Ad Hoc Networks 130 (2022) 102809

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Exploring data collection on Bluetooth Mesh networks

Marcelo Paulon J.V. *, Bruno José Olivieri de Souza, Markus Endler

Pontifícia Universidade Católica do Rio de Janeiro, Rua Marquês de São Vicente, Rio de Janeiro, RJ, Brazil

A R T I C L E I N F O

Keywords:
Wireless sensor networks
Distributed data collection
Mesh networks
Bluetooth mesh

A B S T R A C T

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%.

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 [1] and by [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.

* Corresponding author.
E-mail addresses: mvasconcelos@inf.puc-rio.br (M. Paulon J.V.), bolivieri@inf.puc-rio.br (B.J. Olivieri de Souza), endler@inf.puc-rio.br (M. Endler).

https://doi.org/10.1016/j.adhoc.2022.102809
Received 15 June 2021; Received in revised form 12 January 2022; Accepted 15 February 2022
Available online 3 March 2022

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. [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. The present work is a follow-up of [17], in which we evaluated MAM considering 14 m/s as the collector speed, and did not vary parameters such as the amount of mesh nodes, their mutual reachability, and the general network layout/topology.

In the next section, we present some definitions of concepts used along this work. In Section 3 we present the simulated scenarios considered for tree/forest monitoring as well as other possible applications. Section 4 contains related work in data collection, mesh wireless sensor networks, and BTMesh networking. Then, Section 5 covers BTMesh and its characteristics while Section 6 describes the proposed data collection solution, explaining BTM-R, MAM_0, and MAM_Δ relay algorithms. In Section 7 we then describe the simulation model and evaluation metrics, as well as the software engineering work that was required to perform the simulations. Wrapping up, Section 8 presents and discusses the simulation results and Section 9 concludes by pointing to 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 Section 3.

According to [18], Bluetooth is a wireless protocol for short-range communications that operates in the license-free 2.4 GHz spectrum. The Bluetooth 4.0 protocol specification [19] introduced Bluetooth Low Energy (i.e., BLE), designed as a low-power solution for application control and monitoring [20]. 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 [21], an unprovisioned device is a device that is not part of a Bluetooth Mesh network [22]. 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

[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.

of a forest [23]. 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 [23–27] 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 [28]. Tailings dams monitoring also appears to be a possible application for sensor network data collection [29].

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 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. Section 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 [30], 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.

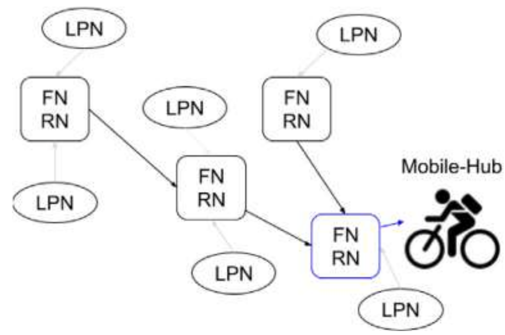


Fig. 1. t1 — Mobile-Hub connected to a sensor network.

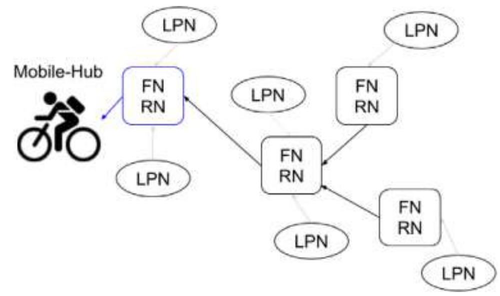


Fig. 2. t2 — Mobile-Hub connected to a sensor network.

Figs. 1 and 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 [16,31], 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.

4. Related work

4.1. Data collection and Mesh WSNs

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

[3] https://www.reuters.com/article/us-vale-sa-bhp-billiton-dam-idUSKCN0SU38I20151106.
[4] https://www.bbc.com/news/business-47432134.

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 [32]. 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 [31] that may provide up to 300 m signal range per device, or short-range communications (technologies such as Zigbee or Bluetooth) that consume less power [33,34]. 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 [35].

4.2. Energy consumption in WSNs

In the context of sensor data collection, radio communications are responsible for most of the energy draw in microcontrollers [36] 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. [37] 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.

With the goal of extending the lifetime of WSNs, [38] 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.

4.3. Mobility in WSNs

[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 Section 6) would be defined as having an asynchronous mobility-independent discovery approach, and a proxy-based routing approach.

[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.

4.4. Bluetooth

Bluetooth is a technology that can be used to form Wireless Mesh Sensor Networks, as described by [39] 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 [40–44]. [41] 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 [12,45,46]. [45] analyzes BTMesh in a real-world environment and reports limitations for message delivery as a result.

Hansen et al. [46] 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 Section 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 [47–52]. The *Harmony* algorithm [52] 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 [53] is a wireless technology already widespread across devices, from home automation to personal gadgets such as wireless headphones. Also, for BLE [54] 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. Data collection for forest/urban hills sensor data could alternatively use ZigBee (802.15.4) or even LPWANs such as LoRa and Sigfox for uplink communication from a sensor node to a base station/data collector. Those technologies provide up to 15 km range and can be relatively cheap to implement. However, the lower data transfer rates (when compared to BLE) makes using LPWANs not suitable for all applications; for instance, while BLE has a physical rate starting at 125 Kb/s, LoRA has a rate of 27 Kb/s and Sigfox 100 b/s. [55] is an in-depth study that compares trade-offs like different data transfer rates and power consumption among IoT technologies such as BLE, 802.15.4, LoRa and Sigfox; it reported that BLE presented higher lifetime and was the best technology for most medium and high data rate scenarios that they tested.

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[5] and Mesh Model[6] 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 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 do not 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 Section 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 Section 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 s) 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.

[5] Mesh Profile Bluetooth® Specification, Bluetooth Technology Website. 2017-07-13.

[6] Mesh Model Bluetooth® Specification, Bluetooth Technology Website. 2017-07-13.

For the latter strategy, the implementations compute incoming packet signatures and check them against an LRU cache. If they are present, they will not be relayed. If they are 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.

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
```

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 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. This parameter can be hard-coded in a real world implementation scenario with a fixed value (e.g. one of the values this work has evaluated in Section 8).

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.

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
if (isDiscoveryMessage(messageBody) == false) then
    if (bestNodeAddress != NULL) then
        hops ← messageHops + 1
        sendMessage(bestNodeAddress, messageBody, hops)
    end if
    return
end if
bool expired ← NOW() > expiry
if (expired == true or messageHops < bestNodeHops) then
    bestNodeAddress ← senderAddress
    bestNodeHops ← messageHops
    expiry ← NOW() + Δ
end if
bluetoothMeshRelay(senderAddress, messageHops, messageBody)
```

7. Simulation

The simulations rely on OMNET++ v5.6.1 and the INET framework. OMNET++[7] is a cross-platform simulator library and framework for discrete events, whereas the INET[8] 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

[7] https://omnetpp.org.
[8] https://inet.omnetpp.org.

