Joint Technology and Route Selection in Multi-RAT Wireless Sensor Networks with RODENT

Brandon Foubert, Nathalie Mitton

To cite this version:

Brandon Foubert, Nathalie Mitton. Joint Technology and Route Selection in Multi-RAT Wireless Sensor Networks with RODENT. IWCMC 2021 - 17th International Wireless Communications & Mobile Computing Conference, Jun 2021, Harbin / Virtual, China. hal-03186044

HAL Id: hal-03186044
https://hal.inria.fr/hal-03186044

Submitted on 30 Mar 2021

Joint Technology and Route Selection in Multi-RAT Wireless Sensor Networks with RODENT

Brandon Foubert and Nathalie Mitton
Inria - Lille, France
{brandon.foubert, nathalie.mitton}@inria.fr

***Abstract*—Wireless Sensor Networks (WSN) are limited by the characteristics of the Radio Access Technologies (RAT) their are based on. We call a wireless multi-hop network composed of nodes able to use several RAT a Multiple Technologies Network (MTN). Nodes must manage the RAT and route selection, in a local and distributed way, with an suitable communication protocol stack. Nodes may share multiple common RAT with multiple neighbors. Thus the devices' heterogeneity of technologies has to be taken into account by each of the stack's layer. In this article, we introduce our custom Routing Over Different Existing Network Technologies protocol (RODENT), designed for MTN. It is capable of dynamically (re)selecting the best RAT and route based on data requirements evolving over time. RODENT is based on a multi-criteria route selection via a custom lightweight TOPSIS method from our previous work [1]. For an evaluation of performance, we implemented a functional prototype of RODENT on Pycom FiPy devices. Results show that RODENT enables multiple data requirements support and energy savings, while increasing effective coverage.**

***Index Terms*—LPWAN, WSN, MTN, RODENT, multi-RAT, heterogeneous, routing, multi-flow, Pycom FiPy**

I. INTRODUCTION

Wireless Sensor Networks (WSN) allow many use cases such as remotely monitoring various metrics [2]. Such networks usually rely on a medium distance Radio Access Technology (RAT) (*e.g.,* IEEE 802.15.4) and a multi-hop path routing. A specific subset of WSN, Low Power Wide Area Networks (LPWAN), usually rely on a long distance RAT (*e.g.,* LoRaWAN) and star topology. This simplifies the network structure and enables wider coverage. When deployed, WSN usually use a single RAT shared by all nodes. The chosen RAT's limits, in terms of coverage and throughput, *de facto* constrain deployment. As an example, the Sigfox operator-based network provide a very large range (up to km) but does not offer a worldwide coverage. Specific data requirements such as large data or delay-intolerant data are impossible to withstand for the most constrained RAT. We also have to consider that outdoor nodes have to bear weather changes such as rain, which heavily alter the reliability of wireless links.

Actual WSN have trouble supporting multiple use-cases because of a lack of flexibility. Numerous RAT with various performances and capabilities are available nowadays [3]. Overcoming the aforementioned issues is possible with the use of Multiple Technologies Networks (MTN) [4]. Nodes could switch RAT and form multi-hop networks, which would extend the range of deployment. Based on the routes' performance and cost in energy, money etc., nodes would be able to select the RAT that best matches their needs. If the selected route's reliability or availability decreases, *e.g.,* because of a change of weather or any node condition, nodes could select a new route with a different RAT. Nodes with several use-cases (*e.g.,* video and humidity monitoring) could use different routes. The resiliency of the network would greatly increase, because in case of a RAT failure, nodes can use another technology.

Nodes should manage the RAT heterogeneity autonomously. But the routing protocols currently available does not support MTN. In this article, we introduce a novel Routing Over Different Existing Network Technologies protocol (RODENT) designed for MTN. It takes every RAT of a node into consideration for the routing process. As an input, it requires a list of the links available between the node and its neighbors. Such links have several associated performances and costs, in terms of energy consumption, delay, bit-rate etc. The routing table of each node is built based on the values from the list of its available links and the values from the routes shared by its neighbors. Nodes select the best route from the set of known routes by means of a custom TOPSIS method from our previous work [1]. The best route's criteria depend on the use case and the associated data requirements to fulfill (*e.g.,* delay deadline, data size).

