

Evaluating bi-directional connectionless BLE for bike-to-bike communication

Pedro Henrique Wo de Alcantara
University of Twente
p.h.wodealcantara@student.utwente.nl
student number: 2147742

Khalil Ben Fredj
Pervasive Systems
University of Twente
Enschede, The Netherlands
k.benfredj@utwente.nl

Geert Heijenk
Design and Analysis of Communication Systems
University of Twente
Enschede, The Netherlands
geert.heijenk@utwente.nl

Yanqiu Huang
Pervasive Systems
University of Twente
Enschede, The Netherlands
yanqiu.huang@utwente.nl

Abstract—As a response to the growing demand for solutions to improve and manage bicycle utilization in cities, this paper aims to design and evaluate the features of bike-to-bike (B2B) communication via Bluetooth Low Energy (BLE). Our proposal is a system that disseminates relevant data from surrounding bikes to aid the cyclist in detecting hazardous obstacles and scenarios. Our system uses bi-directional connectionless communication to broadcast messages containing position information to nearby bikes. To determine the possible use cases of our communication design, we evaluated with practical measurements different sets of BLE parameters and compared them with the analytical model from previous research. By doing so, we confirmed that scan window duration and synchronization are critical factors for optimal message delivery. Moreover, the number of copies sent is a relevant option to reduce energy consumption with minimal effects on message delivery. Our results show that even for time-critical applications B2B via BLE could be an option.

Index Terms—Smart bikes, B2B communication, BLE, connectionless communication, bi-directional communication

I. INTRODUCTION

With the ascending integration of wireless communication into transportation, different solutions for safety, management and sustainability in smart cities have emerged. As bicycles are one of the most sustainable and efficient means of transportation, many cities worldwide have been expanding and improving their bike infrastructure by building new bike lanes and improving cyclists' safety. Cities like Paris are investing 250 million euros into bike infrastructure aiming to make it a "100 per cent cycling city" [1]. With the Netherlands already having a developed cycling infrastructure, resources are being invested in Internet of Things (IoT) solutions to improve bike commuting. Ideas such as vehicle communication with traffic lights and other vehicles, already exist for smart vehicles but are now being translated to bicycles. One example is the "Enschede Fiests" App, which tracks the user's commutes and rewards rides with redeemable points. In special locations, it can also communicate with traffic lights to increase the

chance of getting the green light faster. Regarding "smart" bike technologies, there are different levels of smartness. As shown in [2], level 0 consists of regular bikes, while at the highest level (5), there is a full integration between the data and how traffic authorities change users' behavior based on real-time information. This paper explores level 4 technology, focusing on bike-to-everything (B2X) wireless communication. This paper expands specifically on the use of BLE for B2B communication and how efficient this communication could be.

The research aims to expand on the theoretical model shown in [3] by having a practical evaluation of the characteristics presented. In the paper, it is modeled how the selection of BLE parameters impacts the likelihood of success in transmission and the latency to discover devices. Based on the model, we want to evaluate the same parameters in a practical setting and compare the performance. With practical tests simulating real scenarios, we intend to examine the performance of our system and answer the following question:

"RQ1: How can bike-to-bike wireless communication help improve bicycle commuting safety and efficiency while providing cyclists with real-time surrounding awareness?"

Additionally, we look more into the technical characteristics of the system by analyzing the following sub-questions:

- **RQ1.a:** What are the main BLE parameters to guarantee a reliable communication, and how do they impact energy consumption?
Approach: We will test different sets of parameters and evaluate how they impact the quality of the communication.
- **RQ1.b:** What are the effects of medium obstruction and movement on the quality of the communication?
Approach: Performing field tests with real-life conditions of propagation.

The next sections will be divided as follows. We will start presenting what has previously been researched about the approached topics and the gap filled by our research in the related works section. Then, in background, we will give a brief overview of the main BLE aspects studied. Next, a methods section where we will take a deeper look at the mathematical model we base ourselves on, present the Key Performance Indicators (KPIs) for our use case and explain how we can get the most out of the limited size packet structure. Further on, in section V, we will present the setup implementation and the results from our tests, explaining how they are relevant to our research. Finally, we will end the paper with recommendations for future work and a conclusion.

II. RELATED WORK

BLE technology and IoT in transportation are not new topics, but this paper aims to explore scenarios that have not been studied as much. In the next subsections, we will show works which topic overlaps with ours while pointing out the research gaps we want to address. First, we will dive deeper into our B2B context, showing how smart traffic technologies are being implemented to improve safety and infrastructure management. Further, we will explain how our system differs from the current BLE strategies for the smart cycling context regarding how nodes interact with each other. Following that, we examine the importance of discovery latency, presenting works where this performance has been modeled. Finally, we look into the energy matters to understand why it is important for IoT applications and what to expect when using BLE.

A. Smart bikes communication

As IoT technologies become more accessible, solutions to improve transportation efficiency and civilian safety have emerged. Initially focused on automotive applications, intelligent transportation systems (ITS) were developed to detect vehicles and perform real-time actions based on the current traffic information. Some common applications are: presenting less busy alternative routes, notifying potential dangers and displaying status information on public transport. Nonetheless, to improve sustainability, smart cities realized that they should invest in bicycle infrastructure as it is a "greener" and efficient option for urban transportation. Consequently, as the bike market expands, the demand for the translation of ITS solutions into bicycle traffic scenarios grew. In [4], a variety of use cases are presented for the connection between bicycles and other devices, some of them are: location-based services, riders and bike status, safety notifications and traffic control. In [5] and [6], it is shown how data gathered during rides can be used to improve riders' well-being, not only for efficient transportation but also for health factors such as air quality.

In [2], it is shown that bike smartness aims to achieve the ability to, in real-time, act upon traffic actors. For that to happen, the bike needs to communicate with others nearby. An option is to use cellular connections because it allows the device to exchange information with a main server that can transmit data from all parts of the city as a centralized center

of operation. However, it may come with the trade-off of a larger energy demand. In [7], three potential communication protocols for short-range communication are evaluated: WiFi, ESP-Now and BLE. Although BLE does not have the best performance, the paper points out that the main reason for using the protocol is its small use of energy and compatibility with most devices currently available in the market. Another benefit of BLE is the broadcast mode, which may decrease the latency of information dissemination by removing the need to establish connections.

Our research expands on the specific BLE configurations and parameters that can be tuned to achieve the most out of this protocol, exploring the trade-offs between message delivery, latency and energy use. Also conducting practical trials to evaluate BLE under real cycling scenarios.

B. Connectionless Mode

When working with the communication of moving devices, a stable channel is not the norm. First, taking into account the range of BLE (approximately 100 m), devices may not be in communication range, and even when in range, due to obstacles in the Line of Sight (LoS), packet delivery does not always happen. To counteract the challenging environment factors, solutions like [5] and [6] use a server approach where data is primarily sent to a smartphone device, then to a main datacenter over the internet. However, this process may create latency, becoming inefficient for time-critical applications. With BLE, we can solve the latency issue by having a peer-to-peer link. Yet, as connection under moving circumstances is challenging, the best approach is to use the connectionless mode. In this mode, the nodes can skip the connection process and send the data straight away, turning communication faster and less dependent on channel stability. In [8], an example of the use of BLE connectionless mode is shown. There, the beacons transmit status information if the bike is reported as stolen. In [9], BLE beacons are attached to traffic signs to alert vehicles once they are in range. The latest resembles the B2B scenario, but only transmits data in a single direction.

This project explores a gap regarding the bi-directionality of the connectionless approach, pointing out the new implications that come with such configuration. Bi-directionality means that node will act both as advertiser and scanner, which allows bikes to be equally aware of the surroundings. However, it comes with the implication that if nodes are transmitting at the same time, they do not detect each other.

C. Discovery latency

Especially when considering safety warnings and accident prevention, timing is vital. This implies that information must be propagated at such a speed that the user is still within the reaction time and deceleration range. In [10], practical tests were performed to determine the "comfort zone" for obstacle avoidance, or the distance where a cyclist can comfortably maneuver, for different speeds. The results show the average cyclist needs 9.65 ± 4.66 m for breaking. Since the first step in the stopping process is to receive the warning sign, it is

required that nodes should be detected as soon as possible in the Neighbor Discovery process (NDP). In [11] and [12], NDP is modeled and optimized to decrease discovery latency on connectionless BLE. Those works give an insightful view of which BLE parameters most impact the NDP’s speed and what their trade-offs are with energy consumption. However, both analyses only consider unidirectional communication.

