

# Continuous Athlete Monitoring in Challenging Cycling Environments using IoT Technologies

Esteban Municio\*, Glenn Daneels\*, Mathias De Brouwer†, Femke Ongenaë†, Filip De Turck†, Bart Braem\*, Jeroen Famaey\* and Steven Latré\*

\**IDLab—Department of Mathematics and Computer Science, University of Antwerp—imec, Belgium*  
{esteban.municio,glenn.daneels,bart.braem,jeroen.famaey,steven.latre}@uantwerpen.be

†*IDLab—Department of Information Technology, Ghent University—imec, Belgium*  
{mrdbrouw.debrouwer,femke.ongenaë,filip.deturck}@ugent.be

**Abstract**—IoT-based solutions for sport analytics aim to improve performance, coaching and strategic insights. These factors are especially relevant in cycling, where real-time data should be available anytime, anywhere, even in remote areas where there are no infrastructure-based communication technologies (e.g. LTE, Wi-Fi). In this paper, we present an experience report on the use of state-of-the-art IoT technologies in cycling, where a group of cyclists can form a reliable and energy efficient mesh network to collect and process sensor data in real-time, such as heart rate, speed and location. This data is analyzed in real-time to estimate the performance of each rider and derive instantaneous feedback. Our solution is the first to combine a local body area network to gather the sensor data from the cyclist and a 6TiSCH network to form a multi-hop long-range wireless sensor network in order to provide each bicycle with connectivity to the sink (e.g. a moving car following the cyclists). In this work we present a detailed technical description of this solution, describing its requirements, options and technical challenges. In order to assess such a deployment, we present a large publicly available data-set from different real-world cycling scenarios (mountain road cycle racing and cyclo-cross) which characterizes the performance of the approach, demonstrating its feasibility and evidencing its relevance and promising possibilities in a cycling context for providing low-power communication with reliable performance.

**Index Terms**—CONAMO, 6TiSCH, Industrial IoT, stream reasoning, real-time feedback, cycling.

## I. INTRODUCTION

The use of sport analytics for cycling is already fundamental and widely used, not only in professional events but also in the amateur world. By monitoring cycling metrics such as heart rate, cadence, instantaneous power and speed it is possible to significantly improve the performance of a cycling team and each of its members according to different coaching strategies. Cycling can normally take several hours (an average stage of the *Tour de France* takes around 5 hours) and the eventual result mainly depends on how cyclists distribute their energy and on how cyclists interact with each other. For this reason, successful coaching plays a very important role in cycling. Although traditionally this has been done by radio voice-communication or visual signals, current IoT technologies allow significantly improved monitoring and much more efficient professional coaching.

However, current cycling monitoring techniques rely on infrastructure-based communication technologies (e.g. LTE, LoRa, etc.). But because the high spottiness and scarcity

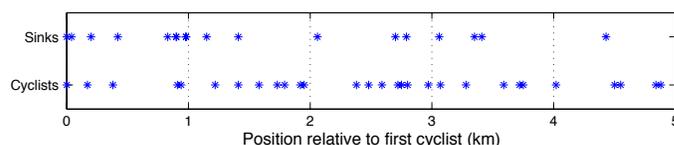


Fig. 1: Position of cyclists and sink points (motorbikes, cars or helicopters) in an actual *Tour de France* stage (Alpe d’Huez on July 25, 2015).

of cellular coverage in mountain areas, professional events deploy literally a fleet of vehicles such as motorbikes, cars and helicopters in order to ensure one-hop signal availability from each rider. For example, Figure 1 represents the number of used sinks in order to ensure sensor data backhauling from a subset of riders. This becomes economically challenging in professional events and impossible for amateur cycling.

In this work we propose instead to leverage state-of-the-art IoT technologies to provide reliable, energy efficient communication and deterministic performance in a dynamic mesh network, in which each bicycle is a node in the network and relays sensor data towards a single sink, where it will be processed in real-time. Sensor data is gathered at each bicycle through ANT+ [1], a popular Wireless Personal-Area Network (WPAN) technology widely used in cycling. This sensor data is then relayed to the sink through 6TiSCH [2], an IPv6-enabled technology that combines the TSCH (Time Slotted Channel Hopping) mode of IEEE 802.15.4e, which regulates medium access, and RPL (Routing Protocol for Low Power and Lossy Networks) (RFC 6550) [3], which builds the routes from each node to the sink. However, deploying these types of networks is challenging. Cycling is highly dynamic and this requires optimization of the 6TiSCH network standard, which until now, has been mostly applied to static networks. In this sense, we argue that by modifying and optimizing the 6TiSCH scheduling and routing layers, sensor data can be successfully delivered while tolerating high levels of dynamicity.

In order to demonstrate this, this article provides an in-depth technical description of the CONAMO project (CONTinuous Athlete MONitoring)<sup>1</sup>, an innovative proof-of-concept system that uses an improved version of the multi-technology platform

<sup>1</sup><https://www.imec-int.com/en/what-we-offer/research-portfolio/conamo>

presented in our previous work [4], in order to gather and relay cycling sensor data towards the sink node and analyze it in real-time using our data analytics back-end [5]. We also provide a large publicly available data-set (containing more than 25000 data points) from two real-world test-beds (mountain road cycle racing as proof-of-concept and cyclo-cross racing as a more thorough study) and present the changes made in the 6TiSCH protocol stack in order to tackle network dynamics. Finally, we show how the data processing performed on the received data can help coaching and improve cycling performance.

The remainder of this article is organized as follows. Section II presents some relevant background on related technologies in the field. Section III describes in detail the CONAMO architecture at different levels: hardware prototypes, networking and data processing machinery. Finally, Section IV presents some results of the experiments performed and Section V concludes this paper.

## II. RELATED WORK

First, this section briefly discusses the related work on monitoring in challenging environments, then introduces the related work on real-time data analytics in sports, and finally gives an overview of the selected wireless technologies.

### A. Connectivity in Remote Environments

A large part of current research on IoT is focused on smart cities and industrial applications where networks are highly accessible, static and depend on a dense infrastructure of interconnected base stations [6]. This is not the case for cycling, which is highly dynamic and usually takes place in isolated and remote environments without proper coverage of infrastructure-based networks.

For these scenarios, solutions based on long-range LPWAN (Low-Power Wide-Area Network) are preferred [7] such as NB-IoT, LoRa and SIGFOX. Although these technologies have a range of multiple kilometers, they do not always offer sufficient throughput for real-time monitoring and still depend on cellular infrastructure, that although sparse, still requires to be attached to a high-speed backhaul network. Also other solutions based on meshed low-power WPANs have been proposed for monitoring purposes in rural and remote areas [8], [9] which use technologies such as Zigbee, DASH7, 6TiSCH or other proprietary solutions [10]. But again, they require a backhaul to provide connectivity to the network. Although some works address the problem of interconnecting remote areas using multi-hop networks [11], their use to provide connectivity to cycling events is not considered due to the sporadic and dynamic nature of this use case.

Cycling monitoring requires a totally different approach, which has to combine the medium and topology variability of a MANET (Mobile Ad-hoc Network) [12], with the low power consumption and deterministic performance of state-of-the-art IoT networks. This is the approach we follow in the present article, previously introduced in our previous work [4]. However, while in that work a general-purpose multi-technology architecture was presented, in this work we

provide an in-depth study on how such an architecture is applied and performs in cycling sports. This includes a further optimization of the previously presented approach in both the hardware and software in order to do so.

Our approach, unlike other cycling platforms, does not rely on a large mobile infrastructure with multiple single-hop networks (deployed on motorbikes, cars, helicopters, etc.) that follow the bikes [13]–[15], nor does it rely on cellular coverage to send sensor data from a traditional cycling application [16], [17]. Instead, it leverages the swarm nature of the groups of bikes to provide easy-to-setup ad-hoc connectivity to sporadic sinks. Our approach is the first that does this.

### B. Data Analytics for Real-time Feedback in Sports

In order to provide real-time personalized cycling feedback based on the current context, intelligent data consolidation and analysis is required of the heterogeneous and voluminous data provided by the various sources [18], [19]. These sources include IoT sensor data streams, background knowledge (e.g. what heart rate thresholds exist), and context information (e.g. sensor and profile data). Ontologies and semantic reasoning can ideally be used to model the data, and perform advanced processing tasks that enable the design of personalized and context-aware algorithms [19]. Stream reasoning [20] platforms, such as C-SPARQL [21], use these techniques to perform the real-time detection of interesting events and patterns in a continuous stream of data, by mapping it on the available background knowledge modeled in an ontology. They place a window on top of the data stream, and registered semantic queries are continuously evaluated as the streaming data passes through the window [22].

To make sure a real-time feedback system can also be used in amateur cycling, it should take into account the limited amount of resources available in the low-end devices, such as tablets, smartphones and GPS devices. While some techniques propose to tackle this by optimizing the format used for exchanging and querying RDF data in memory-constrained environments [23], other works focus on optimizing the semantic reasoning process itself [24]. Works studying the scalability and real-time response of different reasoning approaches in realistic IoT environments with mobile devices demonstrate that Semantic Web technologies are adaptable to the IoT [25].

In existing research, a number of ontology-based context platforms have been proposed [18]. The feedback provided by these platforms is however not sufficiently real-time, personalized, and context-aware. Moreover, existing distributed semantic reasoning frameworks do not target low-end devices [26]. In general, IoT data processing frameworks do mostly not support real-time local data processing for critical applications with limited local resources [27].

