Reliable Sensor Data Gathering with Bluetooth Mesh: An Experimental Study

Hasan Qaq[a,c], Omkar Kulkarni[b], Yuming Jiang[c]
[a]*Norsk Helsenett SF, Trondheim, Norway*
[b]*Nordic Semiconductor ASA, Trondheim, Norway*
[c]*NTNU, Norwegian University of Science and Technology, Trondheim, Norway*

***Abstract*—Bluetooth mesh is a recent addition to the IoT connectivity landscape. It provides a simple and efficient short-range wireless mesh networking solution. This paper evaluates the performance of a Bluetooth mesh-based IoT sensor data gathering network in a real-world office environment. It provides suggestions to improve the reliability of the data gathering process. Specifically, the effect of various Bluetooth mesh protocol-related and non-protocol-related parameters is evaluated. The protocol-related parameters explored in this paper include the Publish Retransmit Count (PRC), the Network Transmit Count (NTC), and the Relay Retransmit Count (RRC). The non-protocol-related parameters include the packet-sending randomization interval and the packet payload redundancy. Through extensive experiments, the results show that a high degree of reliability, 99%, can be achieved with a combined use of these parameters. These results provide insights and shed light on configuring a Bluetooth mesh network to achieve high reliability for sensor data gathering.**

I. INTRODUCTION

Several short-range low-power wireless communication technologies are available today for deploying Internet of Things (IoT) applications. Examples of these technologies include Zigbee, Thread, and Bluetooth Low Energy (BLE). The ultra-low power consumption of BLE [1] makes it a good option for power-limited devices. It has even demonstrated its capability to meet tight Quality-of-Service (QoS) requirements for single-hop real-time industrial IoT applications [1].

Bluetooth mesh (BTM) is a recent technology developed by the Bluetooth Special Interest Group (Bluetooth SIG) [2]. The Bluetooth Mesh Profile specification was released in July 2017. BTM is considered a new milestone technology for IoT [3] and is gaining momentum in various IoT application areas, such as the commercial lighting solutions market [4]. BTM allows many-to-many device communication over BLE radio. Using BTM, thousands of devices can communicate with one another to create a mesh network. This topology greatly enhances the communication range compared to the traditional peer-to-peer connection topology used with Bluetooth devices.

Functional and operational challenges exist in achieving reliable sensor data gathering with BTM, which may be critical for IoT applications. These challenges are related to resource limitations, e.g., energy supply [5], physical barriers, and radio noise in the environment of deployment. Additionally, the sensors are left unattended to do their job. An immediate consequence of these challenges, if not adequately addressed, is an unacceptable packet loss level. Accordingly, for a BTM network to achieve a high-reliability level for data collection the fundamental questions that need to be answered include where to put the sensor data collector or gateway, how to make use of relays, and what parameters may be tuned and how to tune them.

Very few studies have been conducted to evaluate the performance of BTM for *reliable sensor data gathering in real-world environments*. The most relevant ones are [6] and [7]. In [6], it is reported that a packet delivery ratio (PDR) as high as 99.9% can be achieved by enhanced configuration for network layer retransmission and randomization. However, the evaluation was conducted through simulation, and no specific information was provided regarding the simulated environment. In addition, how such enhanced configuration affects PDR is not detailed. The investigation in [7] was on a BTM test network consisting of up to 192 nodes deployed in an area of about 2230 m^2. However, its focus is on latency performance. Although it is reported that the PDR can reach 99%, no detailed information is provided in [7] about in what configurations it is achieved. Furthermore, how other related parameters may impact and be tuned to achieve high PDR is not discussed in [6] and [7].

