MASTER

Robustness of bluetooth low energy in in-vehicle networks

Winkel, T.T.L.

Award date:
2016

Link to publication
ROBUSTNESS OF BLUETOOTH LOW ENERGY IN IN-VEHICLE NETWORKS – AN EXPERIMENTAL STUDY

Master Thesis

Tim Winkel
Student ID: 0828206
Email: Winkel.ttl@gmail.com

Supervisor NXP    Dr. ir. Bart Vermeulen
Supervisor University    Dr. Majid Nabi Najafabadi

JANUARY 22, 2016
NXP SEMICONDUCTORS
Eindhoven
Public version
Abstract

The number of electronic devices in vehicles is increasing. Often these devices have to communicate with each other and the electronic control units (ECUs). Currently, these communications are facilitated through a wired in-vehicle network in the form of a wiring loom. The wiring loom introduces significant disadvantages such as weight, costs and the inability to reach all locations in the vehicle. These disadvantages will become more severe as the number of electronic devices in the vehicle increases.

A possible solution is the development of a wireless in-vehicle network. However, there is no consensus on which communication protocol is optimal. In this thesis we investigate the performance of Bluetooth Low Energy (BLE) in IVNs. We then compare the performance of IEEE 802.15.4(e) TSCH and BLE in the in-vehicle environment. The IEEE 802.15.4(e) TSCH are provided by [1].

Bluetooth Low Energy is a standard created and maintained by the Bluetooth SIG. It is designed to provide the same range as Bluetooth Classic with lower power consumption. Key features of Bluetooth Low Energy are: A star topology network with a schedule of dedicated timeslots, a variable TX power, channel hopping, and standardized application descriptions through the use of profiles.

In this thesis we describe the implementation of a Bluetooth Low Energy test platform with the ability to measure the Message Error Rate (MER), Packet Error Rate (PER) and End to End Latency. The MER represents the performance of the application layer. The PER represents the performance of the physical layer and the latency represents the time required to pass a message between the application layers of the sender and the receiver. We investigate these metrics for different schedules, different packet sizes, different TX powers, the presence of interference from Bluetooth Classic, and the accuracy and predictability of the test setup.

We observe that all investigated node locations have good best case performance. When we move away from the best case, we discover that there is a clear distinction between nodes placed in the passenger cabin and nodes placed outside of the passenger cabin. Nodes inside the passenger cabin have performance similar to the best case under all observed test cases because they are in the same enclosure as the central node. Nodes outside of the passenger cabin suffer from performance deterioration when the TX power is reduced. Increasing the connection interval or the message size has no significant effect on the performance of the BLE links investigated. We noted that Bluetooth Classic interference affects the performance of BLE, however the increase in packet error rate is always smaller than 5%. Finally we observed the accuracy and predictability of the test setup, we noted that the performance of a BLE link is predictable when the RSSI of the received packets is greater than -75 dBm. We conclude that there is no reason to stop research into BLE based IVNs.

In this thesis we attempt to compare BLE to IEEE 802.15.4(e) TSCH. We can only compare general trends and the regions in which links are because the accuracy and predictability of TSCH is never investigated. We observed that both protocols have similar performance in the best case. TSCH has better performance than BLE in the uplink test where TSCH uses broadcasts and BLE uses unicast. But BLE has better performance in the downlink test where both protocols use unicast. We observe that TSCH has better best case latencies than BLE but suffers from harsher penalties when the link quality deteriorates. Overall there is insufficient data to complete the comparison at this point.

Keywords: Bluetooth Low Energy, In-Vehicle Network, IEEE 802.15.4(e) TSCH.
Acknowledgements

This part of my thesis is (chronologically) the last part that I am writing. During my time at NXP there have been moments where I was really worried whether or not I was going to succeed in performing the tasks I wanted to perform. In the end I am very happy that I managed to complete this thesis. This thesis has become larger than I anticipated. However, I feel that is necessary to include the information that I included to ensure that anyone performing future work has a comprehensive and as complete as possible source of information to fall back upon.

There are many people that I would like to thank for their help and support. I would like to start by thanking the TUE, Avans Hogeschool and NXP and their employees for providing me with the knowledge, skills and the opportunity to work on this thesis. My special thanks go to my supervisors: Bart Vermeulen and Majid Nabi for supporting me and occasionally pushing me into the right direction. I would also like to thank the other committee members: Kees Goossens and Pieter Cuijpers for their time, feedback and support. I would like to thank Lars van Meurs and Lulu Chan for providing me with help and feedback. Special thanks go to Gino Knubben of NXP for providing me with his car for the IVN experiments.

I thank my friends from the archery club E.S.H. Da Vinci and my friends from the TUE for supporting me during this period.

I would like to take a moment to thank my father, Nick Winkel, who always supports and believes in me. You are someone I look up to and who I aspire to one day be like.

I also thank the rest of my family: My mother: Monique Winkel, my sisters: Anika and Carmen Winkel, my uncles: Ron Winkel and Raimond Winkel, Aunts: Jaqueline Winkel and Miranda Winkel, my Cousins: Mike, Kevin and Kristin Winkel and last but not least my grandparents: Theo and Ria Winkel and Leo and Eli Dentener. I would not have gotten this far without you, you all mean a lot to me.
Contents

Abstract .......................................................................................................................................................... I
Acknowledgements ....................................................................................................................................... II
Contents ....................................................................................................................................................... III
List of Figures ................................................................................................................................................ V
List of Tables ................................................................................................................................................ VI
List of Equations .......................................................................................................................................... VII
Acronyms ................................................................................................................................................... VIII

1 Introduction .......................................................................................................................................... 1
  1.1 Motivation ..................................................................................................................................... 2
  1.2 Problem Statement ....................................................................................................................... 2
  1.3 Thesis Overview ............................................................................................................................ 3

2 Background ........................................................................................................................................... 4
  2.1 In-Vehicle Electronics and Networks ............................................................................................ 4
  2.2 Bluetooth Low Energy ................................................................................................................... 6
  2.3 The QN9020 SOC ......................................................................................................................... 14

3 Prior Work ........................................................................................................................................... 15
  3.1 Wireless Networks ...................................................................................................................... 15
  3.2 Wireless In-Vehicle networks ...................................................................................................... 17
  3.3 IEEE 802.15.4e TSCH ................................................................................................................... 18
  3.4 Bluetooth Low Energy ................................................................................................................. 21
  3.5 Contributions .............................................................................................................................. 22

4 Test Requirements and Approach ...................................................................................................... 23
  4.1 Test Setup Requirements ............................................................................................................ 23
  4.2 Post Processing Requirements .................................................................................................... 24
  4.3 Test Setup ................................................................................................................................... 24
  4.4 Test Phases .................................................................................................................................. 25
  4.5 Performance Metrics .................................................................................................................... 26
  4.6 Interference ................................................................................................................................ 30
  4.7 Post Processing ........................................................................................................................... 30
  4.8 Expected Results ........................................................................................................................... 31

5 Test Implementation ........................................................................................................................... 34
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Test Messages and State Machine</td>
<td>34</td>
</tr>
<tr>
<td>5.2</td>
<td>Test Setup Interfaces</td>
<td>38</td>
</tr>
<tr>
<td>5.3</td>
<td>Logging</td>
<td>39</td>
</tr>
<tr>
<td>5.4</td>
<td>Other Design Choices</td>
<td>39</td>
</tr>
<tr>
<td>5.5</td>
<td>Interference Generation</td>
<td>40</td>
</tr>
<tr>
<td>5.6</td>
<td>Post Processing</td>
<td>40</td>
</tr>
<tr>
<td>5.7</td>
<td>Software Tools</td>
<td>42</td>
</tr>
<tr>
<td>6</td>
<td>Bluetooth Low Energy Experiments and Results</td>
<td>43</td>
</tr>
<tr>
<td>6.1</td>
<td>Table-top Experiments</td>
<td>43</td>
</tr>
<tr>
<td>6.2</td>
<td>Vehicle Location</td>
<td>43</td>
</tr>
<tr>
<td>6.3</td>
<td>Test Parameters</td>
<td>46</td>
</tr>
<tr>
<td>6.4</td>
<td>IVN Experiments Execution</td>
<td>47</td>
</tr>
<tr>
<td>6.5</td>
<td>IVN Results</td>
<td>48</td>
</tr>
<tr>
<td>7</td>
<td>Comparison with IEEE 802.15.4(e) TSCH</td>
<td>49</td>
</tr>
<tr>
<td>7.1</td>
<td>Comparability</td>
<td>49</td>
</tr>
<tr>
<td>7.2</td>
<td>Best Case Comparison</td>
<td>50</td>
</tr>
<tr>
<td>7.3</td>
<td>TX Power Comparison</td>
<td>50</td>
</tr>
<tr>
<td>8</td>
<td>Conclusions and Future Work</td>
<td>51</td>
</tr>
<tr>
<td>8.1</td>
<td>Conclusions: Bluetooth Low Energy</td>
<td>51</td>
</tr>
<tr>
<td>8.2</td>
<td>Conclusions: Bluetooth Low Energy and IEEE 802.15.4(e) TSCH</td>
<td>51</td>
</tr>
<tr>
<td>8.3</td>
<td>Future Work</td>
<td>52</td>
</tr>
<tr>
<td></td>
<td>Bibliography</td>
<td>53</td>
</tr>
<tr>
<td>A</td>
<td>Appendix: Experiment Pictures</td>
<td>57</td>
</tr>
<tr>
<td>B</td>
<td>Appendix: Measuring the Number of Transmissions</td>
<td>58</td>
</tr>
<tr>
<td>C</td>
<td>Appendix: Accuracy of the Test Setup</td>
<td>59</td>
</tr>
</tbody>
</table>
List of Figures

Figure 2.1: Star topology of a piconet........................................................................................................................................6
Figure 2.2: Client server model.............................................................................................................................................6
Figure 2.3: The Architecture of the Bluetooth Low Energy protocol [21] .................................................................7
Figure 2.4: The state machine of the BLE Link Layer .........................................................................................................8
Figure 2.5: Schematic view of a series of Connection Events.........................................................................................9
Figure 2.6: BLE Link layer packet format [19] ..................................................................................................................10
Figure 2.7: BLE channels [23] ..........................................................................................................................................12
Figure 2.8: BLE channels and Wi-Fi channels [23] ...........................................................................................................12
Figure 2.9: Block Diagram of the channel selection algorithm used by BLE ...............................................................13
Figure 4.1: Network topology of the BLE test ..................................................................................................................25
Figure 4.2: End to end latency example with one retransmission (image is not to scale) ........................................27
Figure 4.3: Post processing steps........................................................................................................................................31
Figure 5.1: Flow chart of the central node .......................................................................................................................35
Figure 5.2: The structure of the universal data format ..................................................................................................41
Figure 6.1: The environment in which the vehicle will be placed ..................................................................................44
Figure 6.2: The Location of the parking lot (the red box) 3 ........................................................................................44
Figure 6.3: IEEE 802.15.4 Wi-Fi environment ..............................................................................................................45
Figure 6.4: BLE Wi-Fi environment ..................................................................................................................................45
List of Tables

Table 2.1: Most popular wired automotive technologies. Information provided by [12] ......................... 4
Table 2.2: Classification of real-time Systems .......................................................................................... 5
Table 2.3: BLE network parameters and their IEEE 802.15.4 counterparts ........................................... 14
Table 3.1: Effects of weather on the 2.4 GHz Band ............................................................................... 17
Table 3.2: Automotive networking requirements as defined by Segers ................................................ 17
Table 3.3: Experimental conditions for the tests involving IEEE 802.15.4(e) TSCH ............................... 19
Table 3.4: Experimental results of IEEE 802.15.4(e) TSCH .................................................................. 20
Table 4.1: Period definitions for the end to end latency calculation ....................................................... 28
Table 4.2: RQ1: The expected effects of changing the connection interval ............................................ 32
Table 4.3: RQ2: The expected effects of changing the packet size ......................................................... 32
Table 4.4: RQ3: The expected effects of changing the TX power ............................................................ 33
Table 5.1: Reactions of the central node to a received message ................................................................. 37
Table 5.2: Reactions of the end nodes to messages received ................................................................. 38
Table 5.4: List of tooling used to create the experimental source code .................................................... 42
Table 6.1: Static Test Parameters .......................................................................................................... 46
Table 6.2: Parameter values for the connection interval, payload size and the TX power ......................... 46
Table 6.3: Experiment runs performed for BLE ...................................................................................... 47
Table 6.4: The locations of the nodes in the IVN setup ......................................................................... 47
List of Equations

Equation 2.1: BLE center frequencies [22]................................................................. 11
Equation 2.2: The first stage of the channel selection algorithm [22] ...................... 13
Equation 4.1: The chance of message loss due to Bluetooth Classic interference ........... 33
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLE</td>
<td>Bluetooth Low Energy</td>
</tr>
<tr>
<td>ECU</td>
<td>Electronic Control Unit</td>
</tr>
<tr>
<td>IVN</td>
<td>In-vehicle network</td>
</tr>
<tr>
<td>LQE</td>
<td>Link Quality Estimator</td>
</tr>
<tr>
<td>PDU</td>
<td>Protocol Data Unit</td>
</tr>
<tr>
<td>SOC</td>
<td>System-on-Chip</td>
</tr>
<tr>
<td>MSE</td>
<td>Mean Square Error</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Rate</td>
</tr>
<tr>
<td>CAN</td>
<td>Controller Area Network</td>
</tr>
<tr>
<td>ETX</td>
<td>Expected Number of Transmissions</td>
</tr>
<tr>
<td>IVE</td>
<td>In-vehicle Environment</td>
</tr>
<tr>
<td>LIN</td>
<td>Local Interconnect network</td>
</tr>
<tr>
<td>MOST</td>
<td>Media Oriented Systems Transport</td>
</tr>
<tr>
<td>PRR</td>
<td>Packet Reception Ratio</td>
</tr>
<tr>
<td>RSSI</td>
<td>Received Signal Strength Indicator</td>
</tr>
<tr>
<td>PER</td>
<td>Packet Error Ratio</td>
</tr>
<tr>
<td>MER</td>
<td>Message Error Ratio</td>
</tr>
<tr>
<td>LAT</td>
<td>End to End Latency</td>
</tr>
</tbody>
</table>
Chapter 1.