Table 1
INET simulation parameters.

Property name	Value
*.radioMedium.backgroundNoise.power	−110 dBm
**.radio.transmitter.power	0.275 mW
**.energyConsumer.typename	"SensorStateBasedEpEnergyConsumer"
**.energyConsumer.offPowerConsumption	0 mW
**.energyConsumer.sleepPowerConsumption	1 mW
**.energyConsumer.switchingPowerConsumption	1 mW
**.energyConsumer.receiverIdlePowerConsumption	2 mW
**.energyConsumer.receiverBusyPowerConsumption	5 mW
**.energyConsumer.receiverReceivingPowerConsumption	10 mW
**.energyConsumer.transmitterIdlePowerConsumption	2 mW
**.energyConsumer.transmitterTransmittingPowerConsumption	100 mW

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. [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. [60] compare ZigBee and BLE 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 250 kbit/s while BLE's is 1 Mbit/s.

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

(a) maximum transmission range is configured for 100 m
(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 nor change any of the BTMesh packet structure for any of the algorithms we propose.

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 Section 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 (u) divided by the total number of sent data packets (s), multiplied by 100:

$$\frac{u}{s} \times 100$$

- Mobile-Hub received packets (bytes): the amount in bytes of packets collected by the Mobile-Hub (a). The charts distinguish between unique (u) and repeated (r) data.
- Energy Draw (Joules): the amount of energy that was drawn (e_i represents the amount of energy consumed by each node i), by all of the network nodes:

$$\sum e_i$$

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 10 ms 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]. Some radio propagation aspects of wireless and mobile networks, such as how to choose and position the antennae at the Mobile-Hub and the sensor nodes of the mesh, and how to model the RF interference in different regions/environments of the mesh (i.e. according to the node density), have not been dealt with by our work, but are an exciting path to explore as future work. Table 1 contains the parameters used for INET's radio and energy consumption model.

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 m was used.

Table 2
MAM result comparison (percentage, Energy Draw = small is better, other attributes = big is better).

Algorithm	2 m/s			6 m/s			14 m/s		
	Delivery Rate (%)	Unique data received (bytes)	Energy Draw (Joules)	Delivery Rate (%)	Unique data received (bytes)	Energy Draw (Joules)	Delivery Rate (%)	Unique data received (bytes)	Energy Draw (Joules)
BTM-R	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
MAM-0	−43.28%	−26.38%	**−12.66%**	−46.90%	−34.80%	**−10.47%**	−40.39%	−25.36%	−16.87%
Δ = 5.0	−24.74%	2.62%	−3.14%	−37.64%	−20.39%	−4.39%	−28.50%	−13.04%	**−17.07%**
Δ = 10.0	13.87%	51.31%	−4.53%	−9.28%	16.74%	−0.17%	−0.92%	21.74%	−13.29%
Δ = 20.0	89.83%	153.00%	2.21%	26.78%	54.81%	8.67%	32.60%	61.23%	−8.63%
Δ = 50.0	110.64%	169.38%	2.47%	46.11%	77.97%	10.28%	43.87%	76.45%	−4.20%
Δ = 100.0	111.82%	**173.54%**	4.63%	46.57%	80.11%	10.90%	50.43%	75.97%	−6.61%
Δ = 500.0	**115.71%**	172.54%	0.53%	**47.57%**	**80.49%**	10.26%	**58.22%**	**82.00%**	−5.45%

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 s, 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 Fig. 3. The following variations values were tested:

- 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): 5, 10, 20, 50, 100, 500

The Mobile-Hub's circular trajectory was across a 400 m radius, at a constant speed of 14 m/s (equivalent to a quadcopter), and covered only a subset of the network nodes. The Mobile-Hub connected to 10 Relay Nodes (20% of all network nodes).

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 $\Delta \geq 20$ 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 Section 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, 10 000, 15 000, 20 000
- Area size limit (in square meters): 400 × 400, 800 × 800, 1000 × 1000
- Node amount: 50, 100, 200
- LPN/FN ratio: 4, 8, 12

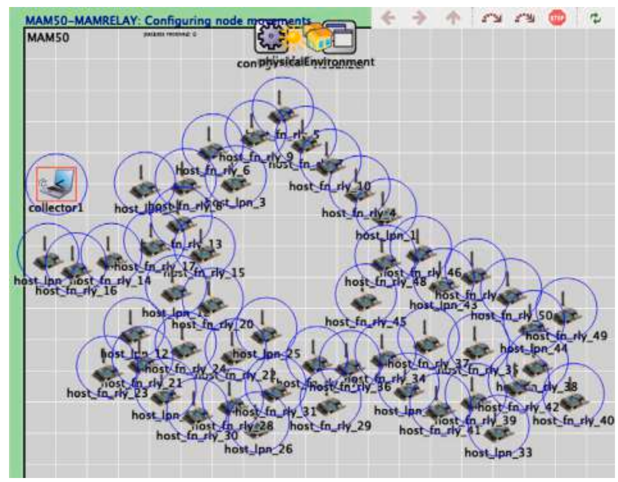


Fig. 3. MAM50 map plotted in the OMNET++ IDE.

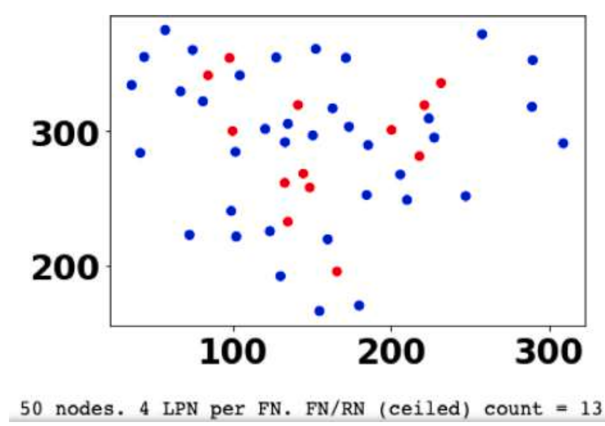


Fig. 4. Random map with 50 nodes spread across an area of 105,000 sqm. Points in red represent Friend+Relay nodes and in blue Low-Power Nodes.

Nodes were randomly placed respecting a minimum distance of 10 m and a maximum distance of 100 m, 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 Section 3, such as tailings dam monitoring — where, in an emergency, sensors could be deployed by being dropped from a plane. Fig. 4 contains one of the randomly generated maps, that contains nodes spread across a 105,000 sqm. Fig. 5 is the same map but as an overlay in a satellite image of a high-risk tailings dam located in Minas Gerais (MG) — Brazil.

Fig. 5. 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).[9]

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.[10]

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 28.8 GB of scalar data.

This work presents and analyzes 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,[11] Pandas[12] and PostgreSQL[13] were crucial for enabling this research to happen in a timely manner.

Table 2 is a percentage comparison between BTM-R and MAM algorithms, and Table 3 provides absolute values for each metric, per run.

