BMSim: An Event-Driven Simulator for Performance Evaluation of Bluetooth Mesh Networks

Zohreh Hosseinkhani, Majid Nabi, *Member, IEEE*

***Abstract*—Bluetooth Mesh (BM) is one of the promising networking technologies for Internet-of-Things (IoT) networks released by Bluetooth SIG in 2017. BM provides a protocol for multi-hop scalable networking of IoT devices over the widely-used Bluetooth Low Energy (BLE) technology. However, the capabilities and limits of this technology are not fully studied to determine the IoT applications for which this technology can be used. One of the barriers towards this is the lack of a suitable BM network simulator for performance investigations under various conditions and settings. This paper presents a full-fledged open-source event-driven simulator (BMSim) for the performance evaluation of BM networks. The accuracy of the developed simulator is verified by real experiments. Also, BMSim is used to perform a comprehensive investigation of the performance of the BM protocol in various network conditions and configuration settings. The impact of several configuration parameters on the BM network performance is studied. Since the simulator is capable of simulating dynamic networks and run-time configurations, a BM network with node mobility is also investigated. The results reveal the importance and necessity of proper BM parameter configuration mechanisms to achieve the required quality-of-service and network efficiency.**

***Index Terms*—Bluetooth Mesh, BLE, Network simulator, Performance study, BMSim.**

I. Introduction

In the Internet of Things (IoT) era, millions of smart devices around the world are getting connected. A variety of applications are relying on the IoT concept for their true operation. Examples are health, transportation, industrial automation, and environment monitoring [1]. The communication technologies for IoT are divided into two categories: short-range technologies like Wi-Fi, Zigbee, Thread, Bluetooth Low Energy (BLE) [2], and long-range technologies such as LoRA and NB-IoT. The availability of the BLE transceivers in almost all daily-used smart devices (e.g., smartphones and tablets) has made it very widely used and accessible [3]. BLE utilizes star topology for communications, making the implementation and setup of BLE networks simple and straightforward. In many applications in which short-range communication, low-energy consumption, and a reliable connection are required, BLE is a very promising option. Examples of this include monitoring the patients' vital signals [4].

Using star topology and lack of multi-hop communication cause limitations for the applicability of BLE in many application domains. The limitation in the number of devices in a BLE network adds to this limitation for large-scale networking. Therefore, Bluetooth SIG released Bluetooth Mesh (BM) [5] in 2017 extending the single-hop BLE to a multi-hop mesh networking solution. The mesh stack sits on top of the physical and data link layers of BLE, using a managed flooding mechanism in the network layer [6]. Robustness, easy implementation, and reliable end-to-end data delivery are advantages of the managed flooding mechanism used in the BM network layer. It is of high interest for many applications to adopt BM technology, due to the high number of connectable devices and improved communication range.

BM networks are rather new networks with high expected potential for many IoT applications. However, their behavior under various conditions and their performance limitations need to be extensively investigated before their wide adoption in such applications. Moreover, there are several parameters in a BM network, and a proper configuration of them may have a great influence on the network performance. Thus, the exploration of network configurations to reach (near-) optimum performance is of paramount importance. In recent years after the first release of the standard in 2017, some researchers have investigated the BM network's parameters to achieve better performance in these networks (e.g., [7] [8] [9]).

In this paper, the eventual goal is to perform a comprehensive investigation of the functionality and performance of the BM technology in various network conditions and topologies. In particular, we are interested to explore the impact of configuration parameters (e.g., advertising interval, scan window, and packet re-transmissions) on the network performance and the trade-off they make in terms of different Quality-of-Service (QoS) metrics. However, to achieve this goal, we first need to have an accurate and scalable BM simulator that supports full network features. The problem is that there was no publicly available BM simulator that can be used for this purpose. Widely used network simulation engines such as NS-2, NS-3, and OMNeT++ do not have models for the Bluetooth mesh stack. Only the MATLAB R2020a edition released a BM simulator in the communications toolbox-library package. However, the MATLAB BM simulator is closed-source, and

Zohreh Hosseinkhani is with the Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran (email: z.hosseinkhani@ec.iut.ac.ir). Majid Nabi is with the Department of Electrical Engineering, Eindhoven University of Technology, 5600 MB Eindhoven, The Netherlands, and also with the Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran (e-mail: m.nabi@tue.nl).

TABLE I: Specification of BMSim compared to popular network simulation engines

Simulators specification	BMSim	NS-3	NS-2	OMNET++	MATLAB (Comm. toolbox)
Event-driven	Yes	Yes	Yes	Yes	No
Publicly-available	Open-source	Open-source	Open-source	Open-source	Closed-source
Modeling language	Python	C++/Python	C++/TCL	C++	MATLAB script
Includes BM protocol stack	Yes	No	No	No	Yes
User learning	Fast	Moderate	Moderate	Moderate	Fast
Scope	BM-Customized	General-purpose	General-purpose	General-purpose	BM-Customized
BM run-time reconfiguration	Supported	-	-	-	Not suppported
BM node mobility	Supported	-	-	-	Not suppported
BM network settings	dynamic	-	-	-	fixed

most parameters and configurations are fixed. It means it is not possible to modify those settings, which is a crucial aspect especially when it is about doing research on the performance of such networks. Adaptive configuration of network parameters at run-time is another requirement that is not supported by the MATLAB BM simulator. Therefore, to relax such limitations, we first developed a full BM simulator (BMSim) in Python which is open-source, extensible, and publicly available with a number of features that are important for research on this technology. BMSim is designed as a stand-alone customized event-driven simulator instead of being developed on top of the existing network simulation engines such as NS-2, NS-3, and OMNeT++. The first reason for this decision is to provide a lightweight simulator with a short learning period for users. Setting up a BM network and simulating it in BMSim is very straightforward and does not need an insight understanding of a simulation engine, which is a time-consuming effort for many users. The other reason is to avoid complexities that come with the general-purpose network simulation engines and be customized for the needs and specifications of the BM protocol. It leads to a faster BM simulation and enables the users to simulate large-scale BM networks with any settings and dynamics in a reasonable simulation time. Table I presents various specifications of BMSim compared to other popular simulators. Note that there is no BM protocol model on top of the simulation engines such as NS-2, NS-3, and OMNeT++ at the time of writing this article; they are included in Table I as indications of why BMSim is developed as a stand-alone BM simulator. To summarize, the key contributions of this paper are as follows.

1) A full-fledged BM simulator is implemented, which can simulate the link layer and network layer of a given BM network in detail. The nodes' features such as relay, friend, and low power can be set. Moreover, the simulator is capable of the run-time configuration of parameters, features, models, and topologies. Therefore, dynamic mobile networks with adaptive features can be implemented and simulated by BMSim. Network performance metrics like end-to-end Packet Delivery Ratio (PDR), latency, and nodes' energy consumption are measured and reported by BMSim. The simulator is open-source, easily extensible, and publicly available online through https://github.com/BMSimulator/BMSim.
2) BMSim is experimentally verified by comparing its performance estimation results with those out of real experiments, performed using Nordic Chip nRF52840 dongles. The results confirm that the developed simulator can estimate the performance of BM networks with good accuracy.
3) Using the developed BM simulator, the performance of BM networks in various settings and configurations is analyzed. The effect of important configuration parameters of BM networks (i.e., advertising interval, scan window, re-transmission, data packet generation rate, and heartbeat massage rate) on network performance is extensively investigated. BM networks with a variety of scales and topologies and in diverse conditions including scenarios with node mobility are considered to explore the trade-off that such parameters make.

The rest of the paper is organized as follows. Section II gives a brief overview of BLE and BM standard technologies. Section III explains the related work on studying the performance of BM networks in the literature. The structure and features of the developed BM simulator are presented in Section IV. Verification of BMSim using real experiments is discussed In Section V. Section VI discusses the performed experiments and the results of BM performance investigations for various network setups. Section VII concludes.