1 Introduction

The number of electronic devices in the world increases every day. There is an exponential increase of the number of electronic devices in almost all environments on the planet. This trend will most likely continue in the near future. Many of these devices require to communicate with other devices in order to function. As electronics get more complex, the need to communicate increases. The automotive environment is no exception to this trend. During the last decade, the number of electronic devices in vehicles has increased to more than 150 devices [2]. The data from these devices is exchanged using an In-vehicle Network (IVN).

Examples of currently used IVN technologies are the Local Area Interconnect [3] (LIN), Controller Area Network [4] (CAN), FlexRay [5] and Media Oriented Systems Transport [6] (MOST). The properties of these technologies vary widely, however they all require the use of wired IVNs. This means that all of these technologies suffer from drawbacks and advantages of a wired IVN.

A modern car can have several kilometers of wiring used by its IVNs. This reveals the main drawback of a wired IVN: its complexity. This complexity results in an addition to vehicle weight, which adversely affects the performance and efficiency of the vehicle, increased maintenance costs, increased material and component costs, increased installation time, limited scalability and the requirement to place wired connections in hard-to-reach places. As in-vehicle electronics get more complex, the problems associated with these issues increase.

One solution for some of these drawbacks is the use of a wireless communication in IVNs. There are already a limited number of in-vehicle applications that use a wireless IVN to transport data. One example of a wireless IVN is a tire pressure management system examined in [7]. However, current wireless IVNs often use a proprietary protocol which could lead to problems unforeseen by the IVNs designers. An example of these unforeseen problems is the security of currently deployed wireless tire pressure monitoring devices [7].

The combination of these issues reveals the need for a one-size-fits all wireless solution for the automotive domain. However there is no consensus on which protocol should be used in this solution. The thesis contributes by examining the Bluetooth Low Energy protocol in an IVN context and comparing these results to a similar experiment performed using IEEE 802.15.4 [1].
Chapter 1. Introduction

1.1 Motivation
Currently, there are only a very limited number of automotive applications that make use of wireless communications. These applications often use wireless solutions because the wiring loom cannot reach the place where the electronic devices are placed. There is no consensus on how wireless solutions should be used in automotive environments. Therefore, these applications often end up using proprietary protocols, which can be expensive to design and less reliable than established wireless protocols. One of the reasons that wireless IVNs do not use well-known wireless protocols is that there is little to no information on the performance and robustness of these protocols in the automotive environments. This, combined with the unique requirements of the automotive environments, presents many challenges for designers.

Wireless IVNs have been a research topic within NXP for several years now where prior work [8] lists several promising wireless protocols: IEEE 802.15.4, Ultra Wideband (UWB), Bluetooth Classic, Bluetooth Low Energy (BLE), Near-Field Communications (NFC), Wi-Fi, and Millimeter wave (mmW). Out of these protocols, the IEEE 802.15.4 [9] and IEEE 802.15.4(e) TSCH [1] technology standards have received the most attention. Other research [10] shows that BLE can outperform IEEE 802.15.4 in the in-vehicle environment. However, data regarding a performance comparison between IEEE 802.15.4(e) TSCH and BLE is unavailable.

Other technologies such as UWB, NFC or mmW are relatively unproven or still evolving. Wi-Fi and Bluetooth Classic are known for relatively high power demands. For this reason, current research is focused on a comparison between the performance of IEEE 802.15.4(e) TSCH and BLE.

1.2 Problem Statement
This thesis addresses the following problems:

1. There is little to no information available on the performance of BLE in in-vehicle networks.
2. There is no comparison of the performance of BLE to IEEE 802.15.4(e) TSCH when operating in an in-vehicle environment.

The goal of this thesis is to provide a fair experimental comparison of the performance of IEEE 802.15.4(e) TSCH and BLE. This comparison includes shared features and features unique to each technology, which will be compared by examining the best cases of both protocols. The comparison between Bluetooth Low Energy and IEEE 802.15.4 is done based on measurements of the same performance metrics in an environment that is as similar as possible. The operation of the Bluetooth Low Energy and IEEE 802.15.4 stacks will be similar in many ways but not equal.

We examine the performance of BLE by analyzing the performance of a BLE network under various combinations of network parameters. Each network parameter is examined in its own research question. We present the following research questions with regards to the performance of BLE technology for wireless IVNs:

RQ 1: What is the effect of increasing the connection interval period on the robustness of a BLE link in an IVN?

RQ 2: What is the effect of increasing the payload size on the robustness of a BLE link in an IVN?

RQ 3: What is the effect of increasing the TX power on the robustness of a BLE link in an IVN?
RQ 4: What is the effect of Bluetooth Classic interference on the robustness of a BLE link in an IVN?

It should be noted that we do not measure the throughput because we do not utilize the links to their maximum capacity. We do not utilize the links to their maximum capacity because the use cases for the IVN network do not require fully utilized network links.

The goal of this thesis is reached by taking the following steps:

1. Become familiar with the features of both Bluetooth Low Energy and IEEE 802.15.4(e) TSCH, analyze the robustness tests performed by Deepak Sudhakar [1] on IEEE 802.15.4(e) TSCH and examine other related work.
2. Plan, implement and perform measurements on the robustness of Bluetooth Low Energy and examine the results. The tests have to be performed such that the strengths of Bluetooth Low Energy can be examined even if IEEE 802.15.4(e) TSCH does not support a similar feature. However the results must still be comparable to the results of IEEE 802.15.4(e) TSCH. The comparison will be based on the common metrics. All experiments regarding BLE are to be performed using the QN9020 chip [11].
3. Propose and implement a generic post processing method for all IVN experiments.
4. Compare, analyze and explain the differences between the experimental results of IEEE 802.15.4(e) and Bluetooth Low Energy, lists their respective strengths, provide a recommendation on the technology best suited within an IVN, and list future research topics.

1.3 Thesis Overview
This thesis has the following general structure:

- Chapter 2 provides information related to the background of the experiments described in this thesis. It briefly discusses real time systems in the in-vehicle environment, introduce the Bluetooth Low Energy wireless networking protocol, and introduces the QN9020 SOC.
- Chapter 3 discusses relevant work done by others. It focuses on prior work done within NXP with special attention to the work on IEEE 802.15.4(e) TSCH.
- Chapter 4 introduces the requirements and approach of the test setup.
- Chapter 5 explains the implementation of the test setup.
- Chapter 6 describes the performed experiments and the conditions under which the experiments were performed. It also discusses a set of smaller experiments, which impact the accuracy of the main set of performed experiments.
- Chapter 7 performs the comparison between Bluetooth Low Energy and IEEE 802.15.4 TSCH
- Chapter 8 concludes the thesis and discusses future work.
- Appendix A will contain the pictures taken to document the test setup.
- Appendix B will discuss the methods used to calculate the number of retransmissions per message
- Appendix C will discuss the accuracy related topics.
Chapter 2

2 Background

This chapter presents background information in the field of IVNs. It also introduces the Bluetooth Low Energy protocol and the QN9020 hardware platform used in this thesis.

2.1 In-Vehicle Electronics and Networks

This section presents some information on the background of in-vehicle electronics. It also discusses the wiring loom that presently performs most of the in-vehicle communications and discuss the potential of wireless IVN.

2.1.1 Nature of In-vehicle Electronics

The number of electronic devices in automobiles has increased exponentially in the last decades. Tuohy et al [12] provide an overview of the most used wired IVN technologies. An overview of these technologies can be found in Table 2.1.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Bitrate</th>
<th>Medium</th>
<th>MAC mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIN</td>
<td>19.2 Kbps</td>
<td>Single Wire</td>
<td>Serial</td>
</tr>
<tr>
<td>CAN</td>
<td>1 Mbps</td>
<td>Twisted Pair</td>
<td>CSMA/CR</td>
</tr>
<tr>
<td>FlexRay</td>
<td>20 Mbps</td>
<td>Twisted Pair/Optical Fibre</td>
<td>TDMA</td>
</tr>
<tr>
<td>MOST</td>
<td>150 Mbps</td>
<td>Optical Fibre</td>
<td>TDMA</td>
</tr>
<tr>
<td>LVDS</td>
<td>655 Mbps</td>
<td>Twisted Pair</td>
<td>-</td>
</tr>
</tbody>
</table>

The authors expand this set with an analysis of an Ethernet based application. They conclude that Ethernet is the most likely candidate for future wired IVNs. However the current Ethernet IEEE 802.3 [13] is not suitable for automotive applications since it is not real time. Additional queueing, timing and scheduling will be required in order to guarantee deadlines. Research into these techniques is currently ongoing. An example of ongoing research would be time triggered Ethernet.

Most of the devices in an IVN environment may be categorized as real-time systems. A real time system is a system that does not only have to present the right result, it also has to present this result at the right moment in time [14]. A key concept in the field of real-time systems is the deadline. A real-time system has two deadlines: The best case deadline and the worst case deadline. The real time system has to present the correct result later than the best case deadline and earlier than the worst case deadline under all circumstances in order to have the property of timeliness. The absolute values of the deadlines are application-dependent. Real-time systems are classified according to the consequences of missing their deadlines. This has been done in Table 2.2.
<table>
<thead>
<tr>
<th>Classification</th>
<th>Description</th>
<th>Automotive Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hard Real-Time System</td>
<td>Missing the deadline can result in major damage to the device itself, the user or the environment</td>
<td>Airbag</td>
</tr>
<tr>
<td>Firm Real-Time System</td>
<td>Missing the deadline does not cause danger or damage, but it does make the result unusable.</td>
<td>Car Stereo Audio Output</td>
</tr>
<tr>
<td>Soft Real-Time System</td>
<td>Missing the deadline does not cause danger or damage, but the value (usefulness) of the result will diminish over time.</td>
<td>Controls of the air-conditioning</td>
</tr>
</tbody>
</table>

The automotive environment houses all 3 kinds of real-time systems. Often these systems need to execute on the same platform and share resources (including communication mediums). This presents unique design challenges in the field of automotive applications that require timeliness.

2.1.2 Wired In-Vehicle Networks

The increasing number of electronic devices present in vehicles need to communicate with each other and the central computer of the vehicle. The electronic devices can be placed throughout the vehicle. The communication is facilitated by the IVN. In wired IVNs, an IVN is created through the wiring loom. Which is a collection of cables running throughout the vehicle.

Currently, the wiring loom is one of the heaviest, complicated, difficult to handle, and expensive electronic components in the vehicle. It can add up to 50 kilograms to the vehicles weight [2] and connects more than 150 sensors and switches. The bulk, weight, and complexity of the wiring loom make this component difficult to handle, maintenance expensive, and its weight implies decreased fuel economy.

Examples of in-vehicle applications that currently use wired IVNs are: the windscreen wipers, the car stereo, the window controls mounted in the doors, the door mirror controls, the dials in the dashboard, the air-conditioning, and the engine management system.

2.1.3 Wireless In-Vehicle Networks

The negative properties of the wiring loom make it desirable to look into alternative solutions. A possibility to mitigate these properties is to utilize a wireless solution where we remove a part of the wiring loom and replace it with a wireless connection. The utilization of a wireless IVN has the potential to reduce wiring loom installation time, decrease vehicle weight leading to better fuel economy, reduce material costs and allows the car electronics to be placed in locations difficult or impossible to reach through wired connections. However, the inherent unreliability, lack of security and hazards to privacy [7] of wireless links present new challenges to overcome.

There are already several examples of commercially available applications that utilize a wireless IVN. Examples of existing wireless applications for vehicles are the remote keyless entry system [15] and the wireless tire pressure management system [7]. These applications often use proprietary protocols. This requires increased resources during design since a protocol has to be developed and can cause problems overlooked by the designers of the application which are difficult to correct in a deployed system. There have been experiments and proof of concepts where part of the wiring loom has been replaced by a Bluetooth link [16], a BLE link [17] and an ultra-wideband link [18]. However, none of these applications are ready for commercial use.
The potential for wireless IVNs lies not in the complete replacement of the wiring loom since some of the applications require timing guarantees that will be extremely difficult or impossible to give. The solution would be a hybrid solution in which a part of the wiring loom is replaced by a wireless network. Primary application candidates for a wireless IVN connection should have relatively long absolute deadlines which should give the wireless link enough time to correct any transmission errors that might occur.

Examples of applications that could utilize wireless IVN technology would be soft real time systems such as user interfaces or sensors of relatively slow changing metrics such as temperature and fluid level sensors.

2.2 Bluetooth Low Energy
Bluetooth Low Energy [19], also known as BLE, Bluetooth Smart, and Bluetooth 4.0, is a technology for wireless personal area networks designed and maintained by the Bluetooth Special Interest Group (SIG) of IEEE. It was officially released in 2010. It aims to provide the same communication range as normal (classic) Bluetooth while using less power. The most significant differences BLE and Classic Bluetooth are the BLE has a reduction in data rate, no support for audio, a different channel designation and simplified state machines. BLE, like Bluetooth Classic, operates in the 2.4 GHz ISM band.