The performances of RODENT are assessed through implementation and experimental evaluation. To this end, we designed a MTN prototype based on Pycom FiPy devices embedding a custom MicroPython implementation of RODENT. Pycom FiPy nodes have multiple RAT available. Results show that with RODENT the network flexibility and reliability are increased. The energy consumption is lowered and data requirements are fulfilled at best, all while maintaining a good Packet Delivery Ratio (PDR). Unlike related work, RODENT provides a dynamic and flexible way to surpass the limitations of classic WSN without needing a dedicated infrastructure on the operator's side.

The rest of this paper is organized as follows: Section II presents the work related to MTN from the literature. Section III introduces the network model and assumptions we based RODENT on. Section IV exposes beforehand information about the methods used for route selection. Section V presents RODENT's inner workings. Section VI details the hardware and firmware used for our MTN prototype. Section VII presents the setup and scenario of the experiments. Section VIII details the results of the experiments. Section IX brings a conclusion to this article and lists future work.

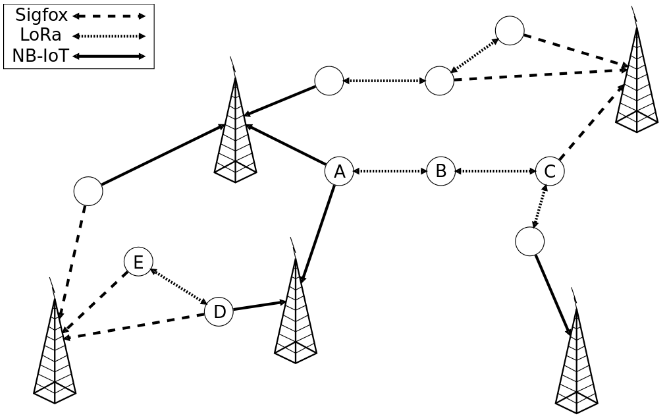

Fig. 1. MTN example.

II. RELATED WORK

To the best of our knowledge, there is few work in the literature about multi-RAT networks. In this section we present the work related to multi-technologies devices.

In [5] the authors propose a multi-RAT IoT architecture for devices. It is based on a network convergence layer managing the multi-RAT aspect from the nodes, and a heterogeneous network controller on the network operator side. An hardware platform for the nodes is also proposed, with specific polling and compression schemes based on Static Context Header Compression (SCHC). This efficiently increases the network flexibility, however it requires a specific virtual network operator. The authors of the next work [6] propose a virtual network operator based on the cloud for multi-modal LPWA networks. The virtual network operator manage the configuration and operations of the heterogeneous LPWAN hardware. As in the previous work, it requires the deployment of a specific infrastructure on the operator side.

The authors of [7] introduce a green path inter-MAC selection protocol. The protocol enables MAC layer path selection, to minimize energy consumption and radio frequency utilization. However, no information about the routing layer is given as this is out of the article's scope. In [8] the authors detail the ORCHESTRA framework. It manages real-time inter-technology handovers. The framework is based on a virtual MAC layer, coordinating every layers from each technology with a single MAC address. As for the previous work, this one also focuses on the link layer and omits the routing layer.

WSN's flexibility is increased in every one of the aforementioned works. But as they require a dedicated infrastructure, it brings in turn a new limitation. In this article, we present a routing protocol adapted to MTN, which greatly increase WSN's capabilities while requesting only multi-RAT nodes.

III. MODEL & ASSUMPTIONS

RODENT's design is based on a specific network model and several assumptions about the communication stack's lower layers. In this section we describe those model and assumptions.

TABLE I
EXAMPLE LINK MATRIX LM_D.