II. Overview of the Bluetooth Mesh Protocol

Bluetooth mesh runs on top of the physical and data link layers of the BLE technology. The BM protocol stack contains several layers each having some duties and responsibilities. The Model layer defines application models which are used by users such as lighting and sensing. The foundation model layer determines models, states, and messages for the configuration and management of the network. The access layer manages how the upper layers use the upper transport layer. Also, the application data format and controlling the application data encryption and decryption are defined by the access layer. The Upper Transport layer encrypts, decrypts, and authenticates application data. Also, it provides confidentiality in the access messages and defines control messages that manage this layer. The lower Transport layer defines how to reassemble and segment upper transport messages. The network layer addresses transport messages and decides whether a message is forwarded to other nodes or rejected. The Bearer layer is responsible for how network messages are sent between nodes. There are two bearers as the GATT bearer and the advertise bearer.

A. Bluetooth Low Energy Core Specification

The physical layer of BLE consists of 40 channels in 2.4 GHz each with 2MHz bandwidth. Three of these channels, namely channels 37, 38, and 39, are advertising channels and the other 37 channels are data channels. The advertising channels are used for broadcasting, device discovery, and connection establishment [10]. The data channels are used for sending and receiving data when the connection is established between the sender and the receiver [11]. The physical layer bit rate is 1 Mbps in Blutooth4.x and it is ranged between 125kbps to 2Mbps in Blutooth5.

B. BM Nodes Features

There are four features defined in the BM standard for each node as *relay*, *friend*, *low power*, and *proxy* [12]. The network designer or any smart management mechanism can enable or disable these features for individual nodes.

Relay: The relay feature is an essential aspect of the protocol for providing multi-hop mesh networking. A relay-enabled node scans the advertising channels to receive data packets from its neighbors and forwards them to other nodes. The low number of relay nodes in a BM network may lead to performance degradation and even network disconnections. On the other hand, a large number of relay nodes may cause high traffic and lots of packet collisions in the network. Therefore, the smart selection of relay nodes is an important decision in BM networks.

Low power and Friend: These two features are used for decreasing energy consumption for energy-constraint nodes called Low Power Node (LPN). One node may not have these two features at the same time. To preserve energy, LPNs are off for the majority of the time. To avoid losing data packets, each LPN should be associated with a node with a friend feature enabled. The friend node receives and stores the packets targeted to the LPNs associated with it. LPNs periodically wake up and send a data poll packet to the associated friend node asking for any packet for them in the buffer of the friend node. If the friend node has some packets in its buffer for the asking LPN, the packets are transferred using an efficient handshaking mechanism.

Proxy: This feature is used for connection between the mesh network and BLE nodes that do not have mesh capability. A node with a proxy feature enabled play as a connection spot to the Mesh for the BLE node.

C. Multi-hop Mesh Networking

Multi-hop data delivery is the main objective of the BM technology. It is realized by setting nodes in the connection-less BLE operational mode in which nodes can exchange packets without making a connection. The multi-hop data transmission is realized using a controlled flooding mechanism in the network layer of the BM stack. In this mechanism, nodes with relay feature scan the advertising channels to receive data packets from their neighbors. Upon reception of a data packet targeting some other nodes than the receiver node, the relay node broadcasts it under certain conditions to all its neighbors. In turn, the data packets reach their destinations with good reliability provided by the flooding algorithm [13].

To control the traffic load and the level of redundancy in data packet transmission, two main mechanisms are employed. Each packet contains the ID of the source node (generator) as well as a sequence number assigned to the packet by the source node. Each relay node maintains a cache to store packets received from other nodes. When the node receives a new packet, it checks if it already has processed a packet from the same source node with an equal or lower sequence number. In that case, the relay node discards the new packet to avoid packets circulating in the network forever. It means relay nodes try to process (buffer and forward) only the latest received packet from each source node.

To further control flooding, the source nodes assign a Time-To-Live (TTL) value to the data packets they generate. By this, the source node indicates the domain around itself that it aims for its data to be flooded over. Each relay node in the path decreases this TTL value by one. A packet is forwarded by the relay nodes as long as its TTL is larger than one. Otherwise, it is discarded. Therefore, the number of hops that a packet can traverse is controlled. The right setting of TTL by the source nodes is very important and has a direct impact on network traffic load and thus its capacity. For that, the source node needs to have an estimation of its current hop distance to the intended destination(s). To provide such estimation, the destination nodes periodically transmit *Heartbeat* messages that are flooded like the data packets. When a source node receives a Heartbeat message originating from a destination node, it gets the number of hops the packet has traveled, thus the distance to the destination.

D. Packet Transmission in BM Networks

In the BM protocol, data packets are sent and received using an advertising/scanning procedure by which three advertising channels are used without any connection establishment between receivers and senders. The non-connectable and non-scannable undirected advertising events of BLE are used for packet exchange in BM [14]. In an advertising event, the sender node broadcasts its data packet in the three advertising channels in a row. The time between two consecutive advertising events is *advertising interval* (T_{adv}), which is an integer multiple of $0.625ms$ in the range of $20ms$ to $10.24s$. Also, nodes scan the three advertising channels in turn, to receive packets from possibly advertising nodes in their neighborhood. The *scan window* ($T_{scanWin}$) is the time duration that a scanning node listens to an advertising channel. Scanning of the next advertising channel starts each *scan interval* ($T_{scanInt}$). If the scan window and scan interval parameters are set to be equal, the scanner node continuously scans the three advertising channels in the same order. Fig. 1 illustrates this operation for a simple network.

To enhance the reliability of link-level packet delivery, packet retransmissions may be activated in the network layer of the generator or relay nodes. In an advertising event, each packet may get retransmitted several times; this number is called *network transmit count* (N_{NTC}) and *relay retransmit*

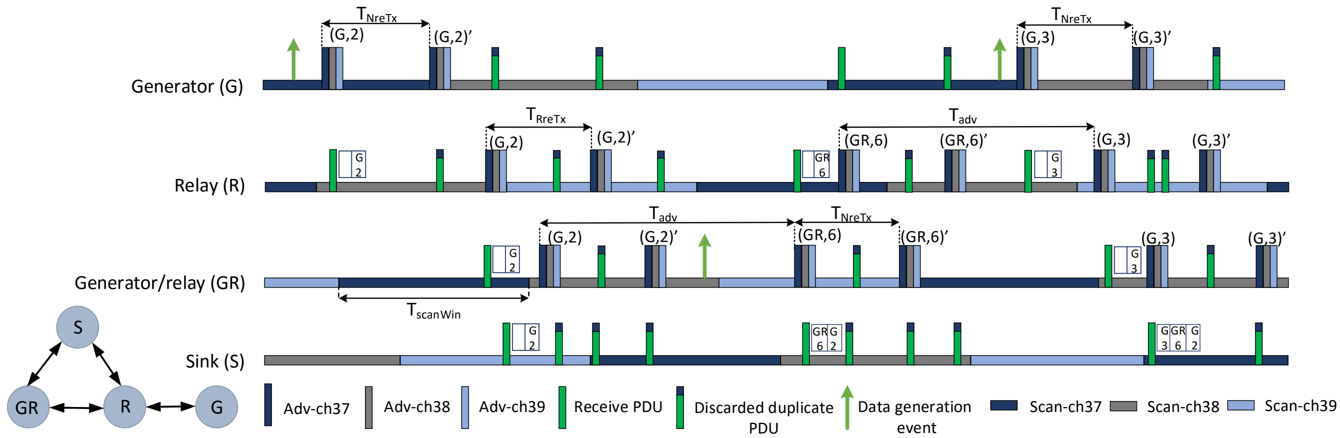


Fig. 1: Illustration of PDU transmission and reception in a BM network ($T_{scanWin} = T_{scanInt} = 40ms$, $T_{adv} = 60ms$, Ntis=Rris=1 (i.e., $20ms \leq T_{NreTx}, T_{RreTx} \leq 30ms$), $N_{NTC} = N_{RRC} = 1$.

count(N_{RRC}) in generator and relay nodes, respectively [15]. Such retransmissions are separated in time by network retransmit interval (T_{NreTx}) and relay retransmit interval (T_{RreTx}) that are specified by two parameters, Network transmit interval steps ($Ntis$) and Relay retransmit interval steps $Rris$, according to Eqn. 1 and Eqn. 2 [4].

$$T_{NreTx} = (Ntis + 1) \times 10ms + rand10 \quad (1)$$

$$T_{RreTx} = (Rris + 1) \times 10ms + rand10 \quad (2)$$

where $rand10$ is a random value in range $[0 \quad 10ms]$.