Existing commercial real-time sports analytics solutions offer many offline post-processing features, but do not focus on real-time, personalized and context-aware feedback. Moreover, they usually do not allow to integrate heterogeneous sensor and data sources. In the presented approach, which is complementary to the offline features, this is all possible. In this way, the data-driven cycling experience is extended from ‘post-event’ to ‘during event’, by analyzing the sensor data in real-time.

### C. Network Technology Selection

This subsection presents an overview of the technologies used in CONAMO: ANT+ for the wireless communication with the sensors and 6TiSCH to build the mesh network.

1) **ANT+ Protocol:** ANT+ [1] is a proprietary protocol specifically designed for health and sport applications. Very popular within the cycling world, it offers very low power consumption and high reliability. This is achieved following a TDMA (Time Division Multiple Access) master-slave approach, in which the sensors are synced with the master device using short frames of 150  $\mu$ s for transmitting 8 bytes. It operates in the 2.4 GHz band with a bandwidth of 1 MHz divided into 125 subchannels. The number of available subchannels and the short duration of the frames, makes the protocol especially robust against internal interference, allowing hundreds of devices to operate in the same area simultaneously [28]. The format and content of the frames is defined by the profiles, each of them designed for one specific goal: heart rate monitor, blood pressure, muscle oxygen monitor, suspension systems, etc.

2) **6TiSCH:** 6TiSCH is becoming one of the most extensive standards for deterministic IoT that combines the IEEE 802.15.4e TSCH (Time Slotted Channel Hopping) link-layer mode with an IPv6-enabled upper stack [2]. The key layer of its architecture is the TSCH layer, which divides the time in slots (typically 10 ms, enough to transmit a packet of maximum 128 bytes and receive a short acknowledgement at 250 kbps), grouping them in slotframes (typically 101 slots per frame). Channel hopping is achieved by assigning a channel offset to each timeslot and pseudo-randomly rotating the transmission channel every time the ASN (Absolute Sequence Number) is increased. This builds the schedule (i.e. when a node has to transmit, receive or sleep). Schedule information is carried in the Enhanced Beacons (EBs), which are required to allow new nodes to discover and synchronize with the network. Although the 2.4 GHz band is the most used, it is also possible to use the subGHz band (e.g. 868 MHz in Europe). In this work we focus on the subGHz band because of its longer range.

6TiSCH also provides the 6top sublayer, which mainly contains the Scheduling Function (SF) and the Sixtop Protocol (6P) [29], the protocol used to manage the schedule in a distributed manner. The default SF in 6TiSCH is the Minimal Scheduling Function (MSF) [30], which allocates or deallocates cells according to the traffic demands using 6P. Additionally there is a housekeeping function that periodically manages the state of the schedule (e.g. to detect collisions).

For routing, 6TiSCH uses RPL, a distributed protocol that builds a Destination Oriented Directed Acyclic Graph (DODAG), where all traffic is directed towards a root node or *sink*. Within the DODAG, each node always has a preferred parent to transmit its packets to. The parent selection determines the multipoint-to-point routes and is a gradient-based decision, where nodes always choose as preferred parent the neighbor which has the lowest perceived rank. Nodes distribute their rank information by broadcasting DODAG Information Object (DIO) messages.

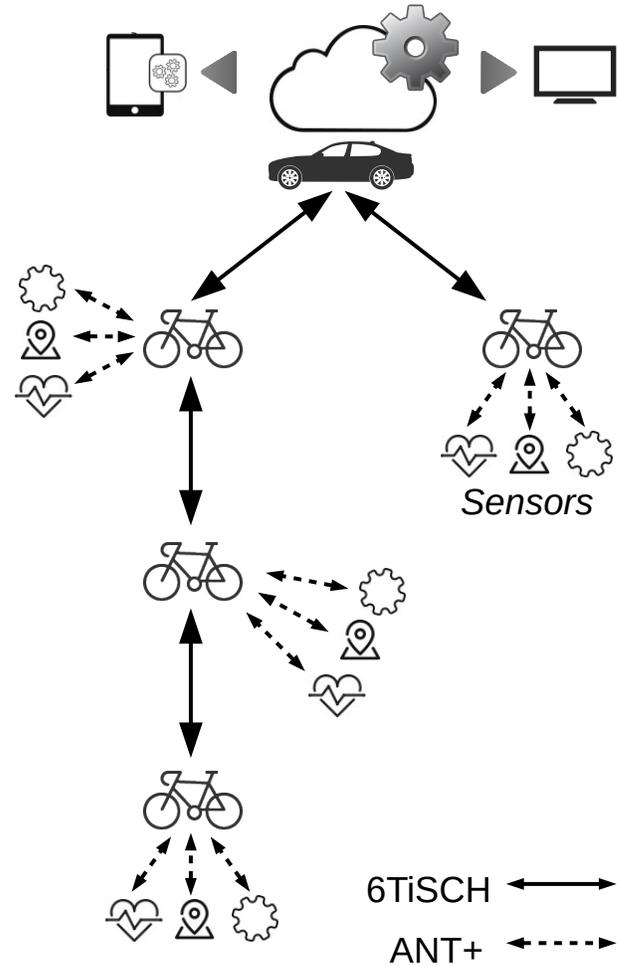


Fig. 2: Example of a CONAMO network.

### III. THE CONAMO ARCHITECTURE

In this section, we first introduce the general picture (Section III-A). Afterwards, we provide details on the prototypes (Section III-B), on the bicycle network (Section III-C) and finally, on the real-time monitoring system (Section III-D).

#### A. Continuous Athlete Monitoring

The CONAMO project encompasses real-time collection and analysis of sensor data during collective cycling to derive instantaneous training feedback. It aims to improve the cycling experience and performance of a group of cyclists by providing them with personalized insights on their performance. This is carried out by building a mesh network where each bicycle becomes a node in the network (as shown in Figure 2). Sensor data is locally collected on the bicycle originating from various sensors such as heart rate monitor, GPS and speed sensor, and subsequently relayed in a multi-hop manner from bicycle to bicycle to a sink node. In the sink node, e.g. a team car, sensor data is individually or jointly processed and a data processing system provides personalized and context-aware feedback in real-time in relation to a specific training or competition plan for coaching purposes. Data is also stored for off-line analysis (e.g. data can also be further relayed to press or jury).



© imec

Fig. 3: CONAMO devices enclosed in their cases.

This particular use case demands full connectivity and real-time monitoring even in areas where cellular connectivity is not present. The difference between CONAMO and other solutions is that the former leverages from the cycling group (e.g. a *peloton*) for relaying the data instead of using a huge amount of mobile infrastructure to cover a cycling event. Although this could also be applicable for professional competitions and *Grand Tours* to reduce costs, it is especially relevant for amateur group rides where the resources are modest. Also, it is highly convenient for ephemeral close-circuit training or competition events such as cyclo-cross, where full connectivity can be provided to the bicycles in the circuit by deploying few additional static devices. Although CONAMO sets a novel approach for monitoring in cycling, it comes with a cost.

The ad-hoc nature of the approach avoids being dependent on cellular coverage, but its highly dynamic nodes add significant complexity. RPL has turned out to over-react to link quality changes, adding instability and disturbing the convergence of the SF [31]. In that sense, RPL and the SF have to be modified to make them compatible with a dynamic environment. Additionally, as previously shown in Figure 1, hops from bicycle to bicycle can be up to 660 meters or even more (e.g. escaped cyclist), so that bicycles require a long transmission range in order to ensure full connectivity among the group. Finally, due to weight constraints in cycling, power consumption must be as low as possible to reduce battery weight. On the data analysis part, the challenge lays on designing the system (ontology and queries) that allows to extract valuable insights in real-time. In the following subsections we detail the CONAMO design that implements the stated approach and overcomes its related problems.

### B. Prototypes

Custom-built devices as the ones shown in Figure 3 have been specifically designed for the CONAMO project. The PCBs (Printed Circuit Board), consist of two modules: the 6TiSCH connectivity and the sensor connectivity.

- **6TiSCH Connectivity:** This module includes a CC2538 system-on-a-chip 32 MHz micro-controller. It is connected by SPI to a CC1200 radio that operates in the 868 MHz band and uses a subGHz antenna of 1.5 dBi.

- **Sensor Connectivity:** All sensor data is collected and aggregated in a Nordic nRF52 system-on-a-chip that includes a 2.4 GHz radio that supports BLE (Bluetooth Low Energy), ANT+, IEEE 802.15.4 based protocols and other proprietary protocols in that band. It can communicate with any commercial sensor that implements ANT+ profiles independently of the vendor<sup>2</sup>. Additionally, the PCB includes an in-built CAM-M8 uBlox GPS that directly communicates with the nRF52 by an I2C (Inter-Integrated Circuit) interface in order to provide continuous geographical location information.

Once aggregated and filtered, the sensor data is periodically sent (1 *pkt/s*) from the nRF52 to the CC2538 by UART in order to be further relayed in the 6TiSCH network. Additional components such as power switches or the CP2105 are included in order to allow easy charging and USB selection to independently flash the nRF52 and the CC2538. Current batteries have 2000 mAh of capacity which, according to our experimental tests, allow at least 2 days of continuous operation. Finally, the cases have been designed to be compatible with a standard device mount used by different vendors such as Xiaomi, GoPro or Sjcam. The total weight of the device including the case is 105 grams.