This paper investigates how to configure a BTM network to achieve high reliability, e.g., 99%, for sensor data gathering. An extensive experimental study has been conducted on a BTM-based sensor data gathering test network for this investigation. The network comprises 100 immobile nodes distributed in a 1400 m^2 office area. The nodes are installed on the ceiling plates, roughly evenly distributed throughout the office floor to emulate the distribution of office lights in the area. One node works as a collector or gateway (GW), which collects data from all other sensor nodes. The sensors periodically generate data packets and send them to the gateway using BTM. The gateway sends the content of the received packets over an Ethernet network to a dedicated server for further analysis. The results are analyzed to investigate the effect of using relays, how different protocol-related and non-protocol-related parameters may affect reliability, and how a high-reliability level may be achieved by applying them jointly.

The rest of the paper is organized as follows. Section II introduces BTM. Section III presents the test setup and environment. The experimental results and discussions are presented in Section IV and Section V, respectively. Section VI concludes the paper.

II. BLUETOOTH MESH

BTM enables the creation of large-scale low-data-rate short-range wireless mesh network [2]. The mesh devices communicate directly with each other without the use of a centralized controller. A *node* in the mesh network is a device that can transmit and receive packets in the network. A BTM node can optionally support one or more additional features [2]: relay, proxy, low power, and friend, as illustrated in Figure 1. The relay node forwards packets to other nodes in the network. The proxy nodes enable communication between a non-mesh device and the BTM network. The friend node forwards packets to and from the Low Power Node (LPN) to the rest of the network.

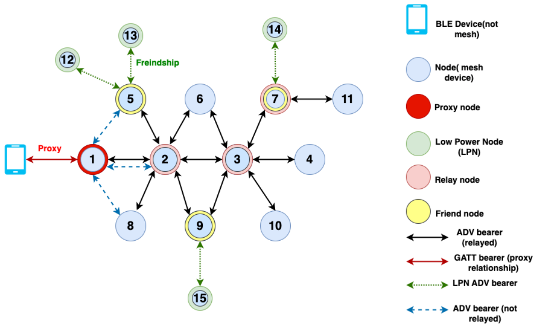


Fig. 1. Node types in a BTM network.

The BTM protocol has a layered architecture. It comprises Foundation Models, Access, Transport (Upper and Lower), Network, and Bearer layers. In BTM, the maximum size of the Access PDU (Protocol Data Unit) is 384 bytes, including a 4-byte Message Integrity Check (MIC) value. If the Access PDU size exceeds 15 bytes, such PDUs are split into multiple Lower Transport PDUs and transferred to the destination using the Segmentation And Reassembly (SAR) mechanism. The SAR mechanism involves sending repetitions of segment PDUs from the sender and sending acknowledgment PDUs from the receiver. If Access PDU can fit into a single Lower Transport PDU, it is sent to the destination without using the SAR mechanism. When Access PDU is unsegmented, the PDU transfer from source to destination is faster and generates lesser traffic. One Lower Transport PDU corresponds to one Network PDU.

To relay packets, BTM uses managed flooding. Managed flooding falls between (uncontrolled) flooding and routing. In managed flooding, the relay node broadcasts packets to all nodes within its radio range. When a packet is relayed, the Time-To-Live (TTL) value assigned to the packet gets decremented by one. A packet is only relayed when its TTL is greater than one. In this way, the maximum number of hops that a packet can traverse can be managed by the originator of the packet. Additionally, the relay nodes cache recently seen packets. If a received packet to be relayed already exists in the cache of a relay node, the packet is discarded by this relay node.