2.2.1 Network
BLE operates in piconets. Each piconet has a star topology as illustrated in Figure 2.1. The node in the center of the star is the master (also known as the central) of the piconet. All the other nodes in the piconet are slaves (also known as end nodes).

BLE communication on a single link takes place according to a client server model. In this link, the master assumes the role of client and the slave assumes the role of server. Key to the use of BLE is the concept of a service. A service is defined as an immutable encapsulation of some atomic behavior of a device and is located on the server. A server can run any number of services. A service is exposed to the outside world through the use of one or more attributes. An attribute is defined as an addressed labeled bit of data. These attributes can be accessed by using their address through a BLE link. All interactions between the client and the server take place through the use of attributes. Behavior on the server side can be triggered by changing the values of one or more attributes.

An example of a service is a light switch. Its attribute is be a boolean value representing on and off. By reading the attribute the status of the light can be recovered and by writing it the status can be

---

Robustness Of Bluetooth Low Energy In In-Vehicle Networks – An Experimental Study
controlled. The Bluetooth SIG has defined several services each of which have their own specifications. These specifications define the attributes the service should expose and the behavior the attribute should trigger. Examples of services defined by the Bluetooth SIG are the battery level service, the proximity service and the heart-rate monitor service.

It should be noted that it is illegal for a node (regardless of its role) to participate in more than 1 piconet. The state machines required to maintain these connections have been deemed too complex for the low power environment in which the nodes should operate. Therefore, the current version of BLE does not support multi-hop communications. Application level solutions [20] have been proposed to implement multi-hop communications, to the best of our knowledge, none have matured to the point where it is used in a commercial product.

A key assumption of BLE is the asynchronous availability of resources. The master is expected to be a full function device with a large amount of available resources while the slave can either be a full function or reduced function battery powered device. Therefore, the master is expected to perform the relatively complex and expensive tasks such as connection maintenance and scheduling.

### 2.2.2 Protocol Architecture

The Bluetooth Low Energy protocol consists of several layers. Each layer has its own function. A schematic overview of the architecture of BLE is presented in Figure 2.3.

- **Physical Layer** handles transmission and reception of bits.
- **Link Layer** handles advertising, scanning, creating and maintaining connections. We will elaborate on this Layer in section 2.2.3.
- **Direct Test Mode** is used for factory testing and calibration of the physical layer.
- **Host Controller Interface** is a standardized interface between host and controller.

---

Robustness Of Bluetooth Low Energy In In-Vehicle Networks – An Experimental Study
In this work we are most interested in the Link Layer in the controller module, since it handles most of the connection-related functions such as creating, maintaining, and terminating connections.

2.2.3 Link Layer

One of the major differences between Classic Bluetooth and Bluetooth Low Energy is the reduced complexity of the BLE state machines of the link layer. This state machine has 5 states as illustrated by Figure 2.4.

Every BLE connection has its own state machine. The status of the state machine is used to track the status of the connection between 2 nodes. When a node is in the connection state, it is connected. Data communication can only take place in the connection state. In all other states, there is no link between the nodes. The states a device assumes are dictated by the role it wishes to play in a connection, listed as follows:

- If a device wishes to assume the role of server, then it enters the advertisement state and begins to send advertisement packets. An advertisement packet typically contains information about the device and the services the device supports. It is possible to query a device for
additional information when the device is in the advertisement state. A connection is initiated when another node responds to this node’s advertisements. It should be noted that it is also possible to advertise without the possibility to connect to the advertising node.

- If a device wishes to assume the role of client, then it will first enter the scanning state. In the scanning state it listens for and collects the advertisements of nearby nodes and if necessary request additional information from them. If the scanning reveals a device to which the node wishes to connect, then it enters the initiating state. In the initiating state the device listens for an advertisement from the target node and then send a connection request to this node. If the target node accepts, a connection is initiated.

- If a device does not wish to assume either roles, it assumes the standby state. No communications can take place in this state.

A connection is always initiated by the client by responding to an advertisement. The server can never initiate a connection. However the server can allow only one node to connect by using targeted advertising. The server can also advertise for the purpose of service discovery only, in this mode it is not possible to create a connection by reacting to advertisements.

### 2.2.4 Connections

A connection between 2 nodes is set up when a client node responds to an advertisement of a server node. During the connection setup, both nodes negotiate on the connection parameters. These connection parameters include the supervision timeout, the slave latency and the connection interval period.

BLE is optimized for low power consumption. This means that BLE devices are intended to sleep for long periods of time. This period is maximized by the use of connection events. A connection event is the moment the first packet out of a group of data packets is transmitted. All data traffic between connected nodes takes place during connection events. A schematic overview of connection intervals can be found in Figure 2.5.

![Figure 2.5: Schematic view of a series of Connection Events](image)

The start of a connection event is known as the anchor point. The server nodes will synchronize themselves to this point whenever they receive the first packet of the connection event.
Connection events are always started periodically by the master based on the negotiated connection interval. During a connection event the master and slave alternate sending packets. Packet acknowledgement is implied in responding to a message. In the absence of a response, the node resends the previous packet until a response is obtained or until the maximum number of retransmissions is achieved. If a node has no data to send, it responds with an empty packet. A connection event ends when both nodes have transmitted at least one packet in which the “More Data” bit in the access address (see Section 2.2.6) is set to zero.

An added benefit of connection events is that, under normal circumstances, multiple connections will not interfere with each other.

The method of connection events closely resembles that of periodic polling. This would mean that both sides are forced to wake up and transmit an empty packet, even if there is no information to exchange. However, there is a key difference defined by the slave latency. Slave latency defines the number of connection events the slave node is allowed to ignore before it is forced to respond to a connection event started by the master.

A connection can be terminated by sending a disconnect message. A connection will also be terminated automatically once the supervision time has elapsed without successful communications.

2.2.5 Retransmissions
The Bluetooth core specification does not specify the maximum number of retransmissions a message is allowed to have. Instead it specifies how the stack should deal with retransmissions during a single connection interval. The behavior is governed by the following three rules:

- Two consecutive packets received with an invalid CRC will always close the connection event
- The master closes the connection event if it does not receive a reply from the slave
- The slave will always reply to a packet from the master. Even if it has an invalid CRC.

Together these 3 rules specify a maximum of 2 transmissions of the same packet in one connection interval.

2.2.6 Packet Format
The Bluetooth core specification [19] defines two types of BLE Link Layer packets: the advertisement packet and the data packet. An advertisement packet is used when the devices are unconnected while a data packet is used when devices are connected. Both packets have the same structure but their payload is different. The format of a BLE Link Layer packet is illustrated in Figure 2.6.

![Figure 2.6: BLE Link layer packet format](image)

The preamble is used for synchronization between the sender and receiver. The access address contains the address of the target device and link layer specific information. The PDU contains the data and the
Message Identification Code –MIC- (if applicable), and the CRC is used for error correction. The shortest packet has a length of 80 bits and the longest packet has a length of 376 bits.

It should be noted that the Bluetooth Core Specification V4.2 [22] specifies that the maximum PDU size is 257 octets resulting in a maximum packet size of 2120 bits. However this is beyond the scope of this thesis.

2.2.7 Traffic Types
Typically, BLE data traffic can be separated into two types:

- **Event Generated:** Event generated data traffic involves the transmission of a single message informing the receiving node of an event registered by the sending node. An example of this type of traffic is a server node informing the client that a battery level threshold has been reached.
- **Request/Reply:** This data traffic involves the client requesting information from the server. This type of data traffic requires the transmission and reception of two messages. An example of this traffic is the central requesting the battery status of the server node.

It should be noted that BLE data traffic is always unicast. There is no concept of broadcasting for the current version of BLE.

2.2.8 Channels
BLE operates in the 2.4 GHz ISM band. As mentioned in section 2.2.2, BLE uses Frequency Shift Keying (FSK) where a positive frequency deviation from the center frequency of 185 kHz corresponds to a 1 and a negative frequency deviation from the center frequency of 185 kHz corresponds to a 0.

The channel designation of BLE is different from the channel designation of classic Bluetooth. Classic Bluetooth defines a total of 80 channels. Every Bluetooth Classic channel has a bandwidth of 1 MHz.

BLE divides the 2.4 GHz ISM band running from 2.400 GHz to 2.480 GHz into 40 channels. Each channel center frequency is separated from the next channel center frequency by two MHz. The center frequency $f_c$ of each channel is calculated according to Equation 2.1.

$$f_c = 2402 + k \times 2 \text{ MHz} \quad \text{for } k = 0, ..., 39$$

*Equation 2.1: BLE center frequencies [22]*

Out of these 40 channels, 3 channels numbered as channels 37, 38 and 39 (2.400 GHz, 2.426 GHz and 2.480 GHz) are used as dedicated advertisement channels. The other 37 channels (numbered from 0 to 36) are only used for data transmissions. The channels of BLE are illustrated in Figure 2.7. The channels 37, 38 and 39 are dedicated advertisement channels and the other channels are data channels.
Advertisement channels are only used for advertising and the initiation of connections. After the connection parameters have been negotiated the newly connected nodes will switch to other channels. The advertisement channels have been selected such that they don’t overlap with the channels used by Wi-Fi. Therefore the interference in these channels should be minimal. The BLE channels and the overlapping Wi-Fi channels are illustrated in Figure 2.8. When advertising, the advertisement channels are used sequentially starting with channel 37. During normal operations all advertisement channels are used for advertising.

2.2.9 Channel Selection and Adaptive Channel Blacklisting

BLE channels are selected using the channel selection algorithm defined in the Bluetooth Core Specification. This algorithm is illustrated in Figure 2.9.
The channel selection algorithm consists of 2 stages:

1. Select the unmapped channel based on the previous unmapped channel used.
2. If the unmapped channel is unused, remap to another used channel.

The first stage of the algorithm computes a channel number referred to as the unmapped channel based on the last used unmapped channel and the hop increment. The hop increment is defined by the Bluetooth Core specification as a random value in the range of 5 to 16. This is done using Equation 2.2.

\[
\text{unmappedchannel} = (\text{lastunmappedchannel} + \text{hopincrement}) \mod 37
\]

*Equation 2.2: The first stage of the channel selection algorithm [22]*

The algorithm then checks if the unmapped channel is used. If it is used then the algorithm will use the unmapped channel. Otherwise the channel is remapped to one of the used channels.

All data traffic during a single connection event takes place in a single data channel. A new data channel will be selected for the next connection interval.

The method described above only describes how a channel should be selected based on the last used channel. The BLE specification does not describe how channels should be marked as used or unused.

### 2.2.10 Parameters

The BLE stack presents us with a number of modifiable parameters. Some of these parameters have an IEEE 802.15.4 counterpart. These parameters are listed in Table 2.3.
Table 2.3: BLE network parameters and their IEEE 802.15.4 counterparts

<table>
<thead>
<tr>
<th>BLE Parameter</th>
<th>Short Description</th>
<th>IEEE 802.15.4 equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Interval</td>
<td>The time between 2 consecutive connection events.</td>
<td>The schedule used</td>
</tr>
<tr>
<td>Payload Size</td>
<td>The number of bytes per packet containing application level data</td>
<td>Payload Size</td>
</tr>
<tr>
<td>TX Power</td>
<td>The amount of energy transmitted by the transmitter</td>
<td>TX Power</td>
</tr>
<tr>
<td>Channel Map</td>
<td>Determine which data channels are used</td>
<td>Random channel hopping</td>
</tr>
<tr>
<td>Maximum number of retransmissions</td>
<td>The maximum number of retransmissions before a packet is dropped</td>
<td>Maximum number of retransmissions</td>
</tr>
</tbody>
</table>

BLE has several other parameters that lack an IEEE 802.15.4 counterpart:

- Slave latency: The number of connection events a slave is allowed to ignore before a mandatory response.
- Supervision timeout: The amount of time that nodes can be out of contact without formally losing the connection.

Any parameters that are not used during the experiments described in this thesis are fixed to a default value.

2.3 The QN9020 SOC
[This section has been removed for confidentiality reasons]
Chapter 3

3 Prior Work

This chapter presents related work concerning the performance and reliability evaluation of BLE, IEEE 802.15.4, and In-Vehicle networks.

3.1 Wireless Networks

The nature of the operation of a wireless network is well understood. Within the scope of this thesis the concept of Link Quality is of importance. This concept is discussed in this section.

3.1.1 Link Quality

The stability and reliability of a wireless link can be expressed in the Link Quality [24]. The link quality expresses the likelihood of successful communications across a wireless link. The quality of a wireless link can be affected by many factors such as the transmission power, range and presence of interference. One of the major factors in the quality of a link is the distance between the sender and the receiver. The quality of a link can be classified as one out of 3 possible regions [24]:

- **Connected Region**: In the connected region, the link quality is very good. The link is symmetric (both the up and downlink have similar properties) and communications are reliable. Links in this region are stable. Typically the message error rate of a link in the connected region is 10% or lower.

- **Transitional Region**: In the transitional region, the link quality is moderate. There is not symmetric and communications are unreliable. Links in this region are unstable and unpredictable. Typically the message error ratio of a link in the transitional region is between 10 and 90%

- **Disconnected Region**: In the disconnected region, the link quality is extremely poor. There is no symmetry and communications are impossible. Links in this region are unstable. Typically the message error ratio of a link in the transitional region is higher than 90%.

The exact boundary of the regions is affected by the environment, the configuration of the transmitter and the Link Quality Estimator used to estimate the link quality. The quality of a link can be used to predict the robustness of a network and provide an upper bound to the throughput of a link [25].

With regards to the transitional region, we would like to highlight 3 key observations made by Baccour et al [26]:

1) Links with either a very low or high average PRRs are more stable then links with moderate average PRRs.