	Energy	*Money*	*Bit-rate*
Sigfox BS	12	102	22
NB-IoT BS	151	87	174
Node E (LoRa)	37	0	72

A. Network model

The nodes taking part in a WSN usually follow one or multiple traffic patterns [9]. Here, we assume that the nodes running RODENT communicate only in a convergecast pattern, from the nodes to the sink. As the nodes of a MTN have multiple RAT, there can be several links between a single pair of nodes. We assume that the network forms a connected graph if we consider every link independently of their RAT. Several data requirements can be fulfilled by the same node (*e.g.,* monitoring and alarm) as long as those requirements are known by every node in the MTN. An example of a MTN is shown in Figure 1. Here, node B (N_B) monitors temperature but is not in the vicinity of a Sigfox or NB-IoT base station. Since N_B has multiple RAT, it can forward its data to N_A or N_C through a LoRa link. They can then offload N_B's data to a base station using another RAT.

B. Data requirements

We want RODENT's nodes to support multiple use cases, *i.e.,* multiple purposes (*e.g.,* monitoring the weather, record videos). Needs and data requirements are different for each use case. To send a video, a RAT with a high bit-rate is essential to ensure low delay and jitter. To send an alarm, a RAT with a short delay is needed, but the bandwidth is secondary. To regularly send a small amount of numerical monitoring data, a RAT with a low power consumption is the priority. Several data requirements can be supported on a single node *e.g.,* regularly sending the amount of rainfall and an alarm in case of a flood. The aim of the route selection is to satisfy at best the nodes' data requirements.

C. Assumptions on communication stack

In this article, we focus on the network layer, specifically routing. We assume that the communication stack's lower layers run protocols suited to MTN. We assume that the physical and link layers can assess the availability and reliability of the links between a node and its neighbors (*i.e.,* nodes or base stations) for every RAT. We consider that those layers gather or estimate information about the performances and costs of each link (*i.e.,* energy consumption, bit-rate, etc.), as WSN radio link quality estimation is a well studied subject [10].

As input, RODENT takes a table or matrix of links, to which we refer to as LM_i for node i. The size of LM_i depends on the number of characteristics considered, the number of available RAT and the number of i's neighbors. As an example, in Figure 1 N_D could have a LM_D such as Table I. LM_D is comprised of every link between N_D and its neighbors as well as the links' characteristics.

TABLE II
EXAMPLE ROUTE MATRIX RM_D.

	Energy	*Money*	*Bit-rate*	*Hops*
Sigfox BS	12	102	22	1
NB-IoT BS	151	87	174	1
Node E (LoRa)	49	102	94	2

IV. BACKGROUND: SELECTION METHOD

In MTN, a single node owns several RAT. This hardens the route selection as nodes must consider many routes over many RAT. In this section, we present RODENT's route selection method based on our previous work [1].

The selection process has to take account of multiple criteria *e.g.,* the energy consumption, delay etc. of each route to meet as much as possible all data specific requirements. Several tools are available in the literature for multi-criteria decision, such as utility and cost functions, Markov chains, fuzzy logic, game theory, data mining and Dempster-Shafer theory. We found the Multiple Attribute Decision Making (MADM) methods to be the most fitting for route selection.

With MADM methods, the problem is formalized as a decision matrix. It is composed of the candidates and their attributes. A set of attributes' values is associated to each candidate, reflecting its performance. The decision matrix serves as input of an MADM method, which outputs the candidates' ranking. Several MADM methods are available in the literature, such as Simple Additive Weighting (SAW), Weighting Product (WP), Analytical Hierarchy Process (AHP) and Gray Relational Analysis (GRA). Among those methods, we find the most interesting to be Technique for Order Preference by Similarity to Ideal Solution (TOPSIS). TOPSIS allows the comparison of each candidate based not only on its closeness to the best theoretical candidate but also on its distance from the worst theoretical candidate.

Classic TOPSIS has two main issues: rank reversal and complex ranking computation algorithm. We use a lightweight TOPSIS for WSN from our previous work [1] which is designed for hardware-constrained devices rand removes rank reversals. We leverage this lightweight TOPSIS method to operate the route selection in RODENT.