### C. The Bicycle Network

The CONAMO devices run OpenWSN [32] on the CC2538 MCU, which currently is the implementation that most closely follows the development of the 6TiSCH standard. However, a number of modifications have been made in the different layers of the stack in order to adapt OpenWSN to the requirements of a highly mobile 6TiSCH network.

1) **PHY and MAC Layer:** In order to obtain a longer range per hop, we integrated OpenWSN with the CC1200 radio to use the 868 MHz band, which has lower propagation loss. In order to enable effective communication between the CC2538 and the CC1200 radio through SPI without losing synchronization with the slotted MAC layer, we have adjusted the durations of the states inside the timeslot and updated the timeslot duration itself to 15 ms. We have kept the transmission rate to 250 kbps, but reduced the bandwidth from 833 KHz to 555 KHz in order to lower the incoming noise and therefore the signal to noise ratio. The modulation has been switched from 2-FSK to 2-GFSK which significantly reduces spurious contents. The resulting energy consumption in the subGHz band is slightly higher to the one of the 2.4 GHz, since besides the CC2538 MCU, it additionally requires to power the CC1200 radio [33].

2) **Sixtop Sublayer:** In the Sixtop sublayer, we have performed modifications in order to change the way schedules are built. Using 6P [29] and starting from MSF [30], we have developed a much more aggressive SF that reacts quicker to topology dynamics by applying the following changes:

- Set the number of minimal cells [34] to 5 to reduce the time of the first 6P allocation in a new neighbor and increase the EB and DIO transmission rate without congesting the broadcast cells.

<sup>2</sup>A full list of tested sensors is in <https://github.com/imec-idlab/conamo>

TABLE I: Parameters modified in scheduling function.

NUMCELLS_MSF	2
LIM_NUMCELLSUSED_HIGH	20
LIM_NUMCELLSUSED_LOW	8
HOUSEKEEPING_PERIOD	1515 ± 250 ms
WAITDURATION_MIN	5000 ms
WAITDURATION_RANDOM_RANGE	5000 ms
CONAMO_MSF_TIMEOUT	6 s

- Modify MSF parameters as shown in Table I for a stable and smooth scheduling performance with a traffic load of 1 *pkt/s* injected per node. We refer to [30] for a full description of these variables.
- Reset the sequence number upon receiving the response code SeqNumErr or CellListErr to react faster to schedule inconsistencies.
- Set the number of requested cells to 4 when no previous allocated cells were present. This is done in order to ease the future 6P transactions and to reduce the delay of the already-stored packets in the queue (if any). In case any cell is already scheduled, the number of requested cells in the new 6P ADD\_REQUESTs will be, as stated in Table I, NUMCELLS\_MSF = 2.
- Modify the MSF housekeeping function to remove cells that have not been used (i.e. successfully received an acknowledgement upon transmitting a unicast packet) in the last CONAMO\_MSF\_TIMEOUT (i.e. 6 s). When at least one cell meets this condition and an alternative parent exists, cells to that neighbor are removed.
- Remove previously dedicated cells (if any) in a parent node if a 6P ADD\_REQUEST is received in any of the minimal cells. Since 6P commands are only sent in the minimal cells when no other dedicated cells exist, by removing the previous dedicated cells before performing a new clean 6P ADD transaction, the synchronization between parent and child schedules will be ensured.

3) **Routing Layer:** The default RPL configuration has been proved to be suboptimal for mobile scenarios [35]. We here introduce a set of changes that allow to improve the routing performance of the network under mobile environments (such as VANETs). Unlike other approaches that require new or modified messages [36], we have opted to modify the following RPL parameters and behaviors to keep compliance with the standard as much as possible:

- Lower DEFAULTLINKCOST from 8 to 1.5 to ease the parent change when a new parent is available. This significantly reduces the rank threshold needed to change to the new neighbor, allowing the child to leave its unreachable current neighbor faster.
- Reset the blacklisted nodes every time a new parent is found in order to allow future parent switches back to the previous neighbor.
- Force rank poisoning in nodes that having child nodes, lose connectivity with their preferred parent. This allows to rebuild the DODAG and to avoid possible loops.

#### D. Real-Time Monitoring

For the monitoring of cyclists, the real-time feedback system presented in our previous work [5] has been further developed on. The main part of this system is a stream reasoning engine running on a low-end device.

The cycling ontology has been updated to ensure that the physiological profile of a cyclist can be unambiguously described. This allows to personalize the data analytics according to each rider’s profile. An overview of how such a profile is now represented in the ontology, is given Figure 4 a).

For the data analytics performed for CONAMO, the physiological profile of a rider mainly consists of the boundaries between the seven different heart rate training zones that exist in cycling [5]. These boundaries are calculated based on a rider’s resting and maximum heart rate, by applying the Karvonen formula [37]. For amateurs, a resting and maximum heart rate are typically estimated based on expertise rules of thumb [5]. Note that in the ontology, the boundaries between the different training zones are explicitly modeled as lower and upper bounds of the existing training zone individuals. This is the biggest change compared to the previous version of the ontology, where each training zone, including its boundaries, was modeled separately. By now linking the two training zones that each boundary is separating, it is enforced that two training zones can never overlap, and that each heart rate value between a rider’s resting and maximum heart rate belongs to exactly one training zone.

Figure 4 b) shows the part of the cycling ontology representing how the different sensors and observations are modeled. It focuses on the heart rate sensor of the rider’s CONAMO device (e.g. Lars), and an observation made with it. This representation uses the patterns defined in the SOSA ontology.

In Figure 5, an overview is shown of the inputs and outputs of the stream reasoning engine. It shows the example of how the heart rate observation, modeled in Figure 4, is combined with the available background knowledge and context data, also modeled in this figure, by the stream reasoning engine, C-SPARQL in this case. This engine executes the registered continuous queries on these inputs. Through these queries, the heart rate value – as well as any values measured by other sensors – is coupled to the corresponding rider, and the corresponding training zone is retrieved. Each continuous query outputs its results on a dedicated data stream.

To perform the real-time monitoring of cyclists, several continuous C-SPARQL queries are used. The generic `getQuantityObservationValue` query, and the similar `getLocationObservationValue` query, retrieve all quantity and location observations, respectively<sup>3</sup> [5]. Moreover, the `getTrainingZone` query retrieves the training zone corresponding to the most recent heart rate observation. It does this by looking for two heart rate thresholds of the corresponding rider, that are the lower and upper bound of the same training zone. If the observed heart rate is between the two bounds, this is the correct training zone.

<sup>3</sup>The used ontologies and queries are available online on <https://github.com/imec-idlab/conamo/>

