run in the background on battery powered mobile devices without limiting the usage of other applications. Although this technology has originally been designed for short-range single hop communications, we show that it is possible to send short messages from one device to another up to a maximum distance of 100 m. By the

* Corresponding author. Tel.: +352661881075.
E-mail addresses: walter.bronzi@uni.lu, ervalt@gmail.com (W. Bronzi), raphael.frank@uni.lu (R. Frank), german.castignani@uni.lu (G. Castignani), thomas.engel@uni.lu (T. Engel).

http://dx.doi.org/10.1016/j.adhoc.2015.08.007
1570-8705/

means of experiments, we evaluate different driving scenarios and investigate the impact of wireless channel interference on BLE communications.

We developed a proof-of-concept mobile application that uses *off-the-shelf* smartphones to show how BLE can be used to send data over multiple hops, which significantly increases the scope of the application.

To further proof BLE in its current state for IVC, we show how co-existence between Wi-Fi and BLE looks in practice and investigate how resilient BLE communications are to interferences on the same Radio Frequency (RF) band coming from IEEE 802.11 devices operating at maximum capacity.

During our experiments, we measured performance in terms of delivery ratio and round-trip time for multiple dynamic vehicular scenarios and static interference scenarios. We also identify and discuss several shortcomings that make the current version of BLE not suitable for all kind of applications. We conclude that BLE can, indeed, be used to exchange information between vehicles while driving. This makes BLE an interesting candidate for specific deployment but, given the obtained results, it cannot be considered as a complete replacement for on-board communication interfaces such as DSRC/802.11p.

The remainder of this paper is organised as follows. In Section 2 we provide a literature review. Next, in Section 3, we present an overview of the BLE protocol stack. In Section 4, we describe how BLE can be used in a vehicular context. The vehicular experimental setup and results are discussed in Section 5. A study on BLE robustness to interferences can be found in Section 6. In Section 7 we conclude the paper and provide directions for future work.

2. Related work

Bluetooth Low Energy (BLE) has been standardised by the Bluetooth *Special Interest Group* (SIG) under the Bluetooth 4.0 specification [5]. Gomez et al. [6] provided a concise overview of the BLE protocol stack and investigated the impact of several critical parameters on its performance. They identified that there exists a trade-off between energy consumption and network performance that depends on several configuration parameters.

As BLE is mainly used for low power communications and applications, several studies have analysed the energy footprint of BLE and compared it to other technologies. The work of Siekkinen et al. [7] investigated the energy consumption of BLE and compared the results with ZigBee 802.15.4. They showed that, indeed, BLE is very energy efficient even considering overhead introduced by additional layers such as IPv6.

Other studies have focused on the security aspects of the BLE protocol. Ryan [8] presented different techniques to eavesdrop on BLE communications. They demonstrated an attack against the key exchange protocol, which compromised the encryption.

Most works evaluate new applications that would benefit from the BLE technology. As an example, Lin et al. [9] proposed a novel low cost blood pressure monitoring system that relies on a BLE link between a smartphone and a blood pressure monitor to retrieve accurate readings. Andersson [10] proposed and evaluated a smartphone application that automatically unlocks a door when a user approaches. By evaluating different scenarios, they show that BLE is a suitable technology for this kind of applications.

Not a lot of research has focused on using BLE for vehicular applications. Recently, Kandhalu et al. [11], proposed a BLE protocol modification that would make it suitable for in-vehicle wireless communications. They enhanced the performance of BLE to guarantee a worst-case latency required by automotive systems. Similarly, Lin et al. [12] evaluated the impact of in-vehicle interference caused by the simultaneous use of different ISM band wireless technologies. They concluded that BLE is the most resilient technology when it comes to interference.

In our previous work [13], which this is an extension of, we investigated if the BLE technology would be suitable for IVC. Our preliminary results showed that BLE can be used to transfer small data packets over distances of up to 100 m. In this work we will provide a more detailed study on the characteristics of BLE under different mobility scenarios. Further, we present a proof-of-concept application that allows sending packets over multiple hops using BLE. To the best of our knowledge, this is the first paper that provides a detailed study on how BLE could be used for IVC.

Additional research has examined the co-existence of Wi-Fi and Bluetooth Classic (BC) [14,15] and other few works introduced BLE in the equation [16,17]. BLE hardware is able to reuse existing BC coexistence features such as passive interference avoidance schemes (e.g. adaptive frequency hopping (AFH)[18]). Moreover, BLE channels have a different spacing compared to BC's (2 MHz for BLE and 1 MHz for BC) and are of two kinds: data and advertising channels. Advertising channels are specifically chosen to be in the least congested zone of the 2.4 GHz band.

3. BLE overview

Bluetooth Low Energy, marketed as Bluetooth Smart, initially introduced by Nokia in 2006 under the name Wibree, was included into the Bluetooth Version 4.0 Core Specification in 2010 [19].

BLE was developed as a single-hop communication technology with a multitude of different applications in mind; healthcare, sport and fitness, consumer electronics, smart homes, security and proximity sensing.

Given the widespread availability of Bluetooth technology it is fair to assume BLE success based on implementation similarities and different markets penetration already present today. The next release of BLE (version 4.1) specifies IPv6 connectivity allowing potential Internet access to all BLE enabled devices, making this technology even more attractive [20]. Some of the BLE features are:

- 1 Mbps Data Rate (RF modulation symbol rate).
- 128 bit AES CCM Security.
- Ultra Low Power Consumption (around 1 μA when sleeping and $<$ 20 mA maximum consumption).
- Low Latency (6 ms from non-connected state).

The BLE protocol stack (see Fig. 1), similar to Bluetooth Classic (BC) is divided into two main parts; the Controller and the Host.

In the Controller we can find the Physical Layer (PHY) and the Link Layer (LL), in the Host the Logical Link Control and Adaptation Protocol (L2CAP), the Attribute Protocol (ATT), the

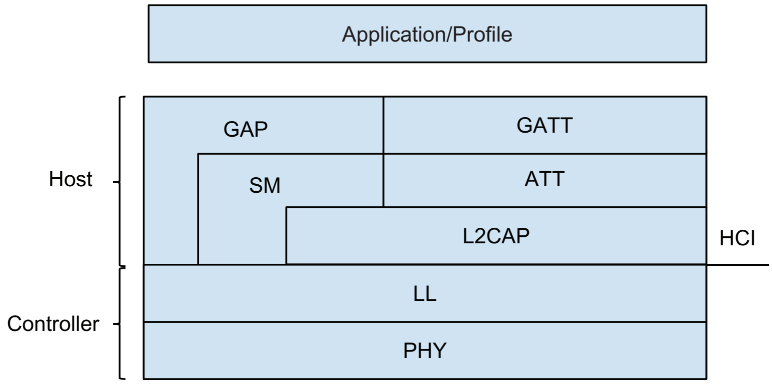


Fig. 1. BLE protocol stack.

