Pontifícia Universidade Católica
do Rio de Janeiro

Marcelo Paulon Jucá Vasconcelos

Opportunistic routing towards mobile sink nodes in Bluetooth Mesh networks

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática.

Advisor: Prof. Markus Endler

Rio de Janeiro
March 2021

Pontifícia Universidade Católica
do Rio de Janeiro

Marcelo Paulon Jucá Vasconcelos

Opportunistic routing towards mobile sink nodes in Bluetooth Mesh networks

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática. Approved by the Examination Committee.

Prof. Markus Endler
Advisor
Department of Informatics – PUC-Rio

Dr. Bruno José Olivieri de Souza
Tribunal Superior do Trabalho – TST

Prof.ª Noemi de La Rocque Rodriguez
Pontifícia Universidade Católica do Rio de Janeiro – PUC-Rio

Rio de Janeiro, March 30th, 2021

Marcelo Paulon Jucá Vasconcelos

Graduated in computer science by the PUC-Rio University.

Bibliographic data

Paulon Jucá Vasconcelos, Marcelo

Opportunistic routing towards mobile sink nodes in Bluetooth Mesh networks / Marcelo Paulon Jucá Vasconcelos; advisor: Markus Endler. – Rio de janeiro: PUC-Rio , Departamento de Informatics, 2021.

v., 63 f: il. color. ; 30 cm

Dissertation (master's) - Pontifícia Universidade Católica do Rio de Janeiro, Department of Informatics.

Includes bibliography

1. Informatics – Dissertation. 2. Redes de Sensores Sem Fio. 3. Coleta distribuída de dados. 4. Redes mesh. 5. Bluetooth Mesh. 6. Reconfiguração Mesh. I. Endler, Markus. II. Pontifícia Universidade Católica do Rio de Janeiro. Department of Informatics. III. Title.

CDD: 004

Acknowledgments

I would like to express my deepest gratitude to my mother Florita Paulon, my father Newton Jucá, and my sister Júlia Paulon, for their unconditional support in all aspects of life.

I extend my gratitude to my beloved girlfriend, Catarina Vianna, for supporting me in both my academic and professional endeavors.

Very special thanks to my advisor, Prof. Markus Endler, for his continuous support since the beginning of my academic journey when I was pursuing my Bachelor's degree, always encouraging me to continue and to overcome roadblocks that come with experimental research.

My sincere thanks also go to Dr. Bruno Olivieri, for his guidance and support in this research as well as in other related projects at the LAC/PUC-Rio laboratory.

I extend my gratitude to my friend and mentor Lucas Bastos Jr., for all his support and invaluable advice since the beginning of my professional career.

Thank you to all my friends for continuously supporting me along the way, as well as to my colleagues at LAC/PUC-Rio and Zwift Inc.

I am very grateful to Prof. Noemi Rodriguez, member of my thesis defense, for reviewing this work and providing important comments and advice.

Finally, I gratefully acknowledge the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), the U.S. Air Force Office of Scientific Research (AFOSR) and PUC–Rio for the financial support.

This study was financed in part by the Coordenação de Aperfeiçoaento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

This study was financed in part by AFOSR grant FA9550-20-1-0285.

Abstract

Paulon Jucá Vasconcelos, Marcelo; Endler, Markus (Advisor). **Opportunistic routing towards mobile sink nodes in Bluetooth Mesh networks**. Rio de Janeiro, 2021. 63p. Master thesis – Department of Informatics, Pontifícia Universidade Católica do Rio de Janeiro.

This work evaluates sporadic data collection on a Bluetooth Mesh network, using the OMNET++ INET simulator. The data collector is a roaming sink node, which could be a smartphone or other portable device, carried by a pedestrian, a biker, an animal, or a drone. The sink node could connect to a mesh network in hard-to-reach areas that do not have internet access and collect sensor data. After implementing Bluetooth Mesh relay extensions, Low Power, and Friend features in OMNET++, we were able to propose and evaluate algorithms for mobility-aware, adaptive, routing of sensor data towards the sink node. One variation of a proposed routing algorithm achieved a 173.54% increase in unique data delivered to the sink node compared to Bluetooth Mesh's default routing algorithm. In that case, there was only a 4.63% increase in energy consumption for the same scenario. Also, the delivery rate increased by 111.82%.

Keywords

Wireless Sensor Networks; Distributed data collection; Mesh networks; Bluetooth Mesh; Mesh reconfiguration;

Resumo

Paulon Jucá Vasconcelos, Marcelo; Endler, Markus. **Roteamento oportunístico em direção a nós sink móveis em redes Bluetooth Mesh**. Rio de Janeiro, 2021. 63p. Dissertação de Mestrado – Departamento de Informatics, Pontifícia Universidade Católica do Rio de Janeiro.

Este trabalho avalia a coleta esporádica de dados em uma rede sem fio Bluetooth Mesh, usando o simulador OMNET++ INET. O coletor de dados é um nó sink em movimento, que poderia ser um smartphone ou outro dispositivo portátil, carregado por um pedestre, ciclista, animal, ou um drone. O nó sink poderia se conectar a uma rede mesh em áreas de difícil acesso onde não há acesso a internet, e coletar dados de sensores. Após implementar extensões ao Bluetooth Mesh, funcionalidades de nós Low Power e Friends no OMNET++, conseguimos propor e avaliar algoritmos para roteamento adaptativo, e com foco em mobilidade, de dados de sensores em direção ao nó sink. Uma variação de um dos algoritmos de roteamento propostos alcançou um aumento de 173,54% na quantidade de dados únicos entregues ao nó sink em comparação ao algoritmo de roteamento padrão do Bluetooth Mesh. Neste caso, houve um aumento de apenas 4,63% no consumo de energia para o mesmo cenário. Além disso, a taxa de entrega aumentou em 111.82%.

Palavras-chave

Redes de Sensores Sem Fio; Coleta distribuída de dados; Redes mesh; Bluetooth Mesh; Reconfiguração Mesh;

Table of contents

List of Abreviations

BLE – Bluetooth Low Energy
BTMesh – Bluetooth Mesh
BTM-R – Bluetooth Mesh Relay
IoT – Internet of Things
IoMT – Internet of Mobile Things
WMN – Wireless Mesh Network
WMNs – Wireless Mesh Networks
WSN – Wireless Sensor Network
WSNs – Wireless Sensor Networks

1
Introduction

Routing in Wireless Sensor Networks (WSNs) to optimize the collection of sensor data has been widely explored over the past years, as reported by a 2011 survey by Di Francesco, Das, and Anastasi [1] and by Djedouboum et al. (2018) [2]. WSNs with access to the Internet are a particular category of Internet of Things (IoT), where IoT devices with sensors may form WSNs to exchange sensor data and actuation commands with any remote machine.

When designing software for connected wireless devices, connectivity intermittence may be considered because smart IoT devices may have to be used in places with limited or variable wireless radio signal or unstable internet connectivity. This connectivity problem is further complicated if mobility is an intrinsic feature of the system and application.

As its primary use case, this work considers the task of monitoring trees in hillside vegetation or an urban forest. More technically, this work considers a scenario where several nodes equipped with sensors are spread in one area of difficult access, where each node monitors the health of a tree with sensors attached to its trunk and foliage, as well as some environmental variables such as temperature and humidity in the proximity of the tree. We further consider that sensor data accumulated at each mesh node can be retrieved by a mobile sink node (i.e., Mobile-Hub) when the sink gets sufficiently close to the mesh node during its continuous movement within the monitored arboretum mesh (i.e., MAM) region.

This could happen as a kind of Participatory Sensing, where pedestrians, bikers, or even animals carry a small portable IoT device or a smartphone that behaves as a Mobile-Hub [3] [4]. Or alternatively, the Mobile-Hubs could be mounted on quad-copters that overfly an urban forest or hillside region and collect the sensor data from certain "visited" WSN nodes.

In any case, the goal is that the Mobile-Hub(s) should be able to collect as much sensor data from the whole network while on the move. But, if the Mobile-Hubs are not supposed to visit every mesh node this requires an agile routing of the sensor data in the WSN towards the direction of the place where the Mobile-Hub is currently "having a rendezvous" with a mesh node.

As can be seen, this use case faces not only the challenges of intermittent connectivity (since the Mobile-Hub is only sporadically connected to the mesh network along its trajectory) but also of the energy constraints of the mesh network, since the nodes monitoring the trees have to be always ready to route their collected sensor data into some direction and, at the same time, minimize their radio activity so as to save as much (battery) energy as possible.

Energy management is a well-known and important topic in WSN implementation since it is a crucial element that will determine the operational lifespan of the entire network, and has been thoroughly explored in the literature covering mobile sink routing for WSNs [5].

On the other hand, routing in WSN and mesh networks is usually done through broadcasting and controlled flooding, which requires the node's radios to be longer active and less efficient [5]. The Ad hoc On-Demand Distance Vector (AODV) [6] is a routing strategy used in the ZigBee protocol that accounts for node mobility, link failures, and packet losses. AODV uses broadcast messages to define routes, but routes are defined without considering the node's battery energy level. More radio usage, in turn, increases the energy consumption considerably on systems where the radio transmission (TX) and reception (RX) activity are much more energy-hungry than the CPU usage or sensor activity.

There are WSN approaches that focus on routing optimization considering energy consumption, such as the "Energy Aware Geographic Routing Protocol for Wireless Sensor Networks" (EAGRP) [7] and the "Low-Energy Adaptive Clustering Hierarchy" (LEACH) [8] from which many other approaches build upon [9]. EAGRP [7] is an improvement on AODV [6], which aims to extend the network lifetime by using energy and positioning information of each node to define routes. LEACH [8] maintains clusters and tries to aggregate data on specific nodes (cluster heads) as a way of saving power and extending the WSN lifetime. Another approach is the "Hybrid, Energy Efficient, Distributed clustering approach for Ad Hoc sensor networks" (HEED) algorithm [10], which selects the cluster heads based on remaining energy levels and communication cost. The survey by Baranidharan and Shanthi [11] compares LEACH, HEED and other approaches focused on WSN energy optimization.

Bluetooth is a wireless technology that can be used for WSNs and may be an interesting option since most commercially available smartphones, as well as many microcontroller devices and system-on-chip devices (SoCs), support it [12] [13]. One example of an SoC that supports Bluetooth is the ESP32. Giacomini et al. (2020) [14] describes the implementation of a Bluetooth routing approach for IoT environments using ESP32 SoCs.

One option for organizing Bluetooth networks is by forming a mesh network with the Bluetooth Mesh standard [12] (which will be referenced as BTMesh in this paper). BTMesh's latest version (5.1) was officially released in 2019 [15], and it tries to achieve more efficient energy draw when compared to other technologies such as Wi-Fi and ZigBee. BTMesh routes packets across the network by adopting a relay strategy that consists of controlled flooding [16].

The main objective of this work is to evaluate BTMesh in a simulated environment as a viable technology for routing sensor data towards a mobile sink node. Additionally, another goal is to design and experiment with a modified version of BTMesh that is tailored for routing towards a mobile sink node, aiming to achieve increased energy efficiency.

Hence, the research questions that this work aims to cover are:

1. MAIN-RQ1: **is BTMesh a viable technology for routing sensor data towards a mobile sink node?**
2. MAIN-RQ2: **can a slightly modified version of BTMesh improve energy efficiency for routing sensor data towards a mobile sink node?**

This work aims to propose two alternatives to BTMesh's default relay algorithm (MAM_0 and MAM_Δ) that may achieve higher energy efficiency as well as higher packet delivery rates and lower energy draw when routing data towards a Mobile-Hub. Those alternatives were evaluated in a simulated data collection context, considering BTMesh's default relay algorithm (which this work will call BTM-R from here on) as a benchmark.

The results indicate that one of the proposed algorithms (MAM_Δ) achieves a higher packet delivery rate to the Mobile-Hub when compared to BTM-R. This delivery rate considers the number of unique data packets received by the Mobile-Hub versus how many packets were generated and sent by all other nodes. This work also evaluated the global energy draw, the number of packets received on the Mobile-Hub, and the end-to-end delay (from each BTMesh sensor to the Mobile-Hub). The MAM_Δ algorithm presented lower end-to-end delay and received more unique data packets than BTM-R. However, in some configurations, it performed worse in terms of energy draw.

In the next chapter, we present some definitions of concepts used along this work. In chapter 3 we present the simulated scenarios considered for tree/forest monitoring as well as other possible applications. Chapter 4 contains related work in data collection, mesh wireless sensor networks, and BTMesh networking. Chapter 5 covers BTMesh and its characteristics. Chapter 6 describes the proposed data collection solution, explaining BTM-R,

MAM_0, and MAM_Δ relay algorithms. Chapter 7 describes the simulation model and evaluation metrics, as well as the software engineering work that was required to perform the simulations. Chapter 8 presents and discusses the simulation results. Chapter 9 concludes by presenting possible ramifications of this work.

2
Definitions

This section defines a few concepts related to Bluetooth Mesh, Wireless Mesh Networks, and the application scenarios described in chapter 3.

According to Madhavapeddy and Tse (2005) [17], Bluetooth is a wireless protocol for short-range communications that operates in the license-free 2.4GHz spectrum. The Bluetooth 4.0 protocol specification [18] introduced Bluetooth Low Energy (i.e., BLE), designed as a low-power solution for application control and monitoring [19]. Unlike Wi-Fi, which offers higher transfer rates and more extensive area cover, BLE is characterized by its low power requirements and low-cost transceiver chips.

Bluetooth Mesh [16] is a network protocol based on BLE that adds mesh networking capability to Bluetooth devices. It introduces the concept of Low power node and Friend node. Low power nodes (i.e., LPNs) are usually not connected to the power grid, relying on battery power. There are periods in which their radio is turned off. Thus they are not always listening for packets, and so they rely on Friend nodes (i.e., FNs) to receive them. LPNs request missed packets to FNs when they wake up.

Friend nodes can be battery-powered or not and can receive and acknowledge messages for LPNs during their sleep periods (when they turn their radio off to save power). They transmit received messages on behalf of LPNs upon their request.

Another important concept for Bluetooth Mesh networks is *Provisioning.* According to the Bluetooth Mesh specification [20], an unprovisioned device is a device that is not part of a Bluetooth Mesh network [21]. The process of adding an unprovisioned device to a Bluetooth Mesh network is called provisioning and is managed by a *provisioner*. The provisioning follows a fixed procedure which is defined in the Bluetooth Mesh specification. Effectively, provisioned devices are Bluetooth Mesh nodes.

Data collector (or also referred to as a Mobile-Hub [3]) is a smart device that is capable of connecting to nodes in the mesh network to receive data and transmit commands or configuration parameters. The data collector could transfer collected data to the internet or to a base station. This work only covers the bidirectional communication between Mesh nodes and Mobile-Hubs.

3
Applications and Simulation Scenario

In 2019, Brazil reached the highest level of deforestation in the Amazon forest since 2008, with 10,000 square km deforested, an area the size of Lebanon[1]. In the first seven months of 2020, more than 13,000 square km of the Amazon forest were burned[2], which is more than eight times the size of London. Scientists and environmental agencies can take early action to prevent wildfires and tree fall by collecting and analyzing sensor data snapshots from different parts of a forest [22]. The snapshots can contain data about temperature, humidity, luminosity, SAP flow (SAP is a fluid transported in tree's xylem cells), and air quality. This data can also be used to identify wildfires that are starting (which may help firefighters control it before they become larger) and to understand how forests change over time.

There are several fire detection approaches that transfer data from sensors to a base station or command center using Wireless Sensor Networks [22] [23] [24] [25] [26] but that specific use case is not be the main focus of this work. Instead, the focus is on evaluating the use of Bluetooth Mesh technology for building energy-efficient routing towards a mobile sink node (that could be applied in a fire detection scenario or in other situations in which a mobile node may need to collect data from a sensor network).