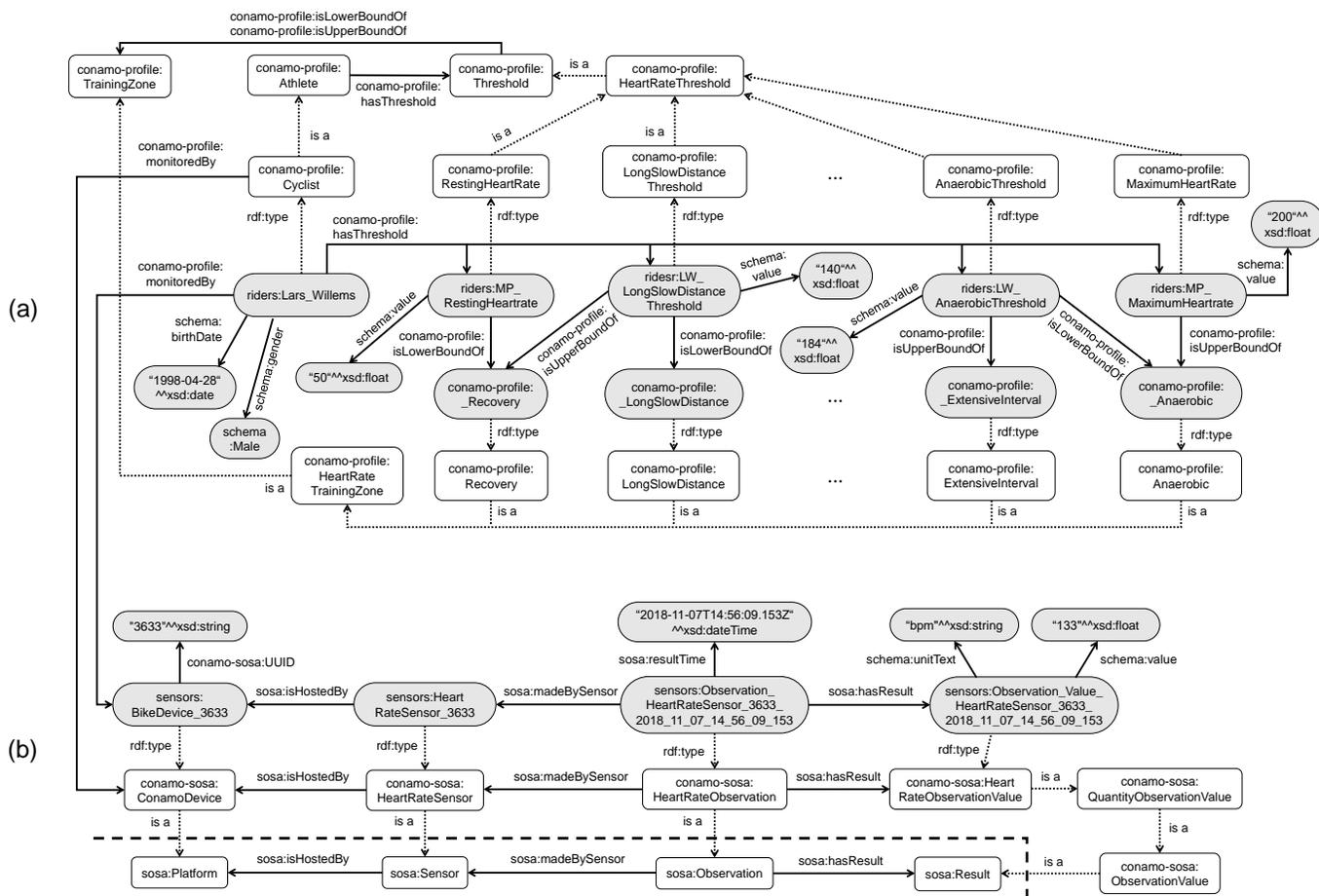


Fig. 4: Overview of the designed cycling ontology: (a) shows how a rider profile is described, (b) shows how sensors and observations are described, focused on the example of a heart rate sensor and observation. The white rectangles represent the ontology concepts, the gray ovals represent individuals, and the arrows represent object and data properties. The concepts separated by the dashed line are part of the SOSA ontology and not of the designed cycling ontology.

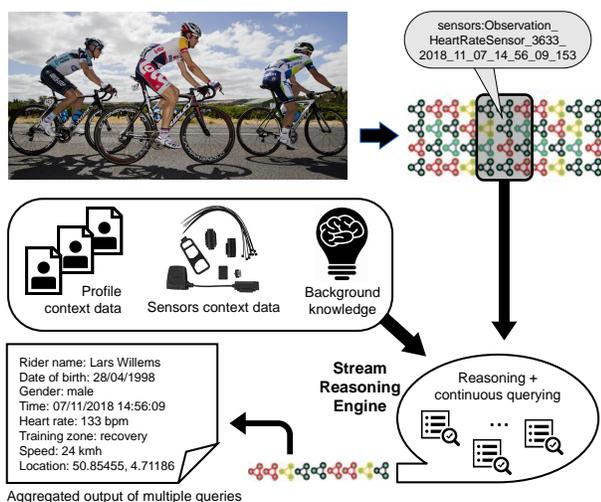


Fig. 5: Overview of the inputs & outputs of the stream reasoning engine of the real-time feedback system. The upper right observation individual refers to the corresponding individual in Figure 4. The background knowledge and context data are also partly shown by the individuals in Figure 4.

### E. Data Visualization

Besides the development of the real-time feedback system, front-end tools have been implemented to ease and improve coaching. These visualizations show feedback and information about the riders' performance, either by directly using the measured data, or by aggregating this data over a certain period of time. In the latter case, additional data analytics components are required, aggregating the real-time processing data on the result output streams of the described continuous queries.

Finally, an application for a tablet was implemented to display the received sensor data received directly from the sink node using BLE. The BLE link is directly established with the sink node using its on-board nRF52 chip.

## IV. EVALUATION

In this section we present: range results in Section IV-A, performance of a dynamic controlled setup in Section IV-B, proof-of-concept results of group cycling events in the French mountains in Section IV-C and more comprehensive results in a cyclo-cross event in Section IV-D, which includes semantic results obtained from the sensor data analysis<sup>4</sup>.

<sup>4</sup>The data-set is publicly available at <https://github.com/imec-idlab/conamo/>

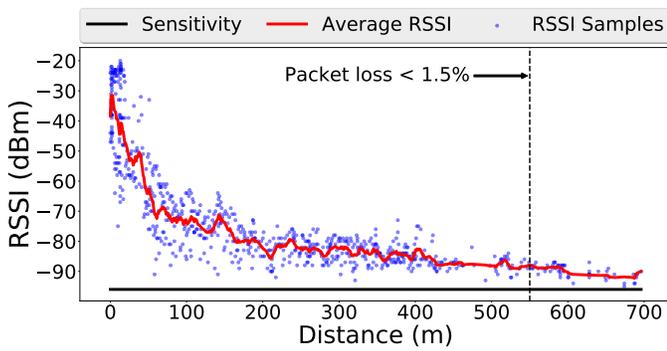


Fig. 6: Obtained RSSI values over distance in the range tests.

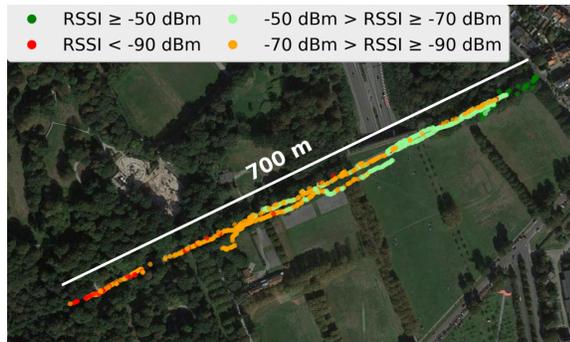


Fig. 7: Distribution of samples in the testing area.

### A. The Need for Long-Range

To increase the effectiveness of the CONAMO approach we aimed for long-range links to cover as much distance as possible. With the PHY configuration described in Section III, we performed range tests in the subGHz band in a controlled Line-of-Sight (LOS) scenario between two CONAMO devices. While the transmission range in the 2.4 GHz band is usually about tens of meters [38], Figure 6 and 7 show that in the subGHz band a maximum range of around 700 meters can be obtained. However, the practical range in which packet loss is lower than 1.5% is 550 meters approximately, where the RSSI is above -90 dBm on average. The sensitivity of the CC1200 radio (-97 dBm) is denoted as a horizontal line in Figure 6.

### B. The Need for Dynamics

We now show the behavior of the prototypes in a controlled multi-hop dynamic scenario. These tests were performed indoors with antenna attenuators (50 dB) in order to emulate the parent switching of a real outdoor scenario.

The tests consist of a mobile probing node (e.g. a bike) connected to different nodes in a linear network (e.g. other bicycles in the road) as shown in Figure 8. The probe node moves from being directly connected to the sink towards the end of the chain (i.e. connected to node 4). Afterwards, the probe node comes back towards the sink and the test stops some time after being again connected to the sink. Data is sent at 1 pkt/s from all nodes and the moving speed of the probe node is 1 m/s (walking speed approximately). This speed is significantly lower than the speed of a bike, but comparable to the relative speed between bikes in a cycling group.

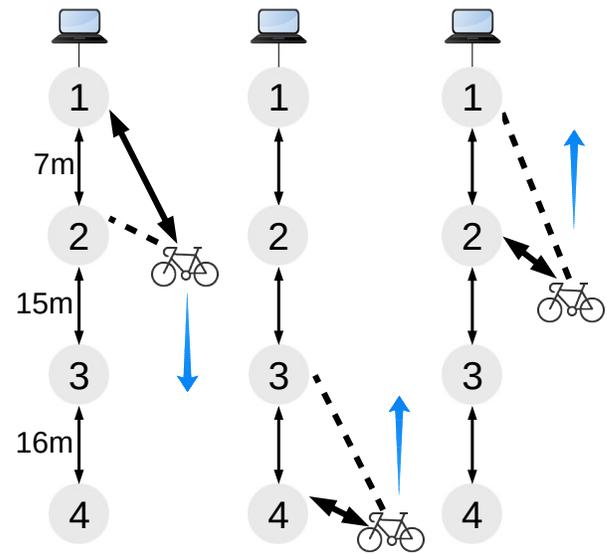


Fig. 8: Topology used for the indoor tests. Black arrows represent parent-child links. Dashed lines represent neighbor links.

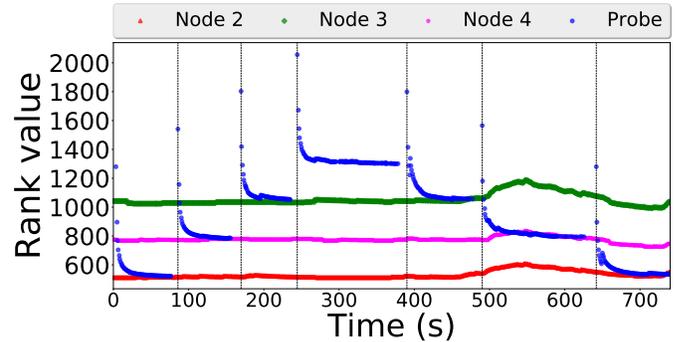


Fig. 9: Probe's rank when moving up and down along the network. Each point is a received packet in the sink node.

When using the CONAMO configuration in OpenWSN as described in Section III-C, some limitations are observed in Figure 9 at every parent switch (i.e. discontinuous vertical lines) due to different reasons. First, since the probe node has an antenna attenuator, in some areas the probe nodes cannot reach any of the nodes in the chain. Also, before triggering a parent switch, the link has to have an effective drop in performance to reach the switching threshold. Finally, after the parent switch, a new 6P transaction has to be performed before starting to send data packets again. This results in an average delay of 12.8 s during the parent switches, which provokes an average of 3.1 dropped packets at each parent switch, because the TX queue is limited to 10 packets.