To provide additional reliability in a mesh network, the Bluetooth Mesh Profile [8] allows for packet repetitions at the application and the network layer. These repetitions of the BTM packets can be controlled by three parameters: PRC, NTC, and RRC. PRC controls the number of replicas of a packet published by a model where each replica has a unique sequence number. The NTC controls the number of transmissions of the Network PDU packets originating from the node, while the RRC controls the number of retransmissions of the Network PDU packets relayed by the relay node. The space between the retransmissions for PRC, NTC, and RRC depends on a specified time interval in milliseconds and can be configured separately for each setting. These settings are called Publish Retransmit Interval (PRI), Network Transmit Interval (NTI), and Relay Retransmit Interval (RRI). The bearer layer applies 10 ms randomization to each packet before sending a packet out on the radio. The PRI setting is configured at runtime, while NTI and RRI can be configured with default values. With the default values of NTI and RRI, each new packet is sent at an average of 25 ms in nRF Connect SDK´s mesh implementation [9]. It is important to note that the ability of the relays to relay many incoming packets depends on the amount of buffers available on the relays and the behavior of relays while dispatching several received packets. The particular version of the implementation, used for testing, enqueues the packets to be sent or relayed and sends them one by one. If the originator (or relay) is configured to have NTC (or RRC) of more than zero, then the next packet in the queue is not transmitted until all retransmissions of the previous packet have been completed. This behavior affects the number of incoming messages that the relay can handle before dropping them due to a lack of buffers.

III. THE BLUETOOTH MESH TEST NETWORK

The topology of the Bluetooth mesh test network is shown in Fig. 2. Hereafter, this network is referred to as the test network. It consists of 100 immobile nodes installed in an area of 1400 m^2 in the office building. Each node is a custom-made circuit board consisting of Bluetooth and Power Over Ethernet (POE) functionality, hereafter referred to as POE development kit (POE DK). It has an nRF52840 chip providing Bluetooth functionality and a Wiznet W5500 chip providing Ethernet functionality. The kits are connected to three stacked POE switches connected to a dedicated computer, hereafter referred to as the remote PC. The test network can be accessed and controlled by using the remote PC.

The test environment has many walls made of concrete or other opaque objects. Many employees work in the testing environment. The test environment also contains Wi-Fi access points and other Bluetooth devices operating on the floor. In Fig. 2, the color of each BTM node indicates the detected Bluetooth traffic on the advertisement channels around this node. It is worth highlighting that the test network does not generate this detected traffic. Higher such traffic implies a higher activity level of other Bluetooth devices around, which will compete or interfere with nodes in the test network using

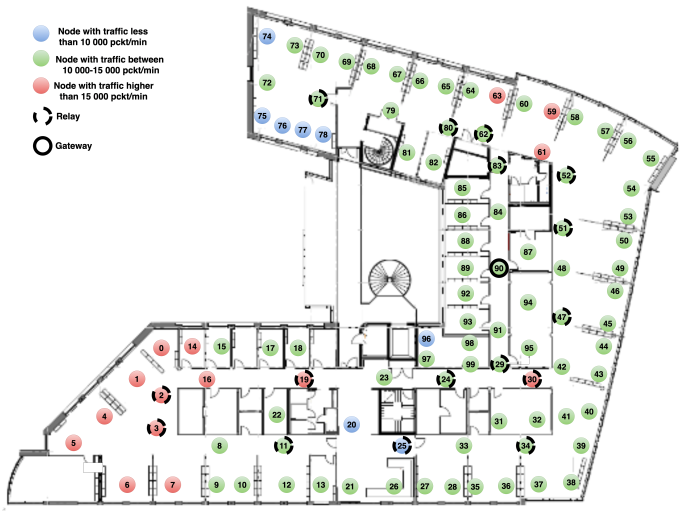


Fig. 2. The sensors, relays, and gateway in the test network.

the Bluetooth radio channel. In the test network, through preliminary investigation, the GW has been chosen to be the node that is more centrally located (see Fig. 2) and receives a higher number of packets from a higher number of sensors, compared to other nodes in the network. Each sensor in the test network is within a radio range of at least one relay.

