An Adaptive Duty Cycling Mechanism for Energy Efficiency in Bluetooth Mesh Networks

Sheida Mehrazi* and Majid Nabi†

*Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran
†Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands
Email: sheydamehrazi@ec.iut.ac.ir, m.nabi@tue.nl

***Abstract*—The Bluetooth Mesh (BM) standard was released in 2017 to expand the application of the Bluetooth Low Energy (BLE) technology for larger-scale multi-hop Internet-of-Things (IoT) networks. To implement multi-hop data delivery, BM defines a relay feature for BLE nodes using a controlled flooding algorithm for data dissemination. By default, the relay nodes persistently scan the advertising channels to receive packets from their neighbors and broadcast. However, it has a big impact on the energy consumption and lifetime of the relay nodes, especially when they have limited energy resources. On the other hand, any constant and homogeneous setting of a duty cycle for relay nodes across the network leads to inefficient performance since the nodes can be in different spatial and temporal conditions. This paper proposes a run-time adaptive duty-cycling mechanism for relay nodes to reduce unnecessary energy consumption and yet collaboratively make a reliable end-to-end data delivery. Each relay node independently decides about its suitable duty cycle based on its local conditions. The proposed method is evaluated using a publicly available BM simulator, showing a significant reduction in energy consumption (up to 57% in the conducted experiments) while preserving the packet delivery performance of the network.**

***Index Terms*—Bluetooth Mesh, BLE, Duty cycling, Wireless sensor networks, IoT.**

I. Introduction

Bluetooth Low Energy (BLE) as a short-range wireless communication technology has gained significant popularity for various Internet-of-Things (IoT) applications. It operates in the 2.4GHz ISM frequency band and provides low-power communication between energy-constrained devices, like wearable gadgets and smart sensors. BLE works in a point-to-point configuration, where two devices directly communicate with each other in a short range with a relatively low data rate (typically 1Mbps). The Bluetooth Special Interest Group (SIG) developed Bluetooth Mesh (BM) [1] to meet the increasing need for large-scale IoT deployments. The traditional point-to-point BLE communication model is insufficient for many IoT applications such as smart homes and buildings in which hundreds of devices may need to be deployed over a vast area. BM enables devices to communicate in a mesh network by introducing a relay feature for nodes. Relay-enabled nodes receive advertising packets from their neighbors and broadcast them to provide multi-hop data dissemination following a controlled flooding approach. This setup allows for multiple paths between source and destination nodes, increasing the reliability of end-to-end data delivery. However, to provide such a reliable multi-hop network, relay nodes are required to constantly scan the three BM advertising channels, leading to high energy consumption.

Duty Cycling (DC) of the scanning task of relay nodes is a potential solution to address the energy consumption issue. Although the nodes are not synchronized in a BM network, DC is still feasible thanks to the inherent redundancy that the flooding mechanism makes in the BM networks. There are already some research efforts in this direction. [2] gives DC a try by setting a fixed DC value for all the nodes in the network and studying the impact on the data delivery performance of the network. However, the main issue in such an approach is that BM networks are typically deployed in an ad hoc way. Thus, the nodes have different circumstances across the network in terms of the neighborhood and the paths they have toward their destinations or the sink of the network. Moreover, such conditions change over time due to all kinds of dynamics in a BM network such as node mobility or interference variation. Therefore, a uniform and constant setting of the DC value for all nodes in a large BM network does not give the best performance. In this paper, we propose a mechanism for run-time adaptive setting of DC value of individual nodes in a BM network according to the conditions of the node. Only local information is used by each node for making the decision about its DC value, which results in a low overhead solution. It is worth mentioning that, to avoid major performance deterioration, relay nodes in our mechanism configure their packet retransmission behaviors based on the DC value of their neighbors.