[9] "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".

[10] "https://github.com/marcelopaulon/PUC-Rio-MSCDIS-MonitoredArboretumMesh".

[11] Jupyter Project — https://jupyter.org.

[12] Pandas — https://pandas.pydata.org.

[13] PostgreSQL — https://www.postgresql.org/.

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 analyzes the results and apparent trade-offs from MAM50's results.

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

Fig. 6 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 chart shows the results, presented as box plots, 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.

BTM-R's median was of 42 ms while MAM_0's median was of 45 ms. However, all of MAM_Δ's medians were lower than BTM-R's, with the best one being 30 ms. Those results indicate that MAM_0 performs worse in terms of end-to-end delay when compared to BTM-R, and that the MAM_Δ algorithm performed better than BTM-R and MAM_0.

8.1.2. Energy draw

Fig. 7 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.

The results (Fig. 7) 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 the tested scenarios was with the MAM_Δ algorithm with Δ = 5, 17.07% less than BTM-R. For MAM_Δ with Δ = 50, most energy was consumed among the alternative algorithms, 4.2% less than BTM-R. Those results indicate that all the proposed alternatives consumed less energy than BTM-R; however, the energy consumption varied according to the algorithm's Δ parameter.

8.1.3. Delivery rate

Fig. 8 presents 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.

The results (Fig. 8) 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 tested scenarios. For MAM_Δ with Δ = 500, the delivery rate was 43.18%, the highest among the tested scenarios, 58.22% greater than BTM-R. Those results indicate that, for the tested scenarios, the proposed alternatives can also achieve higher and lower delivery rates when compared to BTM-R, depending on how the alternative algorithms are parameterized.

Table 3
MAM results (Delivery Rate, Unique Data Received = big is better, Energy Draw = small is better).

Algorithm	2 m/s				6 m/s				14 m/s			
	Packets Sent	Delivery Rate (%)	Unique data received (bytes)	Energy Draw (Joules)	Packets Sent	Delivery Rate (%)	Unique data received (bytes)	Energy Draw (Joules)	Packets Sent	Delivery Rate (%)	Unique data received (bytes)	Energy Draw (Joules)
BTM-R	35 076	8.53	**32 912**	104.30	33 389	3.57	**13 112**	104.30	32 291	6.81	**24 189**	104.30
MAM-0	5765	24.13	15 301	24.74	2451	15.46	4169	22.11	4263	23.55	11 044	23.70
Δ = 5.0	5848	24.47	15 741	25.16	2238	17.34	4268	**21.68**	4035	24.09	10 692	**23.20**
Δ = 10.0	5840	24.64	15 829	25.25	2331	17.76	4554	22.09	4041	23.56	10 472	23.29
Δ = 20.0	5828	25.55	16 379	25.13	2516	16.06	4444	22.23	4230	24.42	11 363	23.44
Δ = 50.0	5676	24.77	15 466	**24.71**	2524	16.72	4642	22.51	4442	**25.91**	12 661	23.69
Δ = 100.0	5687	26.34	16 478	24.99	2311	**18.35**	4664	21.97	4108	23.49	10 615	23.22
Δ = 500.0	5556	**26.98**	16 489	25.04	2588	17.12	4873	22.34	4245	24.03	11 220	23.49

8.1.4. Received packets

Fig. 9 presents 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.

The results (Fig. 9) 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 other tested scenarios. 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 $\Delta = 500$, the amount of unique data received was 16.57k bytes, the highest among the simulations, 82% greater than BTM-R. Those results indicate that 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.

8.1.5. Analysis and tradeoffs — MAM50

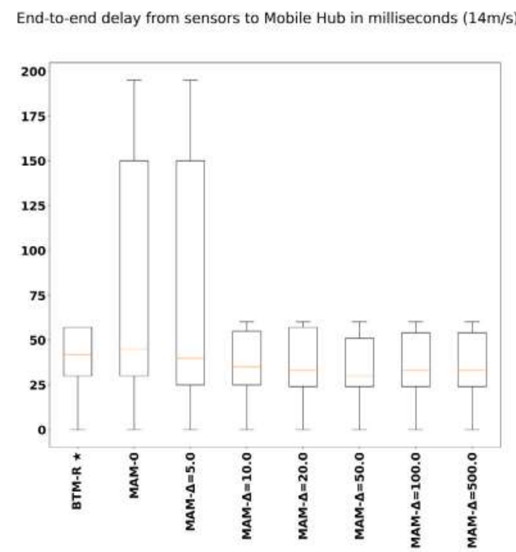
Compared to BTM-R, MAM_0 consumed less energy in the tested scenarios (−16.87%) and lower end-to-end delay in most cases. However, delivery rate (Fig. 8) was lower compared to BTM-R (−40.39%). Also, the received packets in bytes values were lower than BTM-R's (−25.36%). This indicates that, overall, 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 in the tested scenarios (−28.50%). The received packets in bytes were 13.04% lower than BTM-R. The energy draw was lower than BTM-R's in the tested scenarios (−17.07%). This indicates that, in most cases, BTM-R outperforms MAM_Δ with Δ = 5.

For Δ = 10, the delivery rate was lower than BTM-R's (−0.92%). The received packets in bytes were higher than BTM-R in the tested scenarios (+21.74%). The energy draw was lower than BTM-R's in the tested scenarios (−13.29%). This indicates that MAM_Δ with Δ = 10 outperforms BTM-R in terms of amount of data collected and energy draw, however, performs worse regarding delivery rate.

For higher Δ values (≥20.0), MAM_Δ presented delivery rates that were higher than BTM-R in the tested scenarios. The highest Δ value that was simulated (Δ = 500) presented the highest delivery rates: 43.18% (a 58.22% increase compared to BTM-R's rate). In all tested cases, the energy draw was lower compared to BTM-R. The scenario that consumed most energy (Δ = 50) represents only a 4.20% energy draw decrease compared to BTM-R, with a 43.87% delivery rate increase, and a 76.45% received packets in bytes increase. However, in some cases in which the energy draw was lower, the delivery rate and amount of unique received packets in bytes was also lower (MAM_0 and $MAM_{\Delta=5}$).

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).



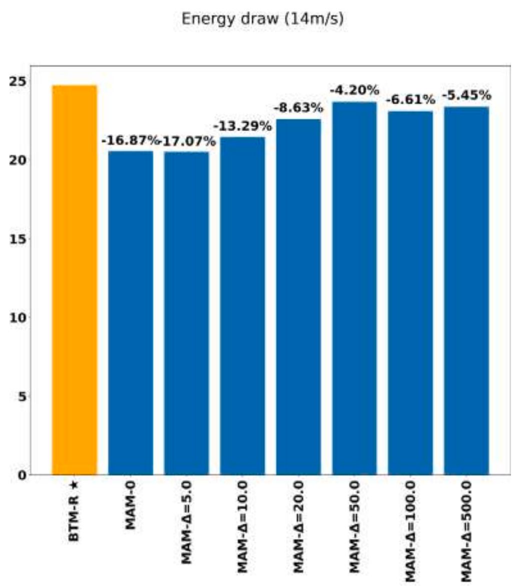
(a) $14m/s$

Fig. 6. Average delay comparison between BTM-R and MAM relay with least-hop route — MAM50.