The test procedure works as follows: (i) The remote PC sends a command over the Ethernet to all the nodes in the test network. The command specifies which node is the gateway (GW), the sending interval (implying how often the sensor nodes send a packet to the GW), and the test duration. (ii) Upon receiving this command, the sensors simultaneously start sending unacknowledged and unsegmented BTM packets to the GW, emulating, for example, sensor measurements. These BTM packets have an available payload of 11 bytes, including bytes that are used for the model message opcode. Of these 11 bytes, a 2-byte field is used as a unique packet tag, and a 2-byte field is used to simulate a random value measured by the sensor, and the remaining bytes are left unused. (iii) When the GW receives the BTM packets from the sensors, it logs the content of each packet and the sender's address. (iv) When the test procedure is over, each sensor logs the number of the BTM packets sent to the GW. (v) The test script on the remote PC parses these logs to identify how many packets were sent by each sensor. For each sensor, the test script calculates the ratio between the number of logged packets from the GW and the number of logged packets from the sensors. This ratio is called the packet delivery ratio (PDR), i.e., reliability.

The tests are conducted in a way where different parameters are used, firstly in isolation from other parameters and then in combination with other parameters, to see the effect of each parameter on the reliability. The parameters, which have a positive effect on reliability when they are used in isolation, are tested later in combinations. The parameters used for testing are PRC, NTC, RRC, sending interval, packet-sending randomization, and payload redundancy.

The transmission power for each node in the network is fixed to 0 dBm, the sending interval is 5 sec, and the test duration is 10 min for each experimental run. This results in about 120 packets being sent from each sensor. The rest of the nodes' configuration options, except for PRC, NTC, and RRC, follow the Bluetooth Mesh implementation in nRF Connect SDK [9]. This includes the NTI and the RRI, which are left to the default value of 20 ms.

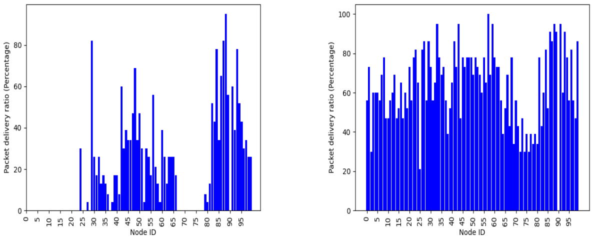
IV. Experimental Results

This section investigates how to achieve reliable data gathering in BTM networks. The test network experiments' results are presented and discussed, focusing on the effect of relays, protocol, and non-protocol-related parameters.

A. Effect of relays

The test network has 100 immobile nodes distributed on 1400 m^2 with concrete walls blocking line-of-sight. Together with other environmental and random factors, many nodes cannot communicate with each other in the test network reliably. For example, Fig. 3(a) shows the packet delivery ratio (PDR) from all the nodes to the GW in the absence of relays. As it is clear from the figure, nearly one-third of the nodes cannot send their packets to the GW directly. Additionally, their PDR to the GW is below 40% for more than half of the nodes. These are unacceptable reliability levels. To address this problem, some number of nodes are configured as relays.

The relays are chosen starting with two relays around the GW from two opposite directions. Then, the number of relays is increased gradually in the two directions till each sensor has at least one close by relay. The number of relays is increased gradually to ensure connectivity between relays while ensuring there are not too many relays which may lead to excessive network traffic. Relays are in direct radio range of at least one other relay. Note that relaying is a feature on a node that can be independently controlled. The relays themselves are also acting as sensors transmitting their data to the GW. The effect of adding relays is shown in Fig. 3(b). The figure indicates that packets sent by all nodes can reach the GW, with more than half of the nodes having a PDR higher than 60%. Throughout the rest of the testing, the arrangement of relays is kept fixed.



(a) Without relay

(b) With relays

Fig. 3. Responses from all the sensors to the GW, node 90

B. Effect of protocol-related parameters

Fig. 3(b) implies that the overall reliability of 63% to send packets to the GW, from all nodes, is still low. This subsection considers the following BTM protocol-related parameters: PRC, NTC, and RRC. Their effect on reliability was tested in isolation and in combination. Table I shows the reliability when each parameter was tested in isolation. The Publish Retransmission Interval (PRI) was set to 50 ms for publish retransmissions. This implies that packets from the model layer are generated one after the other every 50 ms (when PRC is greater than 1).