For node i, its route matrix is referred to as RM_i. Considering route selection, all the routes available for node i are stored in RM_i. The attributes in RM_i are relative to the routes *e.g.,* the expected transmission count, the total energy consumption of the number of hops. As an example, in Figure 1 node D could have RM_D such as Table II. A set of weights relative to each attribute is required as input by TOPSIS. Those represent the importance of each attribute in the ranking process. We call a set of weights a Requirements Vector (RV). For use case x, RV_x is its requirements vector (*e.g.,* $RV_{monitoring}$). Considering route selection, the values of a RV are based on the data requirements nodes have to fulfill (*e.g.,* prioritize speed over energy consumption), and such that $RV\ \{e_n \in RV \mid \sum_{n=1}^{|RV|} e_n = 1\}$. An example of requirements vector is shown in Table III.

TABLE III
REQUIREMENTS VECTORS.

	Energy	Money	Bit-rate
$RV_{monitoring}$	0.6	0.3	0.1
RV_{alarm}	0.1	0.1	0.8

V. ROUTING OPERATIONS

RODENT's distinctive feature is to allow the use of multi-RAT routes. Each route has different performance and costs. In this section, we detail the routing operations performed by RODENT. We introduce the following notations that are used further in the paper. We refer to node i as N_i. Nodes in the direct vicinity of N_i are called neighbors, and we refer to the set of neighbors as $NBR(i)$ for N_i. $NBR(i)_j$ is N_j such that $N_j \in NBR(i)$. $NBR(i)_j$ has at least one link with N_i, which we refer to L^x_{ij} for RAT x. The route of N_i that goes through L^x_{ij} is referred to as R^x_{ij}.

A. Overview

As an example we can consider the operations of N_D and N_E from Figure 1. N_D boots without information about its surroundings. The link layer of N_D probes the links with every RAT and builds LM_D similar to Table I. The network layer then builds the route matrix RM_D based on LM_D. Every direct links from N_D to a base station are inserted in RM_D as single-hop routes. N_E does the same and selects the only route it has toward the Sigfox base-station. This route is advertised and received by N_D through L^{LoRa}_{ED}. N_D's third route is built based on the link's and route's values. RM_D is then similar to Table II. Then for each of its RV, N_D selects a best route, based on the routes' values, independently of the RAT. Here, if we consider Table III, for $RV_{monitoring}$ low energy consumption is favored, thus the route toward the Sigfox base station is selected. For RV_{alarm} a high bit-rate is favored, thus the route toward the NB-IoT base station is chosen. Finally, N_D starts to use and advertise its best routes.

B. Packet structure

The structure of a RODENT's packet is shown in Figure 2. A packet is composed of three parts: *(i)* the header *(ii)* the payload *(iii)* the trailer. RODENT's control data is found in the header. The $Network\ Identifier$ is a two byte value known by every node in the MTN which is used to recognize RODENT's packets. The $Source\ Identifier$ is a two byte value equal to the unique ID of the packet's source node. The $Destination\ Identifier$ is a two byte value equal to the unique ID of the packet's destination node. The $Payload\ Size$ is a one byte value equal the size in bytes of the payload. The $Requirement\ Vector\ Identifier$ is a one byte value indicating the type *i.e.,* use case of the data contained in the payload. The $Route$ is a four byte array with the best route's values *i.e.,* energy, money, bit-rate and number of hops from $N_{Source\ Identifier}$. The payload is a series of $Payload\ Size$ bytes which corresponds to the data shared by the source. Finally, the trailer is a single byte equal to the $CRC8\ Checksum$ of the header and payload parts.

C. Route construction

Here we consider the routing operations of N_i. N_i boots up and begins the construction of RM_i. RODENT uses two data sets: LM_i the link matrix of N_i and the routes shared by $NBR(i)$. The first step of N_i is to search LM_i for any link from N_i to a base station *e.g.,* a Sigfox antenna of LoRaWAN gateway. Such links are directly converted to single hop routes based on the values found in LM_i and stored in RM_i. The second step of N_i is to build the routes passing through the nodes of $NBR(i)$. Considering a route coming from $NBR(i)_j$, N_i adds the attributes' values of its link L_{ij}^x to the attributes' values of the received route. The new route R_{ij}^x is then appended to RM_i.