8.2. MAMSET — Simulations with randomly generated maps

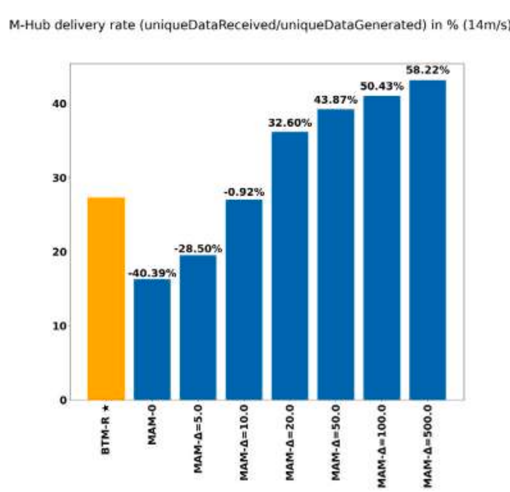
As previously mentioned in Section 7, two additional research questions emerged after obtaining and analyzing MAM50's results:

- 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 $\Delta \geq 20$ values when using different maps and topologies?**



(a) $14m/s$

Fig. 7. Energy Draw (Joules) — MAM50.



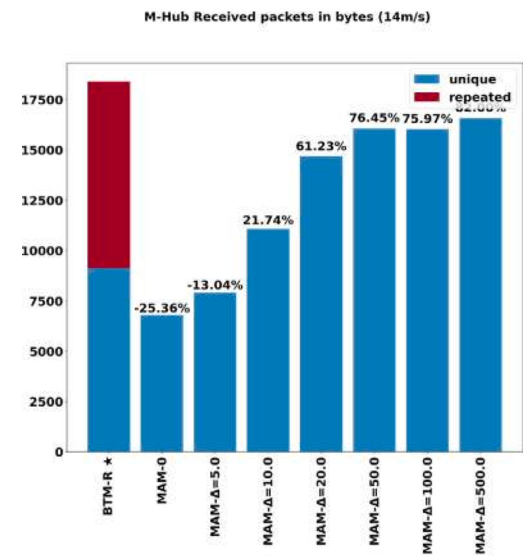
(a) $14m/s$

Fig. 8. Delivery Rate (%) — MAM50.

(given that in MAM50 the delivery rate increased when compared to BTM-R's for all tested $\Delta \geq 20$ 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.



(a) $14m/s$

Fig. 9. Packets received comparison between BTM-R and MAM_{Δ} with least-hop route — MAM50.

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 analyzes the results and apparent trade-offs from MAMSET's results.

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

Fig. 10 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 $\Delta = 100$ (i.e., $MAM_{\Delta=100}$).

The results show a median of 30.11 ms for BTM-R, which is lower than the median of simulations with $MAM_{\Delta=100}$ (30.34 ms). 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.

8.2.2. Energy draw

Fig. 11 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 (182.31 J). As expected, increasing the LPN/FN ratio decreased the Energy Draw of the simulation. BTM-R with presented a median Energy Draw of 78.93 J for a LPN/FN ratio of 8, and of 54.75 J for a LPN/FN ratio of 12. MAM presented a lower median than BTM-R for all LPN/FN ratios (46.73 J for LPN/FN = 4, 46.74 J for LPN/FN = 8, 46.74 J 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.

Fig. 12 shows the Energy Draw (in Joules) across BTM-R and MAM, however, for MAM_{Δ} with $\Delta = 100$ 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 (47.15 J for LPN/FN = 4, 46.78 J for

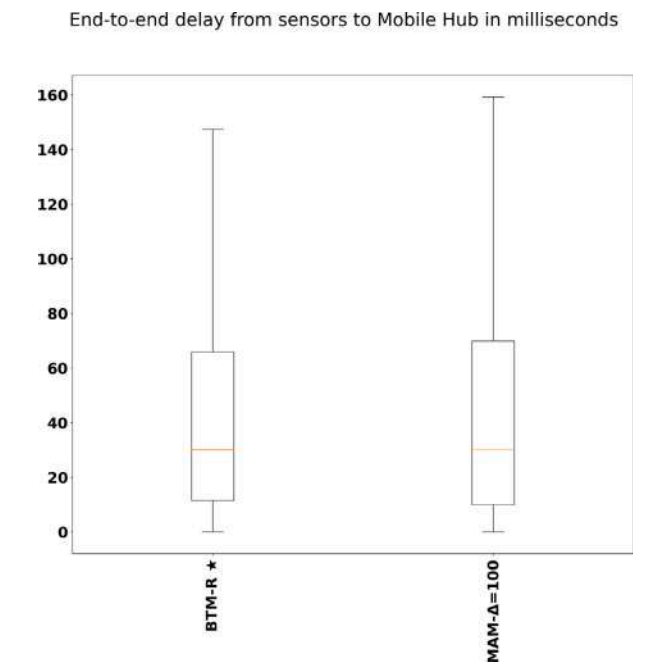


Fig. 10. End-to-end delay from sensors to Mobile-Hub comparison between BTM-R and $MAM_{\Delta=100}$ — MAMSET.

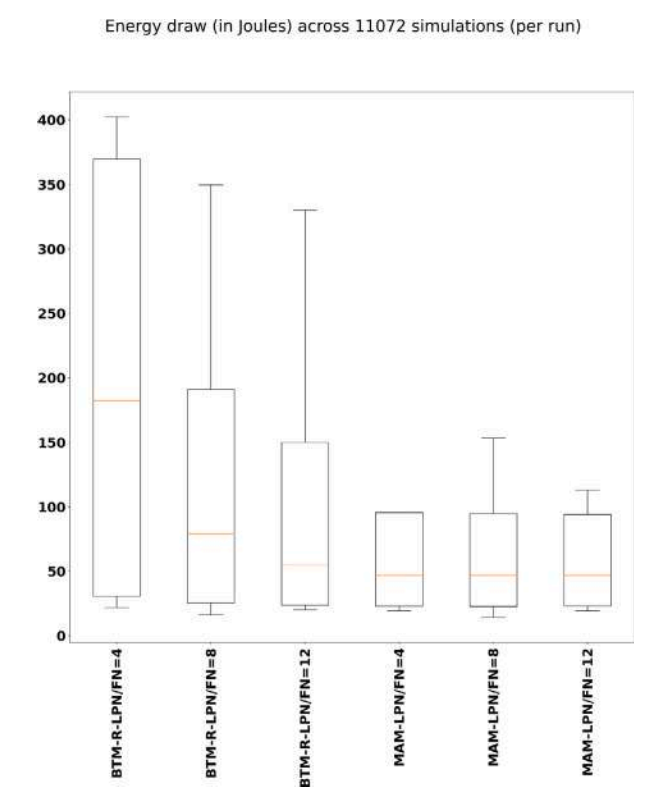


Fig. 11. Energy draw (in Joules) across 11 072 simulations by LPN/FN ratio — MAMSET.