TABLE I
ACHIEVED RELIABILITY USING ONLY ONE OF PRC, NTC, AND RRC

PRC	0	1	2	3	4
PDR	63.4 %	59.3%	51.5%	55.3%	59.5%

NTC	0	1	2	3	4
PDR	63.4 %	68.7%	69.7%	67.6%	65.5%

RRC	0	1	2	3	4
PDR	63.4%	67.7%	70.4%	69.8%	66.1%

Table I shows that using PRC alone *surprisingly* decreases the reliability. With PRC = 2, the reliability drops by more than 12%, compared to when PRC = 0, then it increases as PRC increases but is still lower than what it is when PRC = 0. NTC or RRC increases reliability. The reliability reaches its peak when NTC = 2. However, increasing the NTC after that decreases the reliability, which is still better than when NTC = 0. On the other hand, the reliability reaches its peak when RRC = 2. Like NTC, increasing the RRC after that decreases the reliability, which still is better than when RRC = 0.

Based on the results from Table I, combinations of both NTC and RRC, whose values gave better reliability, were first tested to see how the reliability would be affected. PRC was then tested with some combinations of NTC and RRC to see if it would still negatively affect the reliability. The results are summarized in Table II.

TABLE II
ACHIEVED RELIABILITY WITH A COMBINED USE OF PRC, NTC, AND RRC

NTC	RRC	PRC	PDR	NTC	RRC	PRC	PDR
2	1	0	69.5%	1	1	0	69.3%
2	2	0	70.0%	1	1	1	49.1%
2	3	0	67.1%	1	2	0	73.8%
3	1	0	63.8%	1	2	1	62.5%
3	2	0	66.8%	1	3	0	65.6%
3	3	0	57.3%	1	3	1	51.5%

Table II shows that the reliability increases when using a combination of NTC and RRC but only when PRC = 0. Hence, using only PRC or using PRC with the other two retransmission parameters decreases the reliability. Table II also shows that the highest reliability in the test network was reached when NTC = 1, RRC = 2, and PRC = 0. However, reliability still needs to be higher, and as is seen in the table, increasing NTC and RRC did not increase the reliability.

In summary, using each retransmission parameter alone has an impact on reliability. The reliability drops when using only PRC, while when utilizing only NTC or RRC, it rises to reach a peak and then declines. Figure 4 shows that reliability increases to 69.7% and 70.4% when NTC = 2 and RRC = 2, respectively. It also shows that the reliability decreases to 51.5% when PRC = 2. Using combinations of NTC, RRC, and PRC also impacts reliability. For example, when it is used without NTC or RRC, PRC can cause a decrement between 11% - 20% in reliability. On the other hand, a combination of NTC and RRC gives better reliability, resulting in 73.8% PDR.

C. *Impact of traffic load*

Since each mesh packet occupies finite airtime, sending sensor measurements too frequently will lead to higher traffic and higher collisions due to the flooding nature of Bluetooth mesh. Therefore, different sensor-sending intervals were also explored to determine their impact on reliability. The best combination found from the investigation in the previous subsection is used for the subsequent investigation. Table III shows that the higher the sending interval (implying lower traffic load), the higher the reliability but only to a certain extent. However, the reliability decreases noticeably as the sending interval decreases: Additional packet loss is found in the network due to congestion caused by the increased traffic load.

TABLE III
RELIABILITY UNDER DIFFERENT SENDING INTERVALS

NTC	RRC	PRC	Sending interval(sec)	PDR
1	2	0	8	71.3%
1	2	0	7	71.6%
1	2	0	6	71.8%
1	2	0	5	73.8%
1	2	0	4	60.9%
1	2	0	3	51.5%
1	2	0	2	40.3%

D. *Effect of non-protocol-related parameters*

The retransmission parameters (such as PRC, NTC, and RRC) can enhance the reliability to a certain extent, as observed above. However, the highest reliability is still only 73.8%. Additional ideas have to be introduced to improve reliability further. In the following subsections, two non-protocol-related specific settings are considered. One is *packet-sending randomization*, and the other is *payload redundancy*.