As previous works lack instances where nodes advertise at the same time, this paper evaluates the synchronization aspect. Further, we conduct practical experiments to see those effects in message delivery and latency under real conditions.

D. Energy consumption

To make reliable and long-lasting IoT devices, it is also important to consider energy consumption. Although in some cases designers can rely on a constant source of energy, in many cases the devices are powered by a battery. Therefore, the conscious use of energy resources is essential. With a deeper analysis of the consumption shown in [13], we can affirm that communication processes are energy-intensive, making proper configuration crucial for energy optimization. In BLE, one of the main configuration parameters is the duration of each phase of the communication; consequently, the longer the device remains in a high-consuming phase, the shorter its battery life will be. By examining the impacts of each parameter, as done in [14], it is possible to model the overall energy consumption and estimate the best strategy to set up the parameters. However, trade-offs between latency and battery use need to be evaluated and optimized to have reliable and energy-efficient communication. In [15], the lifetime of the device is evaluated in comparison with the interval of communication. The study shows that by reducing the frequency of packets sent, more energy is saved, but the latency is increased. And, to find the optimal parameters, designers need to take into account the requirements of their use case. A similar approach should be used to determine the specifications of B2B. Once those specifications are listed, the designers can use different methods to perform the optimization. In [16], the solution is to use a back-off scheme based on the success rate of message delivery, adding or reducing delay in between transmissions to increase the chances of a successfully delivered packet. The works mentioned mostly take into account connection-based or unidirectional approaches.

In this paper, we will present how tuning advertising or scanning parameters can impact the energy cost of the system. Then, evaluate the battery life to determine if such a system would be able to accommodate the needs of our use case.

III. BACKGROUND

BLE is a communication protocol that uses the 2.4 GHz band to exchange data. This bandwidth is divided into 40 channels, from which three (37, 38 and 39) are exclusively for devices to advertise their presence and attempt to connect with others. In advertising channels, packets have a standard structure and size so others can receive, authenticate and establish their request for communication. If a connection is established,

each side will take a role (central or peripheral) and access the remaining channels to exchange data. However, on instances where a connection-based approach is not feasible, we can opt for using the connectionless mode. In connectionless mode, the available payload bytes from the advertising packets can be filled with data, which can be accessed by any device listening to the channels. The main benefit is that the information, for instance, the telemetry of a sensor, can be sent to multiple nodes simultaneously (broadcast) and faster since it skips the connection process.

When using connectionless BLE, a device can be in one of three states: scan, advertise or idle. When scanning, it is listening to one of the advertising channels. This state is bounded by two parameters: scanning window (S_w) and scanning interval (S_i). S_i is the period before switching channels (always from 37 to 38 to 39 then back to 37), while S_w is when the device is actively listening to the channel. If $S_w < S_i$, the device will be in an idle state for the remaining time. During the advertising stage, copies the data will be sent to all the advertising channels depending on the Network Transmit Count (N). The main parameter for this state is the advertising window (A_w), which is the period in between copies’ transmissions, being idle until the next transmission. Other relevant parameters are the advertising delay (D_{LL}), which is a random value between 0 and 10 ms added between copies to avoid packet collision, and the Network-to-Link layer delay (D_{N2L}), which is caused by the data injection between layers.

IV. METHODS

In this section, we will summarize the mathematical model our research is based. Next, we will explain the Key Performance Indicators (KPIs) for our use case and how we structure the transmitted packets. In Table I, we outline the main parameters of the model and their symbols.

Symbol	Description
S_w	Scanning window duration
S_i	Scanning interval duration
A_w	Advertising window duration
T_{gen}	Message generation period
D_{AdvDur}	Advertising duration
$D_{LL} \sim \mathcal{U}\{0, 10\}$ ms	Link Layer delay
$D_{N2L} \sim \mathcal{U}\{11, 20\}$ ms	Network-to-Link Layer injection delay
$D_{AdvStart}$	Advertising start delay
$D_{Tx/Rx}$	Mode switch (advertising/scan) delay
N	Number of copies (network count)
$\Phi_{1,2}$	Shift between node 1 and 2 epoch starts
s	State representing the advertising period start

TABLE I: Symbol descriptions used in the analysis.

A. Mathematical model

Although the connectionless mode of BLE has been explored in different studies, [3] focuses on a gap regarding bi-directional communication. Studies like [11] and [12] show a relevant explanation of the importance of BLE parameters in achieving efficient discovery of devices and stable

transmission. However, all cyclists need to be aware of their surroundings, therefore, B2B communication should go both ways. To implement such a feature, differently from previous works, all devices need to act as both an observer and an advertiser. This behavior creates a period named "Blind Time", which is when devices attempt to transmit at the same time. As they cannot scan while transmitting, if their advertising periods overlap, they will not detect each other. [3] proposes dividing the time into fixed-length time-slots and using a Discrete-Time Markov Chain (DTMC) to analytically model the transmission. This allows us to determine the probability of successful message delivery and latency to discovery.

Following the method proposed in the paper, the modeling of the system is composed of analyzing three behaviors: the duration of the advertising state, the start of the advertising state and the synchronization between different nodes' advertising phases. Following the terminology of the paper, we will call *epochs* the interval between message generations and *states* the time-slot where advertising starts within an epoch.

Starting with the duration (D_{AdvDur}), although within each epoch the transmission duration and start are not fixed, we can model it by looking at the estimated duration. To calculate the duration of advertising, we can use:

$$D_{AdvDur} = 2 \times D_{Tx/Rx} + D_{LL,N} + A_{Tx} + \sum_{i=1}^{N-1} (A_w + D_{LL,i}). \quad (1)$$

where A_w is the configured advertising window, A_{Tx} is the time to transmit into all the advertising channels and $D_{LL,i} \sim \mathcal{U}\{0, 10\}$ is the random link layer delay (0 to 10 ms) introduced before transmission at each copy. This equation had a few modifications when compared to the original model. That is because, after oscilloscope analysis (Section V-E), we identified different behavior than the original model. First, that D_{LL} is added before each copy transmission. Then, we found that the device takes a short delay to switch modes ($D_{Tx/Rx}$). Finally, the last A_w , which typically includes A_{Tx} , could be cut short after A_{Tx} is completed.

For determining the start of advertising, we can use:

$$s' \equiv (s + D_{AdvDur} + k^* \cdot S_w) \pmod{T_{gen}} \quad (2)$$

where k^* is the minimum value of ℓ that meets the condition $s + D_{AdvDur} + \ell \cdot S_w \geq T_{gen} + D_{N2L}$. Looking into equation 2, we see that the start mainly depends on two factors: the start of the previous epoch and the length of scan windows (S_w). That is because, as shown in Figure 1, the start depends on how many S_w there are before the next message is generated, the closer the message injection is to the end of a S_w the sooner the message will be sent. Furthermore, we can also calculate the range of possible start times with:

$$\begin{aligned} \min\{D_{AdvStart}\} &= \min\{D_{N2L}\} \\ \max\{D_{AdvStart}\} &= \max\{D_{N2L}\} + S_w. \end{aligned} \quad (3)$$

Since the start is a random variable, this becomes a random process which meets the Markov property, allowing for it to

be modeled as a DTMC. Added to that, even though the following epoch is dependent on the previous state, it does not depend on the epoch's index. Those characteristics allow for an irreducible DTMC.

By modeling the start time of two nodes, we can determine the Message Delivery Ratio (MDR). As successful reception depends on whether transmission and reception are in matching channels, the model calculates MDR by measuring the shift between two nodes' advertising start times. The model considers all possible instances where a packet is received, including when a transmission is only partially affected by blind time (first and third packet in Figure 1). Determining if each packet is received or not, then filtering the copies, we find the message delivery ratio. In [3], an equation to derive MDR is given, but it does not consider $D_{Tx/Rx}$ and that D_{LL} is added before each transmission. Further, the paper states that, having each start time as a state, "we can reach any other state in a finite number of epochs". Therefore, they can also model the reception for any later epoch by examining the sets of state pairs and calculating if they are sufficiently apart.

It is stated that the worst-case scenarios for communication are when nodes advertise simultaneously, leading to a significant blind time effect. To consider all factors impacting the shift, besides determining the start time of advertising at each epoch, we also need to evaluate the time difference between each node's message generation ($\Phi_{1,2}$). That is a random value that depends on when the node has initiated operation. Larger $\Phi_{1,2}$ can be beneficial when it makes nodes de-synchronized to the point where no advertising periods overlap, allowing for optimum reception, but if $\Phi_{1,2}$ is closer to zero, the reception becomes more dependent on the random selection of link to network layer delay, and scan window size.