We further illustrate packet transmissions in a simple BM network in Fig. 1. The topological graph of the network is shown in the bottom left corner of the figure. It is composed of four nodes, a packet generator node (G), a relay node (R), a packet generator and relay node (GR), and a sink (destination) node (S). The used advertising interval is $T_{adv} = 60ms$, the scan window and scan interval are both $T_{scanWin} = T_{scanInt} = 40ms$, one retransmission is enabled for generator and relay nodes, and retransmissions are separated in time by $20ms \leq T_{NreTx}, T_{RreTx} \leq 30ms$ (Ntis=Rris=1). For the relay nodes, the content of their buffer is shown once a packet is added. When a packet gets transmitted by an advertising node in three advertising channels, the pair of (src, seq) on top gives the source node of the packet and the generator sequence number respectively (e.g., (G,2) is a packet generated by node G with sequence number 2).

When node G generates packet (G,2), it transmits the packet in the three advertising channels. Node R is scanning in channel 38 at that time so the packet is received by R in channel 38, and is added to its buffer. Node G retransmits this packet; it is received by R again, but is discarded since it is recognized as a duplicate. We assumed all links are perfect in this example, thus no packet drops. Node R forwards packet (G,2) which is received and processed by nodes GR and S, and received and ignored by G (duplicate). The retransmission of this packet by node R will be discarded by all nodes since it is a duplicate for all. The procedure continues for other packets generated by nodes G and GR in this example.

III. Related Work

Bluetooth SIG released Bluetooth Mesh (BM) in 2017. Since then, researchers have investigated the capacity and performance of this new promising technology in different circumstances and for various applications. Different methods such as experimental evaluation, statistical modeling, and computer simulations are used to evaluate these networks with respect to a variety of performance metrics such as latency, end-to-end Packet Delivery Ratio (PDR), and energy consumption. Also, trade-offs made by different settings of the BM network parameters in terms of various performance metrics are investigated. This section gives a brief overview of such investigations to date.

In [7] different BM networks with diverse configurations are investigated. This paper provides several diagrams showing the effects of configuration parameters on network performance. In particular, it investigates the following four aspects in a BM network as 1) configuration of scanning and advertising events, 2) interference from other wireless technologies (e.g., WiFi), 3) scalability, and 4) importance of random intervals between two consecutive transmissions in different advertising channels [7]. This paper expresses that the used flooding mechanism leads to collisions resulting in limitations for the scalability of the network. Also, it concludes that a smaller value for inter-PDU time reduces the congestion and thus improves end-to-end latency and packet success rate. This work, however, lacks investigation of the impact of configuration parameters such as heartbeat transmission interval and advertising interval with different network types and conditions.

In [9], the packet delivery performance of BM networks is experimentally evaluated. Several experiments are performed by placing BM nodes in an office environment. 29 fixed nodes, 3 mobile nodes, and one base station are deployed. PDR and burst drop for fixed and mobile nodes with different average hop distances are evaluated. It expresses the importance of location and the number of relay nodes in the BM network's performance. Also, the experiments show that saturated relay nodes have an inappropriate effect on

PDR, and suggest that the BM technology is suitable for low generation rate applications mainly because of the inefficient flooding mechanism. This paper does not evaluate latency as a performance metric and does not investigate the impact of parameters such as scan window, packet repetition, advertising interval, and interference in the performed experiments are fixed. A proper setting of such parameters may lead to different results revealing a broader scope for the application of the technology.

In [8] BM network performance is evaluated using statistical as well as experimental approaches. In the experimental evaluation, 22 nRF52832 nodes are placed in a $40m \times 25m$ area. In the statistical approach, Round Trip Time is calculated theoretically based on the number of hops. The impacts of some conditions in BM networks such as network density, hop distance, and back-off period are explained. However, this work does not evaluate reliability metrics like PDR and burst packet loss.

One of the main barriers against a thorough study of the performance of this rather new technology is the lack of a full network simulator that implements the exact behavior of the nodes in a mesh network. This paper first develops such a BM network simulator and then uses it for an extensive investigation of the configuration parameters in a variety of network conditions. Effective parameters such as repetition in relay and generator node, advertising interval, and scan window are tested. Also, trade-offs made by these parameters in networks with diverse configurations (topologies and packet generation rates) and different heartbeat intervals are explored. As an important part, BM networks with node mobility and the role of heartbeat interval in the performance of such networks are studied in this work.

IV. BMSim: the BM network Simulator

The aim here is to design and develop a simulation framework for BM networks, which is lacking for extensive study of the performance of such networks in various conditions. Our main target specifications for such a BM simulator are as follows. First, the simulator needs to be accurate enough that can be trusted for performance evaluation of BM networks. Second, low level details of packet transmission and networking procedures (both medium access and network layers) are truly modeled so that various performance metrics can be estimated. Third, the simulator is modular and flexible so that extensions for the inclusion of any channel/radio/mobility/interference models are possible and straightforward. As the last and very important specification, it is possible to configure the network and its parameters during simulation so that dynamic networks with run-time configurations mechanisms can be simulated. Considering these objectives in mind, we designed and developed BMSim, an open-source and publicly available event-driven Bluetooth mesh network simulator. Python is used as the programming language for the implementation of the simulator. In this section, we first introduce the architecture of the simulator and then discuss its core operation and user interfaces.

Fig. 2 shows the general architecture of BMSim and its inner components. A simulation starts by receiving some high-level network specifications from the user. The *Initializer* module prepares the first snapshot of the node deployment in the specified simulation area according to user wishes. Then the *Updater* module runs to prepare the required inputs for the simulator engine. The updater module runs every T_{update} seconds to support network dynamism as well as run-time configuration changes. The *BM simulation engine* is a discrete event simulator, for which the events and their timings are made according to the BLE and BM standard protocols. This engine continues making and processing events until the requested simulation time (T_{sim}) is reached. The *Logger* module creates the required output files while the simulation engine is running, and finally calculates and reports various performance indicators of the simulated BM network.

A. Initializer Module

The initialize module is the entry point user interface for the simulator by which the user's initial settings are received and compiled into the simulation framework. It gets the number of nodes in the network (network size, N) and the dimensions of the area the nodes should be deployed in. Also, the simulation time (T_{sim}) is the time frame that the user expects the network to be simulated. The initializer distributes N instances of the node model all over the simulation area based on the topology parameters given by the user; it may be either a uniformly random distribution, a regular grid structure, or specifying the x-y coordination of all nodes. The last case is especially used when several simulation runs are expected to be simulated starting from the same deployment. Anyways, the initializer only determines the initial locations of nodes, which remain fixed all over the simulation time if no node mobility is requested (static networks). In the case of having node mobility, the mobility model block of the network updater module determines the new locations of the mobile nodes at each simulation step.

There are a number of parameters related to different models used in the next components of the simulator that are received from the user by the initializer module and are directly passed to the corresponding blocks. Parameters of the mobility/channel/radio/interference models, BM stack initial configuration and node types, and the level of expected logs are other inputs of the initializer module. Note that since the simulator is expected to be easily configurable by plugging in/out various models, the exact list of input parameters will differ. For instance, for link extraction, the user can simply specify the communication range of nodes or the Packet Reception Ratio (PRR) of links between each pair of nodes to have a high level modeling of channel behavior without simulating details of channel and radio. However, to have a more precise simulation, one can use a path-loss model, for instance, for which several parameters such as transmission power of nodes, receiver sensitivity, and channel path-loss exponent need to be set. The same applies to other models such as mobility and interference models.