Generic Attribute Profile (GATT), the Security Manager (SM) and the Generic Access Profile (GAP). The Host-Controller Interface (HCI) processes communications between the Controller and the Host.

The *Physical Layer* takes care of handling incoming and outgoing bits. BLE defines 40 Radio Frequency (RF) channels, three dedicated to advertising (37–39) and 37 data channels. Only one of these data channels is selected for transmitting data using an adaptive frequency hopping (AFH) mechanism to handle wireless propagation issues and interferences.

The *Link Layer* is the medium to establish bidirectional communication between two devices. Two roles are defined in this layer once a connection is established between two devices: Master and Slave. The master (or *central Device*) is usually a device that scans for other devices (smartphone, tablet or laptop). The slave (or *peripheral device*) are devices advertising data (fitness tracker or smartwatch). Slave devices are in sleep mode by default and wake up periodically in order to be discovered and to potentially pair to Master devices.

The Link Layer provides error and flow control. All data packets will have a 24 bit Cyclic Redundancy Check (CRC) code to provide bit error detection, a supervision timeout is in place to detect connection losses due to exceeding range or radio interferences and a stop-and-wait packet flow is used to ensure no packet is lost by acknowledging sequential packet numbers (more details on error recovery are provided in Section 5.3). The *L2CAP protocol* used in BLE is an optimised version of Bluetooth Classic. The main goal of this layer is to encapsulate data channels for the upper layers providing traffic management.

The *ATT protocol* defines the communication between the Master and Slave. The master (central device) is here referred as *Client* and the slave (peripheral device) as *Server*. The Server device advertises data by maintaining a set of attributes. An attribute contains data managed by the GATT. The ATT protocol sets up communication between a server's attribute and a client via a dedicated L2CAP channel.

The *GATT profile* establishes a framework for discovering the data stored in the attribute protocol and exchanging characteristics between devices. Each peripheral will have a set of attributes storing services and characteristics where every characteristic contains a value and a number of properties (see Fig. 2).

As an example, if we have a heart rate monitor acting as peripheral, its GATT Server will contain a *Heart Rate Sensor* service with a *Heart Rate* characteristic. This characteristic will then include the reading of the sensor as value and specify properties for the value such as *Read Only*.

4. BLE for vehicular communications

Bluetooth Low Energy was not developed with vehicular applications in mind. Nevertheless a piconet between one or more vehicles can be engineered using off-the-shelf smartphones fixed to a car's dashboard. We can imagine that in a near future, where entertainment systems in cars will provide a stronger bound between one's smartphone and car (e.g. Apple CarPlay [3]), the BLE interface from the smartphone or from the car itself can be used for fast and reliable communication to neighbour vehicles.

In this paper we aim to show that, at this moment, an exchange of information between multiple moving hops is reliably achievable up to a certain distance between each hop.

Furthermore, different scenarios were put in place to gradually show how connection and link performance is affected by the cars distance and speed. Experiments on the road were conducted for single-hop scenarios and a proof-of-concept for multi-hop communications was developed and tested in our lab. In the following subsections we will describe our single and multi-hop scenarios more in detail.

4.1. Single-hop

For the proposed single-hop scenarios we developed a mobile application (see Section 5a) and used two identical smartphones (iPhone 5S) with the same OS version (iOS 7.1). For all the experiments we fixed both phones inside the vehicle in a phone holder.

The application implements both BLE central and peripheral modes in the same class and handles role switching automatically. Each smartphone will start as central device and will wake up manually its peripheral role via a button on the device itself. Note that once a device switches to peripheral it will still maintain a central role in a different queue allowing the device to discover other peripherals around even whilst broadcasting.

Please cite this article as: W. Bronzi et al., Bluetooth Low Energy performance and robustness analysis for Inter-Vehicular Communications, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.08.007

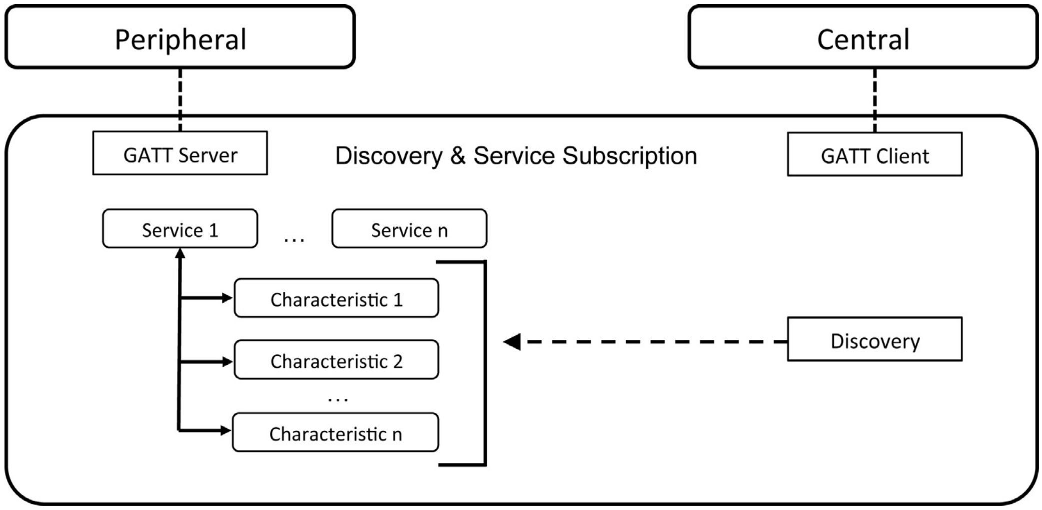


Fig. 2. Generic Attribute Profile (GATT) framework.

We defined three different scenarios to thoroughly benchmark BLE for IVC:

- Constant distance & speed
- Constant speed
- City driving

The first scenario, *constant distance & speed* consisted of two vehicles driving on the highway at constant speed (80 km/h) whilst increasing the distance by steps of 10 m from 20 to 80 m. The second scenario, *constant speed* was conducted with one vehicle parked on the side of the road whilst the second vehicle was driving on the same road at constant speed. The last scenario, *City Driving* consisted of a normal drive through the city of Luxembourg allowing changes in speed, distance and considering local traffic (e.g. other cars between sender and receiver).

4.2. Multi-hop

The multi-hop scenario was conducted in our lab to give a *proof-of-concept* demonstrating its feasibility in a real-life driving scenario.