LPN/FN = 8, 46.52 J 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.

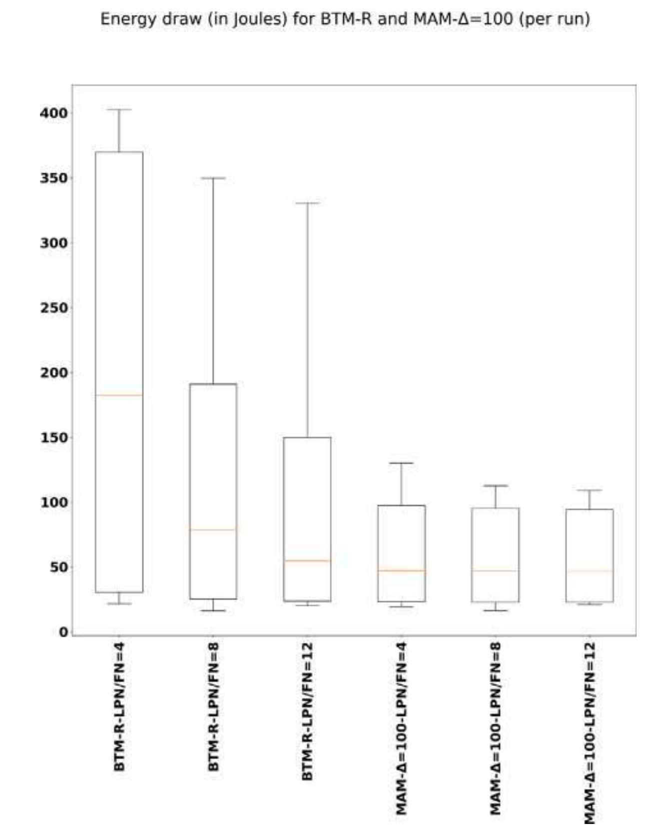


Fig. 12. Energy draw (in Joules) for BTM-R by LPN/FN ratio – $MAM_{\Delta=100}$ – MAMSET.

8.2.3. Received packets

Fig. 13 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 $\Delta = 50$ presented the highest median amount of bytes delivered to the Mobile-Hub (8800 bytes) followed by simulations with $\Delta = 100$ (8778 bytes). BTM-R, however, achieved a median value of 18 315 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 $\Delta = 100$ 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).

8.2.4. Energy efficiency

Fig. 14 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 $\Delta = 100$, 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 181.74 B/J for LPN/FN ratio = 4, 213.66 B/J for LPN/FN ratio = 8, and 253.18 B/J for LPN/FN ratio = 12. MAM_Δ with $\Delta = 100$ presented a median energy

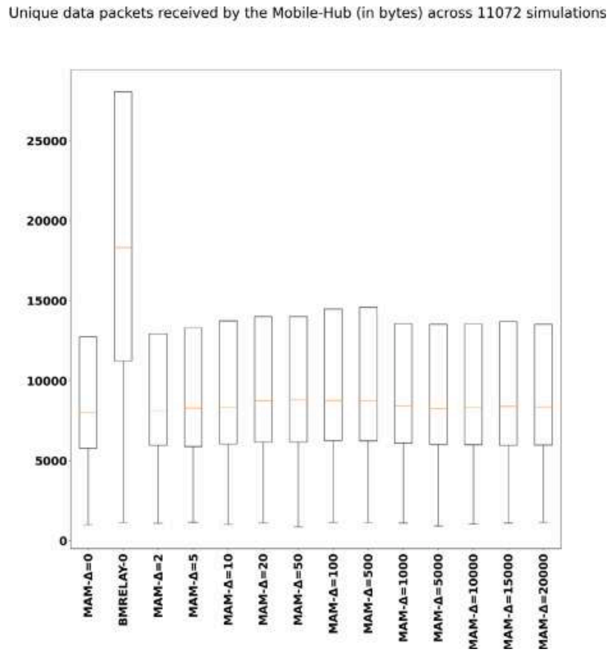


Fig. 13. Unique data received (in Bytes) comparison between BTM-R, MAM_0 and MAM_Δ — MAMSET.

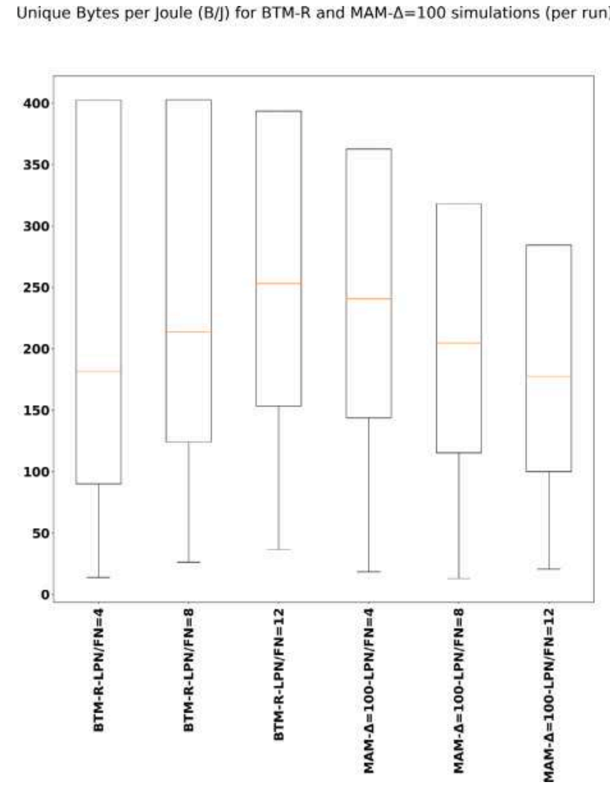


Fig. 14. Unique data received (in Bytes) comparison between BTM-R and $MAM_{\Delta=100}$ — MAMSET.

efficiency of 240.69 B/J for LPN/FN ratio = 4 (32.43% greater than BTM-R), 204.65 B/J for LPN/FN ratio = 8 (4.23% lower than BTM-R), 177.51 B/J for LPN/FN ratio = 12 (29.89% lower than BTM-R). This indicates that MAM_Δ with $\Delta = 100$ 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). Since when using MAM-Δ the network uses fewer nodes to relay packets to the mobile sink node, it is reasonable to expect that MAM would be more energy efficient in some configurations, when compared to BTM-R's flooding approach.

On BTM-R's energy efficiency results, the median increased as the LPN/FN ratio increased, whereas on the results of MAM_Δ with $\Delta = 100$ the median decreased as the LPN/FN ratio increased.

8.2.5. Delivery rate

Fig. 15 contains the delivery rate (in %) between BTM-R and MAM_Δ with $\Delta \geq 20$. 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 $\Delta \geq 20$ 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 $\Delta \geq 20$ 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.