B. Network Updater Module

This module is essential for supporting dynamism as well as the possibility of having run-time adaptations and

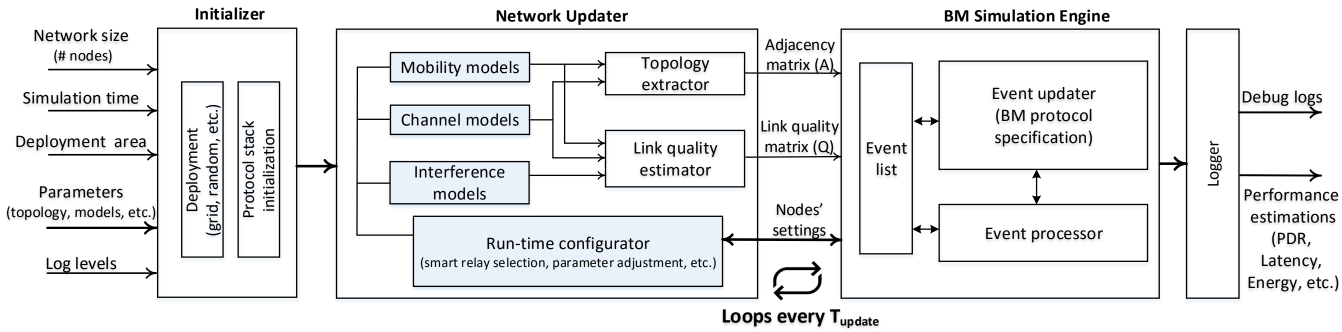


Fig. 2: The architecture of the developed Bluetooth mesh event-driven simulator (BMSim)

(re)configurations. The procedures in this module run every T_{update} to update the network. It is obvious that using lower time intervals for updates results in more smooth changes (e.g., smoother mobility patterns), but leads to higher execution time. On the other hand, to have faster simulation, longer intervals can be used especially when the dynamism level is low or parameter changes are not very frequent. There are two categories of outputs from the network updater module to the simulation engine. The first category of outputs is related to the network topology and link quality, namely the adjacency ($A_{N\times N}$) and link quality ($Q_{N\times N}$) matrices. Each entry $a_{i,j} \in A$ is a binary value that determines if there is a link between nodes i and j while $q_{i,j} \in Q$ gives the probability of successful packet delivery over that link. The topology extractor and link quality estimator blocks receive inputs from the mobility (nodes' positions), channel, and interference models and produce matrices A and Q for the next simulation step.

The other set of outputs generated by the network updater module is the BM protocol parameter settings of each individual node in the network. The settings for a node include the node's features (packet generator, relay, low-power, friend), packet generation interval (T_{gen}), advertising interval (T_{adv}), scan interval ($T_{scanInt}$), scan window ($T_{scanWin}$), network and relay retransmit count, Ntis, Rris, heartbeat message transmission interval (T_{hb}) of the destination nodes, data poll interval for low-power nodes, etc. The values of these parameters are set for the next simulation step based on the adaptation mechanisms that are implemented within the run-time configurator block of the network updater. As an example, determining a subset of nodes in a BM network that are best candidates for being relay is of paramount importance for achieving the required network performance [16]. Relay nodes should be in locations such that isolated nodes are minimized. A large number of relay nodes leads to unnecessary traffic and thus collisions, degrading the network performance [17]. On the other hand, the low number of relays affects the functionality and robustness of the flooding mechanism for data delivery. Thus, some intelligence can be used here to select the best nodes for the relay role; it may be design-time for static networks or run-time for dynamic networks. The current version of BMSim is fortified with a design-time relay selection algorithm based on the graph betweenness and closeness centrality algorithm [18]. Like the other blocks, this can be replaced or extended with other mechanisms for relay selection or adaptations of the other parameters.

C. BM Simulation Engine

The core of the BM simulator is a discrete event simulator that is composed of three blocks. The event updater block runs BM protocol specification, generates events (event id and expiry time), and adds them to the event list. The event processor block controls the simulation time and allows it to elapse only if all events at the current simulation time are processed. The event processor picks an event from the top of the list (the event with the earliest expiry time) and processes it, which may lead to the generation of more events (performed in line with the event updater block). Some of the defined and processed events are data packet generation event (made periodically every T_{gen} by each source node), heartbeat message generation event (made periodically every T_{hb} by destination nodes), relay event (every T_{adv} by each relay node if it has data packet in its buffer), Advertising37, Advertising38, Advertising39, Scan37, Scan38, Scan39, and channel switch event.

D. Logger Module

To closely study the behavior of BM networks and for post-processing toward performance extraction, various log files may be generated during a simulation run. Since the simulator is open-source for the public, users can add logs for whatever parameters and wherever in the protocol execution they aim for. However, there are already some logs produced by the simulator to report important parameters that have the interest of researchers. Such logs have two modes, determined by the user, that control the level of details being logged. If the detailed log flag is on, a separate log file is produced per node that contains details of the behavior of the packet generator nodes and the relay nodes. In the generator nodes' log file, the source ID, packet destination ID, packet generation time, sequence number, and the type of packets (heartbeat or data main packet) are logged for each generated packet. In the relay nodes' log, the advertiser node, the source node, packet TTL and sequence number, advertising and receiving and generation time, type of packets, and the number of packets in the node's buffer are logged. This information help users track the packet's path across the network. If the detailed log flag is off, per node log files are not produced. Anyways, a general log file is generated in every simulation run that contains the required information for evaluating the performance metrics (i.e., PDR, latency, burst packet loss, and energy consumption). This log file includes the following items per delivered packet as source and destination node IDs, packet generation time, packet delivery time to destination, packet type, and packets sequence number. For calculating energy consumption, the time duration that each node has spent in each operation mode (transmission, reception, sleep, and switch) and the source node, is logged. Then energy consumption is calculated based on these timing logs as well as the power consumption profile of the used radio transceiver (provided by the user).

V. Experimental Verification of BMSim

BMSim development is based on the BM specification and is tested by performing many sample networks and tracing the packet exchange between the nodes using the detailed log files. However, to have a higher level of trust in the simulation framework, we verified the simulator by comparing its network performance reports with those of real measurements in an experimental setup. However, this is not a straightforward path since even if the simulator behavior is fully based on BM specification (which is what we aim to verify), there still may be deviations because of the accuracy level of the used models (e.g., the channel model). Also, it is tricky to have exactly the same topology and link behavior in both the simulations and real experiments. Despite these difficulties, we have taken a strategy for experimental verification of the simulator, which is described and reported in this section.

Fig. 3 shows the tabletop setting of the hardware and software setup used in the experiments. Nordic semiconductors PCA10056 development kit and PCA10059 dongles are used, which both include the nRF52840 chipset. SEGGER Embedded Studio and Nordic Semiconductor nRF Connect are used for programming PCA10056 and PCA10059 nodes, respectively. A smartphone with the nRF mesh Android App is used as the provisioner, to provision and configure the nodes [19]. The light switch client and the light switch server firmware are programmed on one PCA10056 kit and ten PCA10059 dongles, respectively. All ten light switch server nodes send packets to the light switch client node periodically every second. The relay feature in all server nodes is active with relay retransmit count=0 and Rris=1. The client node is the destination of all packets and is connected to a PC. The source ID, sequence number, and TTL of every arrived packet at this node are sent to the computer and stored in a log file by the J-link RTT viewer software. By post-processing of the log file, PDR and average TTL of the received packets for each server node are calculated.

For performing experiments, the nodes are deployed in different rooms on a floor of a dormitory building. Such deployment can represent a wide range of indoor applications such as smart home, building monitoring, and indoor smart lighting applications. Although the BM networks deployed for

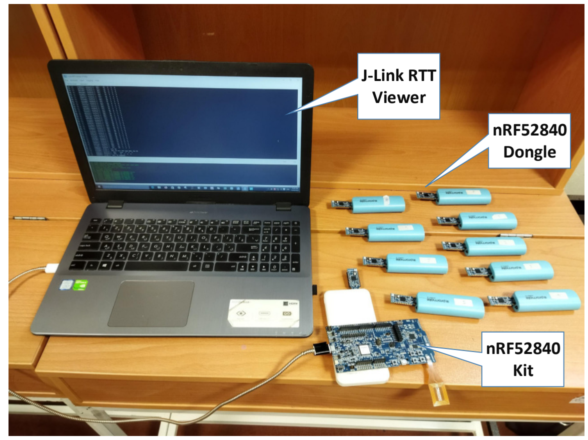


Fig. 3: The table-top hardware and software setup used in the simulator verification experiments.