However, apart from these temporary disruptions, the packet stream flows almost continuously at the expense of transient peaks in the rank value. These peaks are caused by the default rank assigned to the probe after the parent switch occurs, when the link quality has not been yet estimated (i.e. the ETX). After a number of successful transmissions, the rank value of the probe node quickly converges to a stable value ( $RANK_{PARENT} + RANK_{INCREASE}$ ).

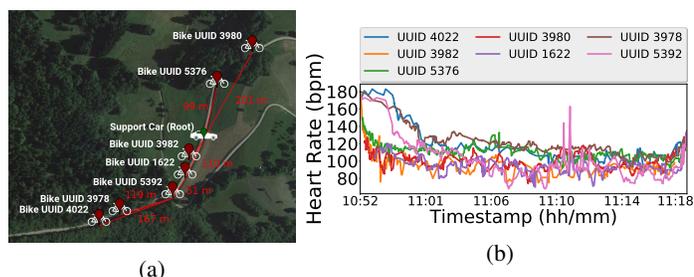


Fig. 10: Example of monitoring for a group of amateur cyclists in the climbing of the Col de l'Épine in the Jura mountains.

By using the default version of OpenWSN (Release 1.14), the connectivity is lost after the probe leaves the coverage range of the sink, and packets are continuously being dropped until the node comes back into its range again. Since the RPL layer is designed for static networks, the rank updates are rather conservative and parent switches only occur until the parent is marked as unreachable (which can take more than 10 minutes). This evidences the unsuitability of the default RPL configuration for a mobile environment.

#### C. Mountain Road Bicycle Racing Use Case

In order to test the prototypes in a real cycling environment we have performed a proof-of-concept field experiment in a real mobile test-beds. The test was conducted in the Jura mountains, in which up to 20 amateur cyclists were carrying CONAMO devices (attached as shown in Figure 12a) in order to monitor heart rate, speed and location. Meanwhile a support car was monitoring the real-time performance of the cyclists from the distance, either connected directly to them (1 hop) or by using other bikes as relays (multi-hop). Although results proved to be highly dependable from the actual topology, successful monitoring was ensured while the bikes were at maximum 5 hops from the car, being the maximum recorded distance between each bike of 432 m.

Figure 10 shows an example of such scenario, where riders' sensor data was collected during approximately 25 minutes while the support car was closely following the cycling group in the last 4 km of a climb. Figure 10a shows the topology at the instant that the car reaches the top of a climb and evidences that complex topologies are formed and can be successfully used to collect the sensor data. Figure 10b shows heart rate values of the riders with a 3.3% of packet loss.

#### D. Cyclo-cross Use Case

Finally, in this subsection we report results for a cyclo-cross training use case in a professional young cycling team training event<sup>5</sup>. Unlike the previous decentralized road cycling use case, cyclo-cross can leverage from a minimal ad-hoc infrastructure to provide better coverage. For this use case, we set up a static mesh network which covers the cyclo-cross circuit, formed with four static CONAMO devices (sink node R and relays 1, 2 and 3) as shown in Figure 11.

<sup>5</sup>Located at ([https://en.wikipedia.org/wiki/Cyclocross\\_Leuven](https://en.wikipedia.org/wiki/Cyclocross_Leuven))

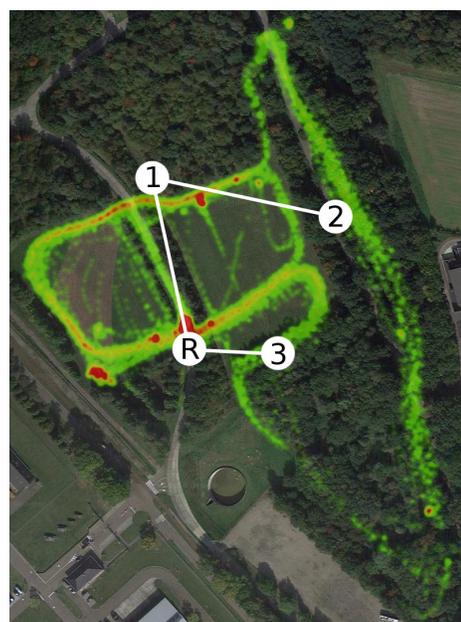


Fig. 11: Heatmap of all location observations measured during the cyclo-cross training. It shows the data measured during the active periods of all riders and the location of the three static CONAMO devices (1, 2 and 3) and the sink node (R).



Fig. 12: CONAMO devices attached to bikes.

These three static devices acting as relays are the same devices attached to the bikes but placed strategically to extend the coverage of the wireless mesh network. The rest of the CONAMO devices were attached to the bicycles as shown in Figure 12b. A total amount of 17 CONAMO devices were attached to bikes, each of them monitoring the heart rate, speed and location of the young riders during their training.

Figure 13 shows the system architecture of the evaluation set-up on the cyclo-cross event. All data collected by the sink node was duplicated on the UART port. One stream was directly sent through BLE to the CONAMO tablet application running on a tablet using the embedded nRF52 chip in the CONAMO device. The second stream was received on a laptop and relayed through Wi-Fi to be consumed by the real-time processing engine on an external Raspberry Pi.

The data processing system includes the stream reasoning engine as well as the additional data analytics components used for the different visualizations. For the evaluation, the C-SPARQL stream reasoning engine is used and it runs the three real-time processing queries discussed in Section III-D every second on a window of 5 seconds.

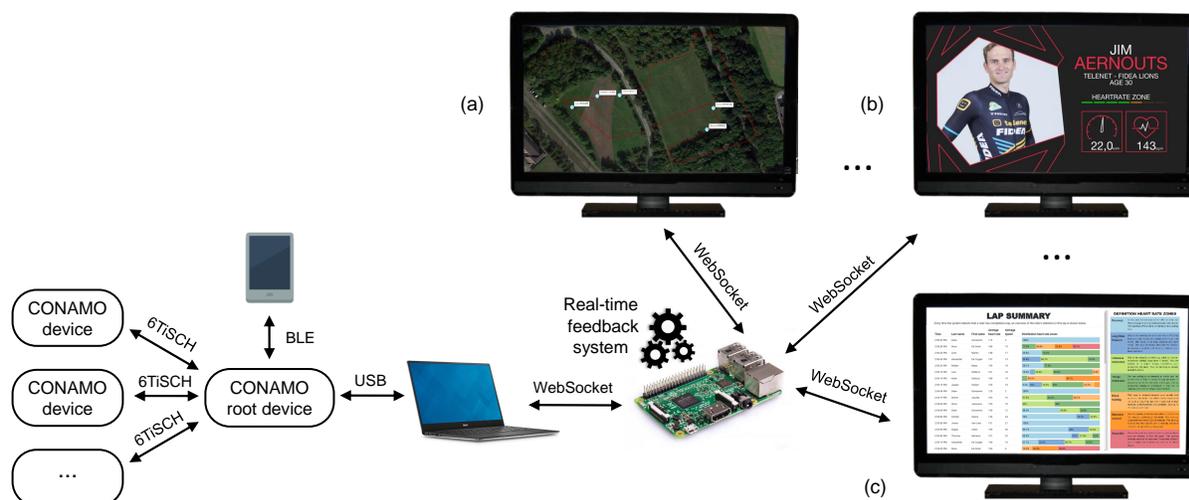


Fig. 13: Architecture of the evaluation set-up of the system on the cyclo-cross training event. The sink (i.e. root device) is connected to a tablet through BLE and attached via a laptop to a Raspberry Pi acting as the gateway. The Raspberry Pi is hosting the real-time feedback system, including the stream reasoning engine and additional data analytics components used for the additional visualizations described in Section III-D. Some examples of designed visualizations are shown in the figure: (a) location from `getLocationObservationValue` query; (b) real-time data from `getQuantityObservationValue` query (speed & heart rate) and `getTrainingZone` query (training zone); (c) aggregated feedback (lap summaries).

1) **Network Performance Results and Discussion:** In order to characterize the network performance we first show how the bikes are connected to the network. The coverage of the sink node and the three static CONAMO devices is illustrated in Figure 14, where each of the samples represents the RSSI value of the packet sent from the rider to the corresponding next hop node. As can be observed in these figures, nodes R, 1 and 2 have a more omni-directional behavior than node 3. This is because while they were mounted on standing poles, node 3 was attached to a tree, which steered its directivity mainly to the south-east (i.e. lower right in Figure 14c).

Additionally, Figure 15 shows the distribution of RSSI values compared to the distance for all samples in the dataset, considering the distance between the bike and its next hop. There are no samples after 272 m since, nodes tend to switch to a better parent rather than keeping a lower quality link with its old parent. Notice that although this scenario is located in an enclosed circuit smaller than 500 m, the link degradation is higher than the one perceived in Section IV-A since the area is covered by trees, many links occur over non-LOS.

The RSSI of each transmission also influences the packet latency per hop. As shown in Figure 16a, the lower the RSSI values are, the higher the latency is. This is mainly because links with lower RSSI require more retransmissions on average, which delay the packet delivery. For example, while for an RSSI of -37 dBm the average 1-hop latency is 0.51 s, for an RSSI of -96 dBm the latency is 1.64 s on average. However, the total latency in the network is mainly affected by the number of hops existing between the bike and the sink. Figure 16b shows the distribution of latencies for each of the three possible hops. As expected, the average and median latency values increase with each hop, and the most representative part of the samples (i.e. inter quartile range) is lower than 2 s.