To evaluate the performance of the proposed mechanism, various BM network setups are tested using BMSim [3], which is an open-source event-driven BM simulator. The behavior of the proposed mechanism is compared with that of BM networks with no DC as well as with constant uniform DC settings. The simulation results show that the proposed adaptive mechanism reduces the energy consumption of relay nodes (up to 57% in some conducted simulations) without any negative impact on the network performance. The mechanism shows to be able to provide a good trade-off between energy consumption and data delivery performance.

The remainder of this paper is structured as follows. Section II presents an overview of the BM standard. Section III discusses related work. The proposed adaptive duty cycling mechanism is presented in Section IV. In Section V, the results of the performance evaluation are presented. Section VI concludes.

II. Bluetooth Mesh Overview

The BLE technology operates in the 2.4GHz ISM band with 40 channels of 2 MHz each. To ensure reliable communication, BLE networks employ channel hopping, switching between channels to reduce the impact of external interferences. Bluetooth Mesh (BM) is a multi-hop networking protocol built on top of BLE, using three BLE advertising channels (channels 37, 38, and 39) for data exchange. Nodes in a BM network may support four types of features each enabling specific tasks. The features are as follows.

Relay node: In a BM network, relay nodes play a vital role by receiving and forwarding packets from other nodes, effectively extending the network's range. They are crucial for ensuring reliable communication between nodes that are not directly within each other's range. Having an adequate number of relay nodes is essential to maintain optimal network performance and prevent disconnections within the BM network. **Proxy node:** Nodes with a proxy feature serve as a gateway, facilitating communication between the BM network and external BLE devices not part of the mesh. **Low Power Node (LPN):** There may be some nodes in a BM network with very limited energy resources. They conserve energy by spending the majority of their time in sleep mode. These nodes do not contribute to multi-hop data forwarding. Moreover, they depend on other nodes (friends) for their reliable reception of data packets, contributing to their prolonged battery life. **Friend Node:** This type of node is used to extend the battery life of LPNs. They act as a mediator for low-power nodes, storing messages on their behalf when those nodes are in a sleep state to conserve energy. When the low-power nodes wake up, friend nodes deliver the accumulated messages to them, ensuring reliable communication and reducing the power consumption of the low-power nodes.

In BM, scanning is the process of listening to the advertising channels to receive packets (advertisements) from other devices. Two parameters of the scan process are *scan window* (*SW*) and *scan interval* (*SI*), which define the time duration a device listens for advertisements and the interval between successive scan windows, respectively. The BM specification recommends equal values for these two parameters, resulting in continuous scanning ($DC = 100\%$). Such continuous scanning is needed in asynchronous networks to maintain connectivity and message exchange between neighboring nodes. By setting a scan interval that is lower than the scan window, duty cycling can be introduced for the relay node in their scan process ($DC = \frac{SW}{SI} * 100\%$).

For multi-hop data propagation, BM employs a controlled flooding mechanism. This protocol offers advantages such as simplicity, high reliability through redundant paths, and no need for routing table calculation. However, to control the redundancy level and reduce unnecessary data propagation,