The application used required a few additions in the implementation such as *Event-Based Role Switching* and the ability to trigger special messages and to handle them accordingly on the receiving device.

4.2.1. Event-based role switching

As shown in Fig. 3 for this application each device will start in central mode and will activate its peripheral role manually on the device itself or upon receiving a message to rebroadcast. In our case the first vehicle on the left will manually trigger a message that will be received by all cars in range. The car in the middle will rebroadcast the messages to the third car (out of range from the first) and will have for a certain period of time both peripheral and central roles active at the same time. Once the middle smartphone receives the first message as a central device it will activate its peripheral manager and rebroadcast messages whilst still receiving new ones from the first car.

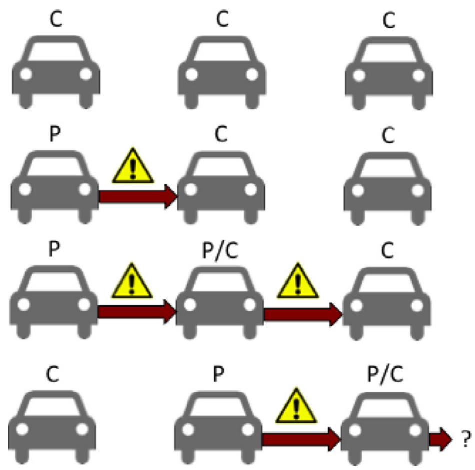


Fig. 3. Event-based role switching (P = peripheral mode, C = central mode).

5. Performance evaluation

For each scenario described in Section 4 we analysed different data taken from our mobile application logs regarding delivery ratio, round-trip time, distance between the vehicles and packet rate as well as end-to-end delay for our multi-hop experiment.

The role of the mobile application is crucial for the performance evaluation since data connection and messages exchanged are directly logged on the devices used for all scenarios. These logs are then extracted after each test run and combined to examine the scenario results.

The application was developed for iOS devices (version 7.1) since it is currently possible to develop code for both central and peripheral modes. For other platforms such as Android, the option of making the phone act as a peripheral device is still unavailable (latest version while writing the paper, Android KitKat 4.4).

Although energy consumption is not a relevant factor in our study it's important to underline BLE's performance regarding this aspect. As the detailed study by Siekkinen et al.

[7] shows in Section 4, the total energy consumption for BLE devices in non-connected states will vary depending on different communication parameters. These parameters, such as connection interval, slave latency and connection supervision timeout, can be optimised for a vehicular scenario to further reduce the power drain. As devices will remain in a connected state only for a short amount of time, few milliseconds depending from the application, we assume the consumption in this state negligible. Since for our experiments an analysis and optimisation of the power usage was not required, we didn't investigate further its impact on the devices available to us.

5.1. Mobile application

As mentioned, the mobile application was developed for iOS devices. The Core Bluetooth framework from the iOS SDK not only gives the possibility to implement both central and peripheral BLE modes but also provides a simple and flexible platform to interact with BLE functionalities. Depending if the user is testing a single or multi-hop scenario, the application will load different parameters.

Testing single-hop scenarios was conducted outdoors hence requiring the iOS Location Manager to take advantage of the device GPS for tracking position and speed. The application for these tests was deployed on two identical iPhones model 5S fixed each on the dashboard of one car. One device was always kept in peripheral mode, having two characteristics (one readable and one writeable) and constantly updating the first one every 100 ms (10 Hz) after successfully pairing with a central device. We choose a beaconing frequency of 10 Hz which meets the requirement of most ITS applications [21].

The readable characteristic contains the peripheral's GPS location, speed and a sequential ID. When the central reads this information it calculates the distance between the two devices using the latitude and longitude coordinates provided from the peripheral. This distance and the relative speed are then displayed on the screen of the receiving device, which allows better controlling of the experiments.

The central device also stores constant readings of the receiving signal strength (RSSI) of the peripheral's messages as well as coordinates from both devices. After the peripheral updates the readable characteristic with a message, the central device reads it and acknowledges the packet by writing its ID into the peripheral's second writeable characteristic.

For multi-hop testing, our application behaves slightly differently. Compared to our single-hop scenarios, where we only need timestamps on the peripheral device, we now need to synchronise internal clocks on all devices regardless of the role. Using the single-hop scenario we can calculate the round-trip time using the peripheral's characteristics by computing the difference between the timestamp of the sent message and the timestamp of the acknowledgement (when the central writes on the second characteristic). For the multi-hop scenario we choose not to have such a characteristic since we are interested in the end-to-end delay which is more relevant for ITS applications.

We synchronised timestamps on all devices by programmatically updating the internal clocks of the iPhones using a local NTP server. Using such a method, connection and disconnection events as well as sent and received messaged are accurately recorded with a precision in the order of 10 ms on all devices.

For this proof-of-concept our three identical iPhones were programmed with different RSSI thresholds and connection settings in order to make sure that at a distance of 0.5 m between each other the first phone would not see the third one and vice versa. As we will see later the second phone will have to switch roles after a certain time to make sure to *rebroadcast* messages coming from the first device.

5.2. Single-hop scenarios

For the performance evaluation of our single-hop scenarios we take into account the logs extracted from our mobile application. Different relevant information is considered for each of our tests:

5.2.1. Constant distance & speed

This scenario was our first step forward from our previous static experiment [22]. Each device is fixed inside a vehicle using a phone holder and both vehicles are driving at constant speed on a stretch of highway. After some trials at different speeds (up to 120 km/h) we found that 80 km/h was not only the safest speed according to the highway code (typical stopping distance comparison: [23]) but also the most appropriate considering the traffic conditions during the day of the tests. Furthermore, the speed at which the vehicles are moving does not impact the quality of the communication link, as distance does, provided that the relative speed is maintained at 0 km/h.

We established some rules for this particular scenario in order to minimise human errors:

- Vehicles speed was fixed at 80 km/h via cruise control and adjusted to match with the help of GPS speeds and relative distance.
- Distance between the cars was constant for the entire duration of each distance step run.
- No other vehicle was to come in between the two cars, disrupting line-of-sight.

The central device, always the car following, is in charge of maintaining a constant distance since its mobile application will have feedback on relative speed and space between the two cars by using the latitude and longitude transmitted by the car in front. The distance between the cars was gradually increased from 20 to 80 m by steps of 10 m. For every step we initiated a communication for at least 180 s. The results of this scenario can be seen in Fig. 4.

We can see that up to 50 m the two devices are able to achieve 100% delivery rate. From this point on the pairing and communication between the two devices is to be considered unreliable as the delivery rate drastically drops to around 40% at 80 m distance. This is due to how the BLE Link Layer handles information regarding connection supervision and which data channels are to be used in case of a reconnection delay due to a change of the interference pattern.