To demonstrate the theory, [3] performs simulations to evaluate the probability of delivery over different $\Phi_{1,2}$. The results show that with zero shift or any multiple of T_{gen} (fully synchronized), we have the worst probability. In addition, it is shown that a solution to improve worst-case occasions is to increase the scanning window. For example, by increasing S_w from 50 ms to 80 ms, we modeled the probability of reception at $\Phi_{1,2} = 0$ to increase from 70% to 80%. The reason is the distribution of start times since, with shorter S_w , the starts of the advertising state are closer to the start of the epoch, increasing the chance of blind time.

In addition, the paper also estimates the discovery latency by modeling the number of epochs needed until the first reception. The model is based on a DTMC with an absorbing state, where the absorbing state is the first successful transmission. Since, when in communication range, each node is in a different state, simulating all possible initial states, they could calculate the average discovery delay for all $\Phi_{1,2}$. Compared with the MDR simulations, we can see that with a higher delivery probability, there is a shorter average discovery latency.

Finally, as pointed out in their conclusion, the simulation focuses on demonstrating the effects of delivery probability and latency, but it requires more analysis to consider factors as propagation and Age of Information (AoI). In their model,

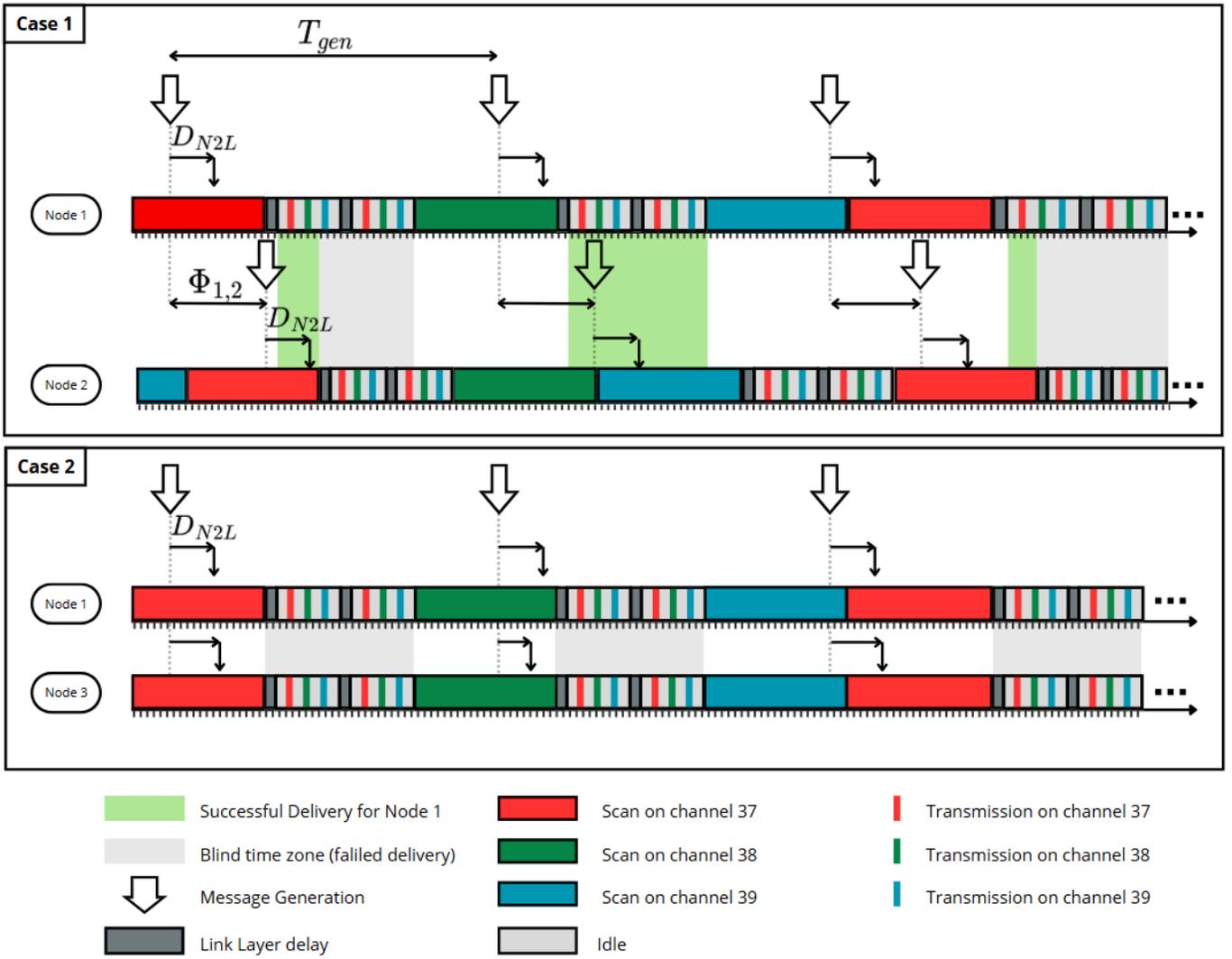


Fig. 1: Communication diagram showing two nodes communicating under different scenarios. In case 1, all packets are received, although there is a small overlap on the first and third transmissions. In case 2, due to blind time, they cannot detect each other ($N = 2$).

they assume the best channel quality, neglecting the impacts of path loss, shadowing and multipath. That is useful to correlate the changes in performance to the tuning of the parameter, but it does not reflect practical scenarios. On top of that, evaluating AoI, the interval between successful receptions, helps determine the relevance of the data received.

B. KPIs

To evaluate the different parameter choices, we choose four KPIs: Message Delivery Ratio (MDR), Discovery Latency, AoI and Energy consumption. Those were selected based on the metrics simulated in [3], recommendations and practical matters, such as battery life. By evaluating these KPIs, we can determine the efficiency of our system and its ability to perform under real cycling conditions. In this section, we will go over each indicator, explain how it is reflected in B2B

applications, and give some context on what the expected results would be.

1) *Message Delivery Ratio*: MDR represents how many unique packets are received successfully. This is the base probability to determine whether another bike will receive the transmitted message. The main natural factors that impact MDR are medium quality, radio propagation and blind time. As those conditions are independent of the system's design, the best approach to counteract them is to increase transmission strength or add redundancy by transmitting more packets. Since the nodes run the same code and BLE schedule, the other limiting factor is synchronization. As mentioned in the previous section, if the node's epochs are fully in sync, reception chances are drastically decreased due to blind time. However, with the added random delays, the node can still receive messages even when in sync. Theory shows that by

having longer scan windows, we can reduce the impact of synchronization.

Keeping a high percentage for this indicator means that more bikes are likely to detect the signal and, consequently, will be able to react to the information. As a side effect, higher probabilities will also result in faster detection.

2) *Discovery Latency*: This metric represents how fast nodes detect each other. It is a crucial parameter to determine the range of applications such communication could have. In transportation use cases, there can be different requirements for discovery latency. For instance, time-critical applications like collision avoidance require a latency lower than cyclists' reaction and maneuvering, while for non-time-critical cases, slower discovery is more acceptable. Similarly to the MDR, we are dependent on the propagation limitations of the environment, but by improving MDR, we are also going to see similar improvements to the discovery latency.

We aim to detect the communicating node within the comfort zone of maneuvering. We chose a zone within 1 to 4 seconds based on [10]. In their trials, the highest speed evaluated was 22 km/h, as we will perform ours at 25 km/h, we chose a similar range to their highest speed.

3) *Age of Information*: AoI is a representation of how up-to-date the data (location on our use case) received are. If too old, it may not be useful as the bike might have moved a considerable distance from the sent location, while if recent, the cyclist could use the information to determine how they cycle. Similarly to discovery latency, this indicator relates to MDR and message generation frequency. It relates to MDR because, if packets are frequently lost, AoI will grow due to longer waiting times between receptions. And, it is dependent on the generation period because it impacts the frequency of "fresh" data. The longer the period, the older the information will be, independent of MDR.

In our experiments, we evaluate the peak AoI, which represents the maximum AoI achieved before the newest reception [17]. We can find this value by comparing the timestamp of the latest reception with the generation time from the previous message received. Under perfect conditions, when MDR is 100%, the peak AoI will be close to the generation period, but with poorer medium quality, it will increase. Peak AoI is a relevant indicator because it can also assist in determining whether a piece of information is still relevant or not. For applications that require a fast response from the cyclist, the peak AoI should not be longer than the reaction time. On the other hand, non-time-critical use cases can have a more flexible performance.