In 2015 and 2019, Brazil experienced two major disasters (Mariana[3] and Brumadinho[4]) in which tailings dams collapsed and killed hundreds of people. Tailings dams can be relatively large in area size, and may require continuous monitoring through sensors such as pressure, water level sensor, and deformation sensors [27]. Tailings dams monitoring also appears to be a possible application for sensor network data collection [28].

Beyond fire detection and tailings dam monitoring applications, different sectors can benefit from data collection using mobile sinks in Wireless Mesh Networks. In agriculture, sensors could be deployed to monitor vast amounts of land. Worker's phones or even drones could be used to gather data from those

[1]https://www.statista.com/statistics/1041354/number-wildfires-brazil/

[2]https://www.bbc.com/news/world-latin-america-53893161#: :text=In%20the%20first %20seven%20months,times%20the%20size%20of%20London.

[3]https://www.reuters.com/article/us-vale-sa-bhp-billiton-dam-idUSKCN0SU38I20151106

[4]https://www.bbc.com/news/business-47432134

sensors. This could be achieved by using a mobile WMN, where sensors can communicate and forward data to the collectors dynamically as they connect to them. Similarly, in manufacturing, deployed sensors in a factory could exchange data to be sent to the web (to a server that controls the factory plant, for example). They could also send data to other nodes with the intent of the nodes taking a quick decision without the information even needing to reach the web.

This work performed simulations considering scenarios in which there are multiple nodes placed on the ground and a single mobile sink node that moves in a circular trajectory. The mobile sink node moves at a constant speed and is sporadically within radio range of some of the ground nodes. The simulations involved varying mobile sink speed, amount of ground nodes, and other simulation parameters. Chapter 7 describes those parameters as well as the simulation model.

The scenarios were inspired by a possible application to monitor urban forests, tailings dams, or hillside vegetation, considering sensor nodes spread over an area without structured cabling and where individual wireless internet access is either unavailable or cost-prohibitive. Also, this work considered that the large size of the monitored area makes it infeasible to have a central wireless router directly connected to each node. In those scenarios, sensor nodes have a bounded distance to at least one other sensor node so that a connected wireless network can be formed.

This work considers that the nodes used in such applications operate on small batteries (with a limited energy supply). Hence, the data collection and routing algorithms should have power saving [5] as a strong requirement.

Since nodes might fail or run out of battery, redundancy is important to keep the network connection if some nodes shutdown. In Wireless Sensor Networks, fault tolerance can be achieved according to the relay nodes placement [29], however, this is out of the scope of this work.

The goal in each simulated scenario is to transfer data from the sensor nodes to the mobile data collector (i.e., Mobile-Hub). The Mobile-Hub [3] is a data sink that moves around the area and connects to some network nodes only sporadically and for a limited period before moving away again. This work's scenarios consider a single Mobile-Hub connected to the network at a time, that is always in movement at a constant speed.

Figures 3.1 and 3.2 illustrate an example scenario in which there is a network with ten sensor nodes and a Mobile-Hub connected to it in different instants (t1 and t2). Nodes labeled *LPN* are considered to be nodes that have low power and are connected to nodes labeled *FN RN*. Those *FN RN* nodes can propagate data for other nodes. In both instants, the arrows indicate the data flowing from all network nodes to the Mobile-Hub. The *FN RN* node marked in blue is the node that is connected to the Mobile-Hub.

This work discusses the communication inside the sensor network as well as between the sensor network and the Mobile-Hub data sink. Communication between the Mobile-Hub and the internet or external devices is out of the scope of this work.

To collect the data from sensors, the Mobile-Hub should use a wireless technology that is available on a wide range of retail smartphones, so that this opportunistic connectivity can be applied to any real-world scenario where a person, animal, or drone hauling a smartphone can be the Mobile-Hub, without needing additional wireless hub or dongle device.

Most commercially available smartphones have Bluetooth and Wi-Fi radio stacks [13]. Both technologies can be used to form a sensor network [16] [30], however, since the Bluetooth Low Energy (BLE) standard has reduced energy consumption features it may be a more interesting option for this work's scenario.

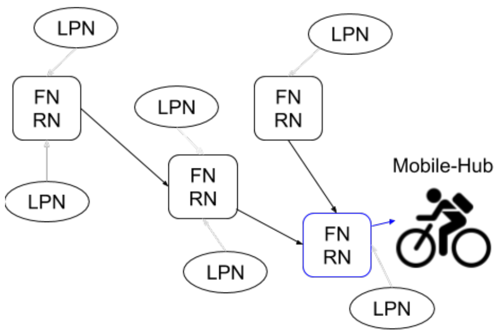


Figure 3.1: t1 - Mobile-Hub connected to a sensor network

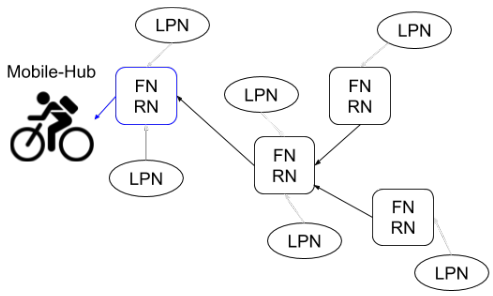


Figure 3.2: t2 - Mobile-Hub connected to a sensor network

4 Related Work

Data collection in sensor networks can be performed by a mobile node connected to the internet (such as a smartphone or a drone). In the ContextNet middleware [4], this is called a Mobile-Hub [3]. The ContextNet Mobile-Hub can connect to nearby sensors (mobile objects) and transmit their data to the internet. Currently, its approach is to connect to each sensor, one by one, gathering stored sensor data and relaying it to a gateway that sends it to a processing server.

Another way of collecting data from a sensor network with a Mobile-Hub is, instead of connecting to all nearby mobile objects, connecting to a single local object that is part of a mesh network and can gather data from other mobile objects [31]. This allows the Mobile-Hub to collect data faster, as the Mobile-Hub will connect to fewer mobile objects, and the mobile objects can send the data closer to the mobile node (which may increase the overall data transmission speed).

Mesh networking in the context of IoT can use technologies such as Wi-Fi Mesh [30] that may provide up to 300m signal range per device, or short-range communications (technologies such as Zigbee or Bluetooth) that consume less power [32] [33]. WMNs can dynamically reorganize and reconfigure. Their nodes can automatically establish and maintain mesh connectivity among themselves, bringing many advantages such as increased reliability and robustness [34].

In the context of sensor data collection, radio communications are responsible for most of the energy draw in microcontrollers [35] so, to save energy, it is important to choose approaches that minimize radio use. Minimizing radio usage in Mesh networks can be achieved through routing approaches that avoid message duplication and re-transmission such as [7] [8].

Several proposals that discuss and evaluate routing in mesh networks have been reported in the literature. Badawy et al. (2009) [36] propose flow control, routing, and resource allocation algorithms for WMNs (wireless mesh networks) considering solar-powered Mesh Nodes. Their work models the problem as a directed graph of Mesh Nodes and apply algorithms to optimize data flow given battery and routing constraints such as message priority. The simulation by Badawy et al. showed that the proposed algorithms might have high computational complexity, suggesting that those algorithms would not be suitable for the concept of Mesh IoMT, where we have hundreds of thousands of devices communicating among them.

Kopják and Sebestyén (2018) [37] compared centralized data collection methods in WMNs without mobility and analyzed their impact on nodes' battery life. It was tested in food quality and safety scenarios using battery-powered nodes connected to temperature sensors.

Adi et al. (2017) [38] described an implementation of a mesh network using Raspberry Pi micro-controllers to perform data collection of temperature and humidity data in rural areas.

With the goal of extending the lifetime of WSNs, Luo et al. (2006) [39] defined a routing protocol called MobiRoute, that supports sink mobility. Through intensive simulations of a mobile data collector and an implementation of MobiRoute, using a simulator named TOSSIM, the authors have shown the feasibility and the benefits of the mobile data collector approach concerning improved network lifetime. Their simulations covered networks containing less than 50 nodes, whereas the present work simulated networks with up to 200 nodes.

Di Francesco et al. (2011) [1] published an extensive survey of WMNs in which mobility is involved. It defined taxonomy for the data collection processes and analyzed data collection works for unmanned aerial vehicles (drones) acting as mobile data collectors. Under this taxonomy, the MAM routing algorithms (described in chapter 6) would be defined as having an asynchronous mobility-independent discovery approach, and a proxy-based routing approach.

Djedouboum et al. (2018) [2] studied the current state of data collection in Wireless Mesh Sensor Networks and analyzed its challenges in the context of Big Data. They also discussed the challenges of data collection when mobility is involved, like contact detection with mobile data collectors, quality of service (QoS), and location detection. The MAM algorithms do not cover quality of service and location detection, which are out of the scope of the present work. However, contact detection is an important part of the routing process and the MAM algorithms would be defined in the context of their research as feature-based routing protocols that rely on route discovery.

Bluetooth is a technology that can be used to form Wireless Mesh Sensor Networks, as described by Todtenberg and Kraemer (2019) [40] survey on Bluetooth multi-hop networks. This survey analyzed over 20 years of research on the topic and involved not only classic Bluetooth technology but also BLE (Bluetooth Low Energy), which is emerging as an excellent - and increasingly adopted - option for IoT and WSN because of its low cost, low energy consumption and well defined GAP/GATT protocols. The survey showed that over 85% of the publications from 1999 to mid-2019 were based on simulations or analytical results, or they were only describing Bluetooth multi-hop networks conceptually. Also, several of the publications analyzed by that survey highlight the need for real-world implementations of those types of networks.

There exist several studies about using Bluetooth as a technology for mesh networking prior to the official BTMesh specification and even the Bluetooth Low Energy protocol, such as [41], [42], [43], [44], [45]. Salonidis et al. (2001) [42] mentioned the idea of choosing specific nodes that will relay messages and uses a leader election approach to form clusters in which each leader is the relay node of a cluster.

After the BTMesh specification was released, some studies and simulations for the BTMesh technology have been explored, such as [12] [46] [47]. Leon and Nabi (2020) [46] analyses BTMesh in a real-world environment and reports limitations for message delivery as a result.

Hansen et al. [47] evaluate three relay selection mechanisms with the intent of reducing the number of relay nodes in the BTMesh network to reduce costs while preserving a certain level of redundancy. Their work is orthogonal to the present work, as it focuses on the BTMesh network formation (in which the network topology is defined), whereas the present work focuses on analyzing routing for data collection without altering the BTMesh network topology.

The authors could not find extensions of BTMesh relay algorithms that could be directly compared to the algorithms this work describes in Chapter 6 - this is - simple extensions to BTMesh routing that can be implemented on top of BLE. For instance, such extensions can be implemented on a microcontroller with BLE support without needing to alter BLE functionality. BTMesh adopts a flooding routing approach and there is an extensive amount of published work on this topic, with optimizations through concurrent-transmission based flooding [48] [49] [50] [51] [52] [53]. The *Harmony* algorithm [53] (2020) was tested through experimental evaluation and, compared to the state-of-the-art at the time of publication, presented 50% higher delivery rates and shorter end-to-end latencies in the presence of harsh Wi-Fi interference. However, such routing algorithms optimizations differ significantly from the algorithms proposed by this work, as they are not designed considering Bluetooth compatibility. The advantage of preserving Bluetooth compatibility is to more easily implement and deploy applications, as many commercially available devices such as smartphones and microcontrollers support Bluetooth [12] [13].

5
Bluetooth Mesh

Bluetooth [54] is a wireless technology already widespread across devices, from home automation to personal gadgets such as wireless headphones. Also, for BLE [55] compatible devices, there is the possibility of using Bluetooth Mesh Networking [16], which allows simultaneous connection across hundreds of connected devices. Those devices can exchange messages and collaborate to automate processes, increase the efficiency of an industrial plant, or bring more comfort to customers that can enjoy plug-and-play home automation.

This work has chosen Bluetooth Mesh (i.e., BTMesh) as the technology to be used to implement and evaluate the different routing approaches because it is a relatively new technology that can be integrated with most smartphones, and the authors wanted to explore it.

BTMesh is a mesh standard based on BLE that allows for many-to-many communication using the wireless Bluetooth protocol. The BTMesh specification was defined in the Mesh Profile[1] and Mesh Model[2] specifications by the Bluetooth Special Interest Group (Bluetooth SIG), and was adopted in 2017.

The standard uses BLE-specific advertising and scanning as underlying mechanisms to achieve flooding-like communication [16]. BTMesh flooding ensures that some nodes in the network, called Relay Nodes, repeat incoming messages so that they are relayed further until their destination is reached. Compared to conventional BLE advertising, BTMesh nodes do not send packets according to advertising intervals but send their packets directly after a random generated back-off time per channel.

To scan the advertisement channels for incoming packets, the mesh nodes use a 100% duty cycle, meaning that they are permanently scanning unless they are sending a packet.

In order to prevent the obvious problems caused by uncontrolled message flooding, BTMesh introduces relay cache features. Only nodes that have the relay feature enabled (i.e., RNs) will forward received messages to neighbor nodes. There is an LRU (least recently used) cache on each relay node that

[1]Mesh Profile Bluetooth® Specification, Bluetooth Technology Website. 2017-07-13
[2]Mesh Model Bluetooth® Specification, Bluetooth Technology Website. 2017-07-13

stores packet signatures and ensures a relay node only relays a specific message once. Also, each message has a Time-To-Live (TTL) field that represents the number of hops. Messages are only relayed if they are not in the cache, and the number of hops is less than 127 (the number 127 is defined by the Bluetooth specification and corresponds to a 1-octet opcode).

The full-time duty cycle to scan the different BLE advertisement channels directly impacts the node's energy consumption, thus requiring some means to support power-sensitive Mesh networks. The BTMesh standard tries to solve this by introducing Friendship and Low Power node features. A BTMesh Friend Node (i.e., FN) has mainly two responsibilities: storing incoming messages for nearby Low Power Nodes (i.e., LPNs) and sending those messages to the LPNs (LPNs periodically query their FNs for new messages). With the Friendship feature, Low Power nodes don't need to stay permanently scanning the network and can save battery power by keeping their radio stack disabled most of the time. Thus, according to the Bluetooth Mesh specification, FNs can function as intermediate storage and opportunistic relay nodes for the other "energy-restricted" mesh nodes that will awake typically only for communication with some FN during short periods to save energy.

Despite these advantages, BTMesh has two main problems considering this work's data collection scenario (described in section 3):

– Network size limitation - message routing is ad hoc, but performed through a controlled flooding approach that limits the number of hops to 127 (which could be insufficient for a vast area network application).
– Power consumption - although the flooding approach results in some flexibility in terms of handling nodes' neighbors change as well as low latency (packets always get sent through the shortest path), duplicate messages are sent through the network and this impacts power consumption.

This work focuses on improving power consumption for data collection scenarios using Bluetooth Mesh, and does not try to overcome the Bluetooth Mesh 127 hop limitation.

The network topology significantly influences how the network will behave, as defining which nodes are relay nodes, friend nodes, or low power nodes is not done dynamically and, if not done correctly, can make the network inefficient and even disconnected (by exceeding the hop limit from one node to another, or by lack of relay nodes that can forward messages between them). This work's configured simulation topologies, described in chapter 7, are connected networks with relay nodes, friend nodes, and low power nodes.

6
Data Collection Algorithms

This section describes the application of BTMesh for collecting data using a Mobile-Hub as described in chapter 3. The first approach uses the standard BTMesh relay implementation (i.e., BTM-R) to direct messages towards the Mobile-Hub. Also, this work proposes two alternative relay algorithms for the specific scenario that is being discussed.

When describing relay algorithms, this work only considers two types of packet: *Discovery packets*, that are used for route discovery; and *Data packets*, that are sent by the Mesh network nodes and contain sensor data that should be relayed to the Mobile-Hub.