2) Links in the temporal region tend to experience short burst of 0% PRR and 100% PRR.

3) The variation mentioned in 2 is caused by changed in the environment characteristics.

We observe that the BLE test setup created in this thesis can have similar properties.
3.1.2 Link Quality Estimators

In this thesis, we perform experiments designed to reveal the quality of a wireless link. The quality of a link can be expressed with a Link Quality Estimator (LQE). In general the performance of a LQE can be expressed in 2 terms: Accuracy and adaptability. The accuracy determines how precise the estimated link quality resembles the real quality of the link. The adaptability resembles how fast the LQE will respond to a change in the environment. Typically a more accurate LQE shows less adaptability [27]. LQEs can be provided either by hardware or software. Wireless sensor networks are known to use the following LQEs:

Hardware LQEs:

- **RSSI** [24]: The received Signal Strength Indicator is the relative received signal strength. It is an indication of the power being received by the antenna.
- **Signal to noise Ratio** [25]: The relative difference in power between the Signal and the noise. Defined as: 
  \[ SNR_{dB} = 10 \log_{10} \left( \frac{P_{Signal}}{P_{Noise}} \right) dB. \]

Software LQEs:

- **Packet Reception Ratio** [24]: The number of packets received compared to the number of packets transmitted.
- **Window Mean Exponentially Weighted Moving Average** [27]: The WMEWMA is generated based on the Packet Reception Ratio and presents a smoother representation of that data.
- **Four Bit** [27]: An approximation of the amount of retransmissions required.
- **Expected Number of Retransmissions**. [27]: The expected number of retransmissions, known as ETX. This LQE uses statistics based on the packet reception ratio to identify the expected number of retransmissions for a packet.
- **Requested Number of Packets** [28]: The RNP. This metric is similar to the ETX but it provides different weights to sequential retransmissions then to discrete numbers of retransmissions.

There is currently no consensus on which of these LQEs provides the optimal performance since they are all unreliable in some situations. However PRR and WMEWMA are considered to be more optimal than the others [27]. Cost wise, RNP is considered to be cheaper than ETX and Four Bit [28].

In this thesis we restrict ourselves to the RSSI and the Packet Error Rate (The inverse of the packet reception ratio). We also introduce the message error rate to quantify the performance of BLE on the application layer. For additional information we refer to section 4.5.

3.1.3 Effect of weather on wireless networks

NXP [29] has investigated the effect of weather on the attenuation of signals in 2.4 GHz bands. They report that weather has a negligible influence on the attenuation of the signals because the peak absorption frequency of water is 22.2 GHz, which is far from the frequencies used by the proposed wireless IVNs. They report the following results:
Table 3.1: Effects of weather on the 2.4 GHz Band.

<table>
<thead>
<tr>
<th>Weather phenomenon</th>
<th>Attenuation (dB/Km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Torrential Rain (4 inches/hour)</td>
<td>0.05</td>
</tr>
<tr>
<td>Thick Fog</td>
<td>0.02</td>
</tr>
</tbody>
</table>

Our experiment are conducted outdoors. Therefore, weather conditions can fluctuate between experiments.

3.2 Wireless In-Vehicle networks

Segers [8] examined the applicability of wireless networks for automotive systems. He notes that most automotive applications are examined by their “classic” domain: the powertrain, safety systems, the chassis, the body, telematics and diagnostics. Each of these domains has its own requirements. He argues that this classification is ineffective since applications should be classified based on their restricting requirements. He identifies several requirements: bandwidth, latency, time or event triggered nature, and real time-requirements. He then proposes 5 new application domains based on their networking requirements. These domains are listed in Table 3.2.

Table 3.2: Automotive networking requirements as defined by Segers

<table>
<thead>
<tr>
<th>Domain number</th>
<th>Description</th>
<th>Example application</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Low bandwidth with guaranteed latency</td>
<td>Airbag</td>
</tr>
<tr>
<td>2</td>
<td>Low bandwidth with no strict latency demands</td>
<td>Windscreen wiper</td>
</tr>
<tr>
<td>3</td>
<td>High bandwidth with tight latency demands</td>
<td>Audio/video streaming</td>
</tr>
<tr>
<td>4</td>
<td>High bandwidth with guaranteed latency</td>
<td>Electronic damping control</td>
</tr>
<tr>
<td>5</td>
<td>High bandwidth with no strict latency demands</td>
<td>Fleet management</td>
</tr>
</tbody>
</table>

He then examines the theoretical applicability of ZigBee, near-field communication, Bluetooth, Wi-Fi, ultra wideband and Millimeter wave. The conclusion of his research is that Wi-Fi, ultra wideband and millimeter wave are the most promising candidates since they have the highest theoretical bandwidth. However, there is little to no information about these technologies in a wireless automotive environment.

We note that Segers’ reclassification of the automotive domains might be too coarse, he classifies requirements on the application-level while we should examine the requirements of each individual sensor or actuator. For example, the engine control system would be classified as domain 1 since it controls the valves in the engine which have strict real-time requirements. These connections would be extremely difficult to convert to wireless connections because of the inherent unreliability of wireless links. However, this engine control system has several sensors with significantly less real-time requirements such as oil temperature sensors. We suspect that these oil temperature sensors can probably be classified as domain 2 since the temperature of fluids is known to change relatively slow which could mean that these sensors might be candidates for a wireless connection.

Rouf et al [7] presents an analysis of a wireless IVN currently in use. They reverse engineer the networking aspects of a wireless tire pressure monitoring system. They note that this IVN uses a proprietary protocol which has several vulnerabilities overlooked by the designers. They note that the IVN uses unencrypted communications which means that the IVN is vulnerable to several types of attacks. They also note that the IVN uses static addresses which raises privacy concerns since these addresses can be used to track the whereabouts of the vehicle.
Several studies [30] [31] [32] [33] have examined in-vehicle signal propagation by either experimental measurements or simulations. Although all experiments notice large power losses in some situations, the concept of a wireless IVNs remains feasible. They also conclude that the signal propagation is heavily dependent on the shape of the car and the materials used in its construction. The experimental data suggests that a gap of more than 20 MHz between 2 channels results in “acceptably” uncorrelated behavior between these 2 channels.

3.3 IEEE 802.15.4e TSCH

Eisner [9] performed measurements with an IEEE 802.15.4 setup in an in-vehicle environment. In most cases, she reached a packet error rate of less than 1%. She also notes that the presence of Bluetooth interference can result in a rise of the packet error rate to 10 to 50%. She recommends a similar test of IEEE 802.15.4(e) TSCH.

Work relating to IEEE 802.15.4e TSCH has been performed by Sudhakar [1]. In his work he examines the robustness of IEEE 802.15.4e TSCH in an in-vehicle environment by examining the following research questions:

1. What is the performance of networks metrics of packet error rate ratio, throughput and latency under shared and dedicated time slots?
2. How do the network configurations of transmit power, retransmissions and packet size influence the performance of a TSCH network?
3. What is the effect of channel hopping on the packet error ratio, throughput and latency?
4. Which benefits and bottlenecks will a TSCH network have in an automotive environment?

These research questions have been answered by performing an experimental analysis using NXP JN5168 dongles [34]. The experiments involved the deployment of a TSCH network in a start topology similar to BLEs star topology (See Section 2.2.1) where the central node (the master in the corresponding BLE network) will assume the role of coordinator with a point to point link to all other nodes known as end nodes. The network operates in 2 test configurations:

1. **Data Dissemination**: The coordinator will periodically broadcast test packets to all end nodes.
2. **Data Collection**: the end nodes periodically unicast packets to the coordinator.

Experiment are conducted through the use of test cases. Each test case involves the execution of both test configurations (data dissemination and data collection) under a set of specified settings. Experiments took place under 2 possible conditions:

1. **Static**: Vehicle parked, engine off, doors closed with no passengers.
2. **Dynamic**: Vehicle parked, engine, radio and air conditioning on, Bluetooth interference present.

The network settings (presented in Table 3.3) depend on the research question that is being investigated. It should be noted that the message size (marked by *) is not a separate test case, it represents the size of a packet consisting of all added headers and a payload of the size mentioned in the field above it. Thus a payload size of 12 bytes results in a message size of 48 bytes.
Table 3.3: Experimental conditions for the tests involving IEEE 802.15.4(e) TSCH

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Test Configurations</td>
<td>Data Dissemination</td>
<td>Data Collection</td>
<td>N/A</td>
</tr>
<tr>
<td>2. Test Conditions</td>
<td>Static</td>
<td>Dynamic</td>
<td>N/A</td>
</tr>
<tr>
<td>3. Payload size (bytes)</td>
<td>12</td>
<td>36</td>
<td>84</td>
</tr>
<tr>
<td>Message size* (bytes)</td>
<td>48</td>
<td>72</td>
<td>120</td>
</tr>
<tr>
<td>4. TX power (dBm)</td>
<td>-24</td>
<td>-12</td>
<td>2.5</td>
</tr>
<tr>
<td>5. Retransmissions</td>
<td>On</td>
<td>Off</td>
<td>N/A</td>
</tr>
<tr>
<td>6. Channel hopping</td>
<td>On</td>
<td>Off</td>
<td>N/A</td>
</tr>
</tbody>
</table>

This work performed measurements which were ultimately processed into the following three performance metrics:

- The packet error ratio
- The throughput
- The end to end Latency

This prior work culminates in the following conclusions:

1. Using IEEE 802.15.4(e) TSCH’s dedicated time slots outperforms using shared time slot in all scenarios resulting in better packet error ratio, superior throughput and a bounded latency because there is no CSMA based system for the dedicated time slots.
2. If the link is in the connected region then changing test cases 3, 5 and 6 does not produce any significant effects unless a change in TX power causes the link to leave the connected region.
3. If the link is in the transitional region:
   a. Increasing transmit power can cause the link to become connected.
   b. Increasing the packet size lowers link quality significantly.
   c. Enabling retransmissions reduces the packet error ratio but does not improve throughput, and increases average latency.
4. Random channel hopping can stabilize an unstable link but it can also destabilize a stable link. Adaptive channel hopping might be required. Similarly adaptive transmission power control can yield good results as well.
5. Certain node locations cause the link to be in the unstable region.

Table 3.4 provides a more detailed list of the experiments and their result.

We note that most of these conclusions make sense, however one significant conclusion is missing: the accuracy of the IEEE 802.15.4(e) TSCH test setup is never investigated. Therefore, the validity of these results is questionable. The lack of accuracy figures also complicates the comparison between IEEE 802.15.4(e) TSCH and BLE.
Table 3.4: Experimental results of IEEE 802.15.4(e) TSCH

(This section has been removed for confidentiality reasons)
3.4 Bluetooth Low Energy

Gomez et al [35] performs measurements on both the energy consumption and the throughput of the CC2540 radio chip. Current measurements are performed using a power analyzer. The data collected using this device is then used to calculate the average current used by the device. They notice that the average current is highly dependent on the connection interval. A connection interval of 7.5 ms (the shortest allowed by the Bluetooth standard) requires an average current of about 10 mA while the longest allowed connection interval of 4 seconds results in an average current of less than 0.1 mA. They use this data combined with an (ideal) battery of 220 mAh to predict device lifetimes of 0 to 14 years. They state that based on the minimum connection interval and bit error rate, the maximum theoretical application level throughput of a BLE device is equal to 236.7 kbps. However, in a practical application the achieved throughput is 58.48 kbps or less. This is caused by invalid CRC checks, limited application, messaging capacity, processing delays and limited number of transmissions during a single connection event.

Siekkinen et al [36] compares the power consumption of ZigBee (CC2530) with the power consumption of BLE (CC2540) using a power monitor. He notes that BLE can reach a transmission rate of more than 500 Kbyte/j while ZigBee reaches 300 Kbyte/j. The authors also perform robustness experiments in which they conclude that BLE reaches a packet error rate of about 40% in severe Wi-Fi interference. It should be noted that the used BLE stack does not deploy adaptive channel blacklisting. Therefore, these results can be pessimistic when compared to the results a BLE stack with adaptive channel blacklisting. The authors also report a transmission rate of 240 Kbyte/j for Wi-Fi, but the origin of these numbers is unclear.

Robustness experiments of BLE and IEEE 802.15.4 in an IVN are performed by Lin et al [10]. They deploy an IVN based on the CC2540 (BLE) and the CC2240 (ZigBee) within a vehicle and measure the robustness of both protocols with and without interference. They report a goodput of 100% for ZigBee and 98% for BLE without any interference. When interference is applied, the goodput of ZigBee can degrade by up to 28% while the goodput of BLE degrades by up to 27%. In general BLE appears to outperform ZigBee in the field of reliability. However, it should be noted that we do not know if the BLE stack used in this experiment uses adaptive channel blacklisting. The measurement results with regards to Wi-Fi interference appear to suggest that they do not use adaptive channel blacklisting. The transmissions of a single Wi-Fi channel cause a packet error rate of approximately 26%. This corresponds to the loss of one message for every four transmitted messages. One Wi-Fi channel overlaps with approximately one fourth of all BLE channels. Therefore, we conclude that these channels are apparently used just as often as the other channels, which is what adaptive channel blacklisting should prevent. Other evidence for this theory is other earlier work mentioned in this section [36] which uses the same hardware (Texas Instruments CC2540 radio chip) and states that it does not support adaptive channel hopping. If it turns out that this data is really obtained without adaptive channel blacklisting then these results can be pessimistic since the adaptive channel blacklisting should restrict the use of the channels overlapping with the Wi-Fi channel in use. This should in turn result in higher goodput with interference.