Although it seems beneficial to decrease the generation period to reduce the average peak AoI, there are practical implications. Faster generation means more packets per second, which could overload the radio spectrum and possibly cause interference and increase packet collision probability.

4) *Energy cost*: The energy consumption is important to calculate the battery life of the devices. As shown in [14], the power needed to scan is similar to the power required to transmit packets. However, considering the duration of

each phase, their difference become more noticeable. That is because scanning is a longer and continuous process, while advertising phases are short peaks of energy followed by idle states where consumption is minimal. One way to reduce its cost is by reducing the overall scanning duty cycles (period of active scanning within an epoch divided by the epoch length). Some methods to change the overall duty cycle are: changing the number of copies sent, the more copies the longer the device will be in advertising mode; or changing the message generation frequency, as it will modify the epoch length.

Optimizing consumption comes with a trade-off in the quality of the communication. Minimizing consumption can have a negative impact on the probability of packets being received because reducing the duty cycle may give the system fewer opportunities for detection. Therefore, choosing the scanning durations should be a balance between efficient reception and viable energy cost.

C. Packet structure

Unlike connection-based applications, we cannot transmit parts of the data over multiple packets. When using connectionless mode, we have two options: include all data in the packet or send additional data in a scan response packet. As scan responses require additional packet transmissions and a more stable transmission medium, our aim is to send all data in a single packet. The main benefit of this strategy is that the observer node does not need to acknowledge the sender to receive the data. Considering the conditions of a moving bike, this strategy avoids cases where not all packets are successfully received. However, there is a trade-off with packet size. An advertisement payload is restricted to 37 bytes (including flags and identifiers), as seen in Figure 2. From those bytes, we have the flexibility to set the Manufacturer's Specific Data Field with a custom structure where we can send the necessary data. Our payload data structure consists of: a packer ID (2 bytes), timestamp (4 bytes), advertising start (1 byte), latitude (4 bytes) and longitude (4 bytes). Because location finding is not part of our scope, the latitude and longitude were implemented as placeholders for future implementations. The remaining data fields were selected to support the test measurements. This structure is flexible to modifications as long as the size is within the boundaries allowed.

V. RESULTS & DISCUSSION

In this section, the results of our tests will be presented. First, we will give an overview of how the BLE and the test setup were implemented. Further, we will go over each test performed, explaining how it was done, what we wanted to determine, and what our system's performance was.

A. Implementation

To implement the communication setup on a bicycle, we opt for two nRF5340 Development Kit (DK) from Nordic Semiconductors. The DK was chosen because it embeds the nRF5340 System-on-Chip (SoC), which is a popular option for IoT applications using BLE. Besides that, as a development

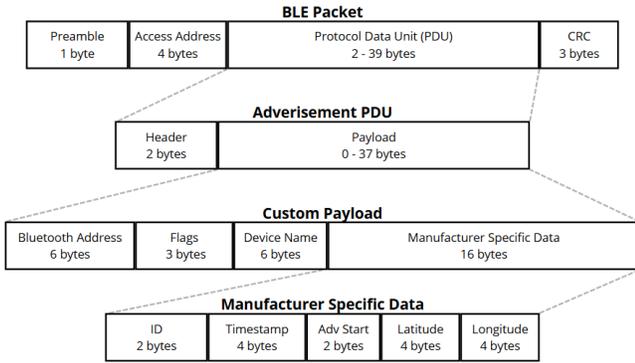


Fig. 2: Packet structure and custom-made structure of the Manufacturer Specific Field for the B2B use case.

board, it offers multiple GPIOs to connect the necessary modules for our measurements, while still leaving empty IOs for future implementations. Regarding hardware (Figure 3a), we also used a microSD module (Adafruit’s MicroSD card breakout board+) to record packet receptions and LED for visualization purposes. To synchronize the nodes, we also use IOs for UART communication.

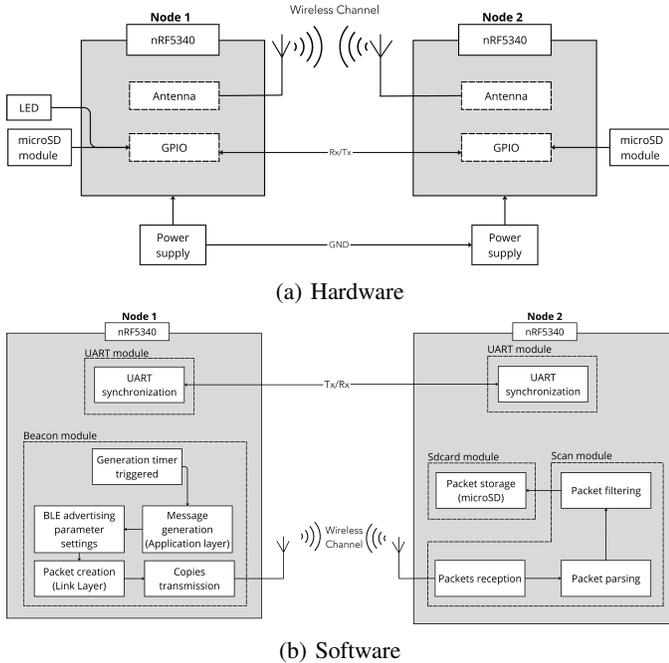


Fig. 3: Block diagrams of the hardware used and how the software modules interact with each other.

On the software side (Figure 3b), the code is divided into C scripts representing each module of the system. The scan module and the beacon module are responsible for the BLE-related functions, settings, and for simulating the Application Layer, generating messages. The SD card module contains the read and write functions for the microSD. The UART module has the UART setup for synchronization, which will

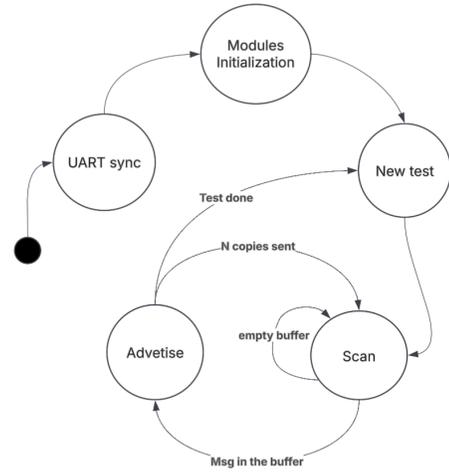


Fig. 4: State machine diagram for the main file loop.

be required to evaluate the epoch shifts. Each module file has its own header file exposing its functions, and we have an additional header file with all BLE parameter values for convenience when tuning. The functions in the files are used in a main script, which initiates modules and loops through a state machine. As shown in Figure 4, after initializing the modules, the system mainly switches between advertising and scanning. However, after a set timer, we create new tests by adding shifts and placing row markers in the saved CSV file. The markers are simply zero rows, but they allow us to automatically distinguish tests in the pre-processing stage. Moreover, it is important to note that scanning is disabled before entering the advertising phase. That is done to avoid unpredictable behavior of scanning delays in between transmissions.

As we want to evaluate the impact of the scan window length and the number of copies sent, we selected the parameter sets in Table II.

Parameter Set	$S_w(ms)$	$N(copies)$	$T_{gen}(ms)$
1	50	3	200
2	80	3	200
3	50	5	200

TABLE II: Parameter sets used in the tests.

In addition to the nRF5340 DKs, we also used an nRF52 DK flashed with Nordic’s nRF Sniffer, allowing us to use that board as a BLE sniffer. That means its only function is to constantly scan BLE channels with 100% duty cycle. This is a useful tool, firstly, to debug communication as we can inspect the content of packets, but also to be the ground truth of measurements by excluding the blind time component.

Pre-processing and processing of the saved CSV files were done using Jupyter Notebooks. As mentioned in section IV-C, the packets contain a timestamp of transmission, advertising start (delta time from message generation to transmission) and an ID for the messages generated. Along with those, the receiving node appends its own timestamp and RSSI. These metrics are saved per packet in each row of the CSV file and

can be used to determine the KPIs after processing. MDR can be found by dividing the number of unique packets received (without counting copies) by the expected number of packets (in our case, one packet per 200 ms). AoI can be measured, accounting for the transmission timestamp and the advertising start, to find when the message was generated. And, discovery latency can be found by observing the reception timestamp.