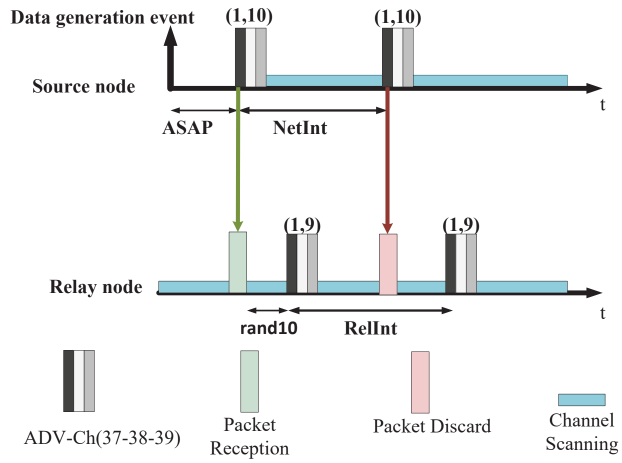

Fig. 1: Retransmission in BM network

BM controls the flooding by means of two services: sequence number and Time-To-Live (TTL) fields in the packets. Sequence numbers together with the ID of the source node are used to avoid loops. Also, each BM message contains a TTL value set by the source node to limit the maximum number of hops the message may travel. Source nodes determine a proper value for TTL based on their estimation of the hop distance to the desired destination. Such estimation is achieved using special packets that the destination nodes periodically broadcast (so-called Heartbeat messages). These messages are flooded all over the network by which each node can estimate its hop distance to the initiating destination node.

The BM specification offers another kind of redundancy in the time domain by packet retransmissions. Note that a transmitter needs to transmit each packet three times in three advertising channels to make sure that the receivers in the range get the packet. However, this process may be repeated a number of times to compensate for channel imperfections due to noise, interference, etc. There are two parameters involved in setting the number of retransmissions, Network Transmit Count (NTC) and Relay Retransmit Count (RRC), which specify the number of retransmissions by the source nodes and the relay nodes, respectively. The time between repetitions is determined by the Network Transmission Interval steps (*NetInt*) or Relay Retransmit Interval Steps (*RelInt*)), using Eqn. 1 and Eqn. 2, respectively. Additionally, a random value ranging from 0 to 10 ms is added to each transmission. The minimum value for these intervals 20 ms.

$$NetInt = (NTC + 1) \times 10ms + rand_{10} \tag{1}$$

$$RelInt = (RRC + 1) \times 10ms + rand_{10} \tag{2}$$

Fig. 1 illustrates packet transmissions in a simple BM network. The generator node sends packets it produces as soon as possible with NTC and RRC both set to one. The producer node sends the packet twice ($NTC = 1$) in the network layer, at an interval determined by *NetInt*. The (x,y) shown on top of packet transmission gives the sequence number (x) and TTL (y) of the transmitted packet. When a packet is received by the relay node, it is stored in the cache and transmitted RRC+1 times (twice in this example since RRC=1) at intervals determined by *RElInt*. During packet forwarding by the relay node, the TTL value is decreased by one.

III. Related work

The topic of energy efficiency in BM networks has recently received a lot of attention. Various methods have been proposed to reduce node energy consumption and enhance network lifetime. The power consumption of relay nodes in a BM network can vary due to factors such as scanning parameters and advertising activities. Given the constant scanning, relay nodes consume substantial energy, which compromises the low-energy principles of BLE. The research presented in [4] discusses the matter of energy consumption in BM networks and proposes a model based on measurements done by the nRF51422 hardware platform. The evaluation of energy consumption levels provided in [1] leads to the conclusion that BM may not be feasible for all IoT applications.

In [5], the authors propose an optional feature in the BM protocol to reduce energy usage in relay nodes. This feature allows nodes to remain in a low-energy scanning mode and switch to continuous scanning through a control message. The technique achieves significant energy savings of around 98% for applications that can tolerate higher data delivery latency. However, its suitability depends on specific use cases, network sizes, and scenarios.

[2] presents two proposals in the context of BM networks. The first proposal uses the C-LEACH clustering method [6] for BM networks. The second contribution of this work that is related to our work is the introduction of a fixed and uniform duty cycle for all relay nodes in the BM network. The goal is to reduce active listening time while ensuring at least one instance of each message is delivered. This is achieved by sending multiple repetitions of any messages at appropriate intervals. The number of repetitions depends on the used DC (e.g., 50% DC requires two repetitions). Combining these proposals shows that reducing the DC value increases collisions and lowers the packet delivery ratio due to repeated transmissions, especially in large networks with high data traffic load. Although a 25% DC provides 78% reduction in energy consumption, using fixed and uniform DCs leads to reduced packet delivery ratio and suboptimal settings.