Because the slotframe duration is  $101 \text{ slots per frame} \cdot 15 \text{ ms slot duration} = 1.515 \text{ s}$ , only 5.6% of the packets have higher delays than  $1.5 \cdot 3 \text{ hops} = 4.545 \text{ s}$ .

On the other hand, the parent changes also have a share in the latency calculation. Although limited to sporadic moments, parent changes can provoke latency increases in some of the packets and even drops. Figure 16c shows the CDF of the introduced delay in the last packet before the switch occurs. The mean parent switch time is 20.30 s and the median is 16.3 s. This time includes the queuing delay old packets suffer after a parent switch when they are enqueued again after failure (if possible) and have to wait until all newly queued packets are sent first to the new parent.

Finally we show in Figure 17 some examples of how packets are delivered through multiple parents during the event. It shows for three different riders how the sensor data, in this case heart rate, arrives to the sink node via different parents. Although some data gaps appear between the parent switches, the disruption is always low enough to feed the input of the real-time processing engine with a quasi-continuous stream.

2) **Data Processing Results and Discussion:** This part of the evaluation focuses on the performance of the real-time processing, done by the stream reasoning engine during the first 17 minutes of the training, where all 17 riders were actively cycling. First, throughout this period, the distribution of the execution times<sup>6</sup> of the three real-time processing queries has been studied. Second, for each observation coming in over the WebSocket connection, the total processing time has been calculated with respect to the processing of this observation by each of the three real-time processing queries. For each query, this total processing time is defined as the

<sup>6</sup>On a Raspberry Pi 3 Model B, (QuadCore 1.2GHz Broadcom BCM2837)

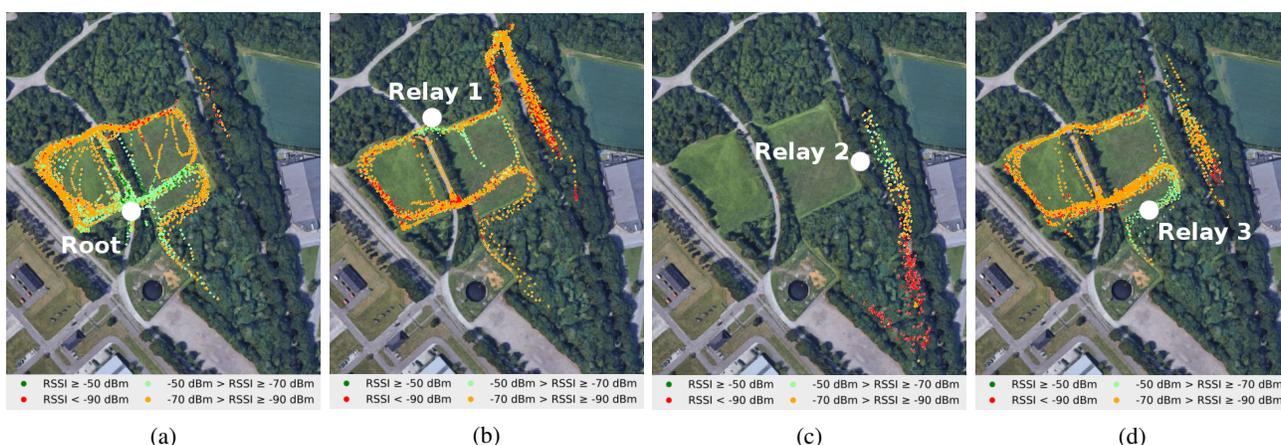


Fig. 14: RSSI samples representing coverage areas of each CONAMO device acting as a relay (1, 2 and 3) and the sink (Root).

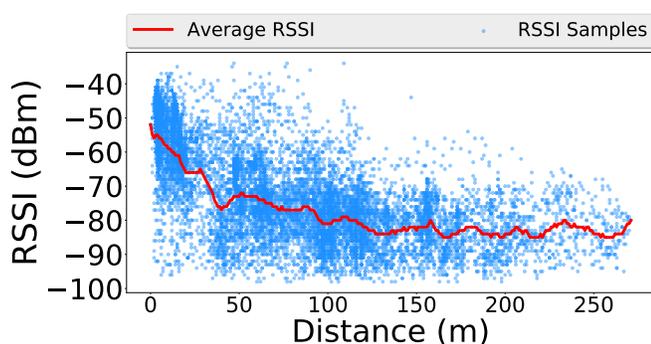


Fig. 15: RSSI vs distance from all obtained data points.

sum of: the preprocessing time, including the conversion of the observation to JSON-LD, the used data format; the waiting time, i.e. the time before the query is evaluated on a window that includes the observation *for the first time*; the query execution time; and the post-processing time, including sending the query results to the different clients. For these evaluations, each query has been executed 1021 times (1 query/s, during  $(17 \times 60 + 1)$  s).

Figure 18a shows a boxplot with the distribution of the execution times of the three real-time processing queries. For all queries, the execution time has a positive skew, i.e. its right tail (upper tail in the figure) is longer and has a significant amount of outliers.

It is clear that the `getTrainingZone` query is more computationally complex compared to the other two. This is caused by the additional processing required to find the training zone corresponding to the measured heart rate. This causes a higher variability in execution times on the low-end device. For all three queries, the execution times are significantly lower than 1 s, which is the period between two query executions.

Figure 18b shows the distribution of the total processing times of observations coming in at the real-time feedback system. These distributions also have a positive skew, but the skewness is much smaller compared to the query execution times. There are again some outliers, but also fewer.

Inspecting these results, a first conclusion is that there is no

significant difference in the distribution variance or skewness for the three queries. Considering the fact that the execution times are part of the total processing times, it is clear that the higher variance of the `getTrainingZone` query execution is partly evened out. Interestingly, the median is higher for the `getLocationObservationValue` query, even though this query has the smallest execution times. This difference is caused by the waiting time. Besides the query execution time, this is the biggest other part of the total processing time. It is inherently dependent on the defined time between two query executions. For this evaluation, the waiting time is always between 0 ms and 1000 ms, as each query executes at a rate of 1 time/s. However, the average waiting time is not necessarily 500 ms, in case the observations are coming in on a pattern with the same period as the query executions, i.e. every second. In this evaluation, this is more or less the case. Hence, for each query, the exact average is different from 500 ms; it is dependent on the difference between the average incoming observation time and the beginning of the query execution, which is defined by the first query execution.

Finally, in general terms, the total processing time is the extra delay that the real-time processing introduces for an observation, before it can be displayed. Given the requirement of real-time monitoring, it is important that this additional delay is not too large. Given these results, it is on average below 1 s, and except for a few outliers, it is always below 2 s. As explained, the exact values can be slightly higher on average in other execution cases. However, it is clear that this stays within the bounds of what is acceptably real-time.

3) **Data Insights:** To generate various data insights, several visualizations were implemented for the cyclo-cross training. First, two visualizations were developed to represent the real-time data results of the continuous queries: one for the location of each rider (Figure 13 a), and one for the real-time quantity and training zone data (Figure 13 b). Second, two examples of aggregated feedback components were implemented for the cyclo-cross use case: a real-time ranking of riders on a route segment with obstacles, based on the training zone distribution on this segment (Figure 19), and a list of statistical summaries for each completed lap of the cyclo-cross track (Figure 13 c).

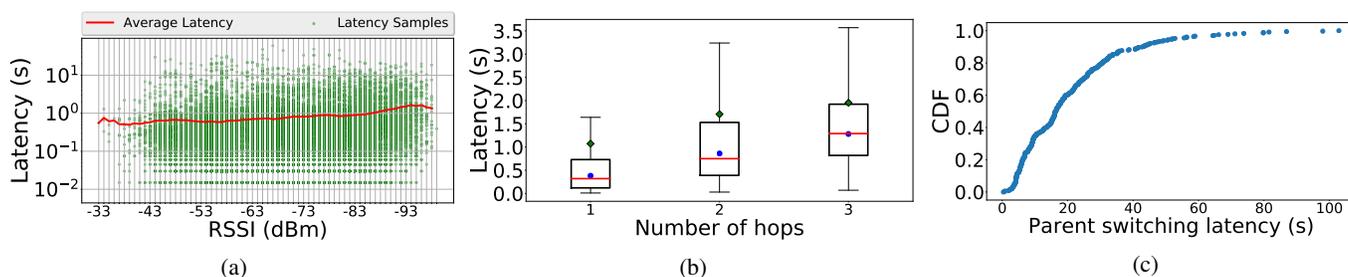


Fig. 16: Latency per hop values on log scale regarding the RSSI (a), End-to-end latency regarding the number of hops (b), and CDF of the latency introduced in the last packet before the parent switch (c).

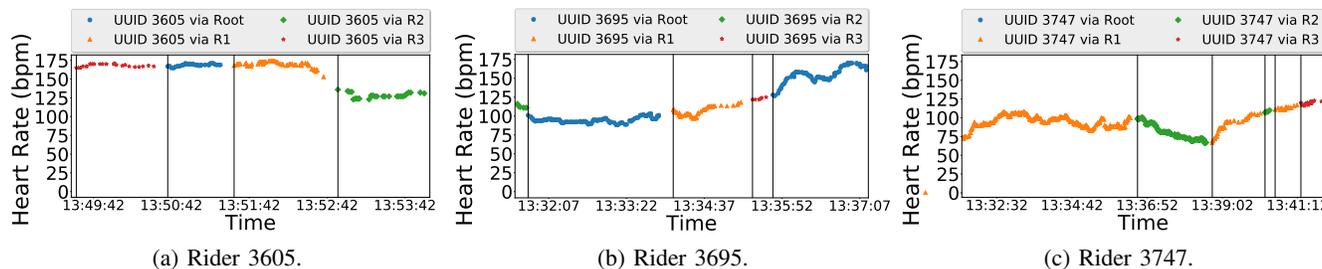


Fig. 17: Heart rate data from different riders arriving at the sink via different parents. Vertical lines represent parent changes.