5.2.2. Constant speed

For the purpose of our second scenario we take a stretch of road of about 300 m, one vehicle will be parked on the side

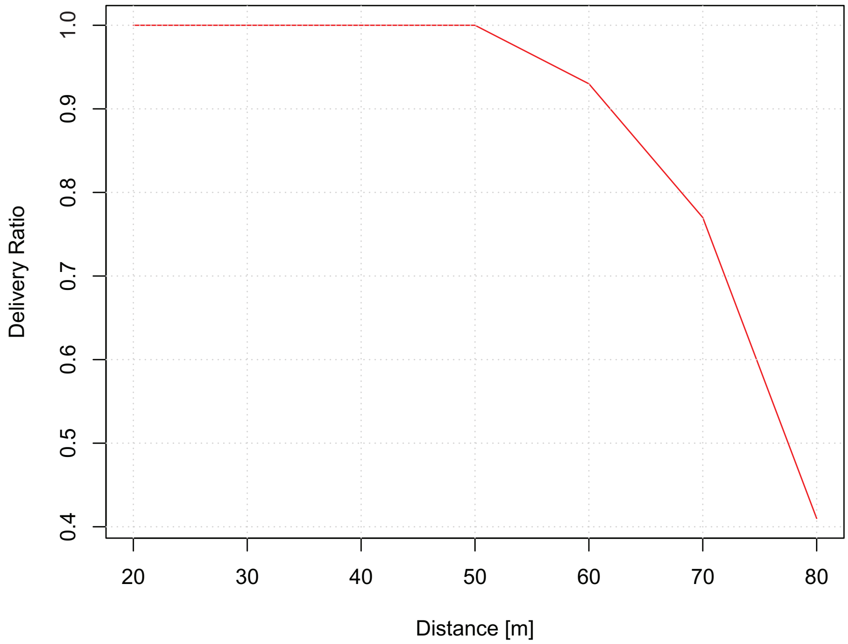


Fig. 4. Delivery ratio at constant speed for an increasing distance.

at the middle mark whilst the second vehicle will pass by at constant speed starting from 20 km/h and repeating with steps of 10 km/h till 60 km/h.

Fig. 5 shows how the pairing and message exchange performs at different speeds. When the car is driving by at 20 km/h, the pairing starts at 110 m with an overall connection time of nearly 50 s before going out or range. This leaves enough time for the peripheral device to transmit 464 packets.

With increasing speed, the number of packets acknowledged and the connected time reduces significantly. At 30 km/h the connection lasts for 19.8 s for 179 packets, continuously reducing till the 60 km/h test where the connects lasted 9.5 s and 85 packets where transmitted.

5.2.3. City driving

This last single-hop scenario does not present driving constraints and is meant to test BLE performance in an urban scenario. For this test, both vehicles drove on a busy road in Luxembourg city continuously varying distance and speed and sometimes having other traffic in-between.

Fig. 6 (a) shows the round-trip time (RTT) distribution and trend line (polynomial regression of degree 3) for different distances.

Depending on different environmental radio interferences impacting the signal propagation, variations in latency are present at any given speed. This graph shows a stable exchange up to 50–60 m with some messages acknowledged also in the range upwards the 90 m.

Fig. 6 (b) shows received signal strength (RSSI) with trend line (also degree 3) for different distances. This reading greatly varies from hardware to hardware. Even whilst having identical phones, it proved to be a challenge to have consistent values at times. Regardless, taking into account interferences with the signal propagation, the overall fit clearly shows a degrading signal strength up to 50–60 m, which confirms the results from our RTT calculations.

Fig. 7 shows the packet rate (10 Hz) and distance between the vehicles over time for the whole duration of the scenario. We can see here how the BLE protocol behaves in a vehicular environment. Almost no messages are lost up to 60 m.

With the distance increasing beyond the 70 m mark, at around 300 s, we notice how the signal quickly degrades up to a point where it is completely lost and the packet rate falls to zero. Communication automatically resumes after the devices are again in range and re-pair.

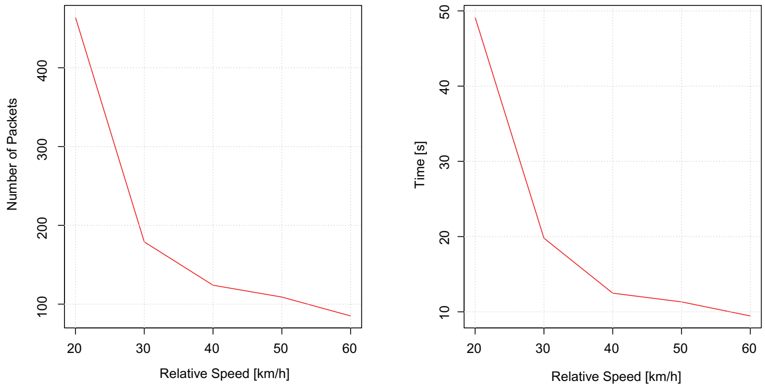
We can also note in this graph how the packet rate fluctuates at times (e.g. between 100 and 200 s) showing how a message is retransmitted after a loss. This behaviour is better understood by looking at how the BLE Link Layer handles retransmissions [6]. Connections at the Link Layer are based on a stop-and-wait flow control mechanism regulated by cumulative acknowledgements providing the protocol with error recovery. Each data channel packet header contains a sequence number (SN) and a Next Expected Sequence Number (NESN). If a packet is received by a device, the NESN of its next packet is increased, serving as negative acknowledgement of the current packet. This behaviour allows a BLE link to provide a robust communication when obstacles (other cars) are to interfere with the line of sight between the two vehicles.

5.3. Multi-hop proof-of-concept

To evaluate our multi-hop scenario we named our three smartphones device A–C (see Fig. 8).

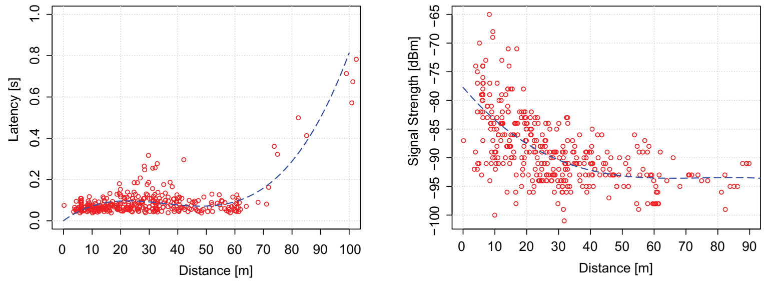
The purpose of this test is to measure how much time an event-triggered message from device A would need to reach device C.