To collect data from the network, the Mobile-Hub sends a discovery packet periodically (every 1 second) while moving around the area. Discovery packets are only generated by the passing Mobile-Hub and may be relayed by relay nodes further into the network. The discovery packet is used to notify the nodes that there is a data sink available to receive data, and those packets eventually reach every network node if they are received by one of the network relay nodes, considering the relay algorithm restrictions (e.g., BTM-R enforces a maximum hop limit). Once any node receives a discovery packet, it should send its sensor data as well as any stored sensor data towards the Mobile-Hub.

The BTM-R determines that the relay nodes only consider the number of hops made so far and whether they have already relayed the message, when evaluating if the message should be relayed.

The proposed alternatives to BTM-R only consider the scenario that was described in section 3 (routing data towards a single Mobile-Hub). This is an important difference between this work and alternative routing technologies for sensor networks that cover routing from any node to another in the network.

Each subsection contains a pseudo-code with a possible implementation of the described relay algorithms. The code would be run on every relay node for each packet they receive, and would receive as input: the sender's address (`senderAddress`), the number of packet hops (`messageHops`), and the packet's content (`messageBody`). Global variables stored in the nodes, available across local executions, are initialized in the pseudo-code's *Init* session.

6.1 BTMesh Relay (BTM-R) - Flooding

BTMesh's original relay algorithm (BTM-R) consists of a controlled flooding approach [56]. The algorithm combines two strategies to manage the network flooding:

1. limit the number of packet hops to 127 (corresponds to a 1-octet opcode, as defined by the specification);

2. avoid the same node relaying a packet multiple times.

For the latter strategy, the implementations compute incoming packet signatures and check them against an LRU cache. If they're present, they will not be relayed. If they're not present, then it will be relayed, and the signature will be cached.

The pseudo-code *Algorithm 1* illustrates BTM-R's implementation logic. Firstly it computes the packet hash and checks if it is present on an LRU cache, storing this information on a variable named recentlyRelayed (lines 1 and 2).

If it was recently relayed (recentlyRelayed == true) or if the number of messages hops is greater than 126, it stops executing, and the packet gets discarded (lines 3-5). Otherwise, the number of message hops is incremented and the packet is relayed through a broadcast (lines 6 and 7).

Two noticeable problems with this approach are: I) due to BTM-R routing approach, the messages may get relayed excessively and delivered multiple times since they are sent through every route possible. If much data is coming from sensors, there may be competition on multiple routes to deliver it. II) the 127 hop limit could be a problem depending on the nodes' topology layout/distribution, possibly making certain nodes unreachable to others.

Algorithm 1 BTMesh Relay (BTM-R)

```
Input: senderAddress, messageHops, messageBody
1: byte hash ← hashMessage(messageBody)
2: bool recentlyRelayed ← isInLRUCache(hash)
3: if (recentlyRelayed == true or messageHops > 126) then
4:     return
5: end if
6: hops ← messageHops + 1
7: broadcastMessage(messageBody, hops)
```

6.2 MAM_0 - Last known route

The MAM_0 algorithm is the first alternative routing algorithm this work designed and evaluated as a viable alternative to BTM-R for Mobile-Hub data routing. MAM_0 is based on a reactive routing strategy that only uses BTM-R's controlled flooding approach for Discovery packet propagation.

With the intent of maintaining a route to the Mobile-Hub, each node sets the last known directly connected node to have access to a Mobile-Hub. This forms a single destination directed acyclic graph (DAG), similar to a tree, which is similar to how routing algorithms such as RPL [57] organize. This information is updated on every node upon each received Discovery packet. With this approach, data packets are then no longer broadcast but are sent to a single node in each step.

The pseudo-code *Algorithm 2* illustrates MAM_0 implementation. On every relay node, a global variable `bestNodeAddress` is initialized as NULL. In line 1, the algorithm checks if the packet is a Discovery packet by looking at its body. If it is a Discovery packet, the global variable `bestNodeAddress` gets set to the packet sender address (line 2), the packet is relayed using BTM-R's mechanism, and the execution stops (lines 3-4).

If the packet is not a Discovery packet, the algorithm assumes it is a Data packet that should be directed towards a Mobile-Hub.

The algorithm continues to execute if the incoming packet is not a Discovery packet. It checks if the global variable `bestNodeAddress` is set (line 6), and if it is, the message gets sent to this address with an incremented number of hops (lines 7-8).

This logic implies that if the variable `bestNodeAddress` is not set, data packets will not be relayed. Also, it implies that if it relays a data packet, it will only be relayed to a single node. It was designed with those characteristics in consideration, with the intent of reducing the number of messages propagated through the network and thus the overall energy consumption.

6.3 MAM_Δ - Reactive least-hop route

The MAM_Δ algorithm also consists of a reactive routing approach in which a DAG is constructed and often updated, but it sets data packet routes to Mobile-Hubs based on its distance (in hops) to the Mobile-Hub. This distance is essentially the number of hops it takes from each node until the Mobile-Hub.

This approach requires a way of knowing in advance the number of hops from each node to the Mobile-Hub (and updating it often as this information

Algorithm 2 MAM_0 - Last known route

Init: bestNodeAddress ← NULL
Input: senderAddress, messageHops, messageBody

```
if (isDiscoveryMessage(messageBody) == true) then
    bestNodeAddress ← senderAddress
    bluetoothMeshRelay(senderAddress, messageHops, messageBody)
    return
end if
if (bestNodeAddress != NULL) then
    hops ← messageHops + 1
    sendMessage(bestNodeAddress, messageBody, hops)
end if
```

changes as the Mobile-Hub moves). This is achieved through the discovery message packet hop information.

Upon receiving a Discovery packet, the relay node evaluates if the sender would be the best destination for sending data to the Mobile-Hub. This evaluation considers the number of hops, as well as an expiration time.

The expiration time is called expiry and, when expired, makes the next Discovery packet have its sender set as the best node regardless of the number of hops. This way, the best routes get preserved for some time, but the logic accounts for them eventually becoming old/invalid. A Δ (milliseconds) parameter is used to control this expiration's length.

If the route is not expired and the number of hops of an incoming Discovery packet is less than the best one, then the algorithm considers it to be the new best destination, and sets its state accordingly (storing the new best sender address and number of hops and resetting the expiry/timeout).

The pseudo-code *Algorithm 3* describes MAM_Δ's implementation. It initializes three global variables: `bestNodeAddress` (as NULL), `bestNodeHops` (as zero), and `expiry` (as zero).

The variable `bestNodeAddress` is the address used to forward data packets, as the relay node considers it to be the best node to reach the Mobile-Hub.

The variable `bestNodeHops` is the number of hops from the current node to the Mobile-Hub. It is stored to decide whether or not the best node should be updated.

The variable `expiry` is the expiration time that was previously described. Upon receiving a Discovery packet, the algorithm will set the best node to the packet sender if the current time is greater than expiry.

In line 1, it checks if the packet is not a Discovery packet. If it's not a discovery packet, then it will be relayed if the best node is set (lines 2-5) and then execution will stop regardless of the best node being set or not (line 6).

If execution proceeds, it means the algorithm is handling a discovery packet. In line 8, it checks whether or not the current time is greater than expiry, and sets this boolean value to a variable called expired.

In line 9, it checks if the value of expired is true or the number of hops is less than the value held in the global variable bestNodeHops. If this check is valid, the bestNodeAddress global variable is set to the sender's address (line 10), the bestNodeHops global variable is set to the number of packet hops (line 11), and the expiry global variable is set to the current time plus the algorithm parameter Δ (line 12).

Line 14 is the final execution step run for all Discovery packets, to relay them using the BTM-R's logic.

Similarly to MAM_0, Discovery packets are still always relayed and Data packets are no longer broadcast (but get sent to a single node).

The advantage of MAM_Δ when compared to MAM_0 is that MAM_Δ temporarily preserves routes considering the distance to the Mobile-Hub. It was designed with the goal of maintaining shorter routes to the Mobile-Hub, which would imply a smaller energy consumption.

Algorithm 3 MAM_Δ - Reactive least-hop route

```
Init: bestNodeAddress ← NULL, bestNodeHops ← 0, expiry ← 0
Input: senderAddress, messageHops, messageBody
 1: if (isDiscoveryMessage(messageBody) == false) then
 2:     if (bestNodeAddress != NULL) then
 3:         hops ← messageHops + 1
 4:         sendMessage(bestNodeAddress, messageBody, hops)
 5:     end if
 6:     return
 7: end if
 8: bool expired ← NOW() > expiry
 9: if (expired == true or messageHops < bestNodeHops) then
10:     bestNodeAddress ← senderAddress
11:     bestNodeHops ← messageHops
12:     expiry ← NOW() + Δ
13: end if
14: bluetoothMeshRelay(senderAddress, messageHops, messageBody)
```

7 Simulation

The simulations rely on OMNET++ v5.6.1 and the INET framework. OMNET++[1] is a cross-platform simulator library and framework for discrete events, whereas the INET[2] framework is a network model library for the OMNET++ environment, that can simulate wired, wireless and mobile networks.

INET contains models for the Internet stack (TCP, UDP, IPv4, IPv6, OSPF, BGP), wired and wireless link layer protocols (Ethernet, PPP, IEEE 802.11). It has support for simulating node mobility, and is intended to be used for designing and validating new protocols as well as exploring different simulation scenarios. The framework supports OMNET++'s simulation features such as parameterization and result recording, and also provides a visualization interface that can be useful for behavior verification and debugging.

To the best of our knowledge, the INET framework lacks any official implementation of the BTMesh standard and BLE. Kajdocsi, Dörömbözi and Kovács (2019) [58] describes a BTMesh partial implementation using the OMNET++ framework. However, the authors state that their simulation model made it impossible to evaluate a network with more than 30 nodes, and also, it did not contain one of BTMesh's core features, Friend Nodes.

The goal of this work's simulations is to evaluate the feasibility and the performance of the proposed variants of relay algorithms in a data collection scenario with one Mobile-Hub. The simulations were initially implemented with 50 fixed nodes and then the model evolved to support a configurable number of nodes. Since energy efficiency is a significant concern and a metric that this work wants to evaluate, it is also included as part of the simulation model.

This work involved creating a model that supported the initial simulation requirements (50 nodes and BTMesh with Friendship feature support), and this was achieved by using INET's IEEE 802.15.4 model as a base, to be used across all simulations. This standard was chosen as it defines low-rate wireless personal area networks (LR-WPANs) like ZigBee [59], which has similar features to BLE. Siekkinen et. al (2012) [60] compare ZigBee and BLE

[1]https://omnetpp.org
[2]https://inet.omnetpp.org

in terms of energy efficiency, and also describe and compare the protocol's lower layers. For instance, they mention that the channel access in ZigBee (802.15.4) is CSMA/CA as opposed to BLE's frequency hopping collision avoidance and that ZigBee's over the air data rate is 250kbit/s while BLE's is 1Mbit/s.

This work's simulation model has the following characteristics that make it similar to BTMesh:

(a) maximum transmission range is configured for 100 meters

(b) transmission rate of 1 Mbps as defined in the BTMesh specification

(c) nodes can be configured as Relay Nodes, which use an algorithm (that can be overridden) upon deciding whether or not to relay received messages

(d) nodes can be configured as Low Power Nodes, which keep their radio off and only enable them occasionally to send and receive messages

(e) nodes can be configured as Friend Nodes, which receive and temporarily store messages for their registered Low Power Nodes

BTMesh messages can be fragmented using Bluetooth's Segmentation and Reassembly mechanism (SAR) and contain up to 384 bytes. Each segment can be 11 bytes long, with up to 3 of the initial bytes being reserved for the opcode. Due to this, we chose to test transmitting data that is only 8 bytes long so that it fits in a single segment (3 bytes of vendor-specific opcode (message type) + 8 bytes of payload = 11 bytes). This has significantly simplified the implementation as we did not need to support SAR.

BTMesh security and provisioning features were not implemented in the simulation model, as those features are out of the scope of this work.

The following metrics were collected and used to compare the different Relay Algorithms described in chapter 6:

- End-to-end delay (ms): the elapsed time in milliseconds from the moment a packet is sent by the source (sensor node) until when it is received by the Mobile-Hub.
- Delivery rate (%): the delivery rate of all of the generated data packets to the Mobile-Hub, this means the number of successfully delivered packets to a Mobile-Hub divided by the total number of sent data packets, multiplied by 100.
- Mobile-Hub received packets (bytes): the amount in bytes of received packets. The charts distinguish between unique and repeated data.

– Energy Draw (Joules): the amount of energy that was drawn, by all of the network nodes.

In some cases, BLE packets being relayed by BTMesh nodes may be lost. This may happen along the Mesh-internal routes due to multiple devices transmitting simultaneously, causing packet collisions, and due to interference caused by any sources. Losing packets can also happen at the last hop towards the Mobile-Hub, which may be drifting away from the Mesh node from which it was receiving relayed packets.

BLE advertising packets, the type of packet used by BTMesh, only implement a simple collision avoidance mechanism. It changes the advertising channels sequentially and also has a random delay between 0 and 10ms for consecutive sends on the same channel, according to Bluetooth Core v5.0 specification [16]. On Bluetooth Core v5.1 specification [15], collision avoidance is slightly improved by allowing advertising channels to be chosen at random instead of sequentially.

This work's simulations account for the possibility of packets being lost, with a radio interference model and CSMA/CA simulation as well as a Mobile-Hub movement model. The radio layer was imported from INET's 802.15.4 model, which uses CSMA/CA not BTMesh's simpler collision avoidance approach [60]. The movement model was imported from INET, called *CircleMobility* [61], in which the node simply moves around a circle at a fixed speed. For this work's 50 node-simulation, a fixed radius of 400 meters was used.

For BTM-R and MAM_0 Relay algorithms, the variation in simulation execution was only the Mobile-Hub speed. For the MAM_Δ, the simulation varied the speed and the algorithm's Δ parameter.

Varying execution time was not very significant in the context of this work, since it is comparing relay algorithms. The execution time only needed to be big enough for the Mobile-Hub to connect to some of the nodes and collect data.

Each simulation was run for 200 seconds, which is the default for OMNET++'s simulations.

Initially, for the 50 node simulation, the network was composed of 13 LPNs and 37 FNs (with all FNs being Relay Nodes), and a single Mobile-Hub that circled around the mesh nodes. This map was manually generated by the authors, and is a connected network corresponding to the screenshot from figure 7.1. The following variations values were tested:

– Relay type: BTM-R, MAM0, MAM_Δ

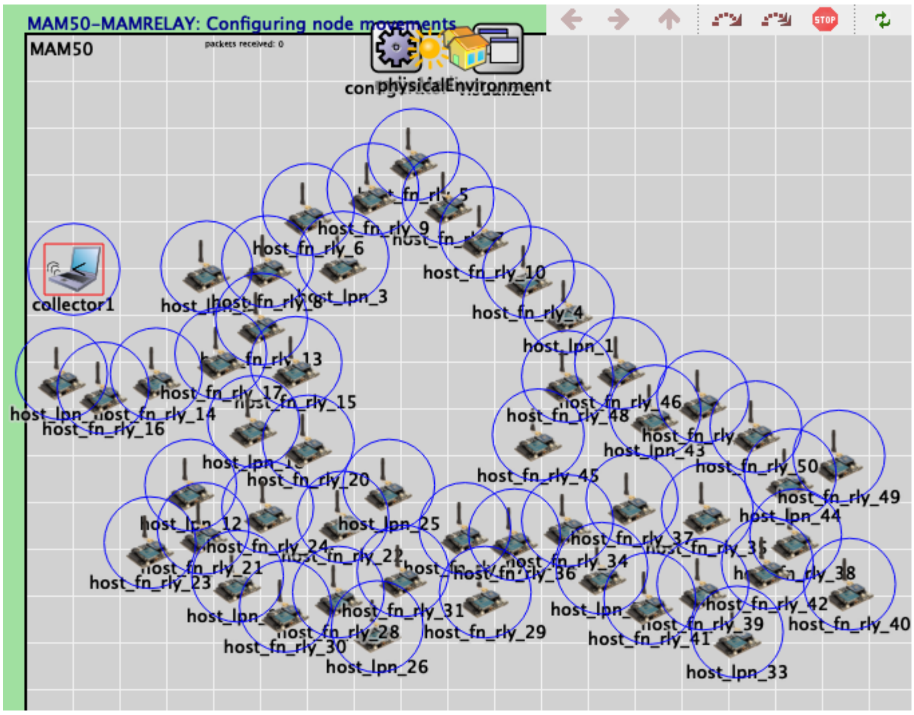


Figure 7.1: MAM50 map plotted in the OMNET++ IDE

- Speed (in meters per second): 2 (equivalent to a pedestrian), 6 (equivalent to a cyclist), 14 (equivalent to a quadcopter)
- Δ (in milliseconds): 5, 10, 20, 50, 100, 500

The Mobile-Hub's circular trajectory was across a 400 meters radius and covered only a subset of the network nodes. On the 14m/s scenario, the Mobile-Hub connected to 10 Relay Nodes (20% of all network nodes) and only to 6 Relay Nodes (12% of all network nodes) on the 6 m/s and 2m/s scenarios.