The proposed mechanism in this paper aims to adjust the DC for each node based on its conditions. Additionally, the mechanism takes into account the dynamic nature of BM networks, where nodes may change their location or be removed from the network. Overall, the proposed solution offers an adaptive approach for introducing duty cycling to the BM nodes, considering both energy efficiency and network performance.

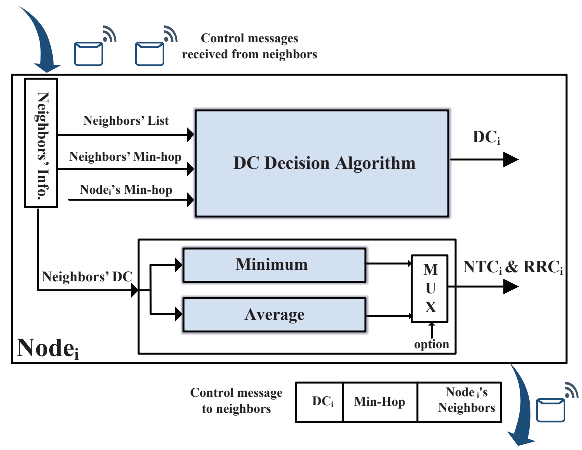

Fig. 2: General behavior of a relay node implementing the adaptive duty cycling mechanism

IV. Adaptive Duty Cycling

This paper proposes a mechanism for setting the DC value of individual relay nodes at run-time instead of having a fixed and uniform DC value for all nodes. In this mechanism, we utilize the local information of nodes about their status in the BM network to determine their DC, which can vary from 100% (SW = SI) to 25% (SW = 0.25 SI). Fig. 2 shows the general structure of the adaptive DC mechanism by showing the behavior of each relay node for making a decision about its DC value (DC_i) at run-time. Each node i uses some parameters that it either receives from its direct neighbors or has them internally and decides about its DC. Also, each node decides about its packet retransmission (NTC_i and RRC_i) based on the selected DC values by its neighbors. In the following, we first explain the DC setting and then discuss how retransmissions are set.

A. DC Setting Mechanism

This research focuses on convergecast wireless sensor networks, where data packets generated by all source nodes need to be collected in a central node. This node uses a DC value of 100% and continuously scans the three advertising channels since it usually does not have energy constraints. To reduce the chance of packet drop, nodes closer to the central node require higher DC values since their function is crucial for delivering data packets to the central node. Therefore, our first decision parameter is the estimated hop distance of each relay node to the central node. This distance is derived from already existing heartbeat messages and thus imposes no overhead for the network.

In BM networks, all neighboring nodes within a node's radio range receive the transmitted packets. Thus, each node i maintains a list of its neighbors (N_i). As the second decision parameter, we analyze the number of common neighbors shared between a node and its neighbors. Choosing this parameter is beneficial because a high number of common

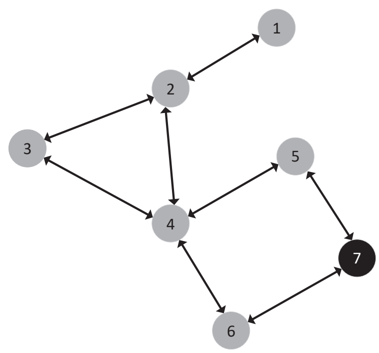

Fig. 3: An example BM network illustrating decision-making about DC value

neighbors reduces a node's responsibility for delivering the received data packets. This is because, with a good chance, the neighboring nodes have already received the same data packet. It leads to less criticality for the node for data dissemination. Another important reason to consider the number of common neighbors is that a node may have multiple neighbors, but only one of them is on the path toward the central node. In such cases, the node can only forward data packets through that specific path. If the node goes to sleep and the other neighbor receives the data packets, it needs to adjust its DC to receive and rebroadcast the data packets from this neighbor with high reliability. Also, this mechanism takes into account the fact that if the number of common neighbors between two nodes is low, the node should check whether the neighboring node is closer to the central node. If the neighboring node is closer, it is less likely that it needs to receive packets from this neighbor and forward them to the central node. In this case, the node can ignore the low number of common neighbors without any effect on its duty cycle.