B. Side by side Test

The side-by-side test is relevant to understanding the impact of blind time (epoch synchronization) on the MDR, removing the effects of imperfect channel conditions (due to pathloss and channel fading). The test was done indoors with the two nRF5340 DKs next to each other (to minimize propagation effects) and the sniffer. Once a test is started, the node will follow the state machine (Figure 4), starting a new test every five minutes (1500 unique packets). To simulate different synchronization, after each test, a 10 ms shift is added to the generation time of one of the nodes. The evaluation of a parameter setting is done after we measure all shifts with the generation period, considering we have $T_{gen} = 200$ ms, then $200/10 = 20$ shifts.

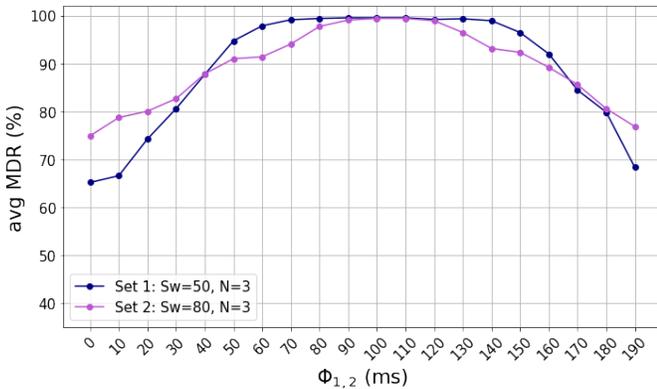


Fig. 5: MDR comparison between $S_w = 50ms$ and $S_w = 80ms$ during side by side test.

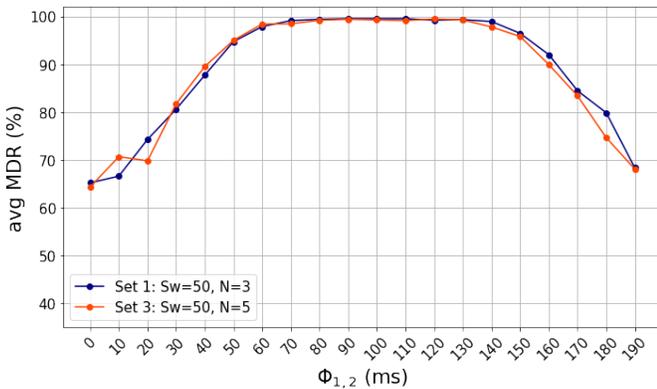
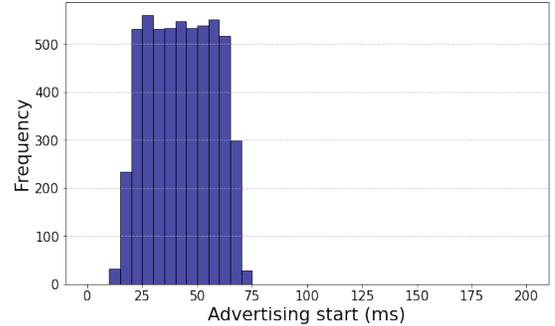
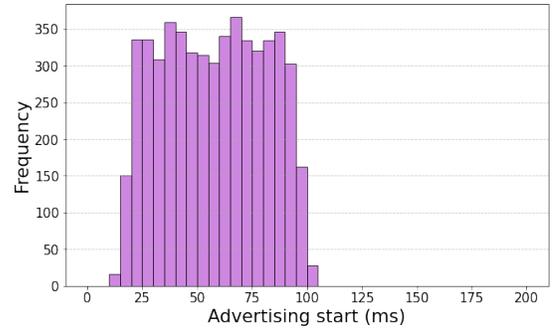


Fig. 6: MDR comparison between $N = 3$ and $N = 5$ during side by side test.

As we can see in Figure 5, a longer scan window gives better performance with closer synchronization as predicted in the model. This difference is explained by the range of time an advertisement can start. In Figure 7, we can see how that range differs between the two sets by making a histogram of their advertising start. That is because, following Equation 3, the maximum start time is increased as we increase S_w . In Figure 6, we also show that modifying the number of copies sent does not influence the trend per shift. Yet, as the duty cycle is reduced, the average MDR tends to be lower with fewer copies.



(a) $S_w = 50ms$



(b) $S_w = 80ms$

Fig. 7: Advertising start (delay between the start of the epoch and the first transmission).

For each measurement, we also observe the peak AoI per shift. As shown in Figure 8, the peak AoI follows a similar trend to its respective MDR measurement, achieving the best performance at around 100 ms shift. We also show in Figure 9 the probability of each peak AoI during a worst-case scenario ($\Phi_{1,2} = 0$), we can see that the longer scanning window from set 2 increases the probability of a lower peak AoI.

Further, we compared the results with the analytical model (Figure 10). As we see, the model results are similar to the experimental results, having a mean absolute error of 2%. Both achieve the best performance with 100 ms shift as the epochs are the most de-synchronized, eliminating the impact of blind time. The model also displays the advantage of the longer scan window for closer shifts, having a similar behavior throughout the shifts.

The overall results from the side by side tests show that the model is right in estimating the impacts of blind time

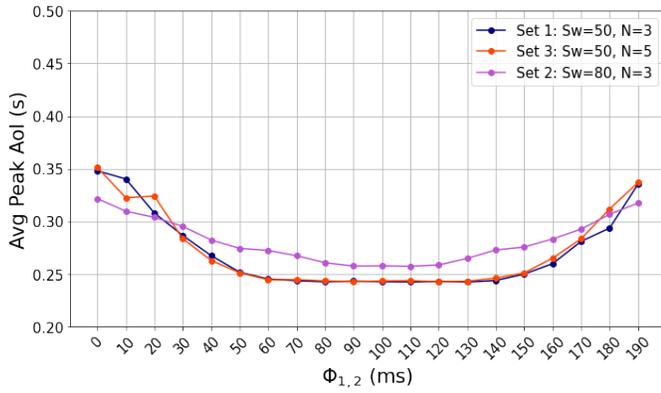


Fig. 8: Average Peak AoI per shift for all parameter sets.

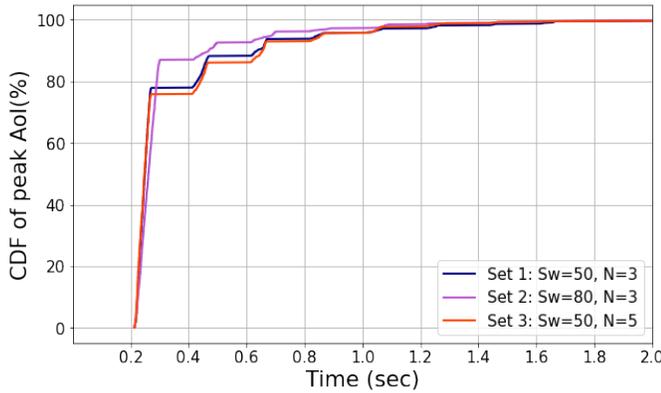


Fig. 9: CDF of peak AoI for each duration showing the probability of a received message having a certain peak AoI ($\Phi_{1,2} = 0$).

and parameter selection. As shown, synchronization between nodes has a significant impact on MDR, with $\Phi_{1,2} = 100ms$ being the best scenario among all parameters tested. The center values of $\Phi_{1,2}$, which are the furthest from the generation times, result in the best performance because the advertising start range got from Equation 3 is sufficiently shifted so that, in any combination of starts, the nodes do not overlap transmissions. Yet, as we cannot determine the initial shift, modifying the scan window is an alternative direction to counteract those effects. Looking into sub-question RQ1.a, we see in the tuning of the scan window a viable direction to maintain reliable communication under non-ideal scenarios.