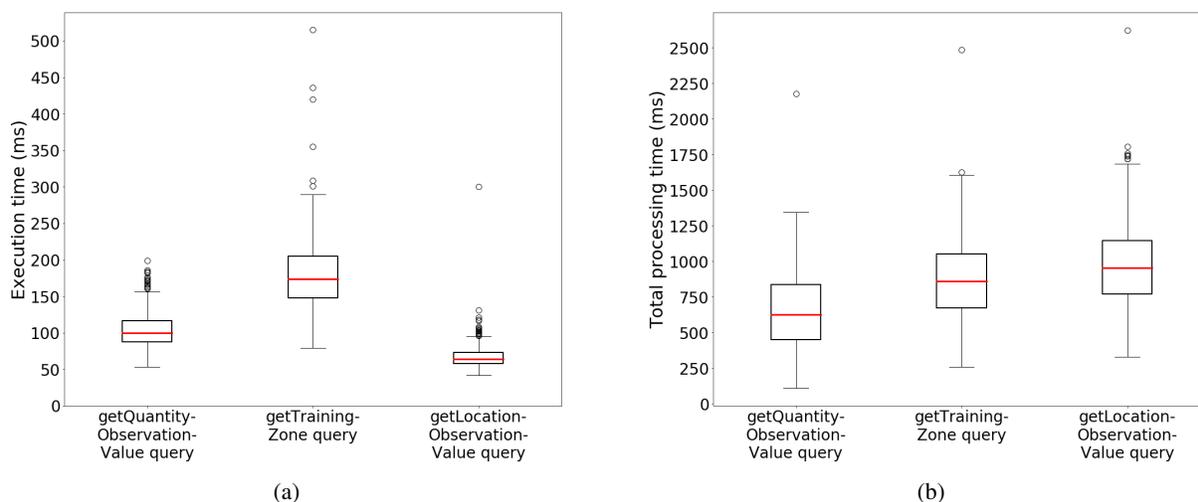


Fig. 18: Data processing results: a) shows the distribution of the execution times of the three real-time processing queries; b) shows the distribution of the total processing times of observations coming in at the real-time feedback system, with respect to the processing of this observation by each of the three real-time processing queries, during the live cyclo-cross use case.

The feedback visualizations allow riders and coaches to gain insights into the performance of the riders. Depending on various context parameters, a different training zone may be preferred for a rider. These parameters can be the current phase of the cycling season (build-up, pre-match training, etc.), the current weather, the current part of the track, or others. Real-time information during a training can then be vital to react to the rider's training zone in a timely manner. For example, the training intensity can be increased or decreased. This suggestion can be provided by the coach, or can be deduced by the rider himself. Using information in the lap summaries, the overall performance can be assessed and compared over the different laps. If a certain training zone is targeted for a

segment of the track, a real-time ranking on this segment can give insight into which riders ride in this zone the most.

In addition to the (aggregated) real-time feedback visualizations, the monitored real-time data, i.e. the outputs of the real-time processing queries, can also be used to generate other insights. Figure 20 shows the evolution over time of speed and heart rate for two different riders. From this figure, it is clear how heart rate and speed are correlated over time, for these riders. Such analyses allow riders or coaches to inspect the training intensity over time. This may for example gain insight into which training pattern is optimal for a rider. Additional sensors such as power and cadence meters can substantially improve these insights. Note that this visualization could also

## REAL-TIME RANKING

Below you can find a real-time ranking of how well the riders perform up to now in the zone around the finish equipped with obstacles, ditches and a running zone. The position of the riders is calculated based on the heart rate zone of each rider.

Position	Last name	First name	Average heart rate	Average speed	Distribution heart rate zones
1	Wout	Janssens	158	16	10.5% 69.2% 19.5%
2	Alexander	De Cuyper	157	17	8.8% 20.9% 39.6% 30.8%
3	Kobe	Dubous	172	18	38.2% 30.9% 25.5%
4	Lars	Willems	148	19	40% 60%
5	Robbe	Maes	152	17	38.6% 47.4% 14%
6	Arne	Martin	158	15	33.7% 9.2% 27.6% 29.6%
7	Milan	Peeters	151	15	22.5% 30.6% 25.2% 19.8%
8	Senne	Jacobs	142	13	43.6% 27.3% 16.4%
9	Seppe	Claes	141	17	43.8% 20% 24.6% 11.5%
10	Jasper	Heylen	155	14	37.5% 14.6% 11.8% 15.3% 13.2%
11	Arthur	Bertels	130	16	66.1% 30.5%
12	Xander	Geens	145	21	45.5% 38.6% 15.9%
13	Noah	Wouters	143	15	60% 32.5%
14	Rune	De Smet	197	15	9.7% 16.1% 66.1%
15	Daan	Goossens	121	15	86%
16	Jonas	Van Laer	134	15	82.6% 12.2%
17	Thomas	Mertens	109	16	86.6%

Fig. 19: Real-time ranking visualization, based on each rider’s training zone distribution in a chosen zone of the cyclo-cross track. The ranks of the riders are based on a ranking score calculated on each rider’s training zone distribution. In this example, the score was higher for riders that were more in the extensive endurance and tempo endurance training zones.

be plotted in real-time, again allowing a coach to react to it during the training. Finally, in Figure 21, the training zone distribution of two other riders is compared between the periods when cycling inside and outside the woods. On the cyclo-cross track, there were some parts which were on an open meadow, while other parts were in the woods. From this comparison, it is clear that the riders were clearly cycling in higher training zones within these woods. This may learn that these parts of the track were more challenging and better suited for a high intensity training. Such real-time location-based analysis gives a coach more insight into the rider’s performance on different parts of the track, in addition to the information provided by the real-time ranking.

## V. CONCLUSIONS

This paper discusses our experience in developing an innovative infrastructure-less IoT-based platform for cycling sports. Our CONAMO platform proposes to use the actual bicycles themselves as nodes in a mobile mesh network in order to gather sensor data from each rider in an ad-hoc manner. This data is processed and analyzed in a sink node using real-time processing techniques on low-end devices, in order to obtain relevant instantaneous personalized feedback, which can for example be used for coaching purposes. CONAMO combines ANT+ technology, which is used to communicate with the different bicycle sensors, and 6TiSCH, which builds the long-range 6LoWPAN network using the TSCH mode of IEEE 802.15.4e in the subGHz band. The combination of the two technologies provides a dynamic, real-time and reliable sensor communication channel that can be used for covering low-budget cycling events such as amateur cycling or professional training without the need of 4G coverage or other infrastructure-based technology. To the best of our knowledge this work is the first one in proposing and deploying such a heterogeneous architecture for cycling monitoring. We report

evaluations in real dynamic scenarios, together with an large open-sourced data set, that show the relevance and promising possibilities of the proposed approach.

## ACKNOWLEDGMENT

Part of this research was funded by the Flemish FWO SBO S004017N IDEAL-IoT (Intelligent DENSE and Long range IoT networks) project and by the Interdisciplinair Coöperatief Onderzoek (ICON) project Continuous Athlete Monitoring (CONAMO). CONAMO is a project realized in collaboration with IMEC, with project support from VLAIO. Project partners are imec, Rombit, Energy Lab and Vlaamse Radio- en Televisieomroeporganisatie (VRT). We thank all the partners. Femke Ongenae is funded by a UGent BOF postdoc grant. We would like to thank the Royal Belgian Cycling League (KBWB) for their help and collaboration. We also thank BOOST for their support in the visualization work.