The next factor for decision is isolated neighbors. If a node has only one neighbor, its data packets can only reach the central node through that neighbor. Therefore, the DC value of the neighbor node is increased to make sure it receives and broadcasts the data packets. The number of common neighbors is used as a decision parameter to address this aspect. Since the isolated node has zero common neighbors with its neighbor and is farther from the central node, the neighbor node takes a higher responsibility for receiving and rebroadcasting the data packets of the isolated node. Consequently, a higher duty cycle is selected for the neighbor node. In this mechanism, isolated nodes have the lowest duty cycle (25%) as they only serve as data producers toward the central node and cannot participate in forwarding data packets from others.

Fig. 3 shows a part of an example BM network, with Node 7 as the central node. Node 1, being an isolated node with only one neighbor, has the least DC. Node 2, serving as the route for Node 1's data packets, uses a high DC. The distance to the central node is a factor in determining DC. Nodes 5 and 6, being one hop away from the central node, bear significant responsibility in delivering data packets to it, and use higher

Algorithm 1: Duty cycling algorithm run by node i

```
2  N_j: Set of neighbors of any node j
4  d_j: Hop distance of any node j to the central node
6  C: An array of the common neighbors of nodes i and its
     neighbors
8  c_ij: Number of common neighbors of nodes i and j
10 for ∀j ∈ N_i do
12 |   c_ij ← |N_i ∩ N_j|  /* # of common neighbors
   |     */
14 |   if d_i > d_j then
15 |   |   c_ij = 5
16 |   C_i ← c_i,j
18 α ← min C_i  /* Find the minimum c_ij in list
     */
19 if 1 ≤ d_i ≤ 2 then
20 |   step = 10
21 else if 3 ≤ d_i ≤ 6 then
22 |   step = 15
23 else
24 |   step = 20
   /* Calculate Duty-Cycle                          */
   _i ← max[(100 − α × step) × 100 , 25%]
   /* Calculate Scan-Interval                       */
26 SI_i ← Scan_window / _i
```

DCs. Node 4 illustrates that having fewer common neighbors makes its role in data relay more crucial, necessitating a higher DC. For instance, when Node 4 is in sleep mode, only Node 3 can receive data from Node 2, increasing the risk of packet loss. Moreover, Node 4's fewer hops to the central node compared to Nodes 2 and 3 show their reliance on Node 4 for data propagation toward the central node. Consequently, Node 4 gets a higher DC value. Despite Nodes 5 and 6 have few common neighbors with Node 4, their shorter hop distance to the central node negates reliance on Node 4. Hence, Nodes 5 and 6 do not influence Node 4's DC.

Algorithm 1 outlines the general process of the proposed method. In this algorithm, each node compiles its neighbor list and shares it with its neighbors. This results in Node i possessing both its own neighbor list (N_i) and its neighbors' neighbor list (N_j , $\forall j \in N_i$) it receives from them. Consequently, the common neighbor count between Node i and its neighbors is computed. Array C, with elements c_{ij}, represents these common neighbor counts between node i and its neighbors ($\forall j \in N_i$). With the number of common neighbors being a key factor in adjusting DCs, this array must consider a minimum count (α) to prevent neighbor data packet loss. If a neighbor node j is closer to the destination than node i, the common neighbor count between i and j in array C is replaced with a predefined maximum value. It is set to 5. When a node has over 5 common neighbors, it can have a lower minimum DC value.