D. Route selection

The route selection in classic WSN is usually trivial, as the route of lowest cost or rank is chosen. In an MTN, a route is a succession of links where each link potentially use a different RAT. Various performances are offered by different RAT. As we aim to support multiple use cases with different data requirements, the route selection must take into account multiple criteria. Section IV introduces the selection method of RODENT. Considering N_i, our lightweight TOPSIS method takes as input RM_i and a requirement vector RV_x relative to use case x. A ranking of the routes comes out of the selection method, and the route on top is the one fulfilling at best the data requirements of use case x. For N_i and for a use case x we refer to the best route as BR_i^x.

E. Route propagation

The propagation of the routes is made with two mechanisms: piggybacking and control packets. To share routes without the need of dedicated transmissions we use piggybacking. In a RODENT packet from N_i and relative to use case x, the header carries the ID number of RV_x and the best route BR_i^x. Because wireless communications share a common medium, N_i overhears the packets from $NBR(i)$. This allows N_i to opportunistically update RM_i. If N_i does not overhears R_{ij}^x from $NBR(i)_j$ anymore (*e.g.,* because of N_j's failure), R_{ij}^x will timeout and will be deleted from RM_i. To keep alive the unused routes, $NBR(i)_j$ can use dedicated control packets. Control packets are regular packets with an empty payload.

VI. IMPLEMENTATION

RODENT's implementation is done on FiPy devices from Pycom [11]. Those devices' specificity is the availability of five different RAT. These nodes form an MTN and offload their data to WiFi and LoRa base stations (BS). The experiments' hardware and firmware are presented in this section.

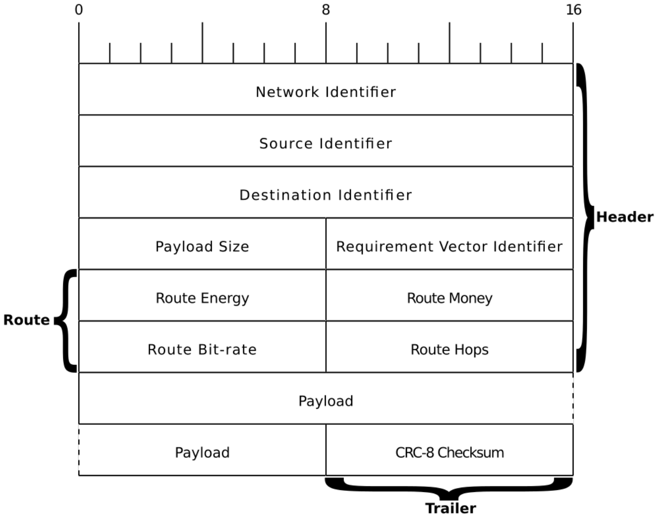

Fig. 2. RODENT packet structure.

A. Hardware

The FiPy nodes from Pycom are made from WSN hardware: CPU ESP32, several wireless RAT, few memory available and low power consumption. The different RAT embedded in a FiPy are LTE-M, NB-IoT, Sigfox, LoRa, Bluetooth Low Energy (BLE) and WiFi. Each one offers a different performance in terms of bit-rate, energy consumption, economical cost, etc., which are considered by the route selection of RODENT. Here we coupled the FiPy nodes with Pytrack sensor shields which offer a GPS, an accelerometer and a micro-USB port.

A B-L072Z-LRWAN1 board [12] is used as a LoRa BS. An Edimax EW-7811Un dongle [13] serves as a WiFi BS. Every device is connected to a Trip Lite U223-007 (7-Port USB Hub). The main computer, a Dell Latitude 5590, powers devices, collects and analyses results.

B. Firmware

Based on the FiPy's firmware (a MicroPython port), we implemented RODENT in Python. After the boot sequence, N_i computes its unique ID. The needed RAT are booted based on LM_i and the building of routes starts. The node's main loop is: *i)* select the best route for each RV_x, *ii)* add the next payload to the transmission buffer *iii)* send every payload from the buffer. The routes received from $NBR(i)$ are appended to RM_i. The payloads received from $NBR(i)$ are inserted into the transmission buffer. The keep track of operations, nodes print the sent packets' data on the serial port. The switch between RV happen upon a press of the Pytrack's button. The implemented RV are $RV_{monitoring}$ and RV_{alarm}.