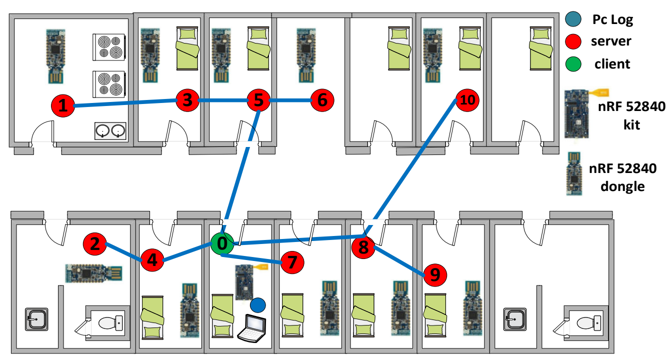


Fig. 4: Deployment of BM nodes in a dormitory environment for BMSim verification tests.

real-world applications are often of larger scale, the performed experiments have been affordable, and they are still can serve out goal in verifying the functionality of the simulator. The floor plan and nodes' placement are shown in Fig. 4. The conditions and dimensions of the deployment environment require multi-hop data delivery to the client node connected to the computer (sink node). Each experiment is conducted for around 10 minutes. After the experiment, the log file is processed to extract performance metrics and to figure out the real topology and connections in the experiment. This latter information is needed as an input to the simulator in order to have similar topologies in simulations and experiments. The blue lines between nodes in Fig. 4 Show such extracted links.

As notions of the network behavior, the end-to-end Packet Deliver Ratio (PDR) and the average TTL of the received packets for each server node are estimated in simulations and measured in the real experiments. The initial TTL for the network in both experiments is set to five. To have statistically more reliable simulation results (because of all random behaviors with the protocol), the simulation is done 10 times and the average of the results are considered. Fig. 5 presents PDR and average TTL for each server node estimated by BMSim and measured in real experiments. Also, the relative error for each node is shown on each bar. These results with

This article has been accepted for publication in IEEE Internet of Things Journal. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2023.3291551

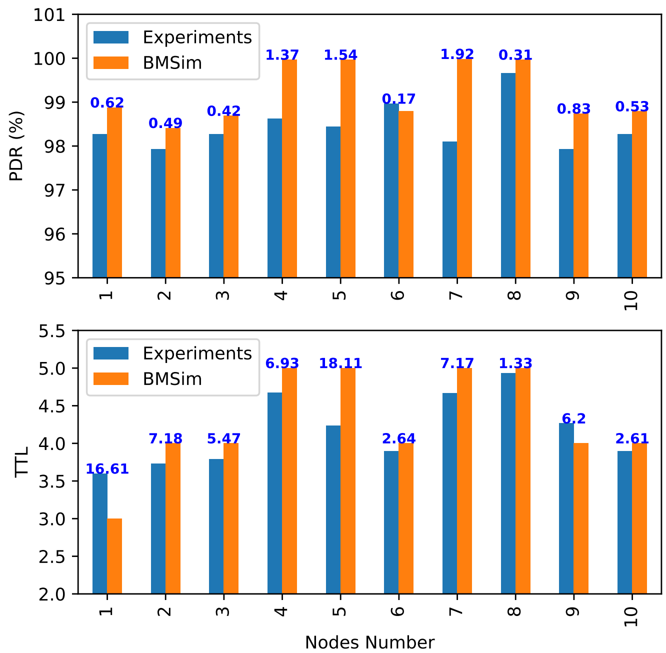


Fig. 5: The achieved performance results produced by BMSim and real experiments. The number on each bar is the relative error.

a maximum error of less than 2% in PDR estimation reveal the accuracy and true functionality of the simulator for the performed experiments. As mentioned, in the BMSim, the communication range of each node is a specific circle, and its radius is an integer number. However, in the real experiments, the communication pattern is unlikely to be a unit disk. Our measurements show that the communication range for the nodes in the test environment varies from 10 to 15 meters. Therefore, in the real experiment, each nodes' neighborhood differs specially and temporally, and is not fixed. It leads to different TTL values over time and for different source nodes; the average values are shown per node in Fig. 5. Also, there are several channels and RF effects that can still cause deviations between the two result sets, explaining even the minor errors in the estimated PDR and the measured values.

VI. BM Performance Study

We use the developed BM simulator to extensively study the performance of BM networks under various conditions and parameter settings. As discussed in Section II, there are a number of configuration parameters for each node in a BM network, a proper setting of which is important for network performance. Such investigations reveal the true capacity of BM networks for various IoT applications. In particular, we perform simulations to study the network performance correlations with five parameters or network settings as advertising interval (T_{adv}), scan window ($T_{scanWin}$), packet retransmissions (N_{NTC} and N_{RRC}), data packet generation intervals (T_{gen}), and heartbeat message interval (T_{hb}) in case of having node mobility. In this section, we first discuss the general setup of the performed simulations. Then, the settings for each simulation scenario are described followed by its results' presentation and analysis.

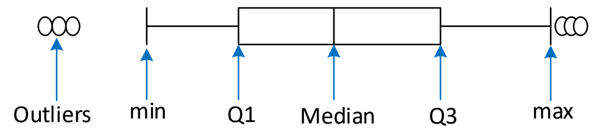


Fig. 6: Characteristics of the spread of data points visualized by a box plot

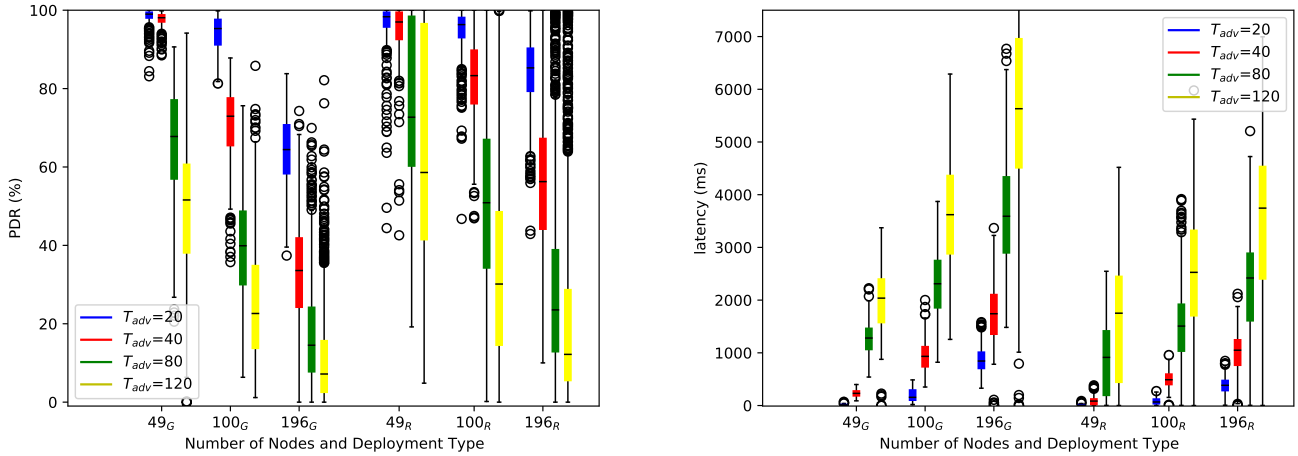
A. Simulation Setups

The simulation setups for the performance study of the BM technology are planned in such a way to be general enough to include a wide range of real-world applications. For that, various network scales, deployment types, and data generation rates are tested. Three BM networks consisting of 49, 100, and 196 nodes have been tested. Such network scales can, for instance, represent different environments such as a mall environment, a building monitoring application, and environment monitoring applications, respectively. Each setup is tested with the grid as well as random deployments. In grid networks, nodes are located in grid positions (7×7, 10×10, and 14×14) with a horizontal and vertical distance of 9 meters between grid points. In randomly deployed networks, nodes are uniformly randomly distributed all over the deployment area of square shape with dimensions of $40m$, $60m$, and $80m$ for the three network scales. The randomly deployed networks are checked for being connected. The communication range of all nodes is set to 10 meters leading to an average node degree of 4 in grid networks. For the sake of simplicity and w.l.o.g., all links are supposed to have 100% packet reception ratio (except if otherwise stated), thus no packet drop is experienced due to channel effects (e.g., multi-path fading, external interference) if nodes are in the communication range of each other.

Using the closeness centrality algorithm [18], One of the nodes in each setup is selected to be the sink of data packets. All other nodes have relay feature enabled so there is no isolated node. The buffer size of each node is set to 6, meaning that the relay nodes can store up to 6 packets from other nodes for getting relayed. However, not all nodes are sources of data packets; two third of the nodes generate data packets every T_{gen}.