1) Effect of packet-sending randomization: Recall that all the sensor nodes begin their transmissions almost simultaneously when the test procedure starts. It was observed that even though the mesh stack applies a built-in 10 ms randomization for the packets being sent out, it is not enough when many nodes try to send packets simultaneously. Less randomization could result in collisions and lead to a loss of reliability. To enhance the reliability, we hypothesized adding additional randomization at the higher layers to check if this could improve the reliability further. To do this, we implemented randomization at the model layer while sending the sensor measurement packets to the GW. We decided to use a randomization interval of 10 seconds. Upon receiving the Ethernet command, which triggers the sensor measurements, each sensor waits for a random time in the interval [0-10] sec. After that, it sends its first packet to the GW. The sensor applies the same randomization interval of [0-10] seconds to subsequent packets.

Table IV shows that using packet-sending randomization boosts reliability. The PDR increased to 94.9% using only this parameter, and to 97.3% when NTC = 1 and RRC = 2, and even to 99.7% when NTC = 2 and RRC = 1 and 98.4% when NTC = 2 and RRC = 2. This is a significant enhancement compared to only using the retransmission parameters. Shorter randomization intervals gave lower reliability.

TABLE IV
RELIABILITY WITH PACKET-SENDING RANDOMIZATION

NTC	RRC	PRC	Reliability
0	0	0	94.9 %
1	2	0	97.3 %
2	1	0	99.7%
2	2	0	98.4%

2) Effect of payload redundancy: Packet transmission using randomization can only partially avoid collisions. Additionally, other Bluetooth devices not belonging to the test BTM network in the environment and other noise present on the same frequencies as the Bluetooth channels can also contribute to the loss. Is there a way to further improve the reliability to counter the effect of these issues and reach a stable reliability level of 99% for sensor data gathering?

Note that the actual payload of a BTM packet sent from a sensor to the GW, as explained above, is 4 bytes. Since the payload field has extra bytes available, another 4 bytes of payload can fit in. Then, the actual payload of each packet would be 8 bytes, 4 bytes for the current measurement, and 4 bytes for the previous measurement. In this way, during the continuous data gathering process, even if one packet is lost, the next packet can still help convey measurement value.

With this change in the packet structure, if we measure the number of unique measurements received at the GW, compared to the number of unique measurements sent by the sensor, we could find out the effect on the reliability of the data gathering at the GW. The results are presented in Table V, where randomized packet-sending intervals in the range [1-8] seconds are used. As seen from Table V, the reliability in the test network increased when using the payload redundancy implementation. A reliability level of at least 99.7% for sensor data gathering was achieved regardless of the combinations of NTC and RRC as long as they are not zero. As a comparison, we found that using the payload redundancy alone gave a reliability of 83.4% when the retransmission parameters were set to zero, and packet-sending randomization was not used.

TABLE V
ACHIEVED RELIABILITY AFTER ADDING PAYLOAD REDUNDANCY

NTC	RRC	PRC	Reliability
1	1	0	99.7%
1	2	0	99.7%
2	1	0	99.99%
2	2	0	99.7%

E. Testing more combinations of different parameter settings

The investigations above have shown that the combined use of payload redundancy, packet-sending randomization, and the retransmission parameters NTC, PRC, and RRC can achieve excellent reliability. However, in this investigation, the results shown in Table V have been based on the “best” possible combinations of the protocol-related parameters. To better understand how these parameters affect reliability, we extend the investigation to consider more combinations with different NTC, PRC, and RRC values. The results are summarized in Fig. 4.