The firmware of the LoRa BS is implemented in C. It listens constantly for LoRa transmissions. LoRa transmissions of RODENT's packets are unpacked and printed on the serial port. The firmware of the WiFi BS is coded in Python. WiFi transmissions of RODENT's packets are unpacked and printed on stdout.

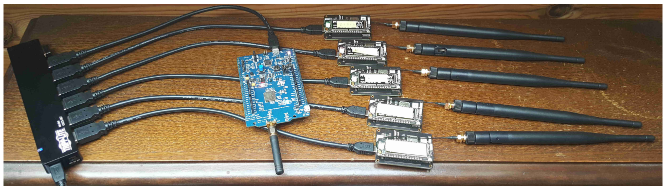

Fig. 3. Experimental setup.

VII. EXPERIMENTS

We use real hardware to run experiments to assess the performance of RODENT. The nodes are configured to unroll a specific scenario and monitor the results. The scenario and setup of the experiments are detailed in this section.

A. Setup

The devices described in Section VI are connected to the USB hub which in turn is connected to the main computer. Nodes and BS are powered at the same time and boot up immediately. Every device is sitting very close the each other as shown in Figure 3. The stdout of the WiFi BS and the serial ports of the nodes and the LoRa BS are monitored by the main computer. The results' computation is done post-experiment.

B. Scenario

The use case we simulate is the monitoring of a farm. In this scenario, we monitor a field used for cultivating crops with nodes. The simulated setup is shown in Figure 4. The useful environmental metrics are measured by five nodes. These nodes have to regularly offload numerical data while saving up power. If a metric becomes off chart (*e.g.,* temperature) and puts the crops at risk, they have to send an alarm.

Five RAT are available on FiPy nodes, but we only use WiFi, LoRa and BLE. Sigfox and LTE-M/NB-IoT aren't open technologies, thus it would considerably harden the experiments. WiFi links require more energy than LoRa and BLE links. Each node N_i shows a different situation. N_1 is the control node and can only use its WiFi link toward the WiFi BS. N_2 have to chose between its WiFi and LoRa links to the base stations. N_3 can either reach the WiFi BS or use its BLE link with N_1 to forward data at a lower energy cost. N_4 have to send monitoring data but also alarms, and can do so through its WiFi and LoRa links with the base stations. N_5 is an isolated node, too far from the WiFi BS to have a direct link. Since farms are usually wide rural environments where tall crops like corn are unfriendly to wireless wave, white zones and isolated nodes are common. But thanks to RODENT, N_5 has the ability to use LoRa to reach N_4.

Three types of experiments were ran: *(i)* RODENT is inactive and only the WiFi links are available, which are shown in blue on Figure 4. *(ii)* RODENT is active and the LoRa and BLE links are available, shown in red and green in Figure 4. *(iii)* RODENT is active, LoRa messages are sent two times, BLE messages three times, to increase the network's reliability. A video of an experiment is available online[1].

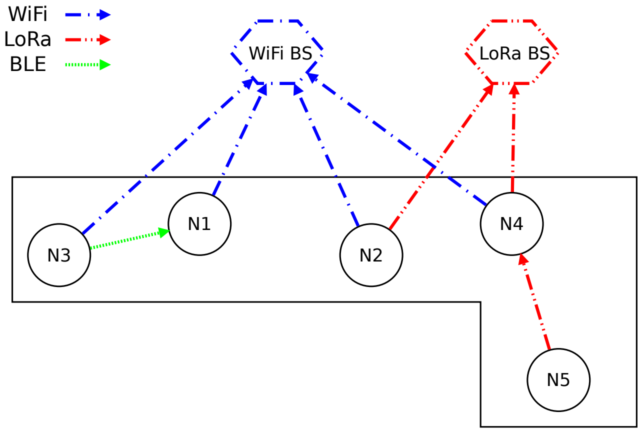

Fig. 4. Farm monitoring scenario.