We have a set of default parameter values that are common between all simulation setups. For investigating the impact of each parameter, several values in a certain range for that parameter are tried while other parameters are set to their default settings. The default parameter values are as follows: $T_{adv} = 20ms$, $T_{scanWin} = T_{scanInt} = 30ms$, $T_{gen} = 1000ms$, $T_{hb} = 4000ms$, with no packet retransmissions.

Simulation time (T_{sim}) for all setups is 10 minutes. Each setup is repeated 10 times with different seeds for the random number generator to have statistically more reliable results (deployments remain the same over repetitions). End-to-end Packet Delivery Ratio (PDR) and latency of delivered data packets are considered as performance metrics. The results (PDR and latency of different nodes and overall 10 repetitions of a simulation setup) are shown as box plots to summarize important characteristics of the spread of the results. Fig. 6



(a) End-to-end packet delivery ratio (PDR)

(b) Latency of delivered data packets

Fig. 7: PDR and latency in random and grid networks with different advertising interval settings

depicts the information presented by a box plot. The middle box contains 50% of all data points which are the two middle quartiles (from Q1 to Q3) together with the median of the data points. Two whiskers (upper and lower whiskers) are located at 1.5 times the length of the box from the top and bottom of the box, respectively. The min and max values shown on the plot are the minimum and maximum of data points within the lower and upper whiskers. Data points outside the whiskers are considered outliers and are shown as black circles. On the horizontal axis of the plots, the combination of network size and deployment type (subscript of G for grid and R for random deployment) are specified.

B. Advertising Interval

Four values of 20, 40, 80, and 120 ms are exercised for the advertising interval. All other parameters are set to their default aforementioned values. Fig. 7 present the results out of BMSim. The first observation here is that increasing advertising interval leads to degraded performance (PDR and latency) in both grid and random networks. The reason is that when relay nodes send packets with longer advertising intervals, a higher delay is experienced in each hop toward the sink node resulting in higher end-to-end latency. Moreover, data packets stay in the relay nodes' buffer for a longer time which causes buffer overflow at some point resulting in packet drop due to buffer capacity. This happens because we are testing networks with many generator nodes that produce data packets with a rather high rate ($T_{gen} = 1000ms$). The network used in a health monitoring application [9] is an example of BM networks with such high data generation rate. Later we discuss that quite better performance is observed when the data packet generation interval increases for the same advertising interval setting. In any case and in every setup, the nodes that are close to the sink node experience good PDR and latency. This is the reason for having such distributed results even when the mean performance is not good.

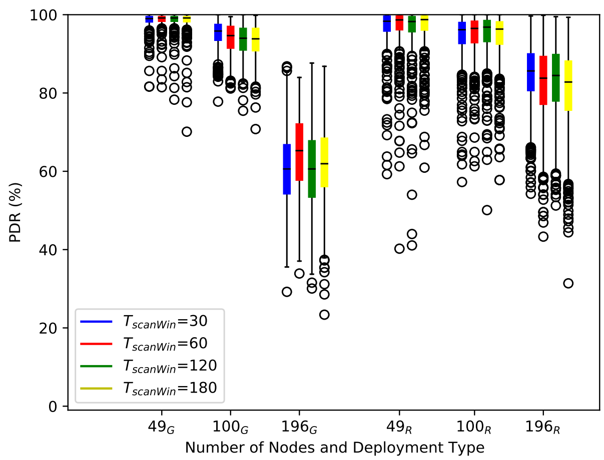
The node degree in the grid networks is overall lower than that in the random networks (we intentionally used a smaller area for random networks to decrease the chance of isolated nodes). Thus, the hop distance between the source nodes and the sink in the grid networks is on average higher compared to that of the random networks. This seems to be the main reason that the random networks have shown better performance compared to the grid networks.

C. Scan Window

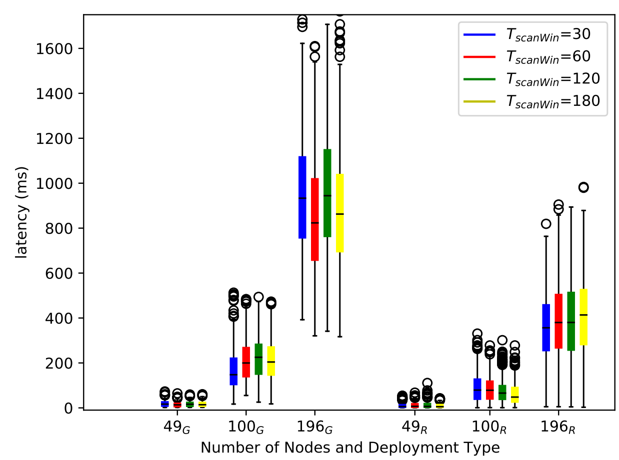
A scanning BM node listens to each advertising channel for a scan window time ($T_{scanWin}$) and then switches to the next channel every scan interval time. In many BM networks including our simulation setups, these two parameters are set equal to each other meaning that the scanning nodes have no duty cycling in their channel scan task. Here we test four different values for the scan window as 30, 60, 120, and 180 ms. All other parameters have their default values. Fig. 8 shows the performance results. As expected, the observation is that the setting of the scan window does not have a clear impact on the performance. It means that scanning each of the three advertising channels for a long time or quickly switching to different channels does not have a visible effect on PDR and latency. The main application of this parameter is then the possibility of introducing a duty cycle for the scanner nodes in order to decrease their radio energy consumption. In that case, the correlation between the scan window, scan interval, and retransmission settings (number and interval) will be important to achieve a proper performance level and avoid packet losses that can lead to degradation of packet delivery performance.

D. Re-transmissions

The BM protocol allows packet retransmissions by the source nodes as well as the relay nodes to compensate for packet drops caused by various channel effects and collisions. As discussed in Section II, the number of retransmissions

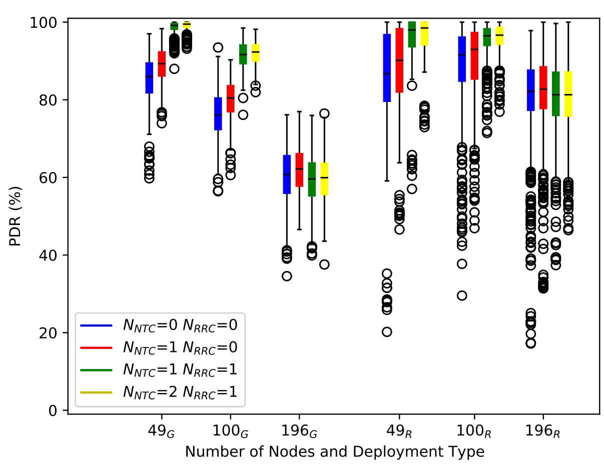


(a) End-to-end packet delivery ratio (PDR)

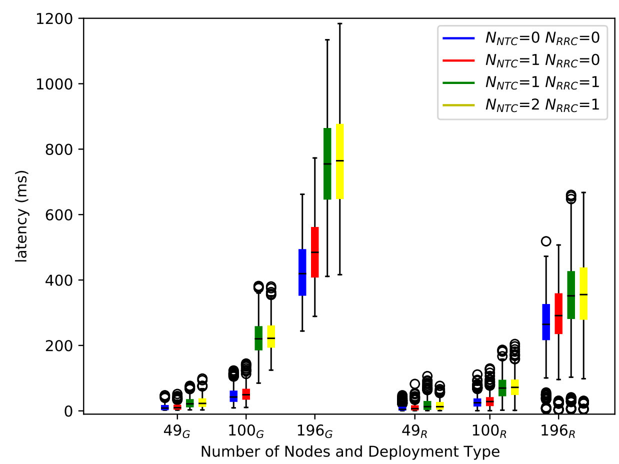


(b) Latency of delivered data packets

Fig. 8: PDR and latency in random and grid networks with different scan window settings configuration



(a) End-to-end packet delivery ratio (PDR)



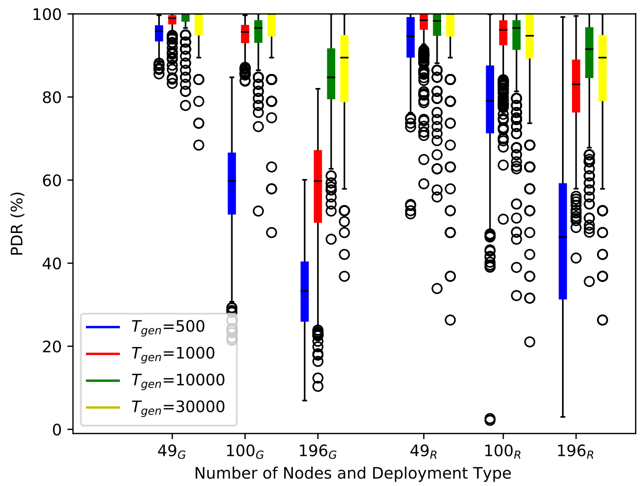
(b) Latency of delivered data packets

Fig. 9: PDR and latency in random and grid networks with different re-transmission settings. The links are supposed to have a non-ideal quality with a packet reception ratio above 30%.