Barge et al [37] perform an experimental analysis on the effect of interference generated by a microwave. They vary both the distance and the power settings of the microwave. They report a strong correlation between the distance to the microwave and the packet losses sustained by the connection. They only note weak correlation between the power settings of the microwave and the packet loss. The
authors reason that this is the case since any interference from the microwave will always overpower the data transmissions across the wireless link. We observe that many potential sources of interference in the in-vehicle environment, such as Bluetooth Classic [19] or Wi-Fi [38], have greater transmission powers than BLE. Therefore we expect to see similar results for the in-vehicle environment.

3.5 Contributions

This section explains the relevance of the work in this thesis, discuss the limitations of the prior work and discuss how this work will reduce some of these limitations.

To the best of our knowledge, current research suffers from the following limitations:

1. Most BLE experiments are performed using the TI CC2540 radio chip. This means that the results can be biased by the implementation of that hardware and its developed protocol stack. We expand this by providing data obtained using the NXP QN9020 radio chip.
2. There is no comparison between the performance of IEEE 802.15.4(e) TSCH and BLE either in or out of the in-vehicle environment. Thus far comparisons have been limited to IEEE 802.15.4 and BLE. We will provide this comparison.
3. There is no known work examining the effect of packet size and transmission power on the quality of a BLE link. We provide this examination.
4. There is only one study that investigates the performance of BLE in an in-vehicle environment. We expand on this study with different hardware setup and additional data points

Summarizing: This thesis seeks to:

- Provide experimental data based on the NXP QN9020 radio chip.
- Increase the number of locations involved in the test of in-vehicle networks.
- Examine the effect of packet sizes and transmission power on the link quality of a BLE link.
- Examine the performance of BLE in an in-vehicle environment.
- Provide a fair comparison between IEEE 802.15.4(e) TSCH and BLE.
Chapter 4

4 Test Requirements and Approach

This chapter discusses the requirements and the approach of the BLE test setup and the post processing that is performed on the logged data. The aim of the experiments is to collect the Message Error Rate, the Packet Error Rate and the latency of BLE wireless links under a specified set of network parameters.

4.1 Test Setup Requirements

The test setup should comply with the following requirements:

1. The results should provide a fair representation of the performance of a typical BLE network in an in-vehicle environment. (section 4.3)
2. The results of the experiments should be comparable to the work reported on IEEE 802.15.4(e) TSCH in [1].
3. The test setup should record the metrics required to accurately calculate the packet error rate, the message error rate and the end-to-end latency since these are the main performance metrics of IVNs. (section 4.5)
4. The test setup should be able to run multiple test cases automatically, without the intervention of humans since collecting the test data can take a long time.
5. The test setups parameters should be programmable without recompiling and reprogramming the nodes involved in the test because a computer equipped with the necessary software to perform these tasks might not be available.
6. The logs generated by the end nodes should be recoverable by the central node without use of a wired connection because there is no wired connection to the end nodes. Creating these wired connection is impractical.
7. The test setup should include controllable interference generation as similar as possible to the interference generation of the IEEE 802.15.4(e) TSCH environment. (Section 4.6)
8. The test setup should be placed in an environment with as low as possible environmental interference to avoid cross contamination. (Section 6.2)
9. The execution time of a single test case in the test setup should be relatively predictable because there is only limited testing time.
10. Every test message on the application level must be directly translatable to a test packet to ensure maximal transparency of the BLE stack.
11. The test setup should allow independent testing of the uplink (client to server) and downlink (server to client). Because performing both tests in a fixed order might not be desirable in some situations where the availability of the testing environment is limited. The test for the uplink is referred to as the data dissemination test and the test for the downlink is referred to as the data collection test.
12. Different links connected to the same nodes should have minimal influence on each other’s performance.
13. The circumstances under which the up and downlink are tested should be as similar as possible.
4.2 Post Processing Requirements

The experiment data of the experiments requires post processing to extract the metrics that present an overview of the network properties. We note that the IVN experiments require many experiments across different test setups involving different protocols. In our work we seek to compare the performance of BLE and IEEE 802.15.4(e) TSCH, thus a universal post processing method is desired to ensure that the results are as similar as possible. A universal approach has the following key benefits:

- Different implementations of the same post processing algorithm can produce different results due to rounding errors and other imperfections of the system performing the post processing. In this universal post processing format we enforce maximal script reuse. Thus the errors caused by the scripts are as similar as possible for all test cases.
- The implementation of a universal data platform saves test setup design time and costs, increases interoperability and makes it easier to compare cross platform results.

This generic post processing method has the following requirements:

1. Post processing should be performed using Matlab to maximize the amount of information that can be extracted from the measurement data.
2. The post processing method should be the same for all test setups involved in the IVN experiments to minimize processing influences on the results.
3. The post processing method should not interfere or hinder with the execution and logging of experiments.
4. The post processing method should allow setup specific data processing alongside “mainstream” data processing because some metrics could be restricted to specific test setups.
5. The scripts used for post processing should be reusable whenever possible for as many test setups as possible to minimize the amount of effort involved in post processing results of a new test setup.
6. The data stored should be easily accessible and readable by the user.

4.3 Test Setup

BLE networks (also known as piconets) always form a star topology. The network used in these experiments are no exception. A graphic representation of the network topology used in the experiments is shown in Figure 4.1.
Chapter 4. Test Requirements and Approach

Figure 4.1: Network topology of the BLE test

The nodes in the network fulfill the following functions:

- **End nodes**: The end nodes are programmed on the QN9020 dongle and are battery powered. These nodes are programmed with the server application. They assume the advertising state once initialized, and advertise until they receive a connection request from the central node. They assume the role of server once the connection setup of the BLE link is completed. The end nodes are scattered across the vehicle to measure the network performance. The test setup can contain a maximum of 8 end nodes at any given time. The end nodes represent small (possibly battery powered) devices that can be placed throughout the vehicle.

- **Central node**: The central node is programmed on the QN9020 development board. The node is programmed with the client application. This node presents the networks user interface. It initializes point to point connections with each of the end nodes. Once connected, it assumes the role of client. This node is connected to the laptop through a UART link. The central node is placed on the dashboard of the vehicle. The central node represents the central computer placed within the vehicle.

- **Laptop**: The laptop uses a UART link to the central node to control the test setup. It will use Tera Term [39] to log the experimental results presented to it by the central node.

In Figure 4.1, the links in the network are represented by a dotted link for wireless links and a solid line for wired links. The BLE link are the source of the experimental results during the experiments while the UART link is used to report and log these results on the laptop.

4.4 Test Phases

We specify separate test cases for the uplink (from the central to the end node) and the downlink (from the end node to the central). The uplink will be tested in the data dissemination test and the downlink will be tested in the data collection test.

Experiments will be performed in runs. A single run covers the execution of both the data dissemination and data collection test over a set of the TX power and message size parameters.
The in vehicle experiments described in this thesis analyze the results of a set of experiment runs. The set of parameters will be constant for all runs but they will take place in different environments or using different connection interval lengths.

4.4.1 The Data dissemination Test
The data dissemination test involves the transmission of a number of test messages from the central to the end nodes. The transmission of a single test packet will have two steps:

1. The central sends the test packet to the end node
2. The end node replies with an empty packet, acknowledging the data packet.

Both steps will take place in the same connection event.

4.4.2 The Data Collection Test
Similarly, the data collection test also involves the transmission of test packets from the end nodes to the central. However, the transmission of a single test packet will have three steps:

1. The central node will poll the end node with an empty packet
2. The end node will reply with a test packet
3. The central node will respond with an empty packet during the next connection event. This packet acknowledges the test packet send in the previous connection event.

Step 1 and 2 will take place during the same (first) connection event while step 3 will take place during the next (second) connection event.

4.5 Performance Metrics
The goal of the experiment is to collect data on the message error rate, the packet error rate and the latency of links. This section describes the methods used to extract these metrics from the raw data. It should be noted that we do not log the throughput because we do not utilize the links to their maximum capacity. The origin of the calculations for the packet error rate is explained in Appendix B

4.5.1 Message Error Rate (MER)
The message error ratio quantifies the number of application-level messages that arrive on the application level of the receiver, compared to the number of messages generated by the sending application. The Message error ratio will be defined as:

$$MER = \frac{SM - RM}{SM} \times 100\%$$

Where $RM$ represents the number of received messages on the receiving side and $SM$ represents the number of messages generated by the sending side.

4.5.2 End to End latency (LAT)
The end to end latency is considered as the time from getting a message from the application layer of the sender to the time of reception of the message by the application layer of the receiver. The end to end latency will be calculated based on measurements explained using Figure 4.2.
The number represent milestones in the transmission of a test message:

1. The moment the application generates a test message (defined as $T_{1tx}$).
2. The moment we record that the stack has entered the message queue (defined as $T_{2tx}$).
3. The anchor point of the first connection interval since the message was queued.
4. The moment the acknowledge begins transmission (defined as $T_{1rx}$).
5. The transmission of the acknowledge fails so the sender has not received an acknowledge when it should have received one. The connection event is closed.
6. The anchor point of the second connection interval since the message was queued.
7. The acknowledge of the first retransmission is sent (defined as $T_{1rx}$).
8. The moment the sender receives the acknowledge and removes the message from the message queue (defined as $T_{3tx}$).
9. The moment the receiver’s scheduler is called.
10. The moment the receiver application processes the message (defined as $T_{2rx}$).

Only moments 1, 2, 4, 7, 8 and 10 can be measured by the test application. Both moment 4 and 7 are defined as $T_{1rx}$ since the receiver does not know at this point if the acknowledge will succeed. The test setup ultimately records the most recent timestamp.
We define the following time periods:

<table>
<thead>
<tr>
<th>Start Moment</th>
<th>End Moment</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10</td>
<td>L</td>
<td>The end to end latency$^1$</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>R</td>
<td>Random amount of time between the moment a message is queued and the first connection event begins</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>$\Delta S$</td>
<td>The amount of time a message spends in the message transmission queue of the sender</td>
</tr>
<tr>
<td>3 (6)</td>
<td>4 (7)</td>
<td>Tm</td>
<td>Transmission time of the test message</td>
</tr>
<tr>
<td>4 (7)</td>
<td>5 (8)</td>
<td>Tack</td>
<td>Transmission time of the acknowledge of the test message</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>$\Delta R$</td>
<td>The amount of time a message spends in the received message queue of the receiver</td>
</tr>
</tbody>
</table>

Time period $R$ is a value that represents the time between the moment a message arrives in the message queue and the anchor point of the first connection interval. This value is random between zero and the length of a single connection interval. It is depending on the moment the test is started in relation to the moment the nodes are connected. This value hinders the comparison between different test cases and is removed from the end to end latency since an efficient application can synchronize to the anchor points to increase the freshness of data.

The latency ($L$) is defined as the time spend between the time between the moment the message is generated and the moment it leaves the send queue of the sending node ($\Delta S$) plus the time between the moment the message arrives at the receiving and the moment it is receives by the application layer of the receiving node ($\Delta R$) minus the randomness ($R$) we discuss earlier. Closer examination of Figure 4.1 reveals that the transmission time of the acknowledge to the data message is added twice. Therefore, we subtract the transmission time of the acknowledge ($Tack$).

$$L = \Delta S - Tack + \Delta R - R$$

The time spend between the time between the moment the message is generated and the moment it leaves the send queue of the sending node ($\Delta S$) is defined as the moment the message leaves the message queue of the sender ($T_{3tx}$) minus the moment the message is generated by the application layer of the sender ($T_{1tx}$).

$$\Delta S = T_{3tx} - T_{1tx}$$

The transmission time of the acknowledge ($Tack$) is defined as the size of the message (80 bytes) divided by the data rate of a BLE link (1Mbps/s).

$$Tack = \frac{10 \text{ bytes} \times 8 \text{ bits/byte}}{1 \text{ Mbps/s}}$$

The time between the moment the message arrives at the receiving and the moment it is receives by the application layer of the receiving node ($\Delta R$) is defined as the time between the moment the

$^1$ Directly measuring the end to end latency is impossible since we cannot synchronize the receiving and sending nodes. See section 2.3.5

Robustness Of Bluetooth Low Energy In In-Vehicle Networks – An Experimental Study
message is consumed by the application layer ($T^r_2$) and the moment the message arrives at the receiver ($T^r_1$).

$$\Delta R = T^r_2 - T^r_1$$

The randomness ($R$) between the moment the message appears in the sending message queue and the first connection interval is defined as the connection interval length ($CI$) modulo of the time spend waiting for a transmission ($\Delta W$). For example

$$R = (\Delta W) \mod CI$$

The time spend waiting for a transmission ($\Delta W$) is defined as the time spend in the message queue ($T^t_3 - T^t_2$) minus the transmission time of the message ($Tm$) and the acknowledge ($Tack$).

$$\Delta W = T^t_3 - T^t_2 - Tm - Tack$$

The transmission time of the message ($Tm$) is defined as the size of the message (a base size of 17 bytes plus the size of the payload $P$) in bits times the transmission time of a single bit.

$$Tm = (P + 17) \times 8 \text{ bits/byte} \times \frac{1}{1 \text{ Mbit/s}}$$

### 4.5.3 Packet Error Ratio (PER)

The BLE stack works with messages from the application level. These messages are converted into one packets and transmitted.

We define the packet reception ratio as:

$$PER = \frac{RGP}{TTP} \times 100\%$$

Where $RGP$ represents the number of good packets received and $TTP$ represents the total number of packets transmitted by the sender. We also define that a link with zero received good packets ($RGP$) has a packet error rate of 1.

We specify in section 4.1 (Requirements) that every message can be directly translated into a single packet. Thus we know that the number of received good packets is equal to the number of received messages: $RGP = RM$

We define the total number of transmitted packets as:

$$TTP = \sum_{i=1}^{Number \of\ messages} (rtx(i) + 1)$$

Where $rtx(i)$ is defined as the number of retransmissions of the message with ID $i$. The added 1 represents the original transmission of the message with ID $i$.