VIII. RESULTS

We measure the topology and Packet Delivery Ratio (PDR). The transmission interval of the nodes is picked randomly in $[2;4]$ seconds. We consider the results of 20 experiments of a duration of 10 minutes each. A small number of experiments is sufficient because the standard deviation is low. A longer duration is not relevant as after only a few messages, the network stabilizes. Since related works' proposals differ a lot from ours, and an increase in flexibility cannot be measured, we do not directly compare their results with RODENT. In this section, we present the experiments' results we've obtained.

A. Topology

The topology of the MTN changes when RODENT is enabled. N_1 can only reach the WiFi BS, thus it keeps the same link. For monitoring, the nodes want to save up as much energy as possible. Thus, N_2 switches from the WiFi link to the LoRa link. N_3 starts to offload its data to N_1 thanks to their BLE link, and N_1 forwards the data to the WiFi BS. N_4 chooses to offload its monitoring data to the LoRa BS, but can still use its route toward the WiFi BS to send or forward alarms, which need a quick RAT even if it costs more energy. N_5 comes out from its isolation and can now forward data through N_4 using LoRa. In turn, N_4 forwards N_5's data to the LoRa BS or WiFi BS depending on the data requirements.

B. Packet Delivery Ratio

The ratio between the total packets received and the total packets sent is known as the Packet Delivery Ratio (PDR). The Figure 5 shows the PDR and its standard deviation for every node. We see that the PDR of N_1 is stable, as it keeps the same route. When RODENT is inactive, the PDR of N_5 is null because it can't offload a single data packet. With RODENT active, the PDR of N_2, N_4 and N_5 is around 80% because they use LoRa. Here nodes use LoRaRAW, without a proper MAC, which make collisions more frequent. With RODENT, the PDR of N_3 is around 60%, because it forwards its data through BLE to N_1. BLE isn't made for single message

---

[1] http://chercheurs.lille.inria.fr/bfoubert/ressources/rodent.mp4

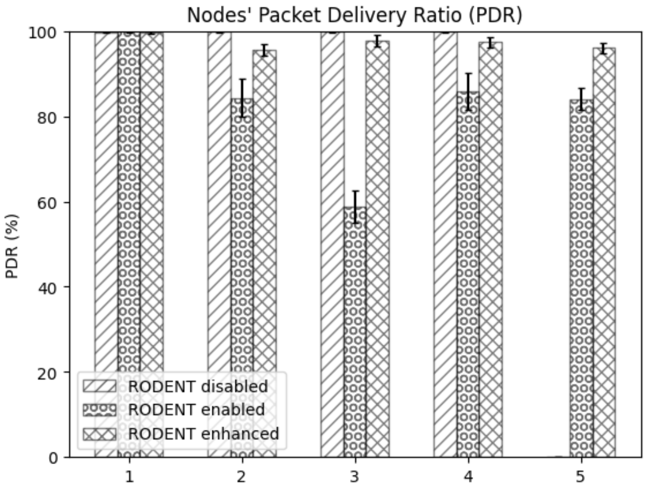

Fig. 5. Packet Delivery Ratio per node.

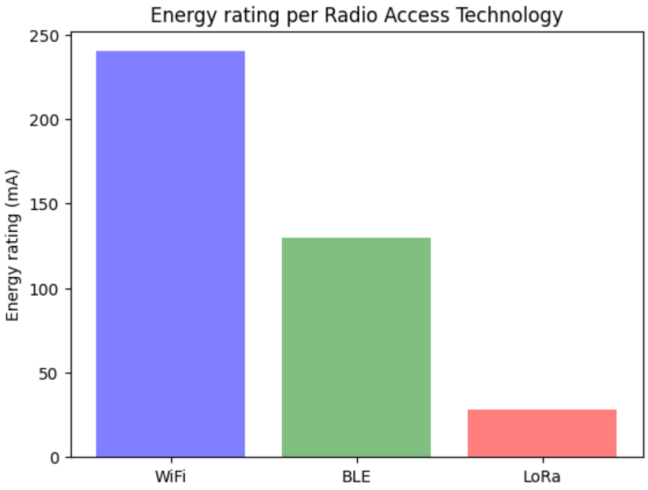

Fig. 6. Energy consumption per RAT.

transmission, thus we tweaked single BLE advertisements to achieve it, hence the packet losses. When enhancing RODENT, every node achieve a better PDR, very close to the PDR obtained with only WiFi.