C. Line of Sight range tests

In this section, we want to evaluate the impact of distance on communication. We used a similar setup to the side by side test, evaluating different shifts. However, instead of being indoors and with the nodes next to each other, the measurements were done outside and with the nodes separated by different distances (10, 20, 30, 40, 60, 80 and 100 meters), and we reduced the test duration to one minute per shift. To simulate bicycle conditions, we placed one node in a handlebar bag (one meter from the floor) attached to a bike and the

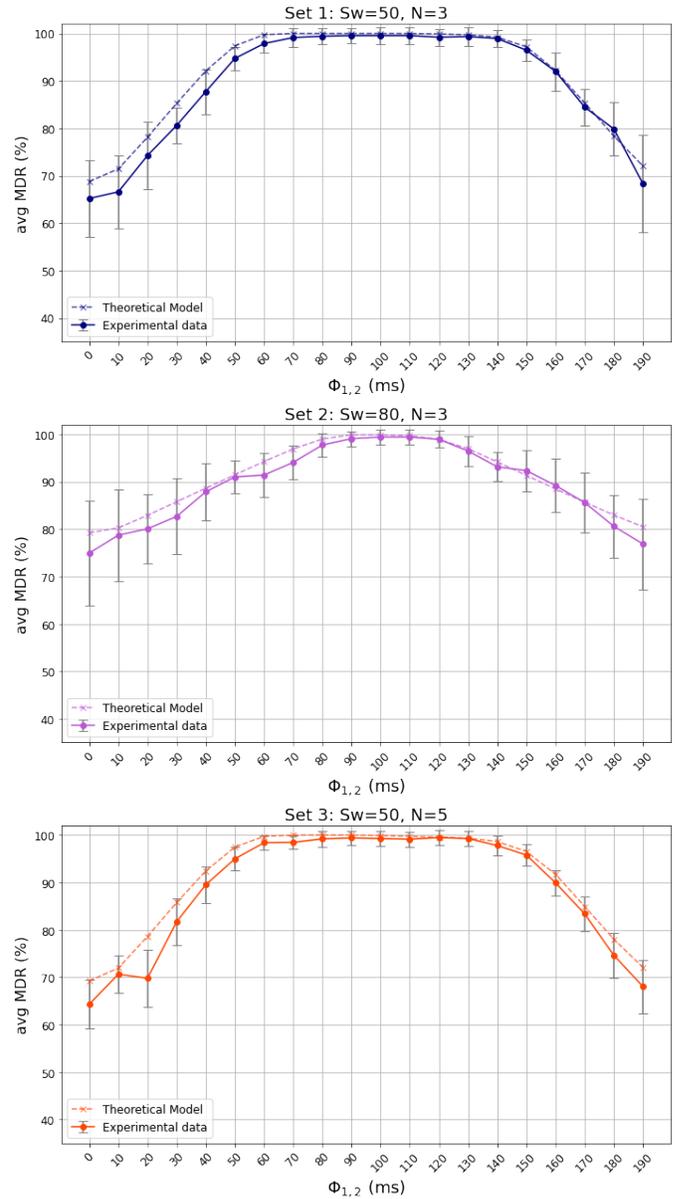
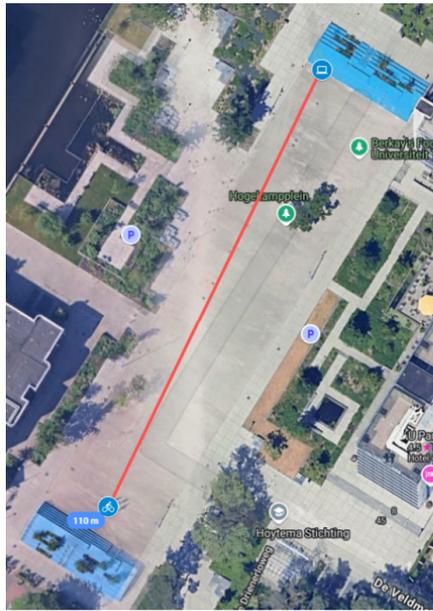


Fig. 10: Comparison between analytical model and experimental data for each parameter set.

other one at the same height. To achieve Line of Sight (LoS) conditions, the location of the test was HogeKampplein (Figure 11a), and all tests were carried out in dry weather.

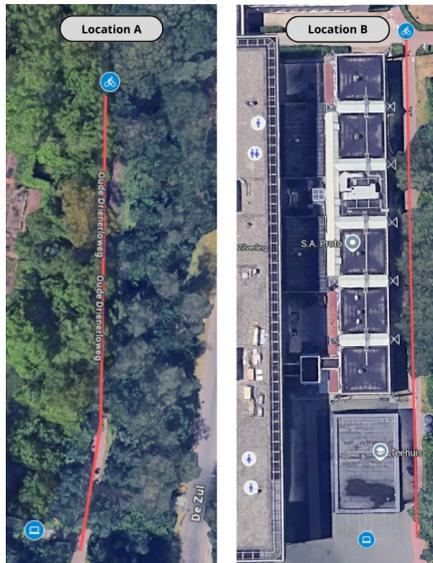
Figure 12 shows the mean and variance of MDR when combining all distance measurements. We observed that the distance between nodes does not have a meaningful impact on MDR. Although the RSSI varies (Figure 13), it does not change a significant amount to impact the reception, resulting in a pattern per shift similar to the side by side.

To verify the performance, we compared the results with a sniffer. There we saw that with a 100% duty cycle, we have the majority (around 99%) of packets are successfully received, which means the change in MDR remains due to blind time.



= Static node connected to computer at similar height
 = Static node at handlebar bag

(a) LoS



= Static node
 = Moving node

(b) Non-LoS

Fig. 11: Satellite view of the test locations. The bike and computer icons represent where the nodes were positioned.

Moreover, the measurements during the range test give us a ground truth reference to analyse the Non-LoS tests. These comparisons lead us to understand the effects of the medium on the quality of the communication (sub-question RQ1.b).

D. Non-LoS tests

In this batch of tests, our goal was to evaluate MDR and Discovery latency under more realistic settings. The measure-

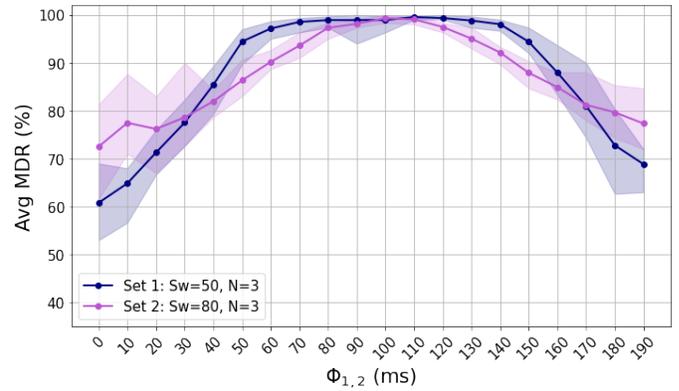


Fig. 12: MDR variance during range test. Solid line represents the mean MDR throughout all distances while the filled-in space shows the range of MDR values per distance.

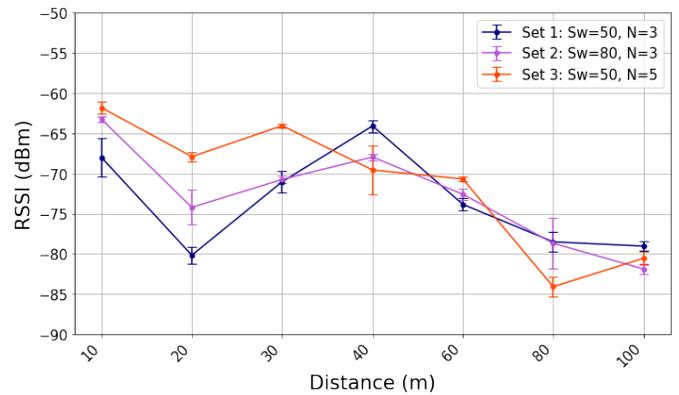


Fig. 13: Average RSSI per Distance during LoS range test.

ment strategy was then to have a moving bike approaching an intersection and a static bike at a (visually) blind spot. The tests were done in two locations, in Figure 11b, where the main difference is that LoS was blocked by vegetation or buildings. We considered the receptions within a 100 m range as it gives the rider enough time to react, but it is not too far, so the information becomes unnecessary. Additionally, because we do not have the instant location of the bicycle in the packet, we use an action camera to record both an LED that blinked whenever a packet was received, and the floor, where we marked the distance from the intersection. Later, we analyze the footage to identify the position of the bike when the first message was received within the 100 m range.

Regarding the code, we made small modifications to the implementation. Since we wanted to focus on a worst-case shift, we tested only when the message generations happen at the same time without any shift. And, because we wanted to measure specific moments, the new test state was now triggered by a button instead of a timer. By doing so, we could disable the recording as the bike was going to the start position and enable it again when the test started.