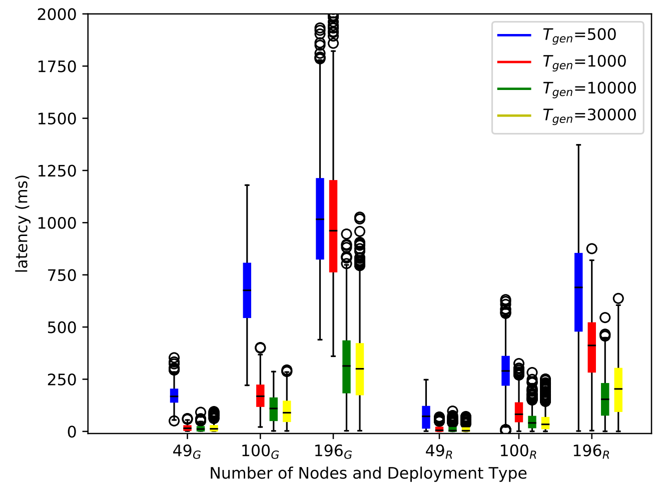
of each packet and the time distance between retransmission events can be set separately for generator and relay nodes. In this part of our study, we aim at investigating the impact of the number of retransmissions on performance. In these experiments Ntis= Rris=1, which means the time distance between consequent retransmissions of a packet by generator and relay nodes is between $20ms$ and $30ms$. The advertising interval is set as $T_{adv} = 120ms$, and other parameters are set as their default values. We try setups having no retransmissions ($N_{NTC} = N_{RRC} = 0$), one retransmission only for generators ($N_{NTC} = 1, N_{RRC} = 0$), one retransmission for generator and relay nodes ($N_{NTC} = N_{RRC} = 1$), and two retransmissions for generators and one for relays ($N_{NTC} = 2, N_{RRC} = 1$).

We first tried the six network deployments with perfect wireless links (PRR=100%). The result of this set of simulations reveals that for BM networks with perfect links, retransmissions only lead to degradation of the performance. It is because retransmissions, especially by relay nodes, increase the network load and lead to collisions. The degradation of the performance for large and dense networks is substantial. It means that the use of retransmissions should be carefully decided to achieve better performance only for links that are in trouble with respect to their connectivity.

To truly investigate the positive impact that retransmissions may have on BM performance, we ran another set of simulations this time with non-ideal wireless links. Here, we considered a random value in the range of 30% to 100% for each link in the network. Fig. 9 presents the results of these experiments, which clearly show the positive influence of the generator and relay retransmissions on PDR and latency. Note that in these simulations, all nodes get equal retransmission



(a) End-to-end packet delivery ratio (PDR)



(b) Latency of delivered data packets

Fig. 10: PDR and latency in random and grid networks with different data packet generation intervals

settings irrespective of their link quality. It means that the achieved gain for nodes with poor links may have faded by unnecessary retransmissions over other nodes (more collisions). This reveals the necessity of proper setting of the parameters per node based on their connection status. Another observation is that retransmission by the generator nodes has quite less negative impact and is able to improve performance while this setting for relay nodes is sensitive and should be carefully set.

E. Packet Generation

The flooding mechanism used in the BM protocol together with limitations associated with the advertising mode (three transmissions in three channels for each packet and short advertising packets) raise concerns about the capacity limits of BM networks and the application scenarios for which this technology can be used. We try a number of different packet generation intervals to investigate this aspect of BM networks. Four generation intervals of 500, 1000, 10000, and 30000 ms are tested. Other parameters have default settings of our setups.

Fig. 10 shows performance results for various packet generation intervals. A low generation interval of half a second leads to very low performance, especially in larger networks. This is again due to the high traffic and packet arrival rate at each relay node that causes the relay buffer to overflow. More collisions due to high traffic and dense networks (especially random networks) are another reason for packet losses. Note that increasing the size of the buffer does not help in this case; it is tested by extensive simulations. The reason is that the buffer will eventually get full since the packet arrival rate is higher than the packet transmission budget. This is while the minimum allowed advertising interval is used in these simulations which provides a higher transmission budget for the relay nodes. Of course, in networks with fewer generator nodes, more traffic from a generator node can be relayed and thus the generation interval can be shorter. For high generation intervals (10 and 30 seconds), no meaningful performance difference is observed since the generation is sparse enough that the tested BM networks are able to pass the generated traffic.

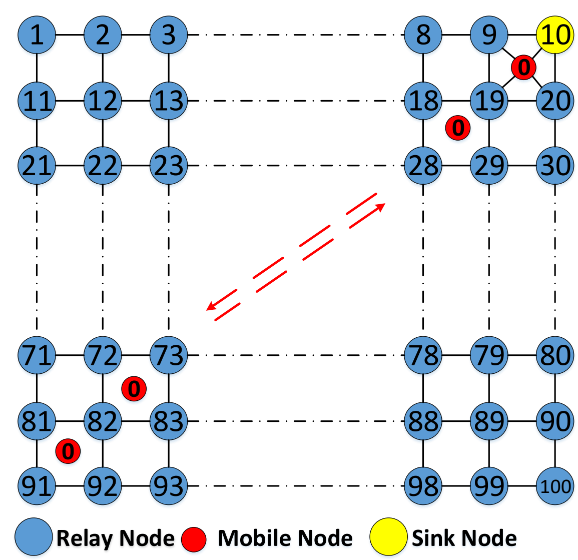
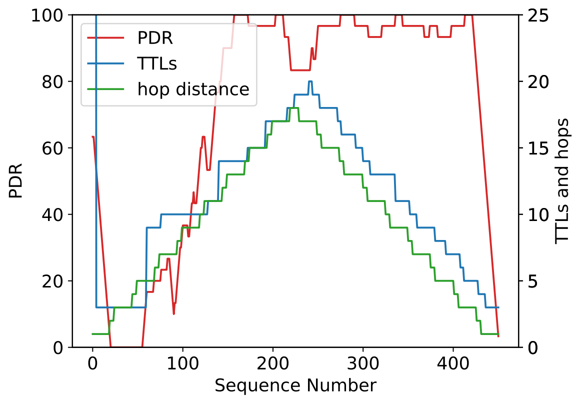


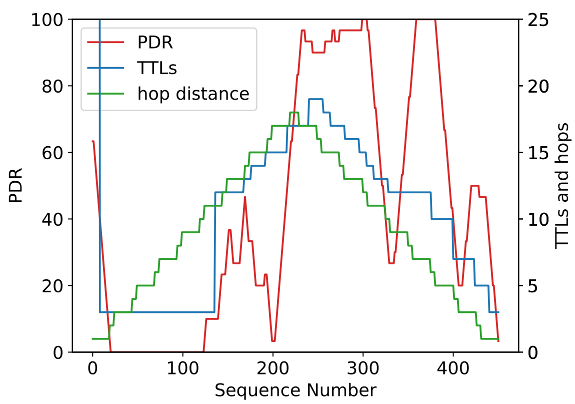
Fig. 11: Setup of the simulations with a mobile node