The node starts its operation with a DC of 100% DC. Then based on its local data, it may reduce DC using α with specific step values, as presented in line 16. Discrete step values depend on each node's distance from the central node, 10 for close nodes, 15 for intermediate nodes, and 20 for farther nodes. These step values can be adjusted to suit specific networks although they generally work well based on our experiments. The DC calculation formula ensures a minimum 25% value to prevent data loss, as values below 25% increase this risk. Once each node's DC is calculated, it is sent to all its neighbors.

We apply a fixed SW to all nodes and vary SI to set the desired DC value. This is influenced by retransmission parameters *NTC* and *RRC* which affect the timing of transmissions. These intervals must not surpass the scan window to ensure packet reception in neighboring nodes' active cycles. If various nodes use different SW settings, neighboring nodes would require knowledge of each other's SW for their parameter adjustment. To address this, all nodes use uniform SW and change SI to set their DC.

B. Retransmission Count Setting

In mesh networks in which the schedules of nodes are not synchronized, duty cycling can cause packet loss. Packet retransmissions can reduce this chance. However, increasing retransmissions can lead to collisions, hindering packet delivery. Thus, there should be a relation between the duty cycle used by neighbors and the number of retransmissions. Since nodes may use different DC values in our solution, nodes share their DCs with neighbors, allowing them to determine the appropriate number of retransmissions based on their neighbors' DCs. This approach eliminates unnecessary retransmissions.

Each node receives a number of DC values (set DC_{ng}) from its neighbors and uses one of the two approaches we envisioned for making decisions about its retransmission. In the first approach, the average of the DC values of neighbors is considered as the base DC value (DC_{base}) to be used for setting retransmission counts ($DC_{base} = \frac{1}{|DC_{ng}|}\sum_{k \in DC_{ng}} k$). The second approach is more conservative taking the minimum DC of the neighbors into account ($DC_{base} = \min_{k \in DC_{ng}} k$). In any case, the retransmission count parameters are set using Eqn.3.

$$NTC = RRC = \lceil \frac{1}{DC_{base}} \rceil - 1 \tag{3}$$

According to the BM standard [1], the number of transmissions for data generator and relay nodes is equal to $NTC + 1$ and $RRC + 1$, respectively. Therefore, in the formula, these parameters are reduced by one. It is preferable to consider the maximum number of repetitions that do not significantly increase network traffic or cause collisions. For example, if a node has an average neighbor DC of 60%, its number of repetitions would be 1. Consequently, this node sends each packet twice, with the configured time interval. When a node computes its transmissions based on the lowest neighbor DC, it results in more transmissions compared to using the average DC. This elevated transmission count enhances data packet reliability in low-traffic networks; yet in larger or high-traffic setups, it can lead to collisions and buffer overflow, reducing the packet delivery ratio.

V. Performance Evaluation

We evaluate the effectiveness of our proposed technique by testing it in different setups and comparing its performance with that of the standard BM [1] (100% scan) and the method in [2], which bears the closest resemblance to our approach (uniform DC values of 25% and 50%). Two variations of our adaptive mechanism are tested. $Adaptive_{avg}$ and $Adaptive_{min}$ stand for the adaptive mechanisms in which the average or minimum of the DC values of the neighbors are considered for retransmission setting, respectively. We use BMSim [3], which is an event-driven BM simulator. We examined three BM networks of different sizes (50, 100, and 200 nodes) distributed over an area of size $50m \times 50m$, $70m \times 70m$, and $100m \times 100m$, respectively. Nodes are deployed randomly, but network connectivity is checked. Link quality between 60% to 100% was considered for more realistic network simulation. This range reflects potential effects like noise, interference, and multi-path fading. The central node is selected using the closeness algorithm [7]. Two-thirds of the nodes act as relays and generators, while the remaining one-third are only relays, determined by mesh graph betweenness [7]. Each source node generates one data packet per second. To have statistically more reliable results, each simulation is repeated ten times, each containing 300 packet generations. We consider three performance metrics, which are the total energy consumption of nodes, average end-to-end Packet Delivery Ratio (PDR), and average latency. For estimating the energy consumption, the current values of the Nordic Semiconductor nRF52840 chip in various operational modes of the transceiver are used.