After 20 repetitions for each location and set, we plotted a Complementary Cumulative Distribution Function (CCDF) to

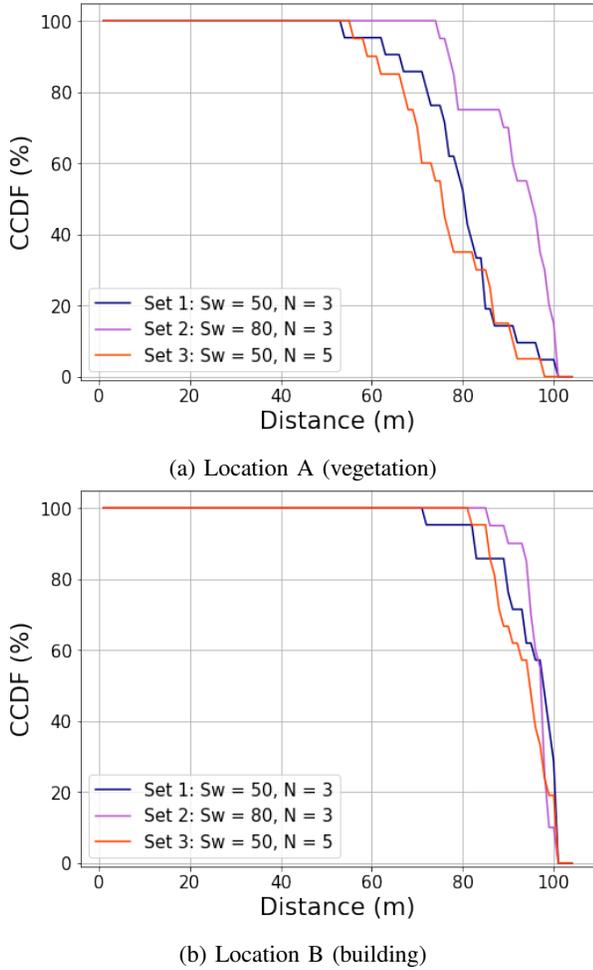


Fig. 14: CCDF of the distance from the intersection for the first received message.

determine the probability of a packet being received at each distance. The CCDF $F(x)$ is found using:

$$F(x) = 1 - P(X > x) \quad (4)$$

where $P(X > x)$ is the probability of the distance X , from the first received message, being higher than the distance x . As expected from earlier results, the configuration with $S_w = 80ms$ showed better probabilities, as seen in Figure 14. That is explained by the minimum epoch shift of the nodes, which is advantageous for longer S_w values. We also observed that vegetation had a greater impact on reception than the building, as the large number of leaves on the path reduces signal strength.

Later, we analyzed the packet arrivals saved in the SD cards. From the CSVs, we plotted the average AoI and the number of packets received per second. In Figure 16, we can see that AoI remained relatively low with values mostly between 400 ms and 1 second, with set 2 having the lowest due to its better reception. The second graph (Figure 15) gives a more in-depth view of reception as the bikes approximate each other by showing the average number of messages per window duration

of one second from the intersection. There, we can see that set 2 tends to receive messages earlier and in larger amounts than the other sets, and that all sets receive messages even before the comfort zone. Also, we observe that the number of detected messages increases as the nodes get closer, as a result of having a better signal strength. Keep in mind that, as $T_{gen} = 200ms$, the maximum number of packets that can be sent per second is 5, not including copies. It is also relevant to point out that set 2 showed a detection range further than 100 m in some tests, but since we chose a range of 100 m, only packets received in this range were considered.

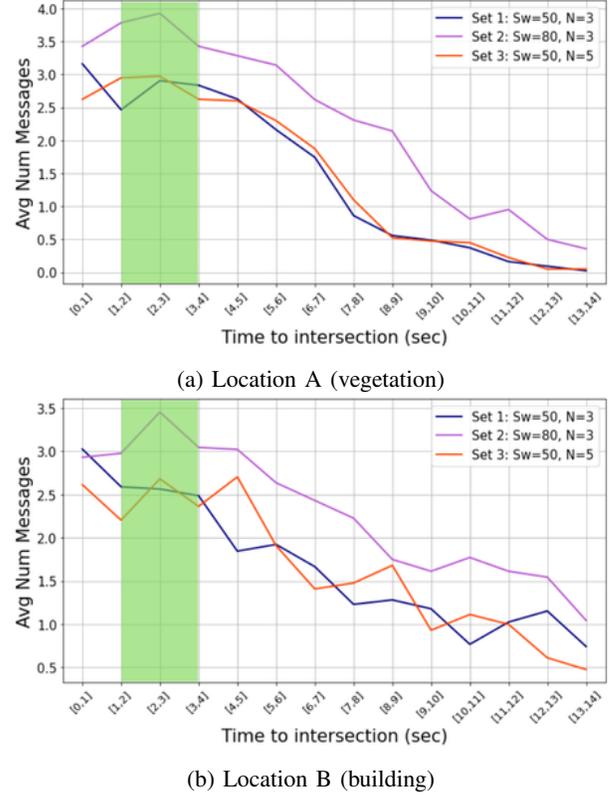


Fig. 15: Average packets per window of one second to the intersection and comfort zone for maneuvering (in green) based on [10].

In summary, the Non-LoS tests helped us identify two key behaviors. First, larger scan windows show a significant advantage over shorter windows on worse-case conditions ($\Phi_{1,2} = 0$). Second, sets 1 and 3 showed very similar results, meaning that we could increase the number of copies on some occasions to decrease the scanning duty cycle and power consumption without having a major impact on communication efficiency, under similar conditions. Furthermore, as we simulate real conditions (obstructed propagation and bike movement), the results demonstrate that B2B wireless communication, under the tested conditions, could allow real-time surrounding awareness. Observing the discovery range and the number of messages per one-second window, most trials demonstrate that the cyclist would receive a warning within and earlier than

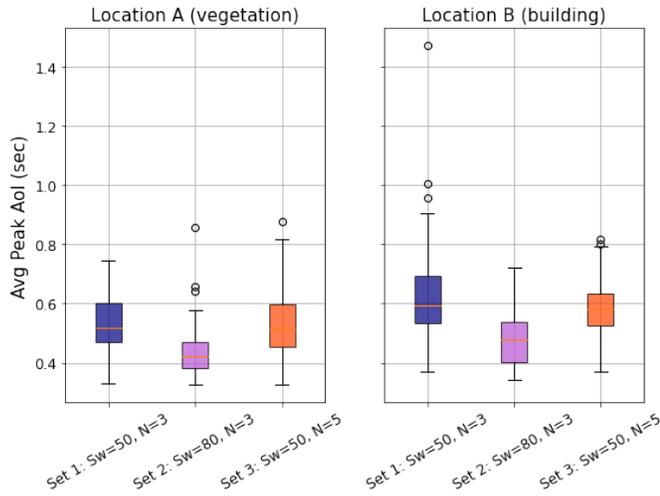


Fig. 16: Distribution of the average Peak AoI during Non-LoS tests.

their comfort zone. Moreover, comparing test locations, we see that the type of obstruction (vegetation or buildings) also impacts the performance of the system.

E. Energy consumption

To determine the energy cost of communication, we measured the current in different stages with an oscilloscope. To perform the measurement, we used the "energy measurement" pins from the nRF5340 DK. By connecting each to both ends of a 10 ohm resistor, we could then use the oscilloscope to see the voltage drop across and calculate the current in each stage. To focus solely on the communication, we modified the code to remove unrelated modules such as the SD card and the LEDs. Since the difference between the parameter sets is only the duration of each phase, we measured the currents for set 1 and calculated battery life for the other sets based on the currents found.



Fig. 17: Oscilloscope screenshot from a packet transmission on the 3 advertising channels.

From the oscilloscope, we identified that the current increases in 3 sections: during scanning, during packet transmis-

sion, and during mode switch. Those sections have an average current of 7 mA while the idle sections have an average of 2.5 mA. Furthermore, we observed that the transmission duration in all channels (in Figure 17) takes approximately 1.3 ms while the mode switch takes 4 ms. Considering the epoch period of 200 ms and 150 mAh coin battery, we can calculate the number of epochs and battery life per charge by multiplying the current by the duration of each section.



Fig. 18: Oscilloscope screenshot from a full advertising duration ($S_w = 50$, $N = 3$).

Using Equation 1, we estimate that the average D_{AdvDur} of each set, which matched approximately to the duration observed on the oscilloscope (Figure 18). By subtracting D_{AdvDur} from T_{gen} , we find the scanning time per epoch. Table IIIa shows the calculated durations and costs per epoch.