We initiate the communication by triggering (in our case with a button) a message simulating, as an example, a hard



(a) Packets Received **(b)** Connection Time

Fig. 5. Packet delivery and connection time for an increasing relative speed.



(a) Round-Trip Time **(b)** Signal Strength

Fig. 6. City scenario results.

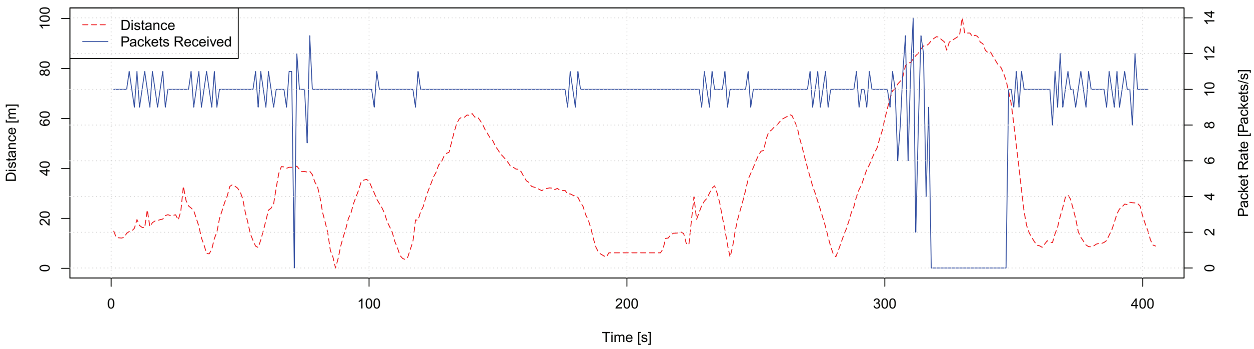


Fig. 7. Distance between peripheral and central vs. packet rate.

braking event. After switching to peripheral mode, device A searches nearby central devices. After pairing, in our case, to device B, device A starts updating a readable characteristic with a sequential ID 100 times at the same frequency as the single-hop experiments (10 Hz). Upon reception of the first message, device B switches itself automatically to peripheral mode and starts looking for new neighbours. Whilst it searches for other central Devices, device B will still keep receiving the rest of the messages from device A. Once device C pairs with device B, it will start forwarding the 100 messages initially sent by device A.

Fig. 3 shows a graphical representation of the proposed scenario.

The performance of this scenario is shown in Fig. 8. At $t = 0$ ms we start advertising with device A in peripheral mode. Device B immediately discovers it ($t = 39$ ms), performs pairing ($t = 95$ ms) and receives the first message ($t = 1370$ ms).

This delay between pairing and reception of the first message is determined by how the central Device discovers and subscribes to the peripheral's characteristics according to the vendor specific implementation of BLE within the Core Bluetooth framework. The pairing process takes around 1.2 s and will need to be performed twice. Once from device A to B and again from B to C. This means that from the instant we trigger the event on the first device it will take roughly 1.37 s for

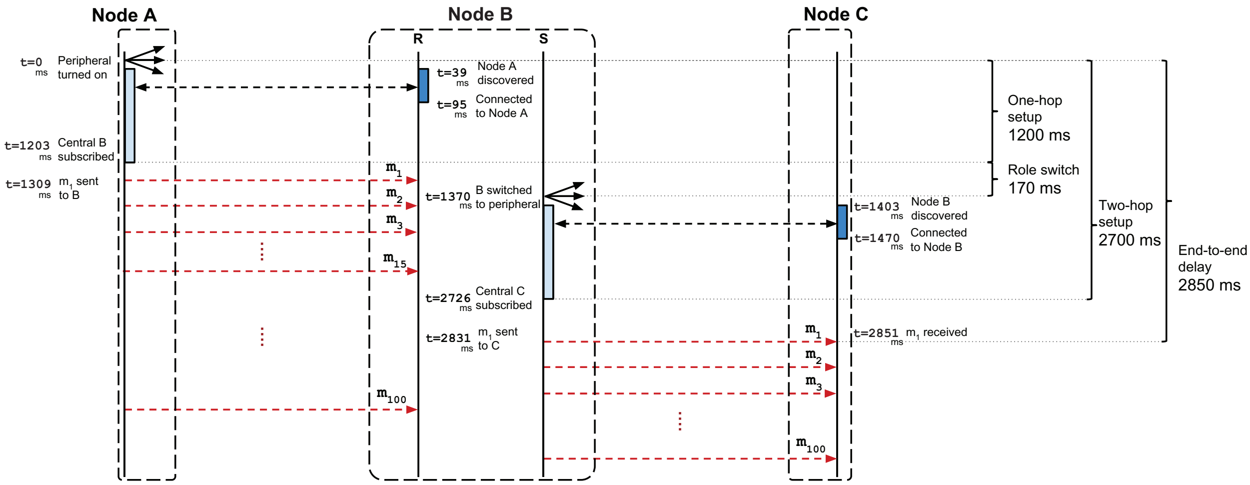


Fig. 8. BLE multi-hop message exchange graph.

device B to acknowledge this message and turn its peripheral manager on.

End-to-end, from the moment we trigger the event to the reception of the first message by device C, it takes around 2.85 s for the two pairing events and the messages to start arriving.

Nodes that are in peripheral mode switch back to central only after the 100 messages are sent to all paired devices.

All three BLE devices are able to handle multiple communication attempts by keeping a local table of nearby neighbours to avoid redundant reconnections. If two simultaneous attempts of connection from two different peripheral devices are received at the same time by a central device then it will have a probabilistic chance of connecting to either based on their duty cycle (probability that the master finds the slave during one of its advertisements [7]). Once a communication link is established multiple connections can coexist thanks to the BLE Frequency Hopping Spread Spectrum (FHSS) and how the protocol handles interferences (More in Section 6).

Communication via BLE can be achieved also without pairing [6] by advertising data through the three advertising channels. Unfortunately at this moment there is no way of doing so on a smartphone. The closest technology for this is Apple's iBeacon [24], which uses advertising channels to transfer information but does not provide the tools to insert a data payload to the messages. Tracking the IDs of sent and received messages is vital for our application to know which message to rebroadcast to a given destination, hence making iBeacon not a reasonable choice.

These results show that BLE, using bidirectional data communication that requires pairing, is not an optimal solution for delay sensitive safety applications such as signalling a sudden emergency brake or a danger on the road ahead. The end-to-end delay is too long, mainly due to the pairing, to apply this technology at high speeds on a highway. Each pairing hop will require, with the current hardware, roughly 1.2 s making its use unpractical. The pairing process can be optimised by reworking Apple's implementation and studying connection specific timeouts in the characteristics discovery phase.