We do not have the access to stack required to directly extract the number transmitted test packets from the stack. Instead we introduce a method to calculate the number of transmissions based on the time spend in the sending message queue of the sending node.

We calculate the number of retransmissions based on the amount of time a message spends in the message queue. We know that a message is retransmitted once for every connection interval it spends.
at the top of the message queue (Appendix B). We also know that this message queue has a size of 1 slot thus any message in the queue will always be at the top of the queue. Thus the time a message spends in the message queue is directly related to the number of retransmissions of a message. This reveals the following formula for the number of retransmissions for the data dissemination test:

\[ rtx(i) = \left( T_3^{tx}(i) - T_2^{tx}(i) \right) \text{div} T_{CI} \]

The data collection test uses the last connection event for the transmission of the acknowledge from the central to the end node. Thus this last connection event should be discarded. The formula for the number of retransmissions for the data collection test becomes:

\[ rtx(i) = \left( T_3^{tx}(i) - T_2^{tx}(i) \right) \text{div} T_{CI} - 1 \]

### 4.6 Interference

This section will discuss the interference. We will limited ourselves to Bluetooth Classic and Wi-Fi because these two protocols are often present in the IVN environment, and they tend to have relatively high TX powers:

- **Bluetooth Classic**: Bluetooth Classic is typically present in the hands-free set of a vehicle. It can also be present in the vehicles stereo system. The main differences between Bluetooth Classic and BLE are: A higher maximum TX power, higher data rates, more channels and smaller bandwidth per channel.

- **Wi-Fi**: Today, Wi-Fi is almost omnipresent in the world. Traffic is generated by computers, cellphones and other devices. Its prevalence combined with its relatively high TX power, high data rates and giant bandwidth make it a primary source of interference for networks using the 2.4 GHz band.

#### 4.6.1 Environmental Interference

There is no feasible location where there will be no environmental interference. The tests are performed on the grounds of the High Tech Campus in Eindhoven. We specify that the location used for the BLE experiments should have minimal environmental interference for both BLE and Wi-Fi.

#### 4.6.2 Controlled Interference

A controlled Wi-Fi interference generator was not available for the BLE experiments described in this thesis. Thus there were no experiments involving controlled generation of Wi-Fi interference. Environmental Wi-Fi interference was present during the experiment but it was not controlled.

We specify that the Bluetooth classic interference for the BLE experiments should be as similar as possible to the interference used for IEEE 802.15.4. The implementation of this interference is described in section 5.5.

### 4.7 Post Processing

We note that the post processing of experimental data has the following three steps:
Figure 4.3: Post processing steps

Where:

- **Parsing**: The extraction of data from the log file into a Matlab processing format.
- **Processing**: The processing of the raw data into the metrics specified in 4.5.
- **Presentation**: The visualization of the metrics into graphs that allow us to draw conclusions.

The requirements for the post processing (Section 4.2) State that the post processing setup should be able to deal with mainstream data and with custom data. Thus we cannot specify a single logging format to be used by all test setups. Instead we propose to achieve universal post processing by specifying the data storage format to be used after the parsing step.

4.8 Expected Results

There have already been some studies regarding the performance of BLE. We discuss these in Chapter 3.

There is no network level simulator for BLE that includes transmission failure mechanisms. A basic BLE network simulator exists [40] but has yet to be extended to the point where it can be used for predicting the robustness of tests. Therefore it is impossible to make supported numerical predictions.

In general we expect that the data dissemination test will have better MRR, PRR and LAT than the data collection test because the acknowledge of the test message is not transmitted in the same connection interval as the test message itself. Instead it is transmitted in the next connection event. This deteriorates the performance when compared to the data dissemination test where the acknowledge is transmitted in the same connection interval as the test message because:

- Connection events are periodic thus waiting for the next connection event will increase the latency.
- Separate connection events utilize different BLE data channels with different channel properties. Different channels can have different performance. When one channel is used then the transmission will fail if this channel is bad. If we use two channels then either of the two has to be bad for the transmission to fail. Thus the test with two channels will have worse performance if we have the same set of channels out of which a subset is bad.

4.8.1 RQ1: Varying the Connection Interval Period

The connection interval period is the time between 2 consecutive connection events. An increased connection interval period will thus result in less connection events in a time window. This effectively reduces the maximum number of transmissions in a time window. We expect that this will only affect the performance of the network when these transmissions are required with the exception of the latency, which will always be affected by the connection interval period. The expected effects of a varying connection interval period are listed in Table 4.2.
Table 4.2: RQ1: The expected effects of changing the connection interval period (RQ1)

<table>
<thead>
<tr>
<th>Metric</th>
<th>Expected effect of changing the connection interval period (RQ1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MER</td>
<td>Messages can only be queued in the message queue when there is room in the queue. Room becomes available after a message has been acknowledged by the receiver. Decreasing the connection interval period results in more opportunities for this to happen within a given time window. Thus the likelihood that there is room in the message queue increases. Resulting in better performance. Increasing the period has the opposite effect.</td>
</tr>
<tr>
<td>PER</td>
<td>The effects on the PER of varying the connection interval period is similar to effects of the MER with the exception of really poor links, here multiple retransmissions without a successful reception can result in a higher PER then the same situation where less retransmissions take place.</td>
</tr>
<tr>
<td>LAT</td>
<td>The connection interval period directly influences the latency since it is the major factor in the queue latency, which is expected to be the dominant factor of the latency of a message.</td>
</tr>
</tbody>
</table>

4.8.2 RQ2: Varying the Packet Size

We expect that the effect of the packet size on the network performance is related to the existing link quality. Relatively good and stable links are expected to have little problems with larger packets but relatively bad and unstable links can have rapidly changing properties which could create problems for larger packets. The expected effects of varying the packet size can be found in Table 4.3.

Table 4.3: RQ2: The expected effects of changing the packet size

<table>
<thead>
<tr>
<th>Metric</th>
<th>Expected effect of changing the packet size (RQ2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MER</td>
<td>Increasing the packet size in a link with the same bit error rate results in a situation where the new packet might require more retransmissions then the old packet. This, in turn, leads to a longer queuing time for messages, which results in a higher MER. If the BER is low then this difference will be small but when the BER increases the likelihood of requiring more retransmissions increases, thus its effect will increase.</td>
</tr>
<tr>
<td>PER</td>
<td>The effects on the packet error ratio is similar to the effects on the message error ratio.</td>
</tr>
<tr>
<td>LAT</td>
<td>The packet size can affect the latency in 2 manners: larger packets take longer and will thus increase the propagation delay and larger packets require more retransmissions, thus increasing the queuing delay. Out of these two, the queueing delay is the dominant factor resulting in an increase in the latency when the packet size is increased and vice versa.</td>
</tr>
</tbody>
</table>
### 4.8.3 RQ3: Varying the TX Power

The effect of changing the TX power is expected to be dominant over all other parameters. In general we expect to see a sharp decline in performance as the TX power becomes lower:

**Table 4.4: RQ3: The expected effects of changing the TX power**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Expected effect of changing the TX power (RQ3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MER</td>
<td>A lower TX power will imply that, on average, more retransmissions are required for the successful reception of a single message. If more retransmissions are required then the queue losses will increase, which would result in a higher Message error ratio. A higher TX power should in general result in the opposite effect, however we expect that there is a ceiling beyond which the performance will stop increasing.</td>
</tr>
<tr>
<td>PER</td>
<td>We expect that the effects of the TX power on the packet error ratio will be similar to the effects on the message error ratio.</td>
</tr>
<tr>
<td>LAT</td>
<td>Decreasing the TX power will likely result in an increased number of retransmissions. This lead to an increased queueing delay which will result in an increased latency. We expect that increasing the TX power will have the opposite effect.</td>
</tr>
</tbody>
</table>

### 4.8.4 RQ4: Bluetooth Interference

Bluetooth Classic has 80 channels with a bandwidth of 1 MHz for each channel. In comparison: BLE has 37 data channels with a bandwidth of 2 MHz each. Both Bluetooth networks need to be transmitting on the overlapping channels at the same time in order to cause problems.

For simplicities sake we assume that both the Bluetooth Classic and BLE nodes are transmitting continuously, that both networks use all of their channels and that any form of interference will always result in packet loss for the BLE link.

Then we note:

- The Bluetooth Classic link will hop channels twice as fast as the BLE link. [21]
- For every BLE channel there will be two overlapping Bluetooth Classic channels.
- BLE links will use 37 out of the 40 channels as data channels.

So the chance that, under the circumstances specified above, the networks will generate interference for each other is calculated in Equation 4.1, resulting in a loss of five messages for every 100 messages transmitted.

\[
\text{Expected packet loss due to Interference} = 2 \cdot \frac{2}{80} \cdot \left( \frac{1}{40} \cdot 37 \right) \approx 0.05
\]

*Equation 4.1: The chance of message loss due to Bluetooth Classic interference*

We note that a BLE transmitter is not capable of transmitting continuously and that it is not always the case that any interference will cause packet loss. Thus the number of messages lost to Bluetooth Classic interference is likely to be lower in reality.

Based on this information we do not expect to see significant increases in the metrics investigated in this thesis due to Bluetooth Classic interference.
Chapter 5

5 Test Implementation

The BLE test platform described in the previous chapter is implemented on the QN9020 BLE radio chips described in section 2.3. This chapter describes the design choices made during the implementation and their consequences. The last sections of this chapter cover the implementation of the BLE interference and the post processing performed using MATLAB.

5.1 Test Messages and State Machine

In this section, the general implementation of the test setup and the state machine are explained.

5.1.1 Generation and Transmission of Test Messages

The test setup can perform two types of tests:

1) The data dissemination test, which tests the uplink.
2) The data collection test, which tests the downlink.

Both test cases essentially follow the same procedure except that the sender and receiver are interchanged. A single test case involves the creation of a set number of test messages. The generation of a test message is performed periodically by an application timer every 100 milliseconds. The test case ends once a set number of messages have been generated.

A message is only transmitted if it is placed in the message queue. A message can only be added to the message queue if there is room in the message queue. If there is no room in the message queue for a newly created message then this message will be discarded.

This approach has been selected because the BLE stack has (theoretically) an infinite number of retransmissions and because we cannot remove a message from the message queue once it has been inserted. This would mean that any test setup with a set number of transmitted packets would need a large amount of time to handle a single bad connection while the other nodes in the network would be idle. This would incur unacceptable waiting time for the other nodes participating in the test. It would also make it impossible to predict the running time of the test because the running time would be dependent on the network properties.
5.1.2 State Machine and Message Types

The flow of the test is controlled by state machines and messages. This section discusses them for both the central and end nodes.

The mechanism that aborts tests has been omitted from the state machine. If a test is aborted both the central and end node will perform the same behavior as a finished test case. After this data is reported the device will return to the Idle state.

5.1.2.1 Central Node

The state machine of the central node is displayed in Figure 5.1.

![Flow chart of the central node](image)

**Figure 5.1: Flow chart of the central node**

The states in Figure 5.1 have the following functions:

- **Boot state**: When the central node boots it will perform a number of steps resulting in the initiation of the BLE stack and the Quintic Private Profile service. When the device has finished booting, it moves to the idle state.
- **Idle state**: In the idle state the connections are initiated and the parameters of the tests are set. When the user is ready to initiate the test he/she sends the start test command which will move
the device to the set parameter state. The connection interval parameters are also set in this state, before the devices connect to the central.

- **Set parameter state**: All parameters except the connection interval are set in this state. All the nodes receive the parameters and confirm them one by one. When all nodes have confirmed the test parameters, the central node will decide between the data dissemination and the data collection test. One of these tests is then started.

- **Data Collection test state**: During the data collection test state, the central node receives test messages from the end nodes. The end nodes will transmit an update when they are finished sending the current batch of messages. When the central node has received an update message from all nodes it will move to the message data collection state.

- **Data Dissemination test state**: The central sends test messages to the end nodes in the data dissemination test state. Once a batch of messages has been generated it waits for the message queue to become empty before moving to the message data collection test state.

- **Message Data Collection state**: In this state, the central node approaches the end nodes one by one to collect the performance data of each message. Once this data is collected, it will be matched to message data produced by the central and printed for the user. The central moves to the test report collection state once all data has been collected.

- **Test Report Collection state**: The test report from each end node is collected in this state. The information of these reports is merged with the test report from the client and stored as the most recent test report. This ensures that there will always be information to report, even if a node disconnects during the test. Once the test reports are collected the device will determine if the required number of test messages have been generated. If this is not the case then the device resumes the test for the next batch of test messages. If this is the case then the device moves on the next case, either the collection of statistics data for the data dissemination test or the print report stage for the data collection test.

- **Get Statistics Data state**: This state is only reached during the data dissemination test where the central device will get the statistics data from the end nodes. The central node already has the statistics data from the data collection test since it is the receiving party in this test. The central will reach the print report state once all the reports are collected.

- **Print Report state**: In this state the test reports are printed to UART. Once all reports are printed the device checks if nodes were lost during the test. If this is the case it tries to reconnect to these nodes. Otherwise, it moves on to the next test case.

- **Reconnect state**: The central attempts to reconnect to lost end nodes during the reconnect state. Once a single reconnect attempt has been made, the device continues to the next test case.