After obtaining the first batch of results (which will be detailed in the next chapter) and analyzing them, some additional research questions emerged:

- MAM50-RQ1: **would the unique received packets be greater than BTM-R's for Δ=100 when using different maps and topologies?**
- MAM50-RQ2: **would the delivery rate be higher than BTM-R's, for Δ>=20 values when using different maps and topologies?**

Hence, aiming to answer MAM50-RQ1 and MAM50-RQ2, we changed the simulation model to also support randomly placed nodes, configurable area size, configurable node amount and configurable proportion of FNs and LPNs (LPN/FN ratio).

The following variations values were tested considering all valid permutations of parameters, with 10 different maps per configuration, resulting in more than eleven thousand simulations that are part of the results detailed in chapter 8:

- Relay type: BTM-R, MAM0, MAM_Δ
- Speed (in meters per second): 2 (equivalent to a pedestrian), 6 (equivalent to a cyclist), 14 (equivalent to a quadcopter)
- Δ (in milliseconds): 0, 2, 5, 10, 20, 50, 100, 500, 1000, 5000, 10000, 15000, 20000
- Area size limit (in square meters): 400x400, 800x800, 1000x1000
- Node amount: 50, 100, 200
- LPN/FN ratio: 4, 8, 12

Nodes were randomly placed respecting a minimum distance of 10 meters and a maximum distance of 100 meters, and were placed so that the network remained connected through relay nodes. The random node placement seemed to make sense considering the described scenarios in chapter 3, such as tailings dam monitoring - where, in an emergency, sensors could be deployed by being dropped from a plane. Figure 7.2 contains one of the randomly generated maps, that contains nodes spread across a 105,000 sqm. Figure 7.3 is the same map but as an overlay in a satellite image of a high-risk tailings dam located in Minas Gerais (MG) - Brazil.

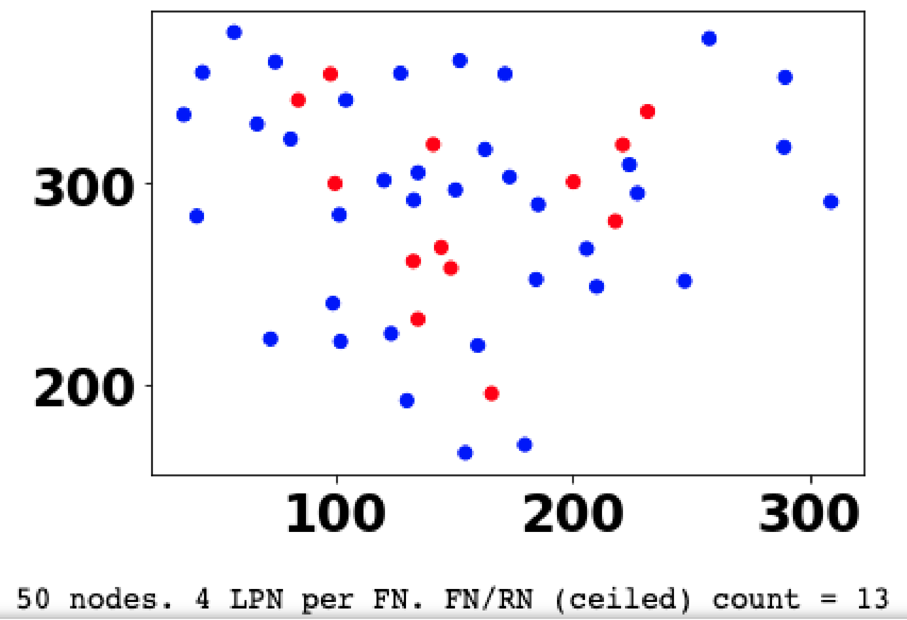


Figure 7.2: Random map with 50 nodes spread across an area of 105,000 sqm. Points in red represent Friend+Relay nodes and in blue Low-Power Nodes.

Figure 7.3: The same random map with 50 nodes, spread across a similar 105,000 sqm area - a high-risk tailings dam located in Barão de Cocais, MG (Brazil).[3]

All of the source code that was used to generate the maps and topologies, the simulation model code, as well as the tools used to generate the charts presented in this paper, is publicly available on GitHub[4].

[3]"High-risk tailings dam data obtained from Agência Nacional de Mineração https://app.anm.gov.br/SIGBM/Publico - Access on February 2, 2021. Images obtained from Google Maps - Map data ©2021 Imagery ©2021, CNES / Airbus, Maxar Technologies"
[4]"https://github.com/marcelopaulon/PUC-Rio-MSCDIS-MonitoredArboretumMesh"

8
Evaluation

The results are divided into two separate sets:

1. MAM50 - the initial simulation, performed by manually placing 50 nodes in an OMNET++ map, containing the results of 32 simulations varying speed and Δ parameters; and

2. MAMSET - a much broader set of simulations varying speed, Δ, node quantity, node placement, area size, and LPN/FN ratio parameters, resulting in a dataset containing more than eleven thousand simulations and 28.8GB of scalar data.

This work presents and analyses the first set of simulations - MAM50 - by comparing individual simulations while presenting and analyzing the latter set - MAMSET - through sliced aggregated data as it would be unfeasible to present and compare all 11,000+ simulations individually on this case.

It is worth mentioning that generating MAMSET was a significant engineering challenge in terms of distributing OMNET++ simulation execution and processing the results. However, detailing those challenges and the implemented solutions is out of the scope of this work. The open-source projects Jupyter Notebook[1], Pandas[2] and PostgreSQL[3] were crucial for enabling this research to happen in a timely manner.

8.1
MAM50 - Preliminary Simulations with 50 nodes

This section presents the four evaluated metrics for MAM50 - End-to-end (sensors to sink) delay, Energy Draw, Delivery Rate, and Received Packets, in the next subsections. Finally, this work analyses the results and apparent trade-offs from MAM50's results.

[1]Jupyter Project - `https://jupyter.org`
[2]Pandas - `https://pandas.pydata.org`
[3]PostgreSQL - `https://www.postgresql.org/`

8.1.1
End-to-end (sensors to sink) delay

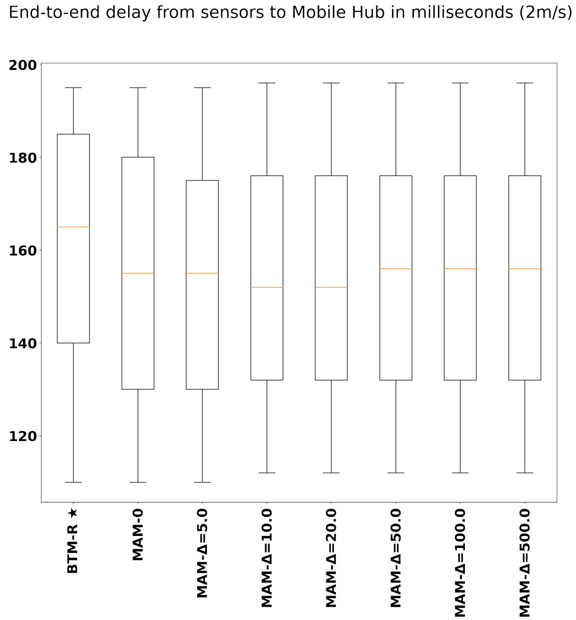
Figure 8.1 shows the end-to-end delay (from the moment data packets are sent by the source node until they reach the Mobile-Hub) in milliseconds. The charts show the results, presented as box plots[4], for BTM-R, MAM_0, and MAM_Δ with varied Δ values. Each box plot representing the data includes the 25% quartile (Q1), median (marked in red), and the 75% quartile (Q3). Outliers have been omitted to facilitate visualization. Higher values indicate that the delay was greater, which means that messages took more time to be delivered to the Mobile-Hub.

For 2 m/s, the results (Figure 8.1(a)) show a median of 165ms for BTM-R, which is greater than MAM_0's 155ms median as well as all of the MAM_Δ delay medians (with the best one of 152ms). Those results indicate that the proposed alternatives performed better than BTM-R in terms of end-to-end delay for the tested cases.

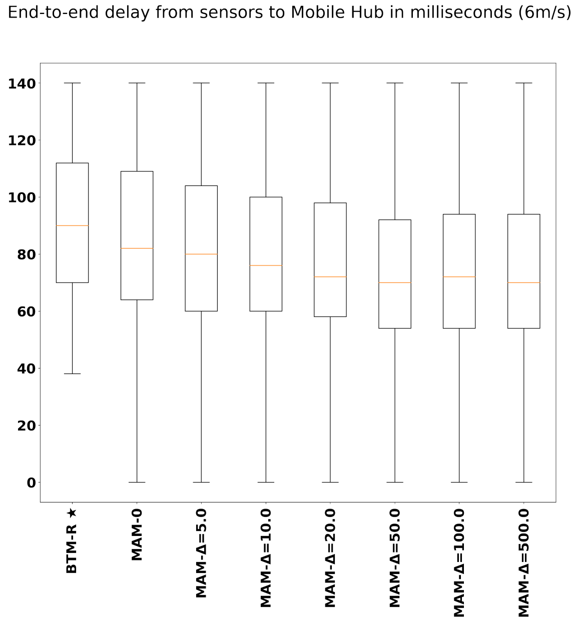
For 6 m/s (Figure 8.1(b)), a similar behavior occurred: MAM_0 and MAM_Δ algorithms also outperformed BTM-R, however, with even lower delays. BTM-R presented a median of 90ms, which was higher than MAM_Δ's medians (the lowest one was of 70ms). Those results reinforced the 2 m/s indication that MAM_0 and MAM_Δ performed better in terms of end-to-end delay, also at a three times higher speed (6 m/s).

For 14 m/s (Figure 8.1(c)), MAM_0 no longer had a lower median than BTM-R's. BTM-R's median was of 42ms while MAM_0's median was of 45ms. However, all of MAM_Δ's medians were lower than BTM-R's, with the best one being 30ms. Those results indicate that MAM_0 performs worse in terms of end-to-end delay when compared to BTM-R at 14 m/s Mobile-Hub speed, and that the MAM_Δ algorithm still performed better with that higher speed.

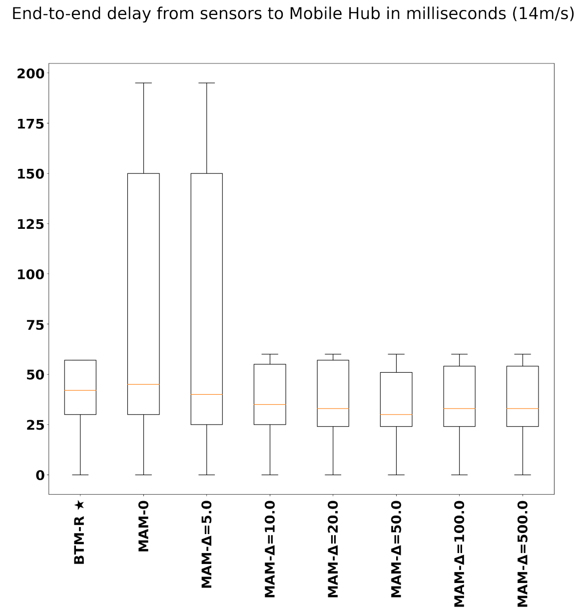
[4] Boxplot - `http://www.stat.yale.edu/Courses/1997-98/101/boxplot.htm`



8.1(a): $2m/s$



8.1(b): $6m/s$



8.1(c): $14m/s$

Figure 8.1: Average delay comparison between BTM-R and MAM relay with least-hop route - MAM50.

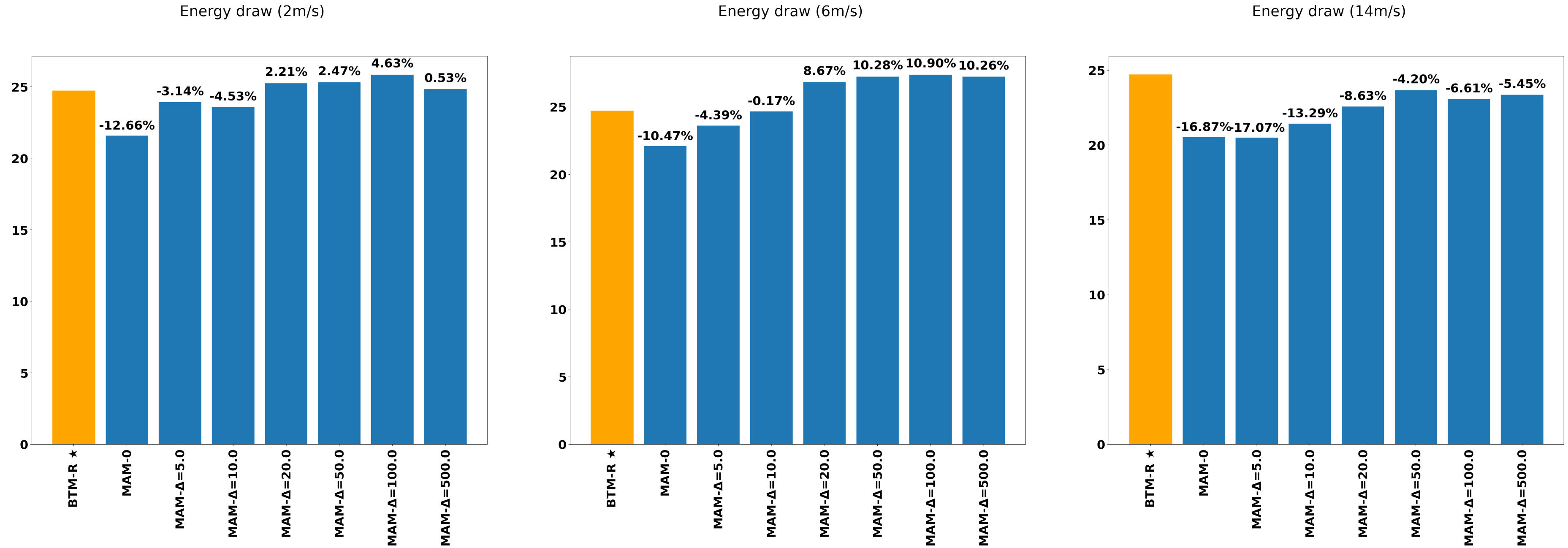
8.1.2 Energy Draw

Figure 8.2 presents the energy draw of all Mesh nodes in Joules. The values were aggregated as a sum, in which the Mobile-Hub energy draw was neglected, and are displayed on a bar chart. The horizontal axis presents each relay algorithm that was used, and the vertical axis contains the energy draw values in Joules. Higher values indicate that more energy was consumed, however, this is not necessarily an indicator of worse overall performance since more messages could have been sent or the simulation presented a higher packet delivery rate. On each alternate algorithm bar, there is a percentage indicating the percentage comparison between each value and BTM-R's.