What BLE provides is a relatively fast, low power and very reliable solution for IVC in dense traffic situations or when the pairing delay is not relevant (e.g. non-safety applications).

6. Robustness analysis

As previously demonstrated, the current version of BLE might mainly be applied to low speed vehicular scenarios (e.g. city intersections, urban environments). We took as a challenge to study its behaviour in a congested RF spectrum by reproducing in our laboratory an interference testbed. Previous work of Giordano et al. [25] shows how, in highly populated areas, we can easily detect more than 200 Access Points (APs) mainly occupying the non-overlapping channels 1,6 and 11. It is thus interesting to study further how a similar environment could affect BLE communications. The interference testbed is composed of 6 Raspberry Pi, 3 of which in AP mode assigned to channels 1, 6 and 11, by doing so we drastically reduce the amount of BLE data channels available to the protocol while still leaving unperturbed the advertising frequencies (Channels 37–39). Amongst other things, we are able to observe how the association time between the two mobile phones varies with the different scenarios.

For the purpose of benchmarking the behaviour of BLE in an environment with a sufficient amount of interference we build a relatively compact (2 m × 2 m) testbed.

As shown in Fig. 9, the setup is composed of 6 Raspberry Pis, three of which acting as access points and the other three as clients. For the first three, we installed *hostapd* and *dnsmasq* to configure the access points in mode IEEE 802.11 g and assigned them different channels (1,6,11). To generate a sufficient amount of traffic to perturb communications we utilised *iperf* from the clients to the APs to create a UDP maximum throughput stream (≈35 Mbps).

For the BLE communication the mobile devices where running modified software from our previous experiment with the purpose of logging packet exchange during a time frame of 1 min. The mobile phones where physically placed

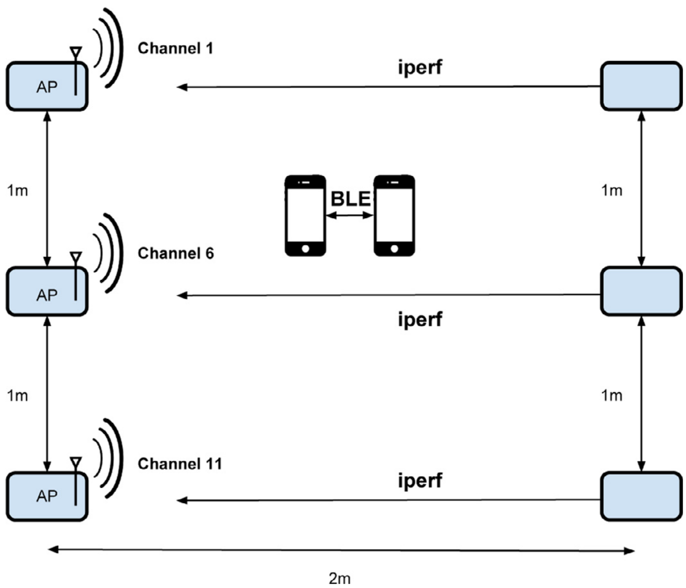


Fig. 9. Raspberry PIs interferences testbed.

in the middle of the testbed to keep the scenario relevant to our purpose and maximise interferences.

We optimised to our best effort the test environment to ensure no other human or environmental interference would spoil the results.

6.1. BLE interferences results

To maximise as much as possible the throughput of the BLE communication we established a constant packet size of 158 Bytes (3 for the header and 155 for the payload) at a rate of 10 Hz with which we can achieve the best reliability and stability given our hardware (≈12.6 Kbps). The packet size is the current maximum size that is achievable with our combination of hardware and software (iPhone6 - iOS8.1).

We defined four different interference scenarios:

- No access point
- 1 access point with iperf traffic (Channel 1)
- 2 access points with iperf traffic (Channel 1,6)
- 3 access points with iperf traffic (Channel 1,6,11)

In Table 1 we find the results of a 1 min BLE communication for each scenario. Within the experimented time frame we also include the association time required for the two devices before exchanging packets.

We notice that, even though the BLE advertising channels are theoretically not congested, the initial association time takes longer in scenarios with higher interference. With no AP we recorded an average association time of 1.5 s which reflect our previous driving scenarios results where we had minimal interference. This behaviour needs to be tested further but can be interpreted as adjacent channel interference [26] as well as environmental reflections affecting the advertising channels.

We also notice that, even though the packet delivery rate is always 100%, the amount of sent and received packet decreases because of the longer association time.

By further investigating the behaviour of RTT in Fig. 10, with the help of a local polynomial regression fitting function, we observe how the setup is seemingly unaffected by interferences coming from only one access point, but by adding a second and particularly a third source of frequency conflict the first batch of packets will be initially delayed and retransmitted later on. In these circumstances we can assume an occupied data channel was picked for communication after the association, justifying the initial higher delay. The chances of landing on a free channel are lower for our 3 APs scenario (only 9 out of 37 channels are free). As soon as the packet exchange starts, for every BLE connection interval (≈30 ms for iPhones) a new better channel will be selected by using the AFH technique. The hopping sequence will repeat until all missing packets are retransmitted and an optimal link is found. For this reason in Table 1 we also included *mean* and *standard deviation* excluding packets until a stable data channel is reached (first ≈ 2.5 s for 0 and 1 AP scenarios, ≈ 5 s for 2 APs and ≈ 10 s for 3 APs).

We conclude from this data that BLE, once a stable configuration is reached, is very resilient to interference.

6.2. Wi-Fi and interferences

To compare how a Wi-Fi setup would perform in a similar environment, we also put two IEEE 802.11g enabled devices through our interference testbed and analysed the results. To replicates a scenario comparable to our BLE tests we implemented a UDP client/server ad-hoc communication between two laptops using sockets. Packet size and data rate where also set similarly to BLE (158 Bytes and 10 Hz) as for the channel it was manually set to Channel 6.

Results of running two different scenarios with minimum and maximum interference (no AP and 3 APs) can be seen in Table 2 and Fig. 11. We notice the disparity between results with and without interference compared to BLE. Other than an increase of RTT there is also noticeable packet loss. These

Please cite this article as: W. Bronzi et al., Bluetooth Low Energy performance and robustness analysis for Inter-Vehicular Communications, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.08.007

Table 1
BLE interferences results

Scenario	Association time (s)	Sent/received packets (delivery ratio)	StDev RTT (s)	StDev RTT (s)[a]	Mean RTT (s)	Mean RTT (s)[a]
0 AP	1.56998	594 (100%)	0.03445	**0.03412**	0.10959	**0.10911**
1 AP	1.71192	592 (100%)	0.03539	**0.03079**	0.10894	**0.10613**
2 APs	2.39517	586 (100%)	0.03924	**0.03368**	0.11014	**0.10663**
3 APs	3.98211	570 (100%)	0.24149	**0.03849**	0.18194	**0.11121**

[a] Excludes packets until optimal data channel is found.