C. Energy consumption

The FiPy nodes from Pycom suffer from some design problems. This makes physical measurement of energy consumption hazardous [14]. To get a general idea, we choose to stick to the energy ratings from the components datasheets [11], [15]. These are showed in Figure 6. BLE needs approximately half-less current than WiFi, and LoRa a tenth. WiFi and BLE offer the same bit-rate with the Pycom's CPU. The bit-rate of LoRa is much slower, which means longer transmission for the same amount of data. However, WiFi and BLE require a lot more traffic control than LoRa, which allows LoRa to use less energy. We can thus assume that RODENT enables significant energy savings.

IX. CONCLUSION

The coverage and other performance of RAT limit WSN deployment. Conceiving multi-RAT devices and deploying MTN can overcome these limitations. In this article, we introduce the novel Routing Over Different Existing Network Technologies protocol (RODENT). It allows multi-technologies routing in MTN. With a RODENT embedded prototype we demonstrate the utility and feasibility of MTN. From the experiments' results we see that RODENT increase the network's reliability, flexibility and energy savings while maintaining a good PDR. This work however lacks a study about the scalability.

Thus, as future work, we plan to run large experiments in real fields to test the scalability and practicality of RODENT. We want to extend RODENT to support downlink communication and precisely measure the energy consumption of nodes. This aim to further increase the nodes' flexibility and use cases (*e.g.,* firmware over the air upgrade). We also wish to design an efficient link layer protocol for precise link reliability assessment adapted to MTN.

ACKNOWLEDGMENTS

This work was partially supported by a grant from CPER DATA, Sencrop, FEDER, I-SITE and Lirima Agrinet project.

REFERENCES

[1] B. Foubert and N. Mitton, "Lightweight network interface selection for reliable communications in multi-technologies wireless sensor networks," in *DRCN*, 2021.
[2] D. Ye, D. Gong, and W. Wang, "Application of wireless sensor networks in environmental monitoring," in *PEITS*, 2009.
[3] B. Foubert and N. Mitton, "Long-range wireless radio technologies: A survey," *Future Internet*, vol. 12, no. 1, 2020.
[4] B. Foubert and N. Mitton, "Autonomous Collaborative Wireless Weather Stations: A Helping Hand for Farmers," *ERCIM News*, no. 119, pp. 37–38, 2019.
[5] J. Famaey, R. Berkvens, G. Ergeerts, E. D. Poorter, F. V. d. Abeele, T. Bolckmans, J. Hoebeke, and M. Weyn, "Flexible multimodal sub-gigahertz communication for heterogeneous internet of things applications," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 146–153, 2018.
[6] J. Hoebeke, J. Haxhibeqiri, B. Moons, M. Van Eeghem, J. Rossey, A. Karagaac, and J. Famaey, "A cloud-based virtual network operator for managing multimodal lpwa networks and devices," in *CIoT*, 2018.
[7] O. Bouchet, A. Kortebi, and M. Boucher, "Inter-mac green path selection for heterogeneous networks," in *Globecom Workshops*, 2012.
[8] T. De Schepper, P. Bosch, E. Zeljković, F. Mahfoudhi, J. Haxhibeqiri, J. Hoebeke, J. Famaey, and S. Latré, "Orchestra: Enabling inter-technology network management in heterogeneous wireless networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1733–1746, 2018.
[9] N. A. Pantazis, S. A. Nikolidakis, and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, 2013.
[10] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: A survey," *ACM Trans. Sen. Netw.*, vol. 8, no. 4, 2012.
[11] *FiPy Specsheet*, Pycom, 2018, version 1.0.
[12] *UM2115 User manual*, STMicroelectronics, 2018, rev 5.
[13] *EW-7811Un Datasheet*, Edimax, 2015.
[14] T. Andersén, "Energy-efficient adaptive sensing in low power wide area networks," Master's thesis, NTNU, 2018.
[15] *ESP32 Series Datasheet*, Espressif Systems, 2020, version 3.3.