For 2 m/s, the results (Figure 8.2(a)) show that MAM_0 energy draw was as low as 12.66% less than BTM-R's. MAM_0 was the algorithm that drew less energy among all others at this Mobile-Hub speed. For MAM_Δ with Δ=100, most energy was consumed among the 2m/s simulations, 4.6% more than BTM-R. Those results indicate that, for 2 m/s Mobile-Hub speed, the proposed alternatives can consume more energy than BTM-R in some cases, as well as less energy in other cases, depending on how the alternative algorithms are parameterized.

For 6 m/s, the results (Figure 8.2(b)) show that MAM_0 energy draw was as low as 10.47% less than BTM-R's. MAM_0 was the algorithm that drew less energy among all others at this Mobile-Hub speed. For MAM_Δ with Δ=100, most energy was consumed among the 6m/s simulations, 10.9% more than BTM-R. Those results indicate that, for 6 m/s Mobile-Hub speed, the proposed alternatives can consume more energy than BTM-R in some cases, as well as less energy in other cases, depending on how the alternative algorithms are parameterized.

For 14 m/s, the results (Figure 8.2(b)) show that MAM_0 energy draw was as low as 16.87% less than BTM-R's. The chart indicates that the lowest energy draw for this speed was with the MAM_Δ algorithm with $\Delta = 5$, 17.07% less than BTM-R. For MAM_Δ with Δ=50, most energy was consumed among the proposed alternatives for 14m/s simulations, 4.2% less than BTM-R. Those results indicate that, for 14 m/s Mobile-Hub speed, the proposed alternatives consumed less energy than BTM-R; however, their energy consumption also varied according to the algorithm's parameterized.



8.2(a): $2m/s$ 8.2(b): $6m/s$ 8.2(c): $14m/s$

Figure 8.2: Energy Draw (Joules) - MAM50.

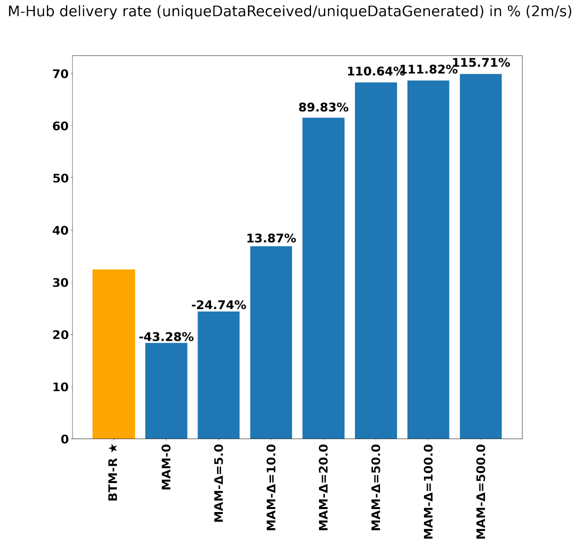
8.1.3 Delivery Rate

Figures 8.3(a)-8.3(c) present the delivery rate (to the Mobile-Hub) of all Mesh nodes data packets, in percentage. The values consider the amount of unique data packets received divided by the amount of unique data generated by each sensor node. The horizontal axis presents each relay algorithm that was used, and the vertical axis contains the delivery rate percentage values. Higher values indicate that more unique messages were delivered successfully to the Mobile-Hub. On each alternate algorithm bar, there is a percentage indicating the percentage comparison between each value and BTM-R's.

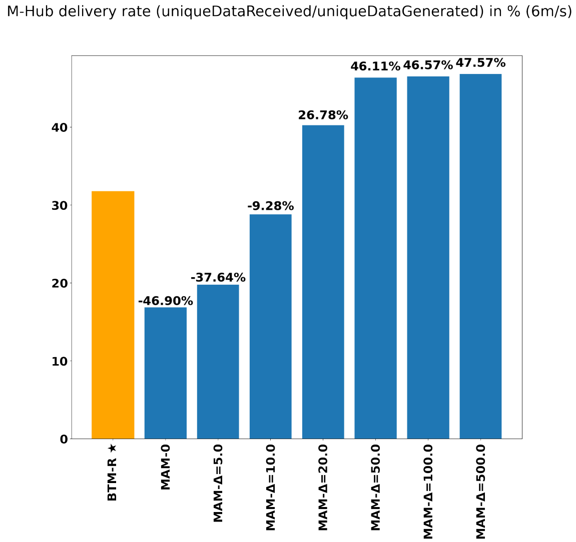
For 2 m/s, the results (Figure 8.3(a)) show that MAM_0 delivery rate was 18.37%, which is 43.28% lower than BTM-R's 32.40% rate. MAM_0 presented the lowest delivery rate among all others at this Mobile-Hub speed. For MAM_Δ with Δ=500, the delivery rate was of 69.89%, the highest among the 2 m/s simulations, 115.71% greater than BTM-R. Those results indicate that, for 2 m/s Mobile-Hub speed, the proposed alternatives can achieve higher and lower delivery rates when compared to BTM-R, depending on how the alternative algorithms are parameterized.

For 6 m/s, the results (Figure 8.3(b)) show that MAM_0 delivery rate was 16.85%, which is 46.9% lower than BTM-R's 31.73% rate. MAM_0 presented the lowest delivery rate among all others at this Mobile-Hub speed. For MAM_Δ with Δ=500, the delivery rate was 46.83%, the highest among the 6 m/s simulations, 47.57% greater than BTM-R. Those results indicate that, for 6 m/s Mobile-Hub speed, the proposed alternatives can also achieve higher and lower delivery rates when compared to BTM-R, depending on how the alternative algorithms are parameterized.

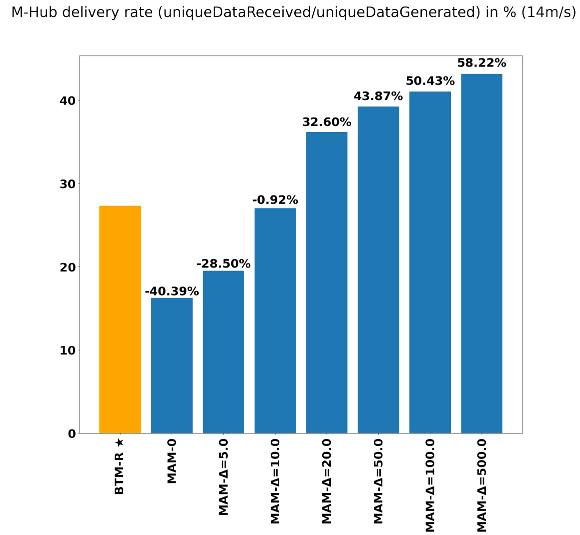
For 14 m/s, the results (Figure 8.3(b)) show that MAM_0 delivery rate was 16.26%, which is 40.39% lower than BTM-R's 27.29% rate, and it was the lowest delivery rate among all others at this Mobile-Hub speed. For MAM_Δ with Δ=500, the delivery rate was 43.18%, the highest among the 14 m/s simulations, 58.22% greater than BTM-R. Those results indicate that, for 14 m/s Mobile-Hub speed, the proposed alternatives can also achieve higher and lower delivery rates when compared to BTM-R, depending on how the alternative algorithms are parameterized.



8.3(a): $2m/s$



8.3(b): $6m/s$



8.3(c): $14m/s$

Figure 8.3: Delivery Rate (%) - MAM50.

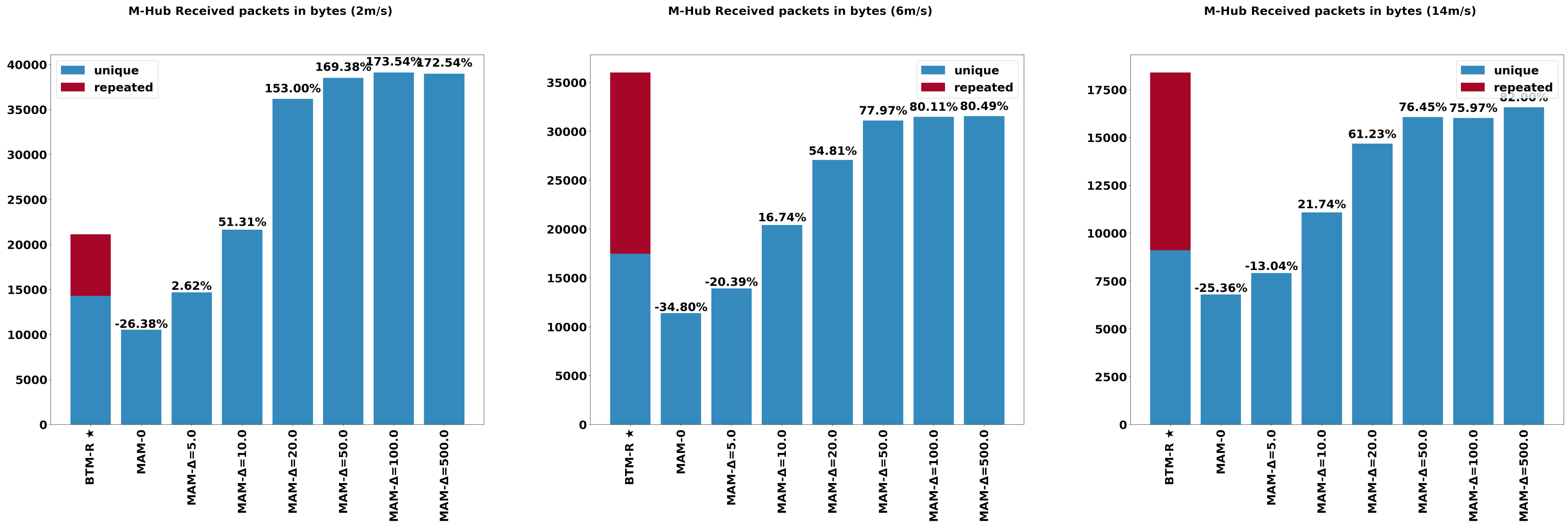
8.1.4 Received Packets

Figures 8.4(a)-8.4(c) present the amount of data packets received by the Mobile-Hub, in bytes. The values consider the amount of unique data packets received and display them on a bar chart indicating how many of them were duplicates (if any). The horizontal axis presents each relay algorithm that was used, and the vertical axis contains the amount of data packets in bytes. Higher values indicate that more messages were received by the Mobile-Hub; however, unique values are painted in blue and repeated values in red. On each alternate algorithm bar, there is a percentage indicating the percentage comparison between each unique value portion and BTM-R's unique value portion.

For 2 m/s, the results (Figure 8.4(a)) show that on the BTM-R simulation, the Mobile-Hub collected 14.3k unique bytes and a total of 21.12k bytes of data. Thus, in this BTM-R simulation, 32.2% of the collected data were duplicate packets. MAM_0 received 10.52k bytes of unique data packets, which is 26.38% lower than BTM-R's, and it was the algorithm with the lowest amount of unique data received among all others at this Mobile-Hub speed. For MAM_Δ, all tested Δ values presented a higher amount of unique data packets collected when compared to BTM-R. With Δ=100, the amount of unique data received was 39.11k bytes, the highest among the 2 m/s simulations, 173.54% greater than BTM-R. Those results indicate that, for 2 m/s Mobile-Hub speed, the proposed alternatives can achieve higher and lower amounts of unique data packets received by the Mobile-Hub when compared to BTM-R, depending on how the alternative algorithms are configured. Also, it indicated that MAM_0 and all tested MAM_Δ algorithms did not result in the delivery of duplicated data packets to the Mobile-Hub at 2 m/s Mobile-Hub speed.

For 6 m/s, the results (Figure 8.4(b)) show that on the BTM-R simulation, the Mobile-Hub collected 17.47k unique bytes and a total of 36.02k bytes of data. Thus, in this BTM-R simulation, 51.4% of the collected data were duplicate packets. MAM_0 received 11.39k bytes of unique data packets, which is 34.8% lower than BTM-R's, and it was the algorithm with the lowest amount of unique data received among all others at this Mobile-Hub speed. For MAM_Δ, with Δ=5, the Mobile-Hub collected 13.91k bytes of unique data, 20.39% less than BTM-R. For all other tested Δ values, results presented a higher amount of unique data packets collected when compared to BTM-R. With Δ=500, the amount of unique data received was 31.54k bytes, the highest among the 6 m/s simulations, 80.49% greater than BTM-R. Those results indicate that, for 6 m/s Mobile-Hub speed, the proposed alternatives can achieve higher and lower amounts of unique data packets received by the Mobile-Hub when compared to BTM-R, depending on how the alternative algorithms are configured. Also, it indicated that MAM_0 and all tested MAM_Δ algorithms did not result in the delivery of duplicated data packets to the Mobile-Hub at 6 m/s Mobile-Hub speed.

For 14 m/s, the results (Figure 8.4(c)) show that on the BTM-R simulation, the Mobile-Hub collected 9.1k unique bytes and a total of 18.4k bytes of data. Thus, in this simulation, 50.5% of the collected data were duplicate packets. MAM_0 received 6.79k bytes of unique data packets, which is 25.36% lower than BTM-R's, and it was the algorithm with the lowest amount of unique data received among all others at this Mobile-Hub speed. For MAM_Δ, with Δ=5, the Mobile-Hub collected 7.92k bytes of unique data, 13.04% less than BTM-R. For all other tested Δ values, results presented a higher amount of unique data packets collected when compared to BTM-R. With Δ=500, the amount of unique data received was 16.57k bytes, the highest among the 14 m/s simulations, 82% greater than BTM-R. Those results indicate that, for 14 m/s Mobile-Hub speed, the proposed alternatives can achieve higher and lower amounts of unique data packets received by the Mobile-Hub when compared to BTM-R, depending on how the alternative algorithms are configured. Also, it indicated that MAM_0 and all tested MAM_Δ algorithms did not result in the delivery of duplicated data packets to the Mobile-Hub at 14 m/s Mobile-Hub speed.



8.4(a): $2m/s$ 8.4(b): $6m/s$ 8.4(c): $14m/s$

Figure 8.4: Packets received comparison between BTM-R and MAM_{Δ} with least-hop route - MAM50.

8.1.5 Analysis and Tradeoffs - MAM50

Compared to BTM-R, MAM_0 consumed less energy across all speeds (-12.66% for 2m/s, -10.47% for 6m/s, -16.87% for 14m/s) and lower end-to-end delay in most cases. However, delivery rate (figure 8.3) was lower compared to BTM-R in all of those cases (-43.28% for 2m/s, -46.90% for 6m/s, -40.39% for 14m/s). Also, the received packets in bytes values were lower than BTM-R's (-26.38% for 2m/s, -34.80% for 6m/s, -25.36% for 14m/s). This indicates that, overall, BTM-R outperforms MAM_0.

The MAM_Δ algorithm was tested with different Δ values. For Δ=5, the delivery rate was lower than BTM-R's across all speeds (-24.74% for 2m/s, -37.64% for 6m/s, -28.50% for 14m/s). At 2m/s, the received packets in bytes were 2.62% higher than BTM-R; however, at 6m/s it was 20.39% lower, and at 14m/s 13.04% lower. The energy draw was lower than BTM-R's across all speeds (-3.14% for 2m/s, -4.39% for 6m/s, -17.07% for 14m/s). This indicates that, in most cases, BTM-R also outperforms MAM_Δ with Δ=5.

For Δ=10, the delivery rate was higher than BTM-R's at 2m/s and lower at 6m/s and 14m/s (+13.87% for 2m/s, -9.28% for 6m/s, -0.92% for 14m/s). The received packets in bytes were higher than BTM-R across all speeds (+51.31% for 2m/s, +16.74% for 6m/s, +21.74% for 14m/s). The energy draw was lower than BTM-R's across all speeds (-4.53% for 2m/s, -0.17% for 6m/s, -13.29% for 14m/s). This indicates that, in most cases, MAM_Δ with Δ=10 outperforms BTM-R.