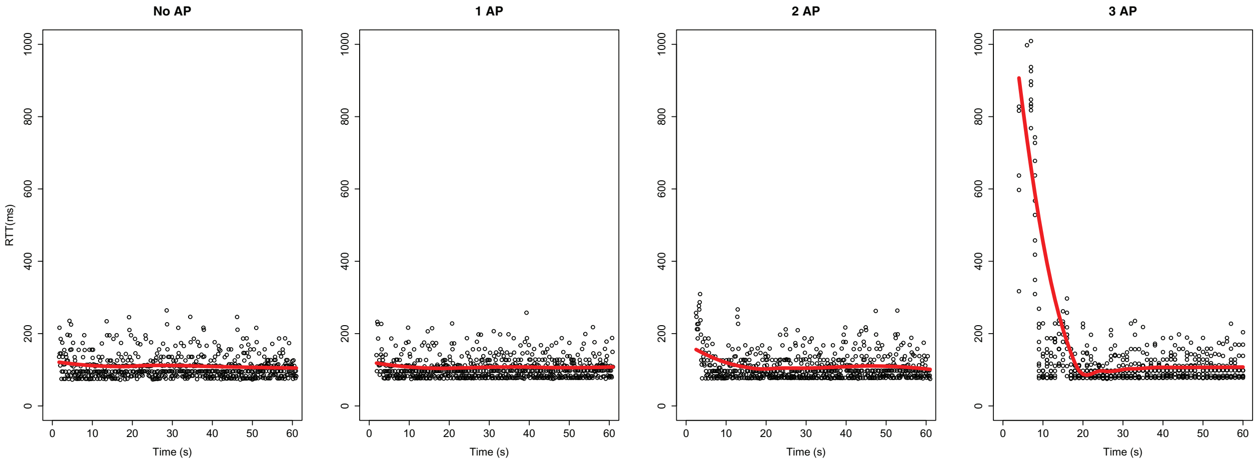


Fig. 10. BLE round trip time.

Table 2
Wi-Fi interferences results.

Scenario	Sent/received packets (delivery ratio)	Standard deviation RTT (s)	Mean RTT (s)
0 AP	600 (100%)	**0.00102**	**0.002212**
3 APs	551 (91.8%)	**0.20778**	**0.108245**

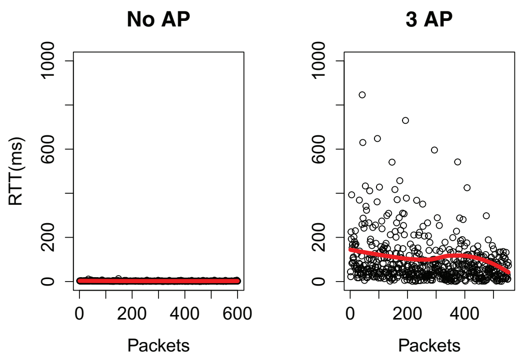


Fig. 11. Wi-Fi round trip time.

tests indicate that although BLE has the disadvantage of having an unoptimised association procedure, protocol characteristics such as the channels width and the use of AFH provide enough interference resistance to leave the packet delay unaffected, after an initial adaptation, even with a partially crowded spectrum.

7. Conclusion and future work

In this paper we studied the potential of *Bluetooth Low Energy* (BLE) for *Inter-Vehicular Communications* (IVC). We described a mobile application specifically developed to work for hardware already on the market that allows for multi-hop communications between moving vehicles.

We measured performance in terms of delivery ratio and round-trip time for multiple single-hop scenarios and end-to-end delay for a multi-hop proof-of-concept application. Furthermore, we analysed the impact on interferences on such technology by examining its coexistence capabilities with other IEEE 802.11g devices on a crowded 2.4 GHz radio spectrum. We have shown how BLE is more resilient than Wi-Fi due to its properties, has more or less the same range and its only weak spot (for low data applications) lies in the time required for two devices to associate.

BLE provides a fast, low power and very reliable solution to IVC in dense traffic situations or as a safe fail solution on top of other technologies (DSRC/IEEE 802.11p). Possible applications include all scenarios where delay sensitive data is not a requirement (e.g. intersection, city traffic management).

As future work, we will continue to investigate how to reduce association delays between devices or even simply broadcasting data using advertising channels to completely eliminate them. By doing so, we would improve BLE likeliness to be used for sensitive IVC applications. Further studies on the 2.4 GHz ISM frequency band can be facilitated by the use of USRPs (Universal Software Radio Peripheral) which can provide continuous frequency coverage throughout the whole spectrum. We will also follow very closely BLE's

Please cite this article as: W. Bronzi et al., Bluetooth Low Energy performance and robustness analysis for Inter-Vehicular Communications, Ad Hoc Networks (2015), http://dx.doi.org/10.1016/j.adhoc.2015.08.007

evolution, as in Bluetooth 4.1, where BLE is said to become IPv6 compatible and able to direct traffic through the Internet along the lines of Wi-Fi. It will be interesting to see how this will affect communications between Bluetooth enabled devices and hopefully make for a more advanced platform for vehicular communications.

Acknowledgement

The authors would like to thank Telindus Luxembourg for their support and their industrial insight on the topic.