The central node responds to incoming messages as described in Table 5.1.
### Table 5.1: Reactions of the central node to a received message

<table>
<thead>
<tr>
<th>Received message ID</th>
<th>Central node reaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) <strong>Server_Parameter_Confirm</strong></td>
<td>The central node leaves the Set parameter state once all end nodes have sent this message.</td>
</tr>
<tr>
<td>2) <strong>Server_Statistics_Data_Reply</strong></td>
<td>Print the received data and either request the next set of data or leave the Get Statistics Data state once all the statistics information has been collected.</td>
</tr>
<tr>
<td>3) <strong>Server_Message_Data_Reply</strong></td>
<td>Match the message data to already known message data, print the results and either request the next set of data or leave the Message Data Collection test state once all message data has been collected.</td>
</tr>
<tr>
<td>4) <strong>Server_Data_Dissemenation_Report_Reply</strong></td>
<td>Match the report data with the central’s own report and store the information so it can be printed later.</td>
</tr>
<tr>
<td>5) <strong>Server_Data_Collection_Report_Reply</strong></td>
<td>Match the report data with the central’s own report and store the information so it can be printed later.</td>
</tr>
<tr>
<td>6) <strong>Server_Test_Message_Data_Collection</strong></td>
<td>Collect and store the performance data relating to this message.</td>
</tr>
<tr>
<td>7) <strong>Server_Data_Ready</strong></td>
<td>The server indicates that it has finished sending a set of test messages and it is ready to report the performance data of these messages. Once all servers have transmitted this message the central node will leave the Data Collection test state.</td>
</tr>
<tr>
<td>8) <strong>Server_Done</strong></td>
<td>The server indicates that it has finished with the data collection test and it is ready to report the performance data of these messages. Once all servers have transmitted this message the central node will leave the Data Collection test state.</td>
</tr>
</tbody>
</table>

#### 5.1.2.2 End Node

An end node has no state machine. Instead it either responds to its own timers or to received messages. The responses to the messages are listed in Table 5.2.
Table 5.2: Reactions of the end nodes to messages received

<table>
<thead>
<tr>
<th>Received message ID</th>
<th>End node reaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>9) Client_Parameter_Message</td>
<td>Adopt the parameters in the message and reply with message 1</td>
</tr>
<tr>
<td>10) Client_Test_Message_Data_Dissemenation</td>
<td>Collect and store the performance data of relating to this message</td>
</tr>
<tr>
<td>11) Client_Request_Statistics_Data</td>
<td>Reply with a message 2, containing the requested information</td>
</tr>
<tr>
<td>12) Client_Request_Message_Data</td>
<td>Reply with a message 3, containing the requested information</td>
</tr>
<tr>
<td>13) Client_Request_Data_Dissemenation_Report</td>
<td>Reply with message 4, containing the test report</td>
</tr>
<tr>
<td>14) Client_Request_Data_Collection_Report</td>
<td>Reply with message 5, containing the test report</td>
</tr>
<tr>
<td>15) Client_DC_Test_Start</td>
<td>Start the timers required for message generation and begin the transmission of test messages for the data collection test.</td>
</tr>
<tr>
<td>16) Client_Abort_Test</td>
<td>Stop all timers and prepare test reports for collection</td>
</tr>
</tbody>
</table>

5.2 Test Setup Interfaces
The test setup can react to 3 outside stimuli: sending or receiving messages, timers and a UART link.

5.2.1 Sending and Receiving Messages
[This section has been removed for confidentiality reasons]

5.2.2 Timers
The QN 9020 provides two types of timers, hardware timers and task timers. The hardware timers are set by changing register values and generate an interrupt when they expire. Task timers are set by the kernel and generate a task when they expire. This task is then handled by the scheduler.

The hardware timer is only used to perform the polling of the message queue since this polling requires a sample frequency of 10 kHz. All other timer related tasks are performed by task timers.

5.2.3 UART
The UART link is used to communicate between the central node and the logging laptop. The test nodes also support UART connections but they are only used for debugging purposes. The main purpose of the UART link is to allow the user to enter commands to the test setup and to report the results of the experiments.

The UART drivers are based on example code provided with the QN9020. This implementation is interrupt-based. Messages are printed using a printf function which stores strings of characters into a send buffer. The UART implementation uses hardware flow control to ensure that no information is lost.

During the development of the test setup we noticed that the UART link can cause the test setup to hang during the reception or transmission of test messages. We suspect that this behavior is caused by the UART related interrupt but we were not able to confirm this. We solved this issue by buffering the test related data until the reporting state. Then the data is printed without any issues.
5.3 Logging

The logging of data is performed using the UART link. The test setup logs scenario related data, message related data and channel related data. Every logging message begins with the word “log” such that they can be easily filtered from other information printed on the UART during post processing. The second word in that line identifies the type of logging message that is being printed. Any other information is printed after these two statements. All information is separated by a comma.

The following information is logged for each scenario:

- **Test Case Parameters** (tcp): Every test case starts by listing the parameters of this test case: The message size, the connection interval length, the number of test messages to be transmitted during the test, the transmission power, the number of connected nodes, the test type (data dissemination/collection) the channel map and the message generation period.

  Example: log,tcp,ms,5,ci,100.00,mm,500,tp,-20,nc,8,tt,dc,chmap,1FFFFFFF,mi,100

- **The link mapping** (map) that presents the addresses of the nodes using the connection handle specified.

  Example: log,map,link,0,source,087CBE0E4982,target,087CBE0E4A4A

- **The report** (rap) of a test containing the number of packets received correctly, the number of bad packets received, the number of application level messages send, the number of application messages received and the number of application level messages not accepted into the message queue due to a lack of room.

  Example: log,rap,link,0,tgood,227,tbad,0,asnd,227,arec,227,arej,273

- **A message logging the end of a test case** (tce)

  Example: log,tce

- For each message (identified by dd for the data dissemination test and dc for the data collection test) we log the link in which the message was transmitted, the message generation timestamp, the message queue entry timestamp, the message queue exit timestamp, the timestamp a message is received, the timestamp a message is processed, the RSSI of this message and the number of receptions of this test message.

  Example: log,dc,link,7,id,478,gen,5889092,qe,5889987,ql,5891987,rec,5907119,proc,5907209,rssi,-71,rec#,1

- For each channel of each link (chinfo) we log the number of packets transmitted on this channel and the number of errors on this channel.

  Example: log,chinfo,link,7,ch,34,tot,131,err,5

Together these logging messages provide all the required information.

5.4 Other Design Choices

[This section has been removed for confidentiality reasons].
5.5 Interference Generation

The experimental data produced in these experiments are compared to similar data produced for IEEE 802.15.4(e) TSCH [1]. Therefore, sources of interference should be as equal as possible. We note that the methods used by [1] are relatively undocumented. We have endeavored to recreate these methods to the best of our ability.

Interference for Bluetooth Classic is generated using a Samsung nexus 4 cell phone and a Marmitek BoomBoom 150 Bluetooth portable speaker [41]. This Bluetooth speaker forms a Bluetooth Classic connection to the cell phone. The cell phone is then used to stream music to the speaker. This generates Bluetooth Classic data traffic. Both the speaker and the cell phone are placed close to the central node to maximize the impact of the interference.

This setup is similar to the setup used in the IEEE 802.15.4(e) experiments with the following two differences because this information was not documented:

1. The version of the OS of the cell phone is unknown.
2. We were unable to determine the music playlist used in the IEEE 802.15.4 experiments.

It is unclear how the type of music affects the data traffic. Thus in order to keep the data traffic as constant as possible, we limited ourselves to a playlist of a single song: “Over the horizon”, the Samsung galaxy S4 theme song.

5.6 Post Processing

Post processing is performed in the three steps introduced in Section 4.7: parsing, processing and presentation.

The result of the parsing step is a universal data format that can be reused for all experiments, including the experiments with other test setups and other wireless protocols. This generalized data format has the shape of a multi-level Matlab data structure such that custom fields can be added without any risk of misidentification of data fields. This thesis is limited to the general structure of the universal data format. For a detailed specification of the universal data structure we refer to the master measurement plan for wireless IVN [42].
The test scenario, label, coordinator and end nodes sections will be added during the parsing step. The data stored in these sections is used during the processing step to create the metrics section.

It is important to note that only the parsing step looks directly into the logs provided by the test setup. The processing step utilizes information provided by the parsing step. This ensures that the processing step can be reused for other logging formats. The presentation step takes the data from this data structure and create graphs based on this data. These scripts can also be reused for other test setups.

The processing step ends with a sanity check. In this step, we check for logical errors. For example: a message cannot be received more times than it is transmitted, a message has to enter the message queue before it can exit the message queue, the packet error rate cannot be smaller than zero and the latency cannot be smaller than zero.
5.7 Software Tools

Table 5.3 lists the software tools used in the creation of the code running on the QN9020 Hardware.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Name</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDE</td>
<td>Keil µVision</td>
<td>5.15</td>
</tr>
<tr>
<td>C Compiler</td>
<td>Armcc</td>
<td>5.05 update 2 (build 169)</td>
</tr>
<tr>
<td>Assembler</td>
<td>Armasm</td>
<td>5.05 update 2 (build 169)</td>
</tr>
<tr>
<td>Linker/Locator</td>
<td>ArmLink</td>
<td>5.05 update 2 (build 169)</td>
</tr>
<tr>
<td>Library Manager</td>
<td>ArmAr</td>
<td>5.05 update 2 (build 169)</td>
</tr>
<tr>
<td>Hex Converter</td>
<td>FromElf</td>
<td>5.05 update 2 (build 169)</td>
</tr>
<tr>
<td>Programmer</td>
<td>QBlue</td>
<td>1.02</td>
</tr>
<tr>
<td>SDK</td>
<td>QN9020 QBlue</td>
<td>1.3.7</td>
</tr>
<tr>
<td>Post Processing Tool</td>
<td>MATLAB</td>
<td>R2015a (8.5.0.197613)</td>
</tr>
<tr>
<td>Source Project (client)</td>
<td>Project easy API (EAPI)</td>
<td>1.0</td>
</tr>
<tr>
<td>Source Project (server)</td>
<td>Project Battery server (BASS)</td>
<td>1.0</td>
</tr>
<tr>
<td>Wi-Fi Analyzer app</td>
<td>Farproc wifi analyzer (<a href="http://wifianalyzer.mobi">http://wifianalyzer.mobi</a>)</td>
<td>3.9.8-L</td>
</tr>
<tr>
<td>Cell phone&lt;sup&gt;2&lt;/sup&gt;</td>
<td>LG Nexus 5</td>
<td>Android OS V5.0</td>
</tr>
<tr>
<td>Bluetooth speaker</td>
<td>Marmitek BoomBoom 150</td>
<td>P/N 27888</td>
</tr>
<tr>
<td>BLE sniffer hardware</td>
<td>Nordic Semiconductor nRF51822 USB dongle</td>
<td>nRF51822</td>
</tr>
<tr>
<td>BLE sniffer software</td>
<td>Nordic Semiconductor nRF51822 BLE sniffer</td>
<td>Windows v1.01.1111</td>
</tr>
<tr>
<td>Wireshark</td>
<td>Wireshark</td>
<td>1.12.6</td>
</tr>
<tr>
<td>BLE interference generator</td>
<td>Samsung Nexus 4 cell phone</td>
<td>OS: Andriod 4.4.3</td>
</tr>
</tbody>
</table>

<sup>2</sup> The cell phone was used in combination with the wifi analyzer app
Chapter 6

6 Bluetooth Low Energy Experiments and Results

In this chapter we discuss the BLE experiments performed in the IVN. We start by providing information on relevant non-IVN experiments, then we discuss the IVN experiment and after this we discuss the results and analysis of the IVN experiments.

6.1 Table-top Experiments

The test setup is used for several experiments in the office environment. The experiments are described in Appendix C. The main goals of these experiments are to:

1. Verify the correct functioning of the test setup
2. Determine the effect of the orientation of the nodes on their results.

We conduct a set of tabletop experiments with the aim to identify errors in the test setup. These tabletop results show that the setup does not display any drift. However, we note that values can deviate from each other with values of less than 0.01 in the packet and message error rates. We also observed that latency is accurate up to a deviation of approximately 300 µS. These deviations are explained as statistical deviations.

We perform an experiment to investigate the effect of the orientation of the sender and the receiver nodes on the results of the experiments. Under the test circumstances, which involved the use of a good link in an environment as constant as possible, we observed no differences between nodes with different orientations.

6.2 Vehicle Location

We have no data on the interference present in the environment during the experiments performed for the IEEE 802.15.4(e) TSCH. In this section we will briefly discuss the environment in which the tests are performed.
Experiments will be performed on a nearby parking lot located at coordinates (51°24'45.4"N 5°27'43.9"E)³ as shown by Figure 6.1 and Figure 6.2.

³ Coordinates and map provided by google maps.
The location for selected for BLE is different from the location selected for IEEE 802.15.4(e) TSCH. Both locations experience different environmental interference. When investigating the environmental interference we limit ourselves to Wi-Fi and Bluetooth Classic interference since they are expected to be the main interferers in these experiments.

6.2.1 Environmental Bluetooth interference
Bluetooth interference in this environment will be by cellphones carried by people passing by the vehicle. As such it is impossible to estimate the Bluetooth interference encountered by the test setup of IEEE 802.15.4 since this was never recorded. Comparing the testing locations, the parking lot utilized by IEEE 802.15.4 is closer to the main paths thus the parking space used for IEEE 802.15.4 may have had more Bluetooth classic interference.

6.2.2 Environmental Wi-Fi interference
The main source of interference in the high tech campus is assumed to be the relatively large concentration of Wi-Fi wireless networks. The presence of Wi-Fi activity has been measured using an app (Table 5.3). We have recorded the presence of Wi-Fi networks for both the BLE and the IEEE 802.15.4 experiment locations. The Wi-Fi activity in the area used for the IEEE 802.15.4 experiments is displayed in Figure 6.3 and the Wi-Fi environment used for BLE is presented in Figure 6.4.