For higher Δ values (>= 20.0), MAM_Δ presented delivery rates that were higher than BTM-R across all speeds. The highest Δ value that was simulated (Δ=500) presented the highest delivery rates across all different speeds: 69.89% successful delivery for 2 m/s (a 115.71% increase compared to BTM-R's 32.40% rate), 46.83% for 6 m/s (a 47.57% increase compared to BTM-R's rate) and 43.18% (a 58.22% increase compared to BTM-R's rate). In most cases for Δ values >= 20.0, the energy draw was higher compared to BTM-R. However, the scenario that consumed most energy (Δ=100 at 6m/s) represents only a 10.90% increase compared to BTM-R, with a 46.57% delivery rate increase, and an 80.11% received packets in bytes increase. It may be an interesting trade-off to spend 10.9% more energy but receive 80.11% more data.

The results indicate that, with the correct tuning (Δ parameter), MAM_Δ may achieve a significantly better performance compared to MAM_0 and BTM-R in terms of unique received packets and delivery rate, as well as energy efficiency (when we consider the amount of energy drawn proportionally to the higher delivery rates and higher unique data packets received).

8.2 MAMSET - Simulations with randomly generated maps

As previously mentioned in chapter 7, two additional research questions emerged after obtaining and analyzing MAM50's results:

- MAM50-RQ1: **would the unique received packets be greater than BTM-R's for Δ=100 when using different maps and topologies?** *(given that in MAM50 results with Δ=100 performed significantly better in terms of unique received packets when compared to other MAM-Δ values and to BTM-R in this particular map and topology)*
- MAM50-RQ2: **would the delivery rate be higher than BTM-R's, for Δ>=20 values when using different maps and topologies?** *(given that in MAM50 the delivery rate increased when compared to BTM-R's for all tested Δ>=20 values)*

As MAMSET consists of more than eleven thousand simulations, it is infeasible to present the results from MAMSET using the same type of visualizations as MAM50. Instead, this work presents MAMSET's data through selected slices and aggregations, with the goal of analyzing the results in regard to the MAIN-RQ1, MAIN-RQ2, MAM50-RQ1 and MAM50-RQ2 research questions.

In addition to the four evaluated metrics for MAM50 - End-to-end (sensors to sink) delay, Energy Draw, Delivery Rate and Received Packets - for MAMSET this work also evaluated another metric: energy efficiency in Bytes per Joule (B/J). B/J has been used in past work that aimed to benchmark energy efficiency of BLE compared to other radio technologies like ZigBee [60]. This metric is calculated by dividing the amount (in bytes) of unique packets delivered to the mobile sink by the amount of Joules that all the network nodes consumed. Higher values in B/J indicate an increases energy efficiency as more unique Bytes are delivered per consumed Joule.

This section presents the five evaluated metrics for MAMSET - End-to-end (sensors to sink) delay, Energy Draw, Delivery Rate, Received Packets, and Energy Efficiency in B/J, in the next subsections. All charts are boxplots similar to the charts presented in section 8.1.1. Finally, this work analyses the results and apparent trade-offs from MAMSET's results.

8.2.1
End-to-end (sensors to sink) delay

Figure 8.5 presents the end-to-end delay (from the moment data packets are sent by the source node until they reach the Mobile-Hub) in milliseconds. The chart's data compares the delay between BTM-R and MAM_{Δ} with Δ=100 (i.e., $MAM_{\Delta=100}$).

The results show a median of 30.11ms for BTM-R, which is lower than the median of simulations with $MAM_{\Delta=100}$ (30.34ms). Although the median delay difference between tests run with BTM-R and $MAM_{\Delta=100}$ is less than a millisecond, those results indicate that the BTM-R performed better in terms of end-to-end delay for the tested cases.

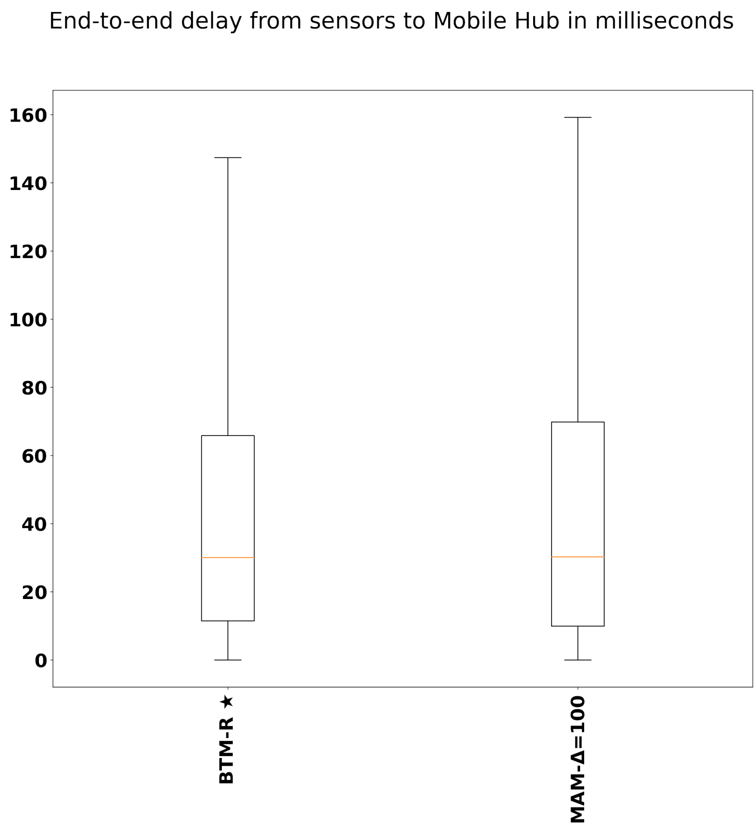


Figure 8.5: End-to-end delay from sensors to Mobile-Hub comparison between BTM-R and $MAM_{\Delta=100}$ - MAMSET.

8.2.2 Energy draw

Figure 8.6 shows the Energy Draw (in Joules) across all 11,072 simulations that this work performed and labeled the results as MAMSET. The data is divided between BTM-R and all other algorithms each of them divided into LPN/FN ratios (of 4, 8, or 12 Low Power Nodes per Friend Node). BTM-R with an LPN/FN ratio of 4 presented the highest median Energy Draw (182.31J). As expected, increasing the LPN/FN ratio decreased the Energy Draw of the simulation. BTM-R with presented a median Energy Draw of 78.93J for a LPN/FN ratio of 8, and of 54.75J for a LPN/FN ratio of 12. MAM presented a lower median than BTM-R for all LPN/FN ratios (46.73J for LPN/FN=4, 46.74J for LPN/FN=8, 46.74J for LPN/FN=12), however, this considers data from MAM_0 and MAM_Δ with all tested Δ values. MAM_0 and MAM_Δ not always presented a low energy draw in previous simulations, as shown in section 8.1.2 which analyzed the same metric for the MAM50 dataset.

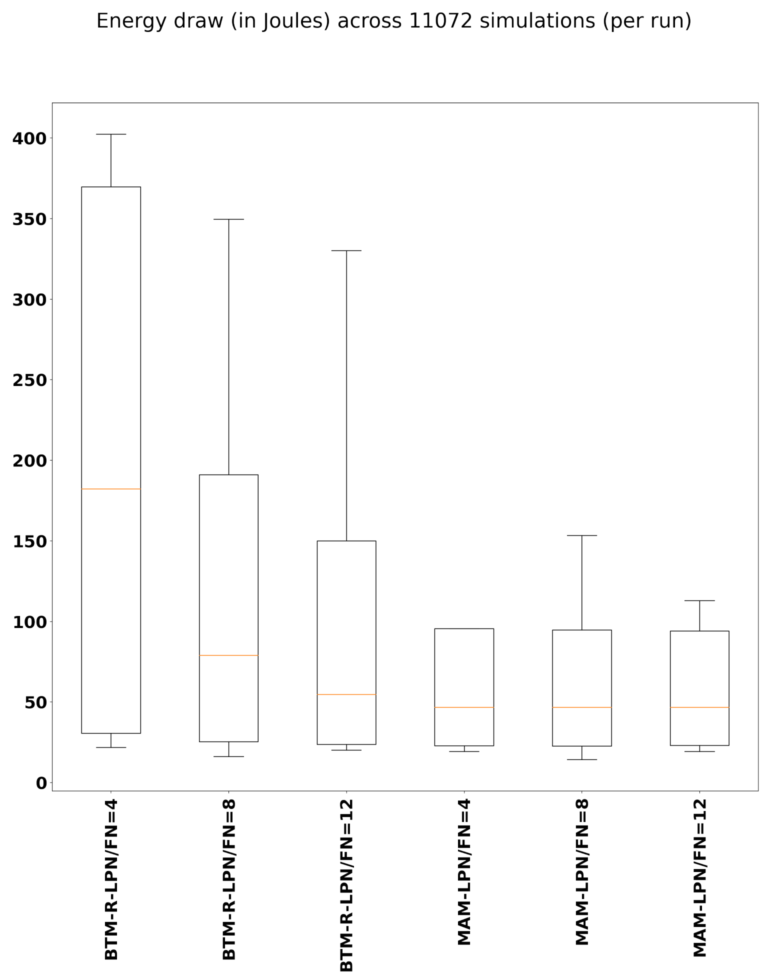


Figure 8.6: Energy draw (in Joules) across 11072 simulations by LPN/FN ratio - MAMSET

Figure 8.7 shows the Energy Draw (in Joules) across BTM-R and MAM, however, for MAM_{Δ} with Δ=100 which presented the highest amount of unique packets delivered to Mobile-Hub in MAM50 (as shown in section 8.1.4. MAM also presented a lower median Energy Draw than BTM-R for all LPN/FN ratios (47.15J for LPN/FN=4, 46.78J for LPN/FN=8, 46.52J for LPN/FN=12). As mentioned in section 8.1.2 where this work analyzed Energy Draw for MAM50, this metric may indicate the algorithm energy efficiency, however, it is important to also analyze the amount of unique data delivered to the Mobile-Hub when comparing routing algorithms.

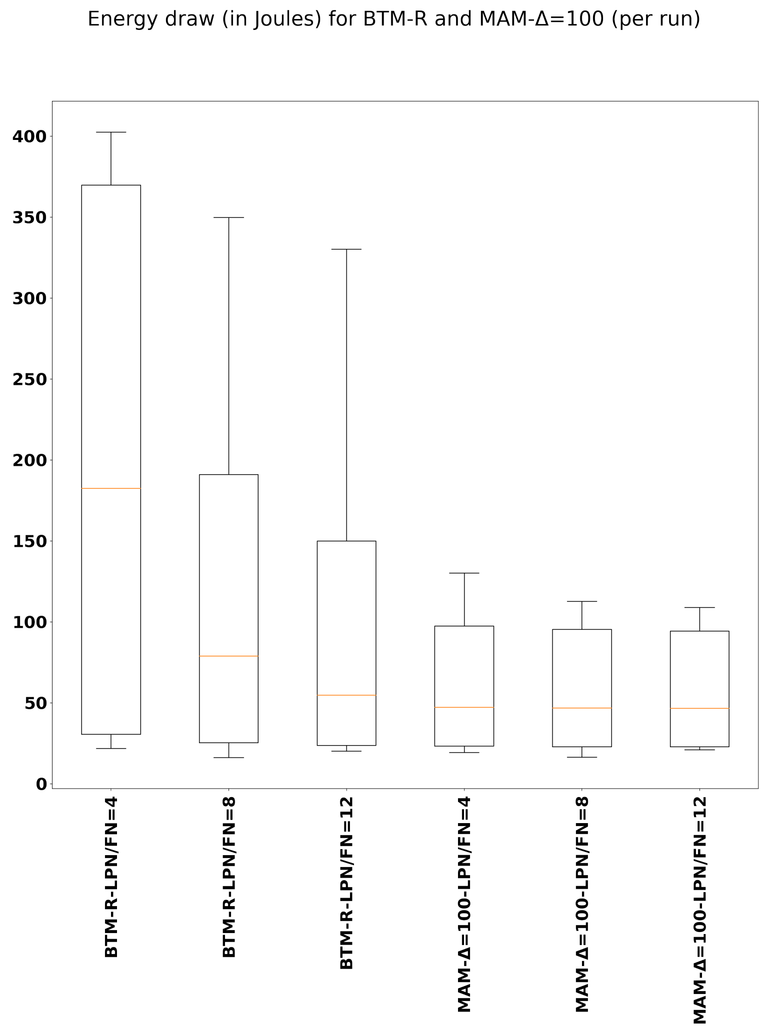


Figure 8.7: Energy draw (in Joules) for BTM-R by LPN/FN ratio - $MAM_{\Delta=100}$ - MAMSET.

8.2.3 Received Packets

Figure 8.8 presents the unique data packets received by the Mobile-Hub (in bytes) across all 11,072 MAMSET simulations, for BTM-R, MAM_0 and MAM_Δ by Δ values. As in MAM50's results (section 8.1.4, MAM0 performed worse in terms of unique data packets received by the Mobile-Hub, with a median of 8002.50 bytes received. Among MAM_Δ, simulations with Δ=50 presented the highest median amount of bytes delivered to the Mobile-Hub (8800 bytes) followed by simulations with Δ=100 (8778 bytes). BTM-R, however, achieved a median value of 18315 bytes delivered to the Mobile-Hub, the highest among other groups, which represents a 2.08 times higher value than MAM_Δ's best group (Δ=500 - 8800 bytes). This result differs significantly from MAM50's delivery rate (described in section 8.1.3), in which MAM_Δ performed up to 173.54% better than BTM-R. Hence, although Δ=100 presented one of the best results among other MAM_Δ Δ values, it does not seem that it achieves a higher amount of unique received packets by the Mobile-Hub in most of the different maps and topologies this work has tested (MAM50-RQ1).

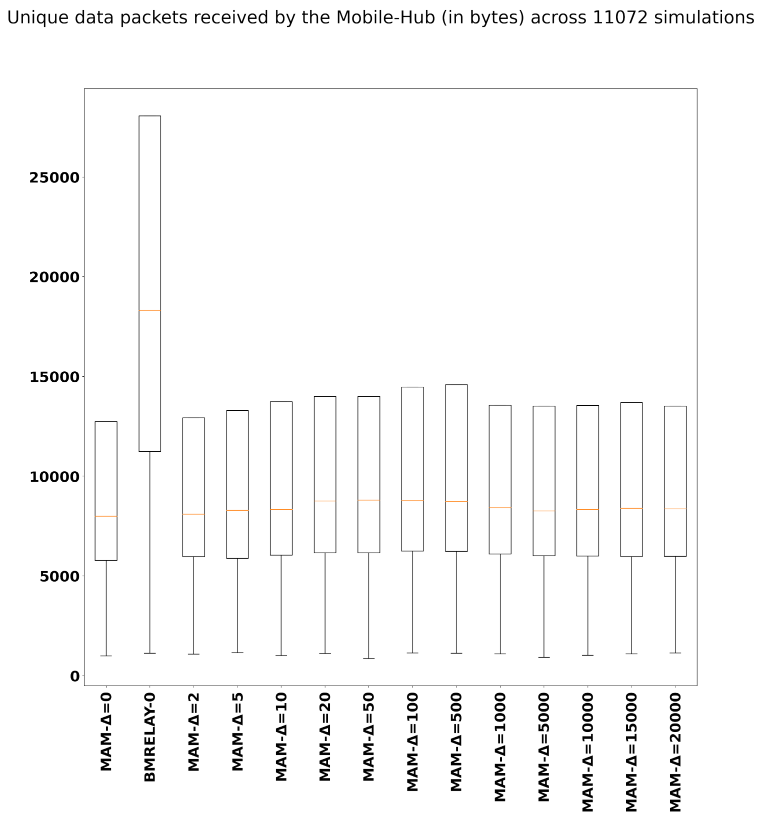


Figure 8.8: Unique data received (in Bytes) comparison between BTM-R, MAM_0 and MAM_Δ - MAMSET.