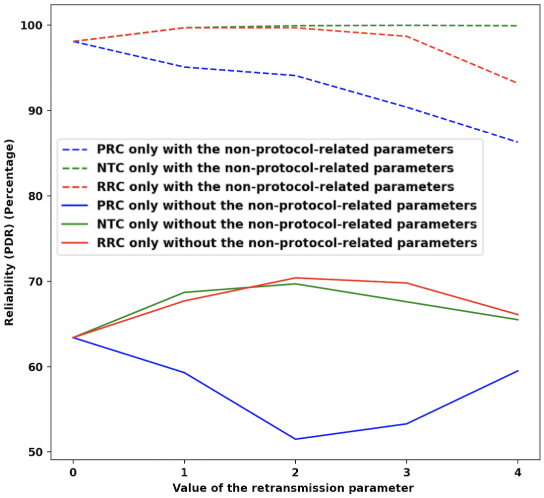


Fig. 4. Impact of NTC, RRC, and RTC on reliability with/without using packet-sending randomization and payload redundancy

Fig. 4 illustrates that using the non-protocol-parameters[1], without the retransmission parameters, i.e., NTC, PRC, and RRC are all set to be 0, boosted the reliability from 63.4% as also indicated in Table I to 98.1%. As a comparison, the figure also illustrates that, without using the non-protocol-related parameters, the best-combined use of NTC, RRC, and PRC could only increase the reliability from 63.4% to a bit over 70%. Fig. 4 also shows that when using packet-sending randomization, payload redundancy, and NTC (> 0) together, a minimum reliability of 99.7% was achieved. Using RRC(> 0) with the non-protocol parameters could also improve the reliability level from 98.1%, but only when RRC=1 or 2. When RRC = 3 or 4, the reliability was lower. Moreover, using only PRC with the non-protocol parameters decreased the overall reliability: The higher the PRC, the lower the reliability.

In summary, the extensive investigation reported above indicates that by combining packet-sending randomization, payload redundancy, NTC, and RRC a reliability level as high as 99% can be achieved with a good margin.

V. DISCUSSION: OBSERVATIONS AND REASONS

A. Effect of protocol-related parameters

Using NTC increases the traffic on the *first hop*. When a packet reaches a relay node, the relay checks its cache before forwarding the packet. If this packet is already forwarded, it will be dropped. Hence, no matter how many replicas of a

---

[1] The sending interval, under the same considered range, showed a marginal effect on reliability when the two non-protocol-related parameters are used.

network layer packet are sent from a sensor to a relay, only one replica will be forwarded. With non-zero to RRC, the traffic could be increased on the *second hop*. This could lead to higher traffic in the network and, thus, higher packet collisions at the GW and the relays, resulting in lower reliability for higher RRC values. By definition, using PRC generally increases the traffic on both the first and the subsequent hops as the model layer sends "x" replicas of the packet. However, each replica carries a new sequence number and causes proportionately higher traffic in the network compared to when PRC was set to 0.

Trade-off: Note that having NTC = y and RRC = z could increase the traffic "y" times and "z" times on the first and the subsequent hops, respectively. This indicates that when protocol-related parameters are used, they can result in more traffic, leading to increased packet drop and collision at the GW and the relays and, subsequently, lower packet delivery probability. On the other hand, their use increases the chance that a given sensor data packet reaches the GW. So, there is a trade-off between increasing retransmissions to offset the effect of collisions vs. too many retransmissions causing more packet losses due to increased collisions. Our experimental study suggests that the trade-off of using NTC, RRC, and PRC differs, and the reliability level can be much improved with proper combinations.

B. Effect of non-protocol-related parameters

Results in the previous section show that, even without the retransmission parameters, only using packet-sending randomization and payload redundancy together could give a high-reliability level of 98.1%. A closer investigation further reveals that using the former alone gave a reliability of 94.5%, while using payload redundancy alone gave 83.4%. Both were better than only using the protocol-related parameters.