Figs. 14 and 15 indicated increased energy efficiency and delivery rates for $MAM_{\Delta\geq20}$ 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 $MAM_{\Delta\geq20}$ 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 (400 sqm) and LPN/FN ratio (4 LPN/FN).

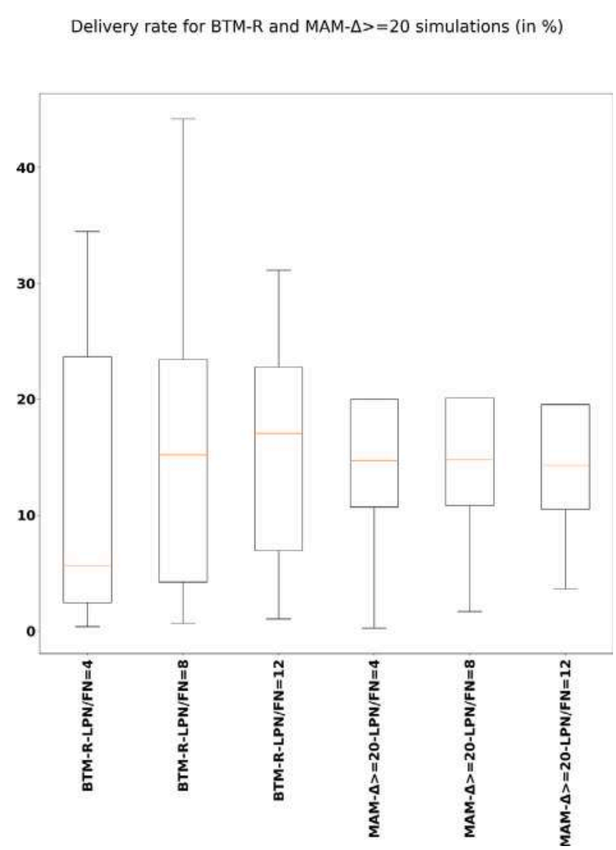


Fig. 15. Delivery rate (in %) comparison between BTM-R and $MAM_{\Delta\geq20}$ — MAMSET.

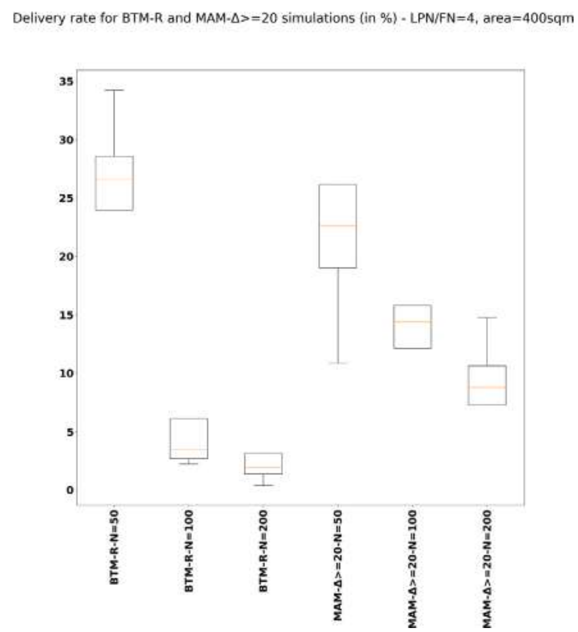


Fig. 16. Delivery rate for BTM-R and $MAM_{\Delta\geq 20}$ with LPN/FN ratio = 4 and area = 400 sqm — MAMSET.

Fig. 16 presents the delivery rate (in %) between BTM-R and MAM_{Δ} with $\Delta \geq 20$ across simulation results in which the LPN/FN ratio = 4 and the max area = 400 sqm. 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 $\Delta \geq 20$ 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 $\Delta \geq 20$ 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 $\Delta \geq 20$ 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).

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} can achieve higher delivery rates than BTM-R across different maps and topologies, as well as increased energy efficiency.

The simulation results discussed in Section 8 were able to partially cover most of the research questions introduced in Sections 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 [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. Exploring a mesh management strategy to dynamically adjust the value of the Δ parameter in accordance with the expected frequency of node visitations is also another possible path for future 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.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was financed in part by AFOSR, United States grant FA9550-20-1-0285. This work has been supported by the Brazilian research agency CNPq.

References

[1] M. Di Francesco, S.K. Das, G. Anastasi, Data collection in wireless sensor networks with mobile elements: A survey, ACM Trans. Sensor Netw. 8 (1) (2011) 7:1–7:31, http://dx.doi.org/10.1145/1993042.1993049, URL http://doi.acm.org/10.1145/1993042.1993049.

[2] A. Djedouboum, A. Ari, A. Gueroui, A. Mohamadou, Z. Aliouat, Big data collection in large-scale wireless sensor networks, Sensors 18 (2018) http://dx.doi.org/10.3390/s18124474.

[3] L.T. Rios, M. Endler, I. Vasconcelos, R. Vasconcelos, M. Cunha, F. Silva, The mobile hub concept: Enabling applications for the internet of mobile things, in: Pervasive Computing and Communication Workshops - PerCom Workshops 2015, 2015, pp. 123–128.

[4] M. Endler, F.S. e Silva, Past, present and future of the contextnet IoMT middleware, Open J. Internet Things (OJIOT) 1 (4) (2018) 7–23.

[5] C. Tunca, S. Isik, M.Y. Donmez, C. Ersoy, Distributed mobile sink routing for wireless sensor networks: A survey, IEEE Commun. Surv. Tutor. 16 (2) (2014) 877–897, http://dx.doi.org/10.1109/SURV.2013.100113.00293.

[6] C. Perkins, E. Belding-Royer, S. Das, RFC3561: Ad Hoc On-Demand Distance Vector (AODV) Routing, ISOC, 2003.

[7] A.G.A. Elrahim, EAGRP: Energy aware geographic routing protocol for wireless sensor networks, Int. J. Comput. Sci. Telecommun. 5 (11) (2014).

[8] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, IEEE Trans. Wirel. Commun. 1 (4) (2002) 660–670.

[9] S.K. Singh, P. Kumar, J.P. Singh, A survey on successors of LEACH protocol, IEEE Access 5 (2017) 4298–4328, http://dx.doi.org/10.1109/ACCESS.2017.2666082.

[10] O. Younis, S. Fahmy, HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Trans. Mobile Comput. 3 (4) (2004) 366–379.

[11] B. Baranidharan, B. Shanthi, A survey on energy efficient protocols for wireless sensor networks, Int. J. Comput. Appl. 11 (2010) 35–40, http://dx.doi.org/10.5120/1616-2173.

[12] M. Baert, J. Rossey, A. Shahid, J. Hoebeke, The bluetooth mesh standard: An overview and experimental evaluation, Sensors 18 (8) (2018) 2409.