8.2.4
Energy Efficiency

Figure 8.9 presents the Energy Efficiency in Bytes per Joule (B/J) metric, which is a result of dividing the Unique Data Received (in Bytes) metric by the Energy Draw (in Joules) metric. This metric may indicate the algorithm's energy efficiency, as it considers both the unique data that was delivered as well as how much energy was required for delivering the data. As other MAMSET metrics, Energy Efficiency is presented as a boxplot. The figure compares BTM-R and MAM_{Δ} with Δ=100, each of them divided into LPN/FN ratios (of 4, 8, or 12 Low Power Nodes per Friend Node).

BTM-R's median energy efficiency was of 181.74B/J for LPN/FN ratio=4, 213.66B/J for LPN/FN ratio=8, and 253.18B/J for LPN/FN ratio=12. MAM_{Δ} with Δ=100 presented a median energy efficiency of 240.69B/J for LPN/FN ratio=4 (32.43% greater than BTM-R), 204.65B/J for LPN/FN ratio=8 (4.23% lower than BTM-R), 177.51B/J for LPN/FN ratio=12 (29.89% lower than BTM-R). This indicates that MAM_{Δ} with Δ=100 may be more energy-efficient than BTM-R for LPN/FN ratio=4, and less energy-efficient than BTM-R for LPN/FN ratio=8, and LPN/FN ratio=12. Hence, it seems that in some scenarios, MAM_{Δ} can be more energy-efficient than BTM-R when routing sensor data towards a mobile sink node (MAIN-RQ2).

On BTM-R's energy efficiency results, the median increased as the LPN/FN ratio increased, whereas on the results of MAM_{Δ} with Δ=100 the median decreased as the LPN/FN ratio increased.

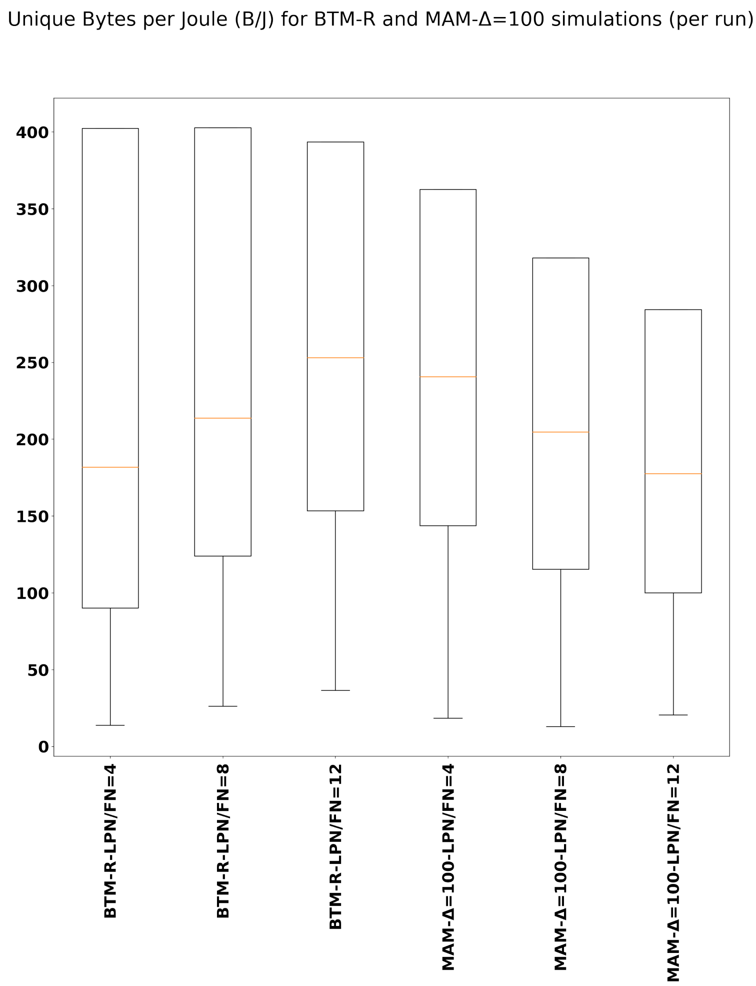


Figure 8.9: Unique data received (in Bytes) comparison between BTM-R and $MAM_{\Delta=100}$ - MAMSET.

8.2.5 Delivery Rate

Figure 8.10 contains the delivery rate (in %) between BTM-R and MAM_Δ with Δ>=20. BTM-R presented a median delivery rate of 5.63% for LPN/FN ratio=4, 15.20% for LPN/FN ratio=8, 17.06% for LPN/FN ratio=12. MAM_Δ with Δ>=20 presented a median delivery rate of 14.70% for LPN/FN ratio=4, 14.77% for LPN/FN ratio=8, 14.30% for LPN/FN ratio=12. This indicates that MAM_Δ with Δ>=20 achieves higher delivery rates for LPN/FN ratio=4 (MAM's median delivery rate was 2.61 times higher than BTM-R's for LPN/FN ratio=4) when compared to BTM-R, and lower delivery rates for LPN/FN ratios of 8 and 12.

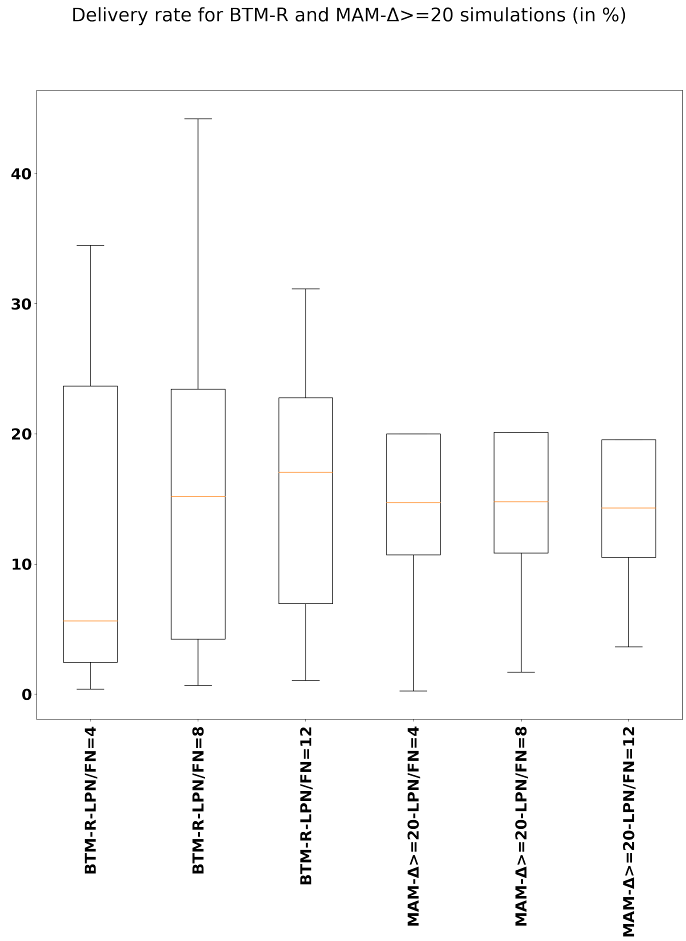


Figure 8.10: Delivery rate (in %) comparison between BTM-R and $MAM_{\Delta>=20}$ - MAMSET.

Figures 8.9 and 8.10 indicated increased energy efficiency and delivery rates for $MAM_{\Delta>=20}$ with the smallest LPN/FN ratio (4 LPNs per FN). A smaller LPN/FN ratio incurs a higher amount of active nodes and increased network activity. The authors hypothesized that increased network activity could be related to increased $MAM_{\Delta>=20}$ efficiency when compared to BTM-

R (this hypothesis will be called *H1* from now on). In order to try to investigate this hypothesis, this work also presents the delivery rate metric using a different slice of MAMSET that aims to select simulations in which there is increased network activity, by picking the smallest tested area parameter (400sqm) and LPN/FN ratio (4 LPN/FN).

Figure 8.11 presents the delivery rate (in %) between BTM-R and MAM_Δ with Δ>=20 across simulation results in which the LPN/FN ratio=4 and the max area=400sqm. The results are divided by the number of nodes (50, 100, or 200 nodes) used in the simulations. It seems reasonable to expect that if *H1* is valid, a higher number of nodes should increase the delivery rate for MAM_Δ with Δ>=20 when compared to BTM-R.

BTM-R presented a median delivery rate of 26.61% for 50 nodes, 3.43% for 100 nodes, 1.94% for 200 nodes. MAM_Δ with Δ>=20 presented a median delivery rate of 22.60% for 50 nodes, 14.40% for 100 nodes, and 8.77% for 200 nodes. This seems to partially corroborate *H1*, as the delivery rate for MAM_Δ with Δ>=20 became significantly higher than BTM-R's as the number of nodes increased. However, MAM_Δ's delivery rate decreased from 14.40% to 8.77% as the number of nodes increased from 100 to 200 nodes. Hence, those results indicate that, in some cases, MAM_Δ can achieve higher delivery rates than BTM-R across different maps and topologies (MAM50-RQ2).

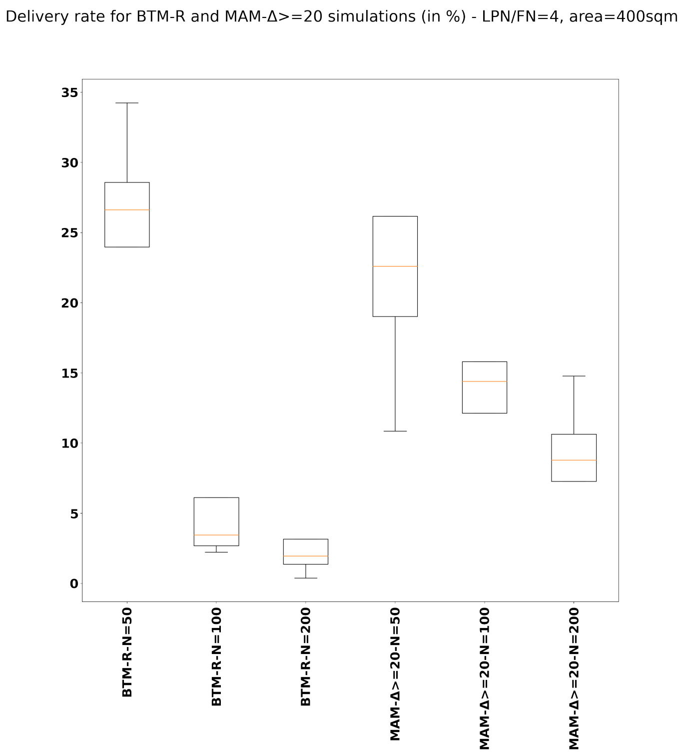


Figure 8.11: Delivery rate for BTM-R and $MAM_{\Delta>=20}$ with LPN/FN ratio = 4 and area = 400sqm - MAMSET.

8.2.6 Analysis and Tradeoffs - MAMSET

MAMSET introduced a much larger amount of information about the MAM_0 and MAM_Δ algorithms, as not only additional simulation parameters affected map and topology generation but also the number of simulations greatly increased (from 24 simulations in MAM50 to more than 11 thousand simulations in MAMSET).

The simulation results changed significantly between MAM50 and MAMSET, exposing more cases in which MAM_Δ performed better and worse than BTM-R. The results indicate that with the correct tuning (Δ parameter) and in certain conditions (smaller areas with a higher amount of active nodes), MAM_Δ may achieve a significantly better performance compared to BTM-R in terms of unique received packets and delivery rate, as well as energy efficiency (when we consider the amount of energy drawn proportionally to the higher unique data packets received, through the energy efficiency in B/J metric).

Assessing whether BTM-R or the modified MAM relay algorithms are suitable for routing data towards a mobile sink node (MAIN-RQ1) depends on the requirements of the application. In some cases, as shown by MAM50's results, it is possible to obtain delivery rates of up to 70% in the tests this work has conducted. In other cases, evaluated through randomized maps and different topologies, delivery rates may be significantly lower but still achieve 25-34% in some cases.

9
Conclusion and future work

This work proposed two alternative approaches to relaying messages in BTMesh networks towards a mobile sink node named Mobile-Hub. The proposed approaches were implemented and compared with the BTMesh standard model on a simulator (OMNET++ INET framework) by executing multiple simulations that collected the following metrics: energy draw, Mobile-Hub data delivery rate, Mobile-Hub amount of data received, and end-to-end delay (time elapsed from the sensor node data being sent to the network until it reaches the Mobile-Hub).

The preliminary results (MAM50), indicated that one of the proposed relay algorithms, MAM_{Δ}, achieved betters results in all of the evaluated metrics when compared to BTMesh's default relay algorithm (BTM-R). Since MAM50 only used a single map and involved varying few simulation parameters, this work also executed additional simulations, with multiple maps and topologies as well as other parameters such as LPN/FN ratio and different Δ values. This other round of simulations (MAMSET) resulted in more than 11 thousand simulations. MAMSET also included an additional metric - Energy Efficiency in Bytes per Joule (B/J) - which was useful to compare BTM-R and MAM_{Δ}.

MAMSET's results indicated that, in some situations, MAM_{Δ} an achieve higher delivery rates than BTM-R across different maps and topologies, as well as increased energy efficiency.

The simulation results discussed in chapter 8 were able to partially cover most of the research questions introduced in chapters 1 and 7. However, the authors believe they were not able to sufficiently engage on MAIN-RQ1 as, to evaluate BTMesh as a viable technology for mobile sink routing, the simulation model would need to be analyzed in terms of fidelity.

Also, as the Bluetooth Mesh specification is relatively new, there are not many published studies of this technology's capabilities neither of its application in the data collection scenarios we evaluate in this work. According to Todtenberg and Kraemer (2019) [62] there are only four reports of a successful establishment of a Bluetooth multi-hop network with more than 30 nodes and only one of them was integrated into a real-world application.

Implementing and conducting field tests using BTM-R as well as MAM_{Δ}

in a microcontroller with BLE radio is currently on the author's future research roadmap, and is made possible by the GrADyS project (partially sponsored by the U.S. Air Force Office of Scientific Research - AFOSR).

Extending the MAM_{Δ} algorithm to handle multiple Mobile-Hubs and heterogeneous data collection by type (e.g., multiple Mobile-Hubs, that subscribe to different data types) should also be an exciting path to explore as a ramification of this work.

In the future, adaptive and opportunistic Mesh routing shall become part of the IoMT middleware ContextNet [63]. This ContextNet Adaptive Mesh Extension (AME) will extend the applicability and reach of the ContextNet approach to supporting connectivity and edge processing for the Internet of Mobile Things (IoMT) in several application fields such as Smart Cities, environmental Monitoring, Precision agriculture, Security, Industry 4.0, and healthcare.

In this approach, we assume that there will be many data-collecting devices, such as smartphones, smartwatches, and drones/UAVs (being used for participatory sensing) and that most smart IoT devices (sensors, actuators, beacons) will probably have only a short-range low-power wireless interface, such as Bluetooth Low Energy (BLE), instead of a low-power Wide-Area connection, such as LoRaWAN, due to the demand to frequently collect and send sensor data. We believe that ContextNet-AME will have many exciting applications in several application fields such as Smart Cities, environmental Monitoring, Precision agriculture, Security, Industry 4.0, and healthcare.

Bibliography

[1] DI FRANCESCO, M.; DAS, S. K.; ANASTASI, G.. **Data Collection in Wireless Sensor Networks with Mobile Elements: A Survey**. ACM Transactions on Sensor Networks, 8(1):7:1–7:31, Aug. 2011.