Fig. 4, Fig. 5, and Fig. 6 present the achieved results for the total energy consumption of nodes, PDR, and latency, respectively. Among the five methods, the constant DC of 25% exhibits the lowest energy consumption, but with a very low PDR. It means that such an approach for duty cycling can lead to deteriorated packet delivery performance. In contrast, the proposed method notably reduces energy usage compared to the default BM (100% scan), while surprisingly it gives better PDR in some cases. This is because the retransmissions in the adaptive DC mechanism compensate for link quality imperfections. However, increasing retransmissions, especially during high network traffic, may lead to more collisions and buffer overflow in relay nodes. As a result, our approach shows a slightly lower median for PDR compared to the default BM. For instance, in a 100-node network, $Adaptive_{avg}$ has 2% lower PDR than 100% scan, which increases to 13% for 200 nodes. The boxplots in Fig. 5 illustrate this, with the median PDR shown by the black line. However, given the energy efficiency gains of the proposed approach compared to default BM, this minor PDR degradation may be negligible.

Comparing $Adaptive_{avg}$ and $Adaptive_{min}$ modes, adjusting the number of retransmissions based on the minimum DC of neighbors has led to more packet transmissions and higher traffic load. It means a higher chance of collisions and buffer overflows in relay nodes, resulting in lower PDR for $Adaptive_{min}$. Another observation is that larger networks have

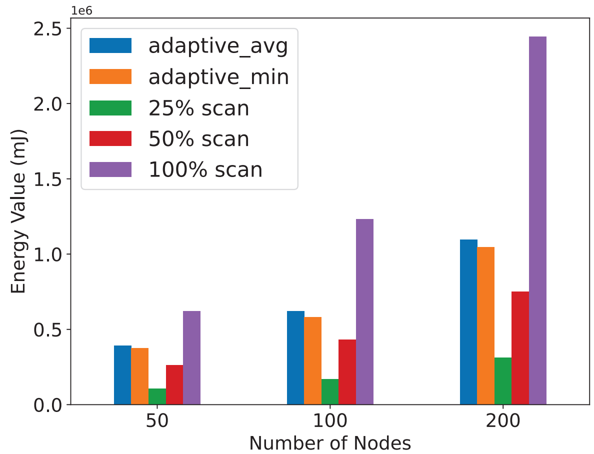

Fig. 4: Total energy consumption of BM nodes.

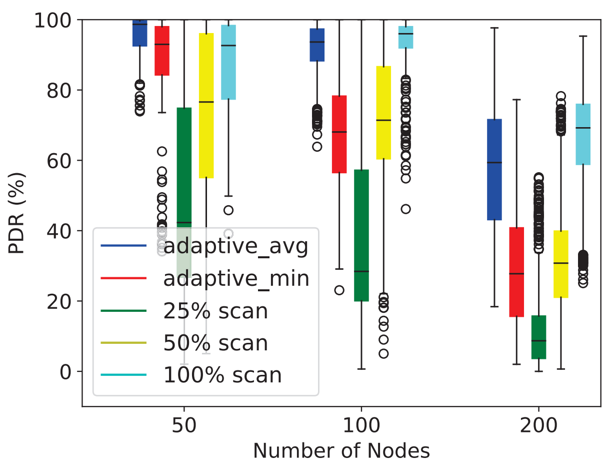

Fig. 5: Achieved end-to-end PDR.

more nodes with lower DCs due to increased neighboring nodes and greater distances from the central node. This leads to more reduction of energy consumption with our adaptive mechanism.

Concerning latency, $Adaptive_{min}$ and 25% scanning show higher delays due to a higher amount of packet transmissions. A comparison between them reveals that the higher Packet Delivery Ratio (PDR) of $Adaptive_{min}$ also contributes to its higher delay. Similarly, in a 200-node network, 50% scanning faces more collisions and buffer overflow than 100% scanning due to more retransmissions, resulting in fewer data packets reaching their destination and a slightly lower delay.