[13] R. Friedman, A. Kogan, Y. Krivolapov, On power and throughput tradeoffs of wifi and bluetooth in smartphones, IEEE Trans. Mobile Comput. 12 (7) (2013) 1363–1376, http://dx.doi.org/10.1109/TMC.2012.117.

[14] E. Giacomini, F. D'Alterio, A. Lacava, F. Cuomo, 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, 2020, pp. 409–414.

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

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

[17] M.J.V. Paulon, B. Olivieri de Souza., M. Endler, Opportunistic routing towards mobile sink nodes in bluetooth mesh networks, in: Proceedings of the 18th International Conference on Wireless Networks and Mobile Systems - WINSYS, INSTICC, SciTePress, 2021, pp. 67–75, http://dx.doi.org/10.5220/0010557000670075.

[18] A. Madhavapeddy, A. Tse, A study of bluetooth propagation using accurate indoor location mapping, in: 2005 International Conference on Ubiquitous Computing - UbiComp 2005, Springer, 2005, pp. 105–122.

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

[20] C. Gomez, J. Oller, J. Paradells, Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology, Sensors 12 (9) (2012) 11734–11753, http://dx.doi.org/10.3390/s120911734, URL https://www.mdpi.com/1424-8220/12/9/11734.

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

[22] Bluetooth SIG, Provisioning a bluetooth mesh network part 1, 2017, URL: https://www.bluetooth.com/blog/provisioning-a-bluetooth-mesh-network-part-1/ access date: 2019-08-15.

[23] Y.E. Aslan, I. Korpeoglu, ö. Ulusoy, A framework for use of wireless sensor networks in forest fire detection and monitoring, Comput. Environ. Urban Syst. 36 (6) (2012) 614–625, http://dx.doi.org/10.1016/j.compenvurbsys.2012.03.002.

[24] N. Grammalidis, E. Çetin, K. Dimitropoulos, F. Tsalakanidou, K. Kose, O. Gunay, B. Gouverneur, D. Torri, E. Kuruoglu, S. Tozzi, A. Benazza, F. Chaabane, B. Kosucu, C. Ersoy, A multi-sensor network for the protection of cultural heritage, in: European Signal Processing Conference - EUSIPCO 2011, Vol. 1 (Eusipco), IEEE, 2011, pp. 889–893.

[25] L. Yu, N. Wang, X. Meng, Real-time forest fire detection with wireless sensor networks, in: Proceedings - 2005 International Conference on Wireless Communications, Networking and Mobile Computing - WCNM 2005, Vol. 2, IEEE, 2005, pp. 1214–1217, http://dx.doi.org/10.1109/wcnm.2005.1544272.

[26] M. Brini, L. Marmo, Wireless sensor networks for early fire detection, Chem. Eng. Trans. 24 (August 2003) (2011) 1153–1158, http://dx.doi.org/10.3303/CET1124193.

[27] B. Son, Y.-s. Her, J. Kim, A design and implementation of forest-fires surveillance system based on wireless sensor networks for South Korea mountains, Int. J. Comput. Sci. Netw. Secur. (IJCSNS) 6 (9) (2006) 124–130.

[28] E. Sun, X. Zhang, Z. Li, The internet of things (IOT) and cloud computing (CC) based tailings dam monitoring and pre-alarm system in mines, Saf. Sci. 50 (4) (2012) 811–815, http://dx.doi.org/10.1016/j.ssci.2011.08.028.

[29] P. Rajesh, N. Manjunathan, A. Suresh, Unmanned dam monitoring system using wireless sensor networks, Int. J. Innov. Technol. Exploring Eng. 8 (5) (2019) 343–347.

[30] G. Xue, Relay node placement in wireless sensor networks, IEEE Trans. Comput. 56 (1) (2007) 134–138, http://dx.doi.org/10.1109/TC.2007.250629.

[31] R. Muhendra, A. Rinaldi, M. Budiman, Khairurrijal, Development of wifi mesh infrastructure for internet of things applications, Procedia Eng. 170 (2017) 332–337, http://dx.doi.org/10.1016/j.proeng.2017.03.045, 2016 Engineering Physics International Conference – EPIC 2016. URL http://www.sciencedirect.com/science/article/pii/S1877705817311566.

[32] B. Nazir, H. Hasbullah, 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, 2010, pp. 624–629, http://dx.doi.org/10.1109/ICCAIE.2010.5735010.

[33] D. Zorbas, P. Raveneau, Y. Ghamri-Doudane, C. Douligeris, The charger positioning problem in clustered rf-power harvesting wireless sensor networks, Ad Hoc Netw. 78 (2018) 42–53, http://dx.doi.org/10.1016/j.adhoc.2018.05.013, URL http://www.sciencedirect.com/science/article/pii/S1570870518302440.

[34] A. Dementyev, S. Hodges, S. Taylor, J. Smith, 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, 2013, pp. 1–4, http://dx.doi.org/10.1109/IEEE-IWS.2013.6616827.

[35] I.F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Comput. Netw. 47 (4) (2005) 445–487, http://dx.doi.org/10.1016/j.comnet.2004.12.001, URL http://www.sciencedirect.com/science/article/pii/S1389128604003457.

[36] O. Landsiedel, K. Wehrle, S. Gotz, Accurate prediction of power consumption in sensor networks, in: The Second IEEE Workshop on Embedded Networked Sensors, 2005. EmNetS-II, 2005, pp. 37–44, http://dx.doi.org/10.1109/EMNETS.2005.1469097.

[37] G.H. Badawy, A.A. Sayegh, T.D. Todd, Fair flow control in solar powered WLAN mesh networks, in: 2009 IEEE Wireless Communications and Networking Conference - WCNC 2009, 2009, pp. 1–6, http://dx.doi.org/10.1109/WCNC.2009.4917720.

[38] J. Luo, J. Panchard, M. Piórkowski, M. Grossglauser, J.-P. Hubaux, Mobiroute: Routing towards a mobile sink for improving lifetime in sensor networks, in: Distributed Computing in Sensor Systems - DCOSS 2006, 2006, pp. 480–497.

[39] N. Todtenberg, R. Kraemer, Survey on bluetooth multi-hop networks, Ad Hoc Netw. 93 (2019).

[40] Ching Law, Kai-Yeung Siu, A bluetooth scatternet formation algorithm, in: 2001 IEEE Global Telecommunications Conference (Cat. No.01CH37270) - GLOBECOM 2001, Vol. 5, 2001, pp. 2864–2869, http://dx.doi.org/10.1109/GLOCOM.2001.965952.

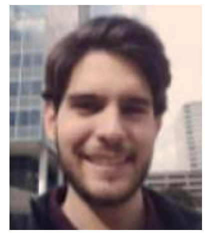
[41] T. Salonidis, P. Bhagwat, L. Tassiulas, R. LaMaire, Distributed topology construction of bluetooth personal area networks, in: Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213) - ComSoc 2001, Vol. 3, 2001, pp. 1577–1586, http://dx.doi.org/10.1109/INFCOM.2001.916654.