## REFERENCES

- [1] Garmin, “This is ANT: the Wireless Network Solution.” [Online]. Available: <https://www.thisisant.com/>
- [2] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, “6TiSCH: deterministic IP-enabled industrial internet (of things),” *IEEE Communications Magazine*, vol. 52, no. 12, pp. 36–41, 2014.
- [3] T. Winter, P. Thubert, and A. Brandt, “RPL: IPv6 Routing Protocol for Low Power and Lossy Networks,” Internet RFC, March, Tech. Rep., 2011.
- [4] G. Daneels, E. Municio, K. Spaey, G. Vandewiele, A. Dejonghe, F. Ongenae, S. Latré, and J. Famaey, “Real-Time data dissemination and analytics platform for challenging IoT environments,” in *Global Information Infrastructure and Networking Symposium (GIIS), 2017*. IEEE, 2017, pp. 23–30.
- [5] M. De Brouwer, F. Ongenae, G. Daneels, E. Municio, J. Famaey, S. Latré, and F. De Turck, “Personalized Real-Time Monitoring of Amateur Cyclists on Low-End Devices,” in *WWW2018, the International World Wide Web Conference*. ACM Press, 2018, pp. 1833–1840.
- [6] P. M. Santos, J. G. Rodrigues, S. B. Cruz, T. Lourenço, P. M. d’Orey, Y. Luis, C. Rocha, S. Sousa, S. Crisóstomo, C. Queirós *et al.*, “PortoLivingLab: An IoT-Based Sensing Platform for Smart Cities,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 523–532, 2018.
- [7] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, “Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios,” *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, 2016.
- [8] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay *et al.*, “A macroscope in the redwoods,” in *Proceedings of the 3rd international conference on Embedded networked sensor systems*. ACM, 2005, pp. 51–63.
- [9] Z. Zhang, S. D. Glaser, T. Watteyne, and S. Malek, “Long-term Monitoring of the Sierra Nevada Snowpack Using Wireless Sensor Networks,” *IEEE Internet of Things Journal*, 2016.
- [10] T. Watteyne, L. Doherty, J. Simon, and K. Pister, “Technical overview of SmartMesh IP,” in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*. IEEE, 2013, pp. 547–551.
- [11] I. Foche-Pérez, J. Simó-Reigadas, I. Prieto-Egido, E. Morgado, and A. Martínez-Fernández, “A dual IEEE 802.11 and IEEE 802.15-4 network architecture for energy-efficient communications with low-demanding applications,” *Ad Hoc Networks*, vol. 37, pp. 337–353, 2016.
- [12] J. Gozávez, M. Sepulcre, and R. Bauza, “IEEE 802.11p vehicle to infrastructure communications in urban environments,” *IEEE Communications Magazine*, vol. 50, no. 5, 2012.
- [13] D. Data, “Dimension Data: Hybrid IT Systems Integrator and Managed Services.” [Online]. Available: <https://www.dimensiondata.com/>
- [14] Quark, “Quark – Know Your Powers.” [Online]. Available: <https://www.quark.com/>
- [15] “Garmin Connect.” [Online]. Available: <https://connect.garmin.com>
- [16] “Strava.” [Online]. Available: <https://www.strava.com>
- [17] “Runkeeper.” [Online]. Available: <https://runkeeper.com>

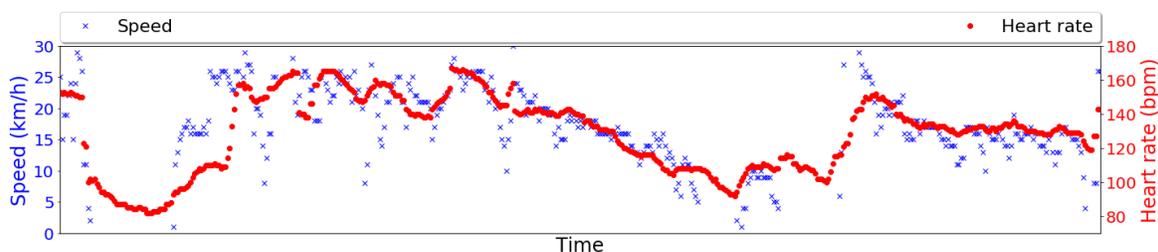


Fig. 20: Evolution over time of speed and heart rate measurements for one rider, shown for his active period during the cyclo-cross training. Two scales are used, in order to overlay the curves of both measurements. This demonstrates the correlation between heart rate and speed over time.

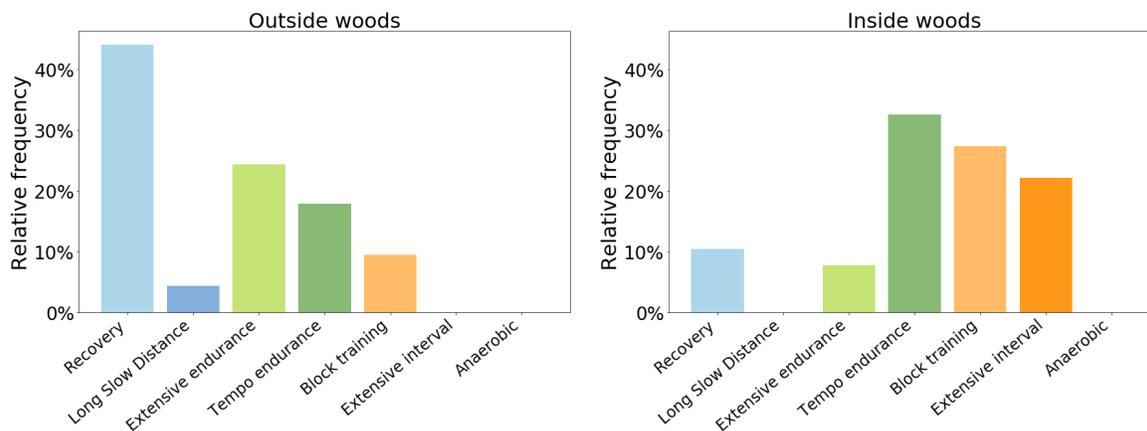


Fig. 21: Example of the heart rate training zone distribution of one rider, measured for his active period during the cyclo-cross training. The distribution is compared between the periods that he was cycling inside and outside the woods of the track.

[18] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 414–454, 2014.

[19] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the Internet of Things: early progress and back to the future," *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 8, no. 1, pp. 1–21, 2012.

[20] D. Dell’Agllo, E. Della Valle, F. van Harmelen, and A. Bernstein, "Stream reasoning: A survey and outlook," *Data Science*, no. Preprint, pp. 1–25, 2017.

[21] D. F. Barbieri, D. Braga, S. Ceri, E. Della Valle, and M. Grossniklaus, "C-SPARQL: a continuous query language for RDF data streams," *International Journal of Semantic Computing*, vol. 4, no. 1, pp. 3–25, 2010.

[22] X. Su, E. Gilman, P. Wetz, J. Riekk, Y. Zuo, and T. Leppänen, "Stream reasoning for the Internet of Things: Challenges and gap analysis," in *WIMS 2016*. ACM, 2016.

[23] V. Charpenay, S. Käbisch, and H. Kosch, "Towards a Binary Object Notation for RDF," in *ESWC 2018*. Springer, 2018, pp. 97–111.

[24] W. Van Woensel and S. S. R. Abidi, "Optimizing Semantic Reasoning on Memory-Constrained Platforms Using the RETE Algorithm," in *ESWC 2018*. Springer, 2018, pp. 682–696.

[25] A. I. Maarala, X. Su, and J. Riekk, "Semantic reasoning for context-aware Internet of Things applications," *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 461–473, 2017.

[26] A. Gyraud, S. K. Datta, C. Bonnet, and K. Boudaoud, "A semantic engine for Internet of Things: cloud, mobile devices and gateways," in *IMIS 2015*. IEEE, 2015, pp. 336–341.

[27] A. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and Q. Z. Sheng, "IoT middleware: A survey on issues and enabling technologies," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 1–20, 2017.

[28] H. Weghorn, "Efforts in developing android smartphone sports and healthcare apps based on bluetooth low energy and ANT+ communication standards," in *Innovations for Community Services (IACS), 2015 15th International Conference on*. IEEE, 2015, pp. 1–7.

[29] Q. Wang, X. Vilajosana, and T. Watteyne, "6TiSCH Operation Sublayer (6top) Protocol (6P)," RFC 8480, Nov. 2018. [Online]. Available: <https://rfc-editor.org/rfc/rfc8480.txt>

[30] T. Chang, M. Vučinić, X. Vilajosana, S. Duquennoy, and D. Dujovne, "6TiSCH Minimal Scheduling Function (MSF)," Internet Engineering Task Force, Internet-Draft draft-ietf-6tisch-msf-02, Mar. 2019, work in Progress. [Online]. Available: <https://tools.ietf.org/html/draft-ietf-6tisch-msf-02>

[31] R. Hermeto, A. Gallais, and F. Theoleyre, "On the (over)-reactions and the stability of a 6tisch network in an indoor environment," in *21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM’18)*, 2018.

[32] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser, and K. Pister, "OpenWSN: a standards-based low-power wireless development environment," *Transactions on Emerging Telecommunications Technologies*, vol. 23, no. 5, pp. 480–493, 2012.

[33] G. Daneels, E. Municio, B. Van de Velde, G. Ergeerts, M. Weyn, S. Latré, and J. Famaey, "Accurate Energy Consumption Modeling of IEEE 802.15. 4e TSCH Using Dual-Band OpenMote Hardware," *Sensors*, vol. 18, no. 2, p. 437, 2018.

[34] X. Vilajosana, K. Pister, and T. Watteyne, "Minimal IPv6 over the TSCH Mode of IEEE 802.15.4e (6TiSCH) Configuration," RFC 8180, May 2017. [Online]. Available: <https://rfc-editor.org/rfc/rfc8180.txt>

[35] K. C. Lee, R. Sudhaakar, J. Ning, L. Dai, S. Addepalli, J. Vasseur, and M. Gerla, "A comprehensive evaluation of RPL under mobility," *International Journal of Vehicular Technology*, vol. 2012, 2012.

[36] F. Somaa, I. El Korbi, C. Adjih, and L. A. Saidane, "A modified rpl for wireless sensor networks with bayesian inference mobility prediction," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International*. IEEE, 2016, pp. 690–695.

[37] M. Kent, *Oxford dictionary of sports science and medicine*. Oxford university press, 2006, vol. 10.

[38] J. H. Yoo, J. H. Lee, and S. H. Cho, "A propagation model in 2.4 GHz ISM band using IEEE 802.15. 4 systems," in *The 17th Asia Pacific Conference on Communications*. IEEE, 2011, pp. 339–343.