The IEEE 802.15.4 environment has considerably more interference than the BLE environment. Thus the location of the BLE experiments is more suitable for the IVN experiment.
6.3 Test Parameters
Our experiments involve a sweep across the TX power, the connection intervals and the message sizes. For all of these test cases we perform both the data dissemination and the data collection test.

For the performed experiments we introduce the following definitions: A test case is the execution of either the data dissemination or the data collection test under one combination of test parameters. A test run is a sweep across a set of combinations of the TX power and the message size. Each connection intervals receives its own runs because they require a reconnect.

6.3.1 Static Network Parameters
The experiments involve the use of a BLE stack under specified circumstances. Most of the environmental and networking parameters remain static for all test cases. The most notable of these parameters are listed in Table 6.1.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message generation interval</td>
<td>100 ms</td>
<td>A new message is generated every 100 ms. Most automotive applications have a maximal message generation interval of 100 ms. Therefore this value can be regarded as a realistic worst case</td>
</tr>
<tr>
<td>Slave Latency</td>
<td>0</td>
<td>Increasing the slave latency would increase the duration of the test.</td>
</tr>
<tr>
<td>Supervision timeout</td>
<td>20 s</td>
<td>When this timeout expires the nodes will automatically disconnect. 20 seconds is the value recommended by the standard [19]</td>
</tr>
<tr>
<td>Message queue size</td>
<td>1</td>
<td>The BLE hardware does not have memory available for a message queue</td>
</tr>
<tr>
<td>Number of test messages per test</td>
<td>1000</td>
<td>0.01 (1%) of the results correspond to 1 second of data traffic</td>
</tr>
</tbody>
</table>

6.3.2 Dynamic Network Parameters
The experiments described in this thesis investigate the following dynamic network parameters: The connection interval, the payload size and the TX power, described in Table 6.2. In total, one run of our experiment requires the execution of 24 test cases.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Interval</td>
<td>50 ms</td>
<td>Two connection events per generated message.</td>
</tr>
<tr>
<td>Connection Interval</td>
<td>100 ms</td>
<td>Realistic maximum of connection interval period. One connection event per generated message.</td>
</tr>
<tr>
<td>Payload size</td>
<td>8 bytes</td>
<td>The payload size required for most target automotive applications</td>
</tr>
<tr>
<td>Payload size</td>
<td>20 bytes</td>
<td>Maximum possible packet payload size specified by the core specification.</td>
</tr>
<tr>
<td>TX power</td>
<td>2 dBm</td>
<td>Maximum TX power allowed by most uncalibrated hardware.</td>
</tr>
<tr>
<td>TX power</td>
<td>-12 dBm</td>
<td>Intermediate value between the minimum and maximum value. This value is also tested by Sudhakar [1] and is the only overlapping TX power value testable with Quintic hardware.</td>
</tr>
<tr>
<td>TX power</td>
<td>-20 dBm</td>
<td>Minimum TX power value allowed by the BLE core specification.</td>
</tr>
</tbody>
</table>
6.4 IVN Experiments Execution

IVN experiments are performed in 2 different vehicles: An Opel Corsa (Error! Reference source not found.) and a Honda Accord (Error! Reference source not found.). Both vehicles are located on the space of the same parking lot. The parking lot is also used by other vehicles and occasionally people walked and drove by the test setup (presumably) with cellphones.

The Opel Corsa experiments were performed on the 8th and 11th of January 2016 and the Honda accord experiments were performed on the 6th and 7th of January 2016. There were occasional rains during the experiment, we assume that the influence of rain on the results of the experiments is negligible [29].

We performed several runs of the test setup in different vehicles and with different combinations of the test setup. We report on the experiment runs listed in Table 6.3:

<table>
<thead>
<tr>
<th>Run ID</th>
<th>Connection interval</th>
<th>Vehicle</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50 ms</td>
<td>Opel Corsa</td>
<td>None</td>
</tr>
<tr>
<td>2</td>
<td>100 ms</td>
<td>Opel Corsa</td>
<td>None</td>
</tr>
<tr>
<td>3</td>
<td>50 ms</td>
<td>Honda Accord</td>
<td>None</td>
</tr>
<tr>
<td>4</td>
<td>50 ms</td>
<td>Honda Accord</td>
<td>Bluetooth Classic interference</td>
</tr>
<tr>
<td>5</td>
<td>50 ms</td>
<td>Opel Corsa</td>
<td>IVN test setup accuracy test</td>
</tr>
</tbody>
</table>

6.4.1 Node locations

The results of the experiments performed using BLE are to be compared to results from IEEE 802.15.4 experiments in a similar environment in chapter 7. Thus the placement of nodes is as identical as possible. The locations of the nodes within the IVN is listed in Table 6.4. Pictures of each location are present in appendix A. The nodes, their roles and locations are explained in Table 6.4.

<table>
<thead>
<tr>
<th>Role</th>
<th>ID</th>
<th>Location</th>
<th>Opel Corsa</th>
<th>Honda Accord</th>
</tr>
</thead>
<tbody>
<tr>
<td>Central</td>
<td>None</td>
<td>On the dashboard of the vehicle, in front of the dials.</td>
<td>Error! Reference source not found.</td>
<td>Error! Reference source not found.</td>
</tr>
<tr>
<td>Server</td>
<td>1</td>
<td>In the front of the engine, slightly lower than the license plate.</td>
<td>Error! Reference source not found.</td>
<td>Error! Reference source not found.</td>
</tr>
<tr>
<td>Server</td>
<td>2</td>
<td>On top of the roof of the car. Near the center of the trunk door.</td>
<td>Error! Reference source not found.</td>
<td>Error! Reference source not found.</td>
</tr>
<tr>
<td>Server</td>
<td>3</td>
<td>In the engine bay, on top of the engine.</td>
<td>Error! Reference source not found.</td>
<td>Error! Reference source not found.</td>
</tr>
</tbody>
</table>
Not every test run features the nodes in the same connection order. This is due to the connection and reconnection mechanism which functions on a first-come-first-serve basis. However this is corrected during post processing where the order of the first recorded experiment is used, with the exception of node 8. In all test cases, node 8 shares the same location and is always connected last because these results are omitted from the test. The reasons for removing the results of node 8 are explained in appendix B. We chose to omit the location of the passenger door because this was the most likely location to present a good performance.

Node 1 has a different location in the Opel Corsa and the Honda Accord because this location was also different in the IEEE 802.15.4 results. So the results for this node cannot be compared. In the Opel Corsa node 1 is placed inside the engine bay near the bottom of the radiator while it is placed on top of the front bumper below the license plate outside of the engine bay for the Honda Accord.

All nodes participating in the experiments are always placed with the same orientation with regards to the central node.

6.5 IVN Results
[This section has been removed for confidentiality reasons]
Chapter 7

7 Comparison with IEEE 802.15.4(e) TSCH

In this thesis we compare the results presented earlier with similar results obtained for IEEE 802.15.4(e) TSCH [1]. In this chapter we start by discussing which of the IEEE 802.15.4(e) TSCH results can be compared with BLE. Then the comparison is performed and conclusions are drawn. In this chapter we will refer to IEEE 802.15.4(e) TSCH as TSCH.

The IEEE 802.15.4(e) TSCH data used in this chapter has been reported in [1]. In this work we introduce a modified parsing script which stores data in the universal storage format described in section 4.7. The packet error rates are recalculated using the same script used to calculate the BLE packet error rate. Therefore, it is possible that the values reported in this chapter deviate from the values reported in earlier work. The end to end latency calculations for IEEE 802.15.4(e) TSCH have not been changed. We try to explain observed behavior whenever possible.

7.1 Comparability

In the work leading up to this thesis we took great care to ensure that the results of the BLE experiments are comparable with the results of the TSCH experiments:

- We use the same vehicles.
- Node locations in the vehicles are similar.
- The nodes used the same power sources.
- We deployed the same message generation rate (100 ms).
- Both networks use the same topology.

The TSCH test setup performs a number of different test cases, which are described in Table 3.3. We note that the following performances can be compared:

1. The best case performance.
2. The performance using different TX powers.

Other network parameters cannot be compared because:

- BLE does not support shared time slots. BLE always has dedicated time slots.
- The smallest TSCH message size reported produces packets (47 bytes) that are larger than the maximum possible packet size of the BLE test setup (37 bytes).
- The BLE stack always performs retransmissions. So we only compare the results to TSCH results with 5 retransmissions.
- The TSCH test setup does not record the message error rate.
- The data dissemination test of the TSCH test setup uses broadcasts. BLE has no broadcasts.
- The effect of Bluetooth Interference is only measured for TSCH in combination with a running engine, air-conditioning and radio. Therefore, interference measurements cannot be compared.
Other relevant factors impacting the comparison include:

- Both protocols are implemented on different hardware platforms with different PCBs and different antennas.
- The vehicles are positioned at different locations for the tests. We observed that the location used for testing the TSCH setup suffers from greater environmental interference. (see section 6.2)
- We have observed that the TSCH experiments are performed with more Wi-Fi interference. However, the effects of this interference are unknown, thus we cannot take it into account in this comparison.

We also note that the accuracy of the TSCH test setup is not investigated. Therefore it is impossible to compare individual values of TSCH and BLE. In this chapter we will restrict ourselves to investigating trends and the regions (connected, transitional or disconnected) in which links are located.

The latencies of the data dissemination test of TSCH cannot be compared to the latencies of BLE because the data dissemination test of TSCH uses broadcasts. Broadcasts are not acknowledged and have no retransmissions. Therefore the latency of the data dissemination test of the TSCH setup is not affected by the packet error rates.

The data collection test of the TSCH setup has 5 retransmissions while the BLE test setup has an infinite number of retransmissions. Thus the latencies of this test can only be compared when the packet error rate is relatively low.

7.2 Best Case Comparison
[This section has been removed for confidentiality reasons]

7.3 TX Power Comparison
[This section has been removed for confidentiality reasons]
Chapter 8

8 Conclusions and Future Work

This chapter discusses the conclusions of this thesis. We discuss the conclusions of the research questions for BLE and the comparison of BLE with IEEE 802.15.4(e) TSCH. We also discuss recommendations for future work and reflect on the work performed in this thesis.

8.1 Conclusions: Bluetooth Low Energy

In this thesis we examined the effect of the performance of Bluetooth Low Energy in the in-vehicle environment. We formulated four research questions where each research question changed one parameter of the BLE network stack. The parameters we changed are: The connection interval period, the message payload size, the TX power and the Bluetooth Classic Interference. We performed experiment runs with different combinations of these parameters.

In general we focus on detecting trends when changing network parameters. When we expect certain changes, they should occur for all nodes (or at least most of them). In chapter 6, we have attempted to explain the trends discovered during these experiments to the best of our ability. These experiments revealed the following insights:

Concluding: We have observed there are IVN node locations where a wireless BLE link delivers good and relatively predictable performance. Based on this information, we conclude that based on currently available knowledge, there is no reason to stop research into the feasibility of wireless IVNs, or wireless IVNs using BLE.

Reflecting on the BLE experiments, we note that performing the measurements on the metrics we have was more difficult than we expected. Both the packet error rate and the end to end latency are difficult to measure. The packet error rate has to be measured indirectly because we cannot count the number of times a message is transmitted and the end to end latency has to be measured separately by the sender and receiver because the nodes cannot synchronize their application layers. In the end we still succeeded in creating a test setup that operates with good precision in tabletop tests. Later we noted that there are the test setup appears to be less accurate when it is deployed in the IVE. We had to schedule some tests in order to test the accuracy of the BLE test setup. We noted that the accuracy difference between the connected region and the transitional region is large. Despite these setbacks, we still managed to perform measurements on the BLE parameters we selected. We can observe the behavior of the BLE stack, view its performance and explain why things happen the way they do. Overall, we are happy with the results we produced for BLE.

8.2 Conclusions: Bluetooth Low Energy and IEEE 802.15.4(e) TSCH

We compare the BLE results presented in this thesis to similar results obtained for IEEE 802.15.4(e) TSCH [1]. We compare both the best case performance and the effect of changing the TX power. Based on this performance, we present the following insights:
Concluding: We have attempted to compare the results of the IEEE 802.15.4(e) TSCH test setup with the BLE test setup. We have examined two best case situations where both protocol stacks show similar performance. We also observed that there are cases where both IEEE 802.15.4(e) TSCH and BLE have better packet error rates when compared to each other. Overall there is insufficient data to select the best protocol stack.

When we reflect on this comparison we note that making the comparison was more difficult than we expected. Many of the parameters (TX power, message size and the number of retransmissions) selected for the IEEE 802.15.4(e) TSCH cannot be selected or are fixed for BLE. Thus we have to approach them to the best of our ability. The lack of any accuracy figures for the IEEE 802.15.4(e) TSCH test setup makes the comparison more difficult. The consequence of this is we have to extremely careful when we make statements on compared data because these statements will rely on a lot of assumptions. We focus on making general comparisons based on the region links occupy or noted trends in the data. The comparison can be done in more detail once the accuracy of the IEEE 802.15.4(e) TSCH test setup is quantified.

8.3 Future Work
[This section has been removed for confidentiality reasons]
Bibliography


[34] NXP Semiconductors, "Data Sheet: JN516x," NXP, Sheffield, 2013.


expo (ICCVE), Las Vegas, NV, 2013.


A. Appendix: Experiment Pictures

[This section has been removed for confidentiality reasons]
B. Appendix: Measuring the Number of Transmissions

[This section has been removed for confidentiality reasons]
C. Appendix: Accuracy of the Test Setup

[This section has been removed for confidentiality reasons]