By considering the three network performance parameters that were investigated, it can be concluded that the proposed adaptive duty cycling mechanism substantially decreases the energy consumption compared to the default BM, while it only has a very minor impact on PDR and latency (even with improvement in some scenarios), particularly in large networks. However, since the primary focus of the fixed DC

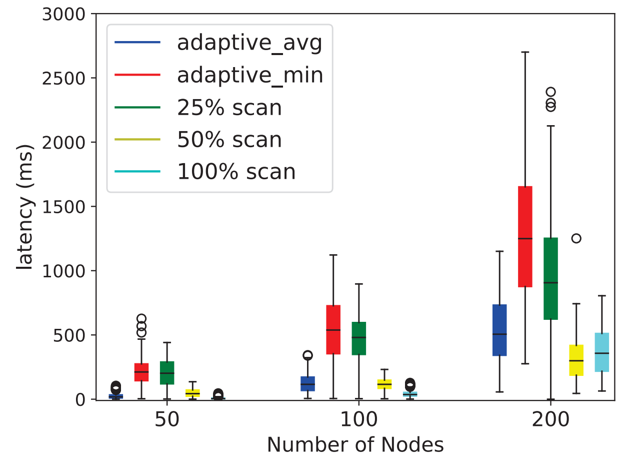

Fig. 6: Achieved average latency.

methods (25% and 50% scan) has been achieving a good reduction in energy consumption, they result in very low data delivery performance.

VI. Conclusion

Continuous scanning of the advertising channels by relay nodes in Bluetooth Mesh (BM) networks leads to high energy consumption for energy-constrained wireless devices. This paper proposed an adaptive duty cycling mechanism for the run-time configuration of the scanning activities of relay nodes. Such nodes make this decision based on their status in the network which may be different for various nodes and may vary over time. To avoid performance degradation due to duty cycling, nodes use retransmissions which are proportional to the duty cycle of the neighbors. The simulation results show that the proposed mechanism provides energy savings with negligible impact on the overall data delivery performance. As a future work, we are working on lightweight machine learning algorithms to make the decision process about duty cycling smarter and more efficient.

References

[1] "Mesh Networking Specifications — Bluetooth® Technology Website," *Bluetooth Mesh*, pp. 1–333, November 2020. [Online]. Available: https://www.bluetooth.com/specifications/mesh-specifications/
[2] J. D. Gotz, O. K. Rayel, and G. L. Moritz, "Improving bluetooth mesh energy efficiency using clustering," *Journal of Communication and Information Systems*, vol. 36, no. 1, pp. 156–165, 2021.
[3] Z. Hosseinkhani and M. Nabi, "BMSim: an event-driven simulator for performance evaluation of bluetooth mesh networks," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 2139–2151, 2024.
[4] S. M. Darroudi, R. Caldera-Sànchez, and C. Gomez, "Bluetooth mesh energy consumption: A model," *Sensors*, vol. 19, no. 5, p. 1238, 2019.
[5] D. Perez-Diaz-de Cerio, J. L. Valenzuela, M. Garcia-Lozano, Á. Hernández-Solana, and A. Valdovinos, "BMADS: BLE mesh asynchronous dynamic scanning," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2558–2573, 2020.
[6] F.-H. N. A. Yektaparast and A. Sarmast, "An improvement on leach protocol (cell-leach)," *2012 14th IEEE international conference on advanced communication technology (ICACT)*, pp. 992–996, 2012.
[7] A. Syarif, A. Abouaissa, and P. Lorenz, "Operator calculus approach for route optimizing and enhancing wireless sensor network," *Journal of Network and Computer Applications*, vol. 97, pp. 1–10, 2017.