[2] DJEDOUBOUM, A.; ARI, A.; GUEROUI, A.; MOHAMADOU, A.; ALIOUAT, Z.. **Big data collection in large-scale wireless sensor networks**. Sensors, 18, 12 2018.

[3] RIOS, L. T.; ENDLER, M.; VASCONCELOS, I.; VASCONCELOS, R.; CUNHA, M.; SILVA, F.. **The mobile hub concept: Enabling applications for the internet of mobile things**. In: PERVASIVE COMPUTING AND COMMUNICATION WORKSHOPS - PERCOM WORKSHOPS 2015, p. 123–128, Mar. 2015.

[4] ENDLER, M.; E SILVA, F. S.. **Past, Present and Future of the ContextNet IoMT Middleware**. Open Journal of Internet Of Things (OJIOT), 1(4):7–23, Sept. 2018.

[5] TUNCA, C.; ISIK, S.; DONMEZ, M. Y.; ERSOY, C.. **Distributed mobile sink routing for wireless sensor networks: A survey**. IEEE Communications Surveys and Tutorials, 16(2):877–897, 2014.

[6] PERKINS, C.; BELDING-ROYER, E.; DAS, S.. **Rfc3561: Ad hoc on-demand distance vector (aodv) routing**. ISOC, 2003.

[7] ELRAHIM, A. G. A.. **Eagrp: Energy aware geographic routing protocol for wireless sensor networks**. International Journal of Computer Science and Telecommunications, 5(11), 2014.

[8] HEINZELMAN, W. B.; CHANDRAKASAN, A. P.; BALAKRISHNAN, H.. **An application-specific protocol architecture for wireless microsensor networks**. IEEE Transactions on Wireless Communications, 1(4):660–670, 2002.

[9] SINGH, S. K.; KUMAR, P.; SINGH, J. P.. **A Survey on Successors of LEACH Protocol**. IEEE Access, 5:4298–4328, 2017.

[10] YOUNIS, O.; FAHMY, S.. **Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks**. IEEE Transactions on Mobile Computing, 3(4):366–379, 2004.

[11] BARANIDHARAN, B.; SHANTHI, B.. **A survey on energy efficient protocols for wireless sensor networks**. International Journal of Computer Applications, 11:35–40, 12 2010.

[12] BAERT, M.; ROSSEY, J.; SHAHID, A.; HOEBEKE, J.. **The Bluetooth Mesh Standard: An Overview and Experimental Evaluation**. Sensors, 18(8):2409, 2018.

[13] FRIEDMAN, R.; KOGAN, A.; KRIVOLAPOV, Y.. **On power and throughput tradeoffs of wifi and bluetooth in smartphones**. IEEE Transactions on Mobile Computing, 12(7):1363–1376, 2013.

[14] GIACOMINI, E.; D'ALTERIO, F.; LACAVA, A.; CUOMO, F.. **Blues: A self-organizing ble mesh-network paradigm for iot environments**. In: 2020 IEEE 21ST INTERNATIONAL SYMPOSIUM ON "A WORLD OF WIRELESS, MOBILE AND MULTIMEDIA NETWORKS" - WOWMOM 2020, p. 409–414, 2020.

[15] BLUETOOTH SIG, I.. **Bluetooth core specification version 5.1 feature overview**. URL: *https://www.bluetooth.com/bluetooth-resources/bluetooth-core-specification-v5-1-feature-overview/*, Jan. 2019.

[16] BLUETOOTH SIG, I.. **Bluetooth core specification version 5.0**. URL: *https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=421043*, Jan. 2016.

[17] MADHAVAPEDDY, A.; TSE, A.. **A study of bluetooth propagation using accurate indoor location mapping**. In: 2005 INTERNATIONAL CONFERENCE ON UBIQUITOUS COMPUTING - UBICOMP 2005, p. 105–122. Springer, 2005.

[18] BLUETOOTH SIG, I.. **Bluetooth core specification version 4.0**. URL: *https://www.bluetooth.com/specifications/specs/core-specification-4-0*, June 2010.

[19] GOMEZ, C.; OLLER, J.; PARADELLS, J.. **Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-Power Wireless Technology**. Sensors, 12(9):11734–11753, 2012.

[20] BLUETOOTH SIG. **Mesh profile specification 1.01**. URL: *https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=457092*, 2019. Access date: 2019-08-15.

[21] BLUETOOTH SIG. **Provisioning a bluetooth mesh network part 1**. URL: *https://www.bluetooth.com/blog/provisioning-a-bluetooth-mesh-network-part-1/*, 2017. Access date: 2019-08-15.

[22] ASLAN, Y. E.; KORPEOGLU, I.; ULUSOY, Ö.. **A framework for use of wireless sensor networks in forest fire detection and monitoring**. Computers, Environment and Urban Systems, 36(6):614–625, 2012.

[23] GRAMMALIDIS, N.; ÇETIN, E.; DIMITROPOULOS, K.; TSALAKANIDOU, F.; KOSE, K.; GUNAY, O.; GOUVERNEUR, B.; TORRI, D.; KURUOGLU, E.; TOZZI, S.; BENAZZA, A.; CHAABANE, F.; KOSUCU, B.; ERSOY, C.. **A multi-sensor network for the protection of cultural heritage**. European Signal Processing Conference - EUSIPCO 2011, 1(Eusipco):889–893, 2011.

[24] YU, L.; WANG, N.; MENG, X.. **Real-time forest fire detection with wireless sensor networks**. Proceedings - 2005 International Conference on Wireless Communications, Networking and Mobile Computing - WCNM 2005, 2:1214–1217, 2005.

[25] BRINI, M.; MARMO, L.. **Wireless sensor networks for early fire detection**. Chemical Engineering Transactions, 24(August 2003):1153–1158, 2011.

[26] SON, B.; HER, Y.-S.; KIM, J.. **A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains**. International Journal of Computer Science and Network Security (IJCSNS), 6(9):124–130, 2006.

[27] SUN, E.; ZHANG, X.; LI, Z.. **The internet of things (IOT) and cloud computing (CC) based tailings dam monitoring and pre-alarm system in mines**. Safety Science, 50(4):811–815, 2012.

[28] RAJESH, P.; MANJUNATHAN, N.; SURESH, A.. **Unmanned dam monitoring system using wireless sensor networks**. International Journal of Innovative Technology and Exploring Engineering, 8(5):343–347, 2019.

[29] XUE, G.. **Relay Node Placement in Wireless Sensor Networks.** IEEE Transactions on Computers, 56(1):134–138, 2007.

[30] MUHENDRA, R.; RINALDI, A.; BUDIMAN, M.; KHAIRURRIJAL. **Development of wifi mesh infrastructure for internet of things applications.** Procedia Engineering, 170:332–337, 2017. 2016 Engineering Physics International Conference – EPIC 2016.

[31] NAZIR, B.; HASBULLAH, H.. **Mobile sink based routing protocol (msrp) for prolonging network lifetime in clustered wireless sensor network.** In: 2010 INTERNATIONAL CONFERENCE ON COMPUTER APPLICATIONS AND INDUSTRIAL ELECTRONICS - ICCAIE 2010, p. 624–629, 2010.

[32] ZORBAS, D.; RAVENEAU, P.; GHAMRI-DOUDANE, Y.; DOULIGERIS, C.. **The charger positioning problem in clustered rf-power harvesting wireless sensor networks.** Ad Hoc Networks, 78:42–53, 2018.

[33] DEMENTYEV, A.; HODGES, S.; TAYLOR, S.; SMITH, J.. **Power consumption analysis of bluetooth low energy, zigbee and ant sensor nodes in a cyclic sleep scenario.** In: 2013 IEEE INTERNATIONAL WIRELESS SYMPOSIUM - IWS 2013, p. 1–4, 2013.

[34] AKYILDIZ, I. F.; WANG, X.; WANG, W.. **Wireless mesh networks: a survey.** Computer Networks, 47(4):445–487, 2005.

[35] LANDSIEDEL, O.; WEHRLE, K.; GOTZ, S.. **Accurate prediction of power consumption in sensor networks.** In: THE SECOND IEEE WORKSHOP ON EMBEDDED NETWORKED SENSORS, 2005. EMNETS-II, p. 37–44, 2005.

[36] BADAWY, G. H.; SAYEGH, A. A.; TODD, T. D.. **Fair flow control in solar powered wlan mesh networks.** In: 2009 IEEE WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE - WCNC 2009, p. 1–6, Apr. 2009.

[37] KOPJÁK, J.; SEBESTYÉN, G.. **Comparison of data collecting methods in wireless mesh sensor networks.** In: 2018 IEEE 16TH WORLD SYMPOSIUM ON APPLIED MACHINE INTELLIGENCE AND INFORMATICS (SAMI), p. 155–160, Feb. 2018.

[38] ADI, H.; KOM, S.; SAKTI, P.; KOM, S.; KOM, M.; AMRON, S.. **Data gathering in rural area using publish/subscribe over wireless**

mesh network. In: 2017 INTERNATIONAL CONFERENCE ON SUSTAINABLE INFORMATION ENGINEERING AND TECHNOLOGY - SIET 2017, p. 241–245, 2017.

[39] LUO, J.; J., P.; M., P.; GROSSGLAUSER, M.; J.-P., H.. **MobiRoute: Routing Towards a Mobile Sink for Improving Lifetime in Sensor Networks.** In: DISTRIBUTED COMPUTING IN SENSOR SYSTEMS - DCOSS 2006, p. 480–497, 2006.

[40] TODTENBERG, N.; KRAEMER, R.. **Survey on Bluetooth multi-hop networks.** Ad Hoc Networks, 93, 2019.

[41] CHING LAW; KAI-YEUNG SIU. **A bluetooth scatternet formation algorithm.** In: 2001 IEEE GLOBAL TELECOMMUNICATIONS CONFERENCE (CAT. NO.01CH37270) - GLOBECOM 2001, volumen 5, p. 2864–2869 vol.5, 2001.

[42] SALONIDIS, T.; BHAGWAT, P.; TASSIULAS, L.; LAMAIRE, R.. **Distributed topology construction of bluetooth personal area networks.** In: PROCEEDINGS IEEE INFOCOM 2001. CONFERENCE ON COMPUTER COMMUNICATIONS. TWENTIETH ANNUAL JOINT CONFERENCE OF THE IEEE COMPUTER AND COMMUNICATIONS SOCIETY (CAT. NO.01CH37213) - COMSOC 2001, volumen 3, p. 1577–1586 vol.3, 2001.

[43] PETRIOLI, C.; BASAGNI, S.; CHLAMTAC, M.. **Configuring bluestars: multihop scatternet formation for bluetooth networks.** IEEE Transactions on Computers, 52(6):779–790, 2003.

[44] PETRIOLI, C.; BASAGNI, S.; CHLAMTAC, I.. **Bluemesh: degree-constrained multi-hop scatternet formation for bluetooth networks.** mobile Networks and Applications, 9(1):33–47, 2004.

[45] LI, X. .; STOJMENOVIC, I.; WANG, Y.. **Partial delaunay triangulation and degree limited localized bluetooth scatternet formation.** IEEE Transactions on Parallel and Distributed Systems, 15(4):350–361, 2004.

[46] LEON, E. D.; NABI, M.. **An experimental performance evaluation of bluetooth mesh technology for monitoring applications.** In: 2020 IEEE WIRELESS COMMUNICATIONS AND NETWORKING CONFERENCE - WCNC 2020, p. 1–6, 2020.

[47] HANSEN, E. A. J.; NIELSEN, M. H.; SERUP, D. E.; WILLIAMS, R. J.; MADSEN, T. K.; ABILDGREN, R.. **On relay selection approaches in**

bluetooth mesh networks. In: 10TH INTERNATIONAL CONGRESS ON ULTRA MODERN TELECOMMUNICATIONS AND CONTROL SYSTEMS AND WORKSHOPS - ICUMT 2018, p. 1–5, Nov. 2018.

[48] MAGER, F.; ZIMMERLING, M.. **Mixer**. PCI-Paint and Coatings Industry, 2016(AUGUST), 2016.

[49] MA, X.; ZHANG, P.; LI, X.; TANG, W.; WEI, J.; THEEL, O.. **DeCoT: A dependable concurrent transmission-based protocol for wireless sensor networks**. IEEE Access, 6:73130–73146, 2018.

[50] FERRARI, F.; ZIMMERLING, M.; THIELE, L.; SAUKH, O.. **Efficient network flooding and time synchronization with Glossy**. Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks - IPSN 2011, p. 73–84, 2011.

[51] YUAN, D.; HOLLICK, M.. **Ripple: High-throughput, reliable and energy-efficient network flooding in wireless sensor networks**. Proceedings of the IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks - WoWMoM 2015, 2015.

[52] CHENG, L.; NIU, J.; LUO, C.; SHU, L.; KONG, L.; ZHAO, Z.; GU, Y.. **Towards minimum-delay and energy-efficient flooding in low-duty-cycle wireless sensor networks**. Computer Networks, 134:66–77, 2018.

[53] MA, X.; ZHANG, P.; LIU, Y.; BOANO, C. A.; KIM, H. S.; WEI, J.; HUANG, J.. **Harmony: Saving Concurrent Transmissions from Harsh RF Interference**. Proceedings - IEEE International Conference on Computer Communications - INFOCOM 2020, 2020-July:1024–1033, 2020.

[54] BISDIKIAN, C.. **An overview of the Bluetooth wireless technology**. IEEE Communications Magazine, 39(12):86–94, 2001.

[55] DARROUDI, S. M.; GOMEZ, C.. **Bluetooth low energy mesh networks: A survey**. Sensors (Switzerland), 17(7), 2017.

[56] BLUETOOTH SIG, I.. **An intro to bluetooth mesh part 2**. URL: *https://www.bluetooth.com/blog/an-intro-to-bluetooth-mesh-part2/*, Aug. 2018.

[57] WINTER, T.; THUBERT, P.; BRANDT, A.; HUI, J. W.; KELSEY, R.; LEVIS, P.; PISTER, K.; STRUIK, R.; VASSEUR, J.-P.; ALEXANDER, R. K.;

OTHERS. **Rpl: Ipv6 routing protocol for low-power and lossy networks.** rfc, 6550:1–157, 2012.

[58] KAJDOCSI, L.; DÖRÖMBÖZI, A.; KOVÁCS, J.. **Development of bluetooth mesh core stack using omnet++**. In: 2019 IEEE 17TH INTERNATIONAL SYMPOSIUM ON INTELLIGENT SYSTEMS AND INFORMATICS (SISY), p. 23–28, 2019.

[59] GISLASON, D.. **Zigbee Wireless Networking**. Newnes, USA, pap/onl edition, 2008.

[60] SIEKKINEN, M.; HIIENKARI, M.; NURMINEN, J. K.; NIEMINEN, J.. **How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4**. 2012 IEEE Wireless Communications and Networking Conference Workshops - WCNCW 2012, p. 232–237, 2012.

[61] VARGA, A.. **Circlemobility - inet framework**. URL: *https://doc.omnetpp.org/inet/api-current/neddoc/inet.mobility.single.CircleMobility.html*, Jan. 2020.

[62] TODTENBERG, N.; KRAEMER, R.. **A survey on bluetooth multi-hop networks**. Ad Hoc Networks, 93:101922, 2019.

[63] ENDLER, M.; BAPTISTA, G.; SILVA, L. D.; VASCONCELOS, R.; MALCHER, M.; PANTOJA, V.; PINHEIRO, V.; VITERBO, J.. **Contextnet: Context reasoning and sharing middleware for large-scale pervasive collaboration and social networking**. In: PROCEEDINGS OF THE WORKSHOP ON POSTERS AND DEMOS TRACK, PDT '11, p. 2:1–2:2, New York, NY, USA, 2011. ACM.