Phase	Duration (ms)	Energy cost (10^{-6} mAh)
Mode switch (high phase)	3.4	6.6
Mode switch (low phase)	0.6	0.4
Copy transmission	1.3	2.5
Idle ($N = 3$)	52.3	36.3
Idle ($N = 5$)	99.7	69.2
Scanning ($N = 3$)	137	266.3
Scanning ($N = 5$)	87	169.1
TOTAL ($N = 3$)		324.1
TOTAL ($N = 5$)		264.8

(a) Average energy cost in a single epoch

N	Epochs	Battery life
3	462,820	25.7 h
5	566,465	31.5 h

(b) Overall battery life estimation based on energy consumption

TABLE III: Energy cost and battery life analysis for different N values.

From the Table IIIa values, we can determine the overall battery life for a coin battery. Table III shows we would be able to do 462,820 epochs when $N = 3$ and 566,465 epochs with $N = 5$. Multiplying by the duration of the epoch we get 25.7 hours and 31.5 hours, respectively. The battery life shows that with any of the parameter sets tested, the system would comfortably have a full day of continuous use before requiring to be charged, or multiple days if disabled

when the bike is not in use. Besides that, we see from the oscilloscope measurements the impact of parameter tuning on energy consumption (sub-question RQ1.a). We see that increasing the advertising phase via the number of transmitted copies is a viable way to change energy use. On the other hand, changing the scan window has no impact on the consumption per epoch.

VI. FUTURE WORK

Although the test performed already shows a positive behavior of our design, there are still some aspects that need to be studied before having a final system. Initially, due to limited resources, we were not able to significantly increase the number of B2B nodes to evaluate the impact of multiple nodes communicating simultaneously. For a final implementation, we should know how to adjust parameters to reduce the impact of medium load, since in a real scenario, there would be multiple bicycles using our system.

Furthermore, as pointed out in our implementation, we did not implement real-time locational data on the packet. Although we reserved a space in the packet to transmit location coordinates, such a feature was not implemented. For future development, GNSS and IMU data could be added to fill the packet and allow the receiving node to estimate where the signal is coming from. Additionally, pre-loaded location information and medium sensing could be used to adapt communication parameters to optimize medium use, message delivery, or energy consumption at runtime.

Finally, despite the energy measurements made, further work could be done on optimizing energy consumption involving all modules of the system. Knowing the energy cost of external sensors and being able to estimate the position, we could reduce or modify the frequency of scanning and transmission. Besides that, based on researching the impact of medium load, we should find alternatives to reduce energy use without increasing the number of packets sent by having a scan window smaller than the scan interval, or increasing the advertising window.

VII. CONCLUSION

In summary, the paper gives a practical overview of the performance of B2B communication, evaluating characteristics such as node synchronization and transmission propagation. We base ourselves on previous work done with the analytical modeling of this use case, expanding on their discoveries by conducting practical trials. And, by comparing sets of parameters, we analyze how tuning the system can change the MDR, latency, and battery life.

To conclude, the current system achieves significant performance for use on bicycle commuting scenarios. Communication efficiency, latency and range show considerable reduction as conditions become more complex with less LoS or non-optimal synchronization. Yet, the results demonstrates that the minimum requirements for time-critical applications are still met, and packets are able to carry useful information to improve bike rides in a city environment. Moreover, test

outcomes present the scan window and the number of copies as important tunable parameters that affect delivery success and energy consumption. Increased scan window showed significant improvement in MDR under worst-case synchronization. While, under our test, increasing the number of copies increased in 6 hours the battery life without major effects on reception. With further work on implementing location finding and parameter optimization, our design has the potential to be a beneficial addition to smart city infrastructure.

VIII. ACKNOWLEDGMENT

With this paper, I am pleased to conclude my Master's thesis as part of the Smart Connected Bikes group. I would like to extend my sincere gratitude to my daily supervisor, Khalil Ben Fredj, for his continuous support and guidance throughout the thesis with our weekly meetings and test sessions. I am also grateful to Dr. Ing. Yanqiu Huang and Prof. Dr. Ir. Geert Heijenk for joining on the committee and for their valuable feedback and time.

Finally, I would like to thank my friends for their encouragement and motivation, and my family for their immeasurable support and the many opportunities they have given me throughout my journey in the Netherlands.

REFERENCES

- [1] Euronews. (2024) Cycling is now more popular than driving in the centre of paris, study finds. Accessed: 2024-09-17. [Online]. Available: <https://www.euronews.com/green/2024/04/12/cycling-is-now-more-popular-than-driving-in-the-centre-of-paris-study-finds>
- [2] K. G. Georgios Kapousizis, Mehmet Baran Ulak and P. J. M. Havinga, "A review of state-of-the-art bicycle technologies affecting cycling safety: level of smartness and technology readiness," *Transport Reviews*, vol. 43, no. 3, pp. 430–452, 2023. [Online]. Available: <https://doi.org/10.1080/01441647.2022.2122625>
- [3] K. Ben Fredj, G. Heijenk, and Y. Huang, "Modeling of mutual connectionless ble discovery in the context of b2x communications," in *IEEE Vehicular Networking Conference*, 2025, 16th IEEE Vehicular Networking Conference, VNC 2025, VNC 2025 ; Conference date: 02-06-2025 Through 04-06-2025. [Online]. Available: <https://vnc2025.ieee-vnc.org/>
- [4] O. B. Piramuthu, "Connected bicycles—state-of-the-art and adoption decision," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 987–995, 2017.
- [5] S. M. Kumar, A. Mohanty, R. Raman, M. Muthulekshmi, and A. Barve, "Smart biking: Iot-connected cycling gear for training and safety," in *2023 Second International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, 2023, pp. 652–656.
- [6] F. Oliveira, D. G. Costa, L. Lima, and I. Silva, "ibikesafe: A multi-parameter system for monitoring, evaluation and visualization of cycling paths in smart cities targeted at cycling adverse conditions," *Smart Cities*, vol. 4, no. 3, pp. 1058–1086, 2021. [Online]. Available: <https://www.mdpi.com/2624-6511/4/3/56>
- [7] N. Elhousseiny, M. Sabry, and H. Soubra, "B2x multiprotocol secure communication system for smart autonomous bikes," in *2023 IEEE Conference on Power Electronics and Renewable Energy (CPERE)*, 2023, pp. 1–6.
- [8] N. Papadakis, N. Koukoulas, I. Christakis, I. Stavarakas, and D. Kandris, "An iot-based participatory antitheft system for public safety enhancement in smart cities," *Smart Cities*, vol. 4, no. 2, pp. 919–937, 2021. [Online]. Available: <https://www.mdpi.com/2624-6511/4/2/47>
- [9] D. Manasreh, S. Swaleh, and M. D. Nazzal, "Evaluation of ble beacon technology for time critical i2v communication to support cav deployment on urban roadways," *Internet of Things*, vol. 24, p. 100932, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S254266052300255X>

- [10] O. Lee, A. Rasch, A. L. Schwab, and M. Dozza, "Modelling cyclists' comfort zones from obstacle avoidance manoeuvres," *Accident Analysis Prevention*, vol. 144, p. 105609, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0001457519317294>
- [11] G. Shan and B.-h. Roh, "Performance model for advanced neighbor discovery process in bluetooth low energy 5.0-enabled internet of things networks," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 12, pp. 10965–10974, 2020.
- [12] P. H. Kindt, S. Narayanaswamy, M. Saur, and S. Chakraborty, "Optimizing ble-like neighbor discovery," *IEEE Transactions on Mobile Computing*, vol. 21, no. 5, pp. 1779–1797, 2022.
- [13] B. Rana, Y. Singh, and P. Singh, "A systematic survey on internet of things: energy efficiency and interoperability perspective," *Transactions on Emerging Telecommunications Technologies*, vol. 32, 2020.
- [14] B. Luo, J. Gao, and Z. Sun, "Energy modeling of neighbor discovery in bluetooth low energy networks," *Sensors*, vol. 19, no. 22, 2019. [Online]. Available: <https://www.mdpi.com/1424-8220/19/22/4997>
- [15] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11734–11753, 2012. [Online]. Available: <https://www.mdpi.com/1424-8220/12/9/11734>
- [16] T.-T. Yang and H.-W. Tseng, "Improving discovery process toward user engagement based on advertising extensions in bluetooth low energy networks," *IEEE Transactions on Mobile Computing*, vol. 21, no. 9, pp. 3176–3192, 2022.
- [17] M. Costa, M. Codreanu, and A. Ephremides, "On the age of information in status update systems with packet management," *IEEE Transactions on Information Theory*, vol. 62, no. 4, pp. 1897–1910, 2016.