[42] C. Petrioli, S. Basagni, M. Chlamtac, Configuring BlueStars: multihop scatternet formation for bluetooth networks, IEEE Trans. Comput. 52 (6) (2003) 779–790, http://dx.doi.org/10.1109/TC.2003.1204833.

[43] C. Petrioli, S. Basagni, I. Chlamtac, BlueMesh: degree-constrained multi-hop scatternet formation for bluetooth networks, Mob. Netw. Appl. 9 (1) (2004) 33–47.

[44] X. Li, I. Stojmenovic, Y. Wang, Partial delaunay triangulation and degree limited localized bluetooth scatternet formation, IEEE Trans. Parallel Distrib. Syst. 15 (4) (2004) 350–361, http://dx.doi.org/10.1109/TPDS.2004.1271184.

[45] E.D. Leon, M. Nabi, An experimental performance evaluation of bluetooth mesh technology for monitoring applications, in: 2020 IEEE Wireless Communications and Networking Conference - WCNC 2020, 2020, pp. 1–6, http://dx.doi.org/10.1109/WCNC45663.2020.9120762.
[46] E.A.J. Hansen, M.H. Nielsen, D.E. Serup, R.J. Williams, T.K. Madsen, R. Abildgren, On relay selection approaches in bluetooth mesh networks, in: 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops - ICUMT 2018, 2018, pp. 1–5, http://dx.doi.org/10.1109/ICUMT.2018.8631214.
[47] F. Mager, M. Zimmerling, Mixer, PCI-Paint Coat. Ind. 2016 (AUGUST) (2016) http://dx.doi.org/10.1145/1096419.1096444.
[48] X. Ma, P. Zhang, X. Li, W. Tang, J. Wei, O. Theel, DeCoT: A dependable concurrent transmission-based protocol for wireless sensor networks, IEEE Access 6 (2018) 73130–73146, http://dx.doi.org/10.1109/ACCESS.2018.2877692.
[49] F. Ferrari, M. Zimmerling, L. Thiele, O. Saukh, Efficient network flooding and time synchronization with glossy, in: Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks - IPSN 2011, IEEE, 2011, pp. 73–84.
[50] D. Yuan, M. Hollick, Ripple: High-throughput, reliable and energy-efficient network flooding in wireless sensor networks, in: Proceedings of the IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks - WoWMoM 2015, IEEE, 2015, http://dx.doi.org/10.1109/WoWMoM.2015.7158133.
[51] L. Cheng, J. Niu, C. Luo, L. Shu, L. Kong, Z. Zhao, Y. Gu, Towards minimum-delay and energy-efficient flooding in low-duty-cycle wireless sensor networks, Comput. Netw. 134 (2018) 66–77, http://dx.doi.org/10.1016/j.comnet.2018.01.012.
[52] X. Ma, P. Zhang, Y. Liu, C.A. Boano, H.S. Kim, J. Wei, J. Huang, Harmony: Saving concurrent transmissions from harsh RF interference, in: Proceedings - IEEE International Conference on Computer Communications - INFOCOM 2020 2020-July, 2020, pp. 1024–1033, http://dx.doi.org/10.1109/INFOCOM41043.2020.9155423.
[53] C. Bisdikian, An overview of the bluetooth wireless technology, IEEE Commun. Mag. 39 (12) (2001) 86–94, http://dx.doi.org/10.1109/35.968817.
[54] S.M. Darroudi, C. Gomez, Bluetooth low energy mesh networks: A survey, Sensors 17 (7) (2017) http://dx.doi.org/10.3390/s17071467.
[55] E. Morin, M. Maman, R. Guizzetti, A. Duda, Comparison of the device lifetime in wireless networks for the internet of things, IEEE Access 5 (2017) 7097–7114, http://dx.doi.org/10.1109/ACCESS.2017.2688279.
[56] I. Bluetooth SIG, An intro to bluetooth mesh part 2, 2018, URL: https://www.bluetooth.com/blog/an-intro-to-bluetooth-mesh-part2/.
[57] T. Winter, P. Thubert, A. Brandt, J.W. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J.-P. Vasseur, R.K. Alexander, et al., RPL: IPv6 routing protocol for low-power and lossy networks, RFC 6550 (2012) 1–157.
[58] L. Kajdocsi, A. Dörömbözi, J. Kovács, Development of bluetooth mesh core stack using OmNET++, in: 2019 IEEE 17th International Symposium on Intelligent Systems and Informatics (SISY), 2019, pp. 23–28, http://dx.doi.org/10.1109/SISY47553.2019.9111635.
[59] D. Gislason, Zigbee Wireless Networking, Pap/Onl ed., Newnes, USA, 2008.
[60] M. Siekkinen, M. Hiienkari, J.K. Nurminen, J. Nieminen, How low energy is bluetooth low energy? Comparative measurements with ZigBee/802.15.4, in: 2012 IEEE Wireless Communications and Networking Conference Workshops - WCNCW 2012, IEEE, 2012, pp. 232–237, http://dx.doi.org/10.1109/WCNCW.2012.6215496.
[61] A. Varga, CircleMobility - INET framework, 2020, URL: https://doc.omnetpp.org/inet/api-current/neddoc/inet.mobility.single.CircleMobility.html.
[62] N. Todtenberg, R. Kraemer, A survey on bluetooth multi-hop networks, Ad Hoc Netw. 93 (2019) 101922, http://dx.doi.org/10.1016/j.adhoc.2019.101922, URL http://www.sciencedirect.com/science/article/pii/S157087051930126X.
[63] M. Endler, G. Baptista, L.D. Silva, R. Vasconcelos, M. Malcher, V. Pantoja, V. Pinheiro, J. Viterbo, 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, ACM, New York, NY, USA, 2011, pp. 2:1–2:2, http://dx.doi.org/10.1145/2088960.2088962, URL http://doi.acm.org/10.1145/2088960.2088962.



Marcelo Paulon J.V. is a Doctoral student in the Department of Informatics of PUC-Rio. He received his MSc degree in Computer Networks and Distributed Systems (2021) and his BSc degree in Computer Science (2018) from the same university. Marcelo has 8+ years experience with software development of high availability systems for different industries such as Gaming, Oil and Gas (O&G) and TV/Entertainment.

Bruno José Olivieri de Souza got his DSc from PUC-Rio (2019). Since then, he is a postdoc fellow at the PUC-Rio, and a collaborator at the LAC/PUC-Rio laboratory, coordinating the GRADYS project and guiding undergraduate and graduate students on their research. He has 6+ years of experience in UAV control beyond 10+ years in Information Security experience.

Markus Endler got his Dr. rer nat from TU Berlin (1992) and the Professor Livre-docente (habilitation) from University of Sao Paulo (2001). Since then, he is Associate Professor of Informatics at PUC-Rio, Brazil, where he is also Principal Investigator of the Laboratory for Advanced Collaboration. His main research interests include Pervasive Computing, scalable middleware and IoT systems. He has supervised 13 doctoral and 34 master theses, and has co-authored more than 170 scientific papers.