F. Node Mobility and Heartbeat Messages

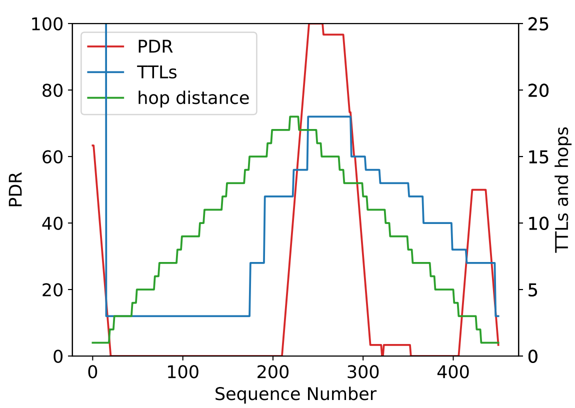
As discussed, source nodes in BM networks may use the estimated hop distance to their intended destinations using the heartbeat messages that are periodically generated by the destination nodes and are flooded throughout the network. The TTL value for each generated packet is then set based on this estimation. The heartbeat message generation interval (T_{hb}) should be set based on the expected level of dynamism in the network. For mostly static networks, the hop distances will remain the same for long periods of time thus there is no



(a) Heartbeat interval = 4000 ms



(b) Heartbeat interval = 8000 ms



(c) Heartbeat interval = 16000 ms

Fig. 12: Hop distance, used TTL values, and the achieved PDR of a mobile node with different heartbeat intervals

need for frequent heartbeat messages (joining nodes should be anyway considered). On the other hand, in networks with a form of node mobility, the distance between source nodes and destinations may change over time, which necessitates more frequent heartbeat messages to update the source nodes about their distance. In this part of our study, we set up a network with a mobile node and try different values of heartbeat (T_{hb}) as 4, 8, and 16 seconds to investigate its impact on the network performance. Thanks to the BMSim architecture (the network updated module), run-time network dynamism and configuration can be simulated with any intended patterns.

A grid network consisting of 100 nodes and one mobile node is considered in this setup. Fig. 11 depicts this network. The sink node (in yellow with ID=10) is located in the upper right corner of the grid. The mobile node (in red with ID=0) is initially located next to the sink; it moves with a speed of 0.5 meters per second straight toward the bottom right corner and returns to its initial position. The movement path is shown in Fig. 11.

Fig. 12 presents the PDR and used TTL for the mobile node. As a reference, the real hop distance of the mobile node to the sink node is also depicted; it is calculated based on the node's speed and the distance between nodes. The horizontal axis of the plots is the sequence number of generated packets by the mobile node. The shown PDR values over time are based on recent packet delivery in the past thirty sequence numbers.

Fig. 12a shows that $T_{hb} = 4000ms$ is performing well for the tested mobility. The used TTL by the mobile node has been able to quickly enough follow the real hop distance. PDR does not experience major drops when the node goes farther away from the sink node. This is while for the other two plots that use longer heartbeat intervals, the used TTL had a delay in following the hop distance leading to sudden drops in PDR, sometimes even until disconnection for a period of time. This result reveals the importance of the heartbeat message setting according to the network dynamism. Note that unnecessarily frequent generation of heartbeat messages can lead to high traffic overhead in the network that will result in degradation of the network performance.

VII. Conclusion

Bluetooth mesh is a networking technology for IoT, providing robust and scalable multi-hop end-to-end data delivery based on BLE technology. This paper presents an event-driven Bluetooth mesh network simulator (BMSim), which is open-source and publicly available online through https://github.com/BMSimulator/BMSim. The developed simulator is capable of simulating dynamic networks and run-time adaptive mechanisms and is extensible to include simulation models for various channel, radio, and protocol models as plugins. BMSim is verified using experiments with Nordic radio modules. As another contribution of this paper, the simulator is used to investigate the impact of the settings of several parameters on the network performance and to realize the capacity and limits of the protocol. In particular, advertising interval, scan window, packet retransmission settings, packet generation interval, and heartbeat message generation rate are investigated.

The results of the performed study reveal the importance of proper configuration settings in fully utilizing the capabilities of the BM networks. Thus, future work in this domain is to develop design-time or distributed run-time configuration mechanisms for BM networks. For all such research, BMSim presented in this paper can be used as a tool for performance evaluation and comparison.

REFERENCES

[1] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad hoc networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[2] "Core Specifications | Bluetooth® Technology Website," *Bluetooth Low Energy*, pp. 1–3256, November 2020. [Online]. Available: https://www.bluetooth.com/specifications/bluetooth-core-specification

[3] J. Yin, Z. Yang, H. Cao, T. Liu, Z. Zhou, and C. Wu, "A survey on bluetooth 5.0 and mesh: New milestones of IoT," *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 3, pp. 1–29, 2019.

[4] D. Pérez-Díaz-De-Cerio, M. García-Lozano, A. V. Bardají, J.-L. Valenzuela *et al.*, "Bluetooth mesh analysis, issues, and challenges," *IEEE Access*, vol. 8, pp. 53 784–53 800, 2020.

[5] "Mesh Networking Specifications | Bluetooth® Technology Website," *Bluetooth Mesh*, pp. 1–333, November 2020. [Online]. Available: https://www.bluetooth.com/specifications/mesh-specifications/

[6] S. M. Darroudi and C. Gomez, "Bluetooth low energy mesh networks: A survey," *Sensors*, vol. 17, no. 7, p. 1467, 2017.

[7] R. Rondón, A. Mahmood, S. Grimaldi, and M. Gidlund, "Understanding the performance of bluetooth mesh: reliability, delay, and scalability analysis," *IEEE Internet of Things journal*, vol. 7, no. 3, pp. 2089–2101, 2019.

[8] M. Baert, J. Rossey, A. Shahid, and J. Hoebeke, "The bluetooth mesh standard: An overview and experimental evaluation," *Sensors*, vol. 18, no. 8, p. 2409, 2018.

[9] E. De Leon and M. Nabi, "An experimental performance evaluation of bluetooth mesh technology for monitoring applications," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.

[10] J. Tosi, F. Taffoni, M. Santacatterina, R. Sannino, and D. Formica, "Performance evaluation of bluetooth low energy: A systematic review," *Sensors*, vol. 17, no. 12, p. 2898, 2017.

[11] S. M. Darroudi, C. Gomez, and J. Crowcroft, "Bluetooth low energy mesh networks: A standards perspective," *IEEE Communications Magazine*, vol. 58, no. 4, pp. 95–101, 2020.

[12] D. Hortelano, T. Olivares, and M. C. Ruiz, "Reducing the energy consumption of the friendship mechanism in bluetooth mesh," *Computer Networks*, vol. 195, p. 108172, 2021.

[13] 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.

[14] D. Perez-Diaz-de Cerio, Á. Hernández-Solana, M. García-Lozano, A. V. Bardají, and J.-L. Valenzuela, "Speeding up bluetooth mesh," *IEEE Access*, vol. 9, pp. 93 267–93 284, 2021.

[15] 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.

[16] E. A. Hansen, M. H. Nielsen, D. E. Serup, R. J. Williams, T. K. Madsen, and R. Abildgren, "On relay selection approaches in bluetooth mesh networks," in *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*. IEEE, 2018, pp. 1–5.

[17] M. Reno, R. Rondón, L. L. Bello, G. Patti, A. Mahmood, A. Lombardo, and M. Gidlund, "Relay node selection in bluetooth mesh networks," in *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*. IEEE, 2020, pp. 175–180.

[18] 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.

[19] P. Pierleoni, A. Gentili, M. Mercuri, A. Belli, R. Garello, and L. Palma, "Performance improvement on reception confirmation messages in bluetooth mesh networks," *IEEE Internet of Things Journal*, 2021.

Zohreh Hosseinkhani received the B.Sc. and M.Sc. degrees both in computer engineering from the Isfahan University of Technology, Isfahan, Iran, in 2014 and 2016, respectively. She is currently working toward her Ph.D. in computer engineering at the Electrical and Computer Engineering Department of the Isfahan University of Technology, Isfahan, Iran. Her research interests include Internet of Things, wireless sensor networks, Bluetooth mesh, and machine learning.

Majid Nabi (Member, IEEE) received the B.Sc. degree in computer engineering from Isfahan University of Technology, Isfahan, Iran, in 2001, the M.Sc. degree in computer engineering from Tehran University, Tehran, Iran, in 2007, and the Ph.D. degree in electrical and computer engineering from Eindhoven University of Technology (TU/e), Eindhoven, The Netherlands, in 2013.

He is currently an Assistant Professor at the Department of Electrical Engineering, TU/e and Isfahan University of Technology. His research interests include efficient and reliable networked embedded systems, low-power wireless sensor networks, and Internet of Things.