References

[1] M. Kracheel, W. Bronzi, A wearable revolution: is the smartwatch the next small big thing? IT ONE Mag. 2014 7 (2014) 52. (December).
[2] O. Altintas, F. Dressler, H. Hartenstein, O.K. Tonguz, Inter-Vehicular Communication – Quo Vadis (Dagstuhl Seminar 13392), Dagstuhl Rep. 3 (9) (2013) 190–213.
[3] Apple Inc., Apple CarPlay – The best iPhone experience on four wheels, URL: https://www.apple.com/ios/carplay/ (accessed February 2015).
[4] Bluetooth Special Interest Group, Mobile Telephony Market, URL: http://www.bluetooth.com/Pages/Mobile-Telephony-Market.aspx (accessed January 2015).
[5] Bluetooth Special Interest Group, Bluetooth Specification Version 4.0, URL: https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=229737 (accessed December 2014).
[6] C. Gomez, J. Oller, J. Paradells, Overview and evaluation of bluetooth low energy: an emerging low-power wireless technology, Sensors 12 (9) (2012) 11734–11753, doi:10.3390/s120911734.
[7] M. Siekkinen, M. Hiienkari, J. Nurminen, J. Nieminen, How low energy is bluetooth low energy? comparative measurements with zigbee/802.15.4, in: Proceedings of the Wireless Communications and Networking Conference Workshops (WCNCW), 2012 IEEE, 2012, pp. 232–237, doi:10.1109/WCNCW.2012.6215496.
[8] M. Ryan, Bluetooth: with low energy comes low security, in: Proceedings of the Presented as part of the Seventh USENIX Workshop on Offensive Technologies, USENIX, Berkeley, CA, 2013. URL: https://www.usenix.org/conference/woot13/workshop-program/presentation/Ryan.
[9] Z.-M. Lin, C.-H. Chang, N.-K. Chou, Y.-H. Lin, Bluetooth low energy (BLE) based blood pressure monitoring system, in: Proceedings of the International Conference on Intelligent Green Building and Smart Grid (IGBSG), 2014, 2014, pp. 1–4, doi:10.1109/IGBSG.2014.6835225.
[10] T. Andersson, Bluetooth low energy and smartphones for proximity-based automatic door locks (2014).
[11] A. Kandhalu, A. Xhafa, S. Hosur, Towards bounded-latency bluetooth low energy for in-vehicle network cable replacement, in: Proceedings of the International Conference on Connected Vehicles and Expo (ICCVE), 2013, 2013, pp. 635–640, doi:10.1109/ICCVE.2013.6799869.
[12] J.-R. Lin, T. Talty, O. Tonguz, An empirical performance study of intra-vehicular wireless sensor networks under wifi and bluetooth interference, in: Proceedings of the Global Communications Conference (GLOBECOM), 2013 IEEE, 2013, pp. 581–586, doi:10.1109/GLOCOM.2013.6831134.
[13] W. Bronzi, R. Frank, G. Castignani, T. Engel, Bluetooth low energy for inter-vehicular communications, in: Proceedings of the Vehicular Networking Conference (VNC), 2014 IEEE, IEEE, 2014, pp. 215–221.
[14] Sue White, Wi-Fi and Bluetooth coexistence, URL: http://www.ecnmag.com/articles/2012/03/wi-fi-and-bluetooth-coexistence (accessed February 2015).
[15] Jeff Paynter, Can Bluetooth and 802.11b/g/n Wi-Fi Devices Coexist?, URL: http://intelligenthospitaltoday.com/can-bluetooth-and-802-11bgn-wi-fi-devices-coexist/ (accessed February 2015).
[16] Phil Smith, Comparing Low-Power Wireless Technologies, URL: http://www.digikey.com/en/articles/techzone/2011/aug/comparing-low-power-wireless-technologies (accessed February 2015).
[17] Raajit Lall, Ca se bouscule dans le spectre 2,4 GHz !, URL: http://www.ni.com/newsletter/51685/fr/ (accessed February 2015).
[18] N. Golmie, O. Rebala, N. Chevrollier, Bluetooth adaptive frequency hopping and scheduling, in: Proceedings of the 2003 IEEE Conference on Military Communications - Volume II, in: MILCOM'03, IEEE Computer Society, Washington, DC, USA, 2003, pp. 1138–1142.
[19] Wikipedia, Bluetooth Low Energy, URL: http://en.wikipedia.org/wiki/Bluetooth_low_energy (accessed November 2014).
[20] 6LoWPAN Working Group, IPv6 over BLE, URL: http://tools.ietf.org/html/draft-ietf-6lowpan-btle-12 (accessed November 2014).
[21] J. Ploeg, B. Scheepers, E. van Nunen, N. van de Wouw, H. Nijmeijer, Design and experimental evaluation of cooperative adaptive cruise control, in: Proceedings of the 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2011, 2011, pp. 260–265, doi:10.1109/ITSC.2011.6082981.
[22] R. Frank, W. Bronzi, G. Castignani, T. Engel, Bluetooth low energy: an alternative technology for vanet applications, in: Proceedings of the 11th Annual Conference on Wireless On-demand Network Systems and Services (WONS), 2014, 2014, pp. 104–107, doi:10.1109/WONS.2014.6814729.
[23] Department for Transport Driver and Vehicle Standards Agency, United Kingdom, The Highway Code: typical stopping distances, URL: https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/312249/the-highway-code-typical-stopping-distances.pdf (accessed June 2015).
[24] Apple Inc., Apple iBeacon – Indoor proximity system, URL: https://developer.apple.com/ibeacon/ (accessed December 2014).
[25] E. Giordano, R. Frank, G. Pau, M. Gerla, Corner: a radio propagation model for vanets in urban scenarios, Proc. IEEE 99 (7) (2011) 1280–1294, doi:10.1109/JPROC.2011.2138110.
[26] E.G. Villegas, E. Lopez-Aguilera, R. Vidal, J. Paradells, Effect of adjacent-channel interference in IEEE 802.11 WLANs, in: Proceedings of the 2nd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2007. CrownCom 2007, 2007, pp. 118–125, doi:10.1109/CROWNCOM.2007.4549783.

Walter Bronzi is a Ph.D. student working at the Interdisciplinary Center of Security Reliability and Trust (SnT) for the VehicularLab in Luxembourg. Prior to beginning the Ph.D. program, Walter successfully completed the Bachelor and Master of Computer Sciences at the University of Luxembourg. Walters main interests lie in vehicular networks, short-range radio technologies and mobile app development.

Raphael Frank is a Research Scientist at the Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg and head of the VehicularLab. Frank received his Ph.D. in Computer Science from the University of Luxembourg in 2010. He is currently involved in several European and national research projects. His research interests include wireless networks and mobile computing.

German Castignani is research associate at the Interdisciplinary Centre for Security, Reliability and Trust (SnT). He received his Computer Sciences Engineer degree at the University of Buenos Aires (FIUBA, Argentina) and a Ph.D. in Computer Sciences at Institut Mines-Telecom, Telecom Bretagne (Rennes, France). His research interests include vehicular and wireless networks, mobility management, Intelligent Transportation Systems (ITS) and advanced driving assistance systems.

Thomas Engel is Professor for Computer Networks and Telecommunications and Vice-Director of the Interdisciplinary Center for Security, Reliability and Trust (SnT) at the University of Luxembourg. Thomas Engel is member of the Information and Communication Security Panel ICS of NATO and Civil High-Level Expert for Electronic Communications (representing Europe) of NATO CEP/CCPC. Since 2009 he is chairman of the National IPv6 Council Luxembourg and appointed Vice-Director of the newly created Interdisciplinary Center for Security, Reliability and Trust (SnT) at the University of Luxembourg.