There are two underlying reasons. One is that the packet-sending randomization reduces the concurrency of packets competing for radio channel resources, reduces buffering, and reduces processing overhead at the GW and relays. As a result, there is an increased possibility for the GW and relays to process the packets that reach them because they reach them at different times. The other reason is that, with payload redundancy, the sensor data is carried by piggybacking in more packets without increasing the traffic load. Consequently, even if some packets carrying the data could be lost, the data can still be recovered at the GW as long as one of the two packets reaches GW.

C. Achieving 99% reliability of data collection

In [6] and [7], it is reported that 99% reliability can be achieved with BTM. However, in [6] [7], no information is provided about configuring and setting BTM parameters to reach this reliability level. Our experimental study reveals that, for our test network, the reliability level that can be achieved using BTM protocol parameters, for the stated use case, is only a bit over 70%. A high-reliability level can only be ensured with combined use of additional, non-protocol-related mechanisms, contrary to what is mentioned by the reports [6] [7] that are only based on BTM protocol-related parameters. Thus some care is needed when designing and configuring a BTM network to achieve a high-reliability level.

As a final remark, to cover the high number of parameter combinations, the results were obtained mainly by running the test multiple times with a duration of 10 minutes. We have also tested for longer duration as high as multiple hours, particularly for the combined use of packet-sending randomization and payload redundancy with NTC and RRC tuned. The obtained reliability was always higher than 99%, regardless of the duration. This implies that this combined configuration is a proper approach to achieving 99% reliability.

VI. CONCLUSION

In this paper, we explored the use of Bluetooth mesh technology in a large-scale sensor data gathering network. There is a fundamental challenge to ensure reliable sensor data gathering, i.e., how to configure the network to achieve high reliability? In this paper, a set of findings have been obtained through an extensive experimental study on a BTM test network, which shed light on addressing the challenge. Firstly, relays play a critical role. The location of the relays should be chosen in a way where packets from any node can reach the gateway, and at the same time, the traffic in the network is kept as low as possible. Secondly, non-protocol-related parameters, namely packet-sending randomization and payload redundancy, can significantly improve the reliability of data gathering. Tuning only the protocol-related parameters may fail to achieve the desired high reliability in this use case. Thirdly, the protocol-related parameters' effect on reliability can differ significantly, and one may even have a negative impact. Specifically, for our test network, we found that using NTC and RRC can increase reliability only to a certain extent. Overall, our investigation has shown that sensor data gathering with BTM can be made highly reliable by balancing network traffic and redundancy. In particular, packet-sending randomization and judicious design of packet formats to compensate for losses can help to attain reliability higher than 99%.

REFERENCES

[1] R. Rondon, M. Gidlund, and K. Landernäs, "Evaluating bluetooth low energy suitability for time-critical industrial iot applications," *International Journal of Wireless Information Networks*, vol. 24, 09 2017.

[2] "The Bluetooth SIG website," https://bluetooth.com/.

[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*, vol. 15, no. 3, pp. 1–29, 2019.

[4] K. MAXWELL, "How wireless technology is enabling the adoption of commercial connected lighting," *Bluetooth Market Research Note*, 2020.

[5] A.-V. Vladuta, I. Bica, V.-V. Patriciu, and F. Pop, *Green, Pervasive, and Cloud Computing*, 2017, ch. Reliable Data Collection for Wireless Sensor Networks Using Unmanned Aerial Vehicles, pp. 323–337.

[6] P. Di Marco, P. Skillermark, A. Larmo, and P. Arvidson, "Bluetooth mesh networking," *Telefonaktiebolaget LM Ericsson*, 09 2020.

[7] Silicon Labs, "An1137: Bluetooth mesh network performance. rev. 0.2."

[8] "Mesh profile bluetooth specification v1.0.1," *Bluetooth SIG Inc.*, 01 2019.

[9] "nRF Connect SDK v1.7.1," https://developer.nordicsemi.com/,.