Journal of
Sensor and Actuator Networks

MDPI

Article

QoS Enabled Heterogeneous BLE Mesh Networks

Subho Shankar Basu *, Mathias Baert and Jeroen Hoebeke

IDLab, Department of Applied Engineering, Ghent University—imec, 9052 Gent, Belgium; mathias.baert@ugent.be (M.B.); jeroen.hoebeke@ugent.be (J.H.)
* Correspondence: subho.basu@ugent.be

Abstract: Bluetooth Low Energy (BLE) is a widely known short-range wireless technology used for various Internet of Things (IoT) applications. Recently, with the introduction of BLE mesh networks, this short-range barrier of BLE has been overcome. However, the added advantage of an extended range can come at the cost of a lower performance of these networks in terms of latency, throughput and reliability, as the core operation of BLE mesh is based on advertising and packet flooding. Hence, efficient management of the system is required to achieve a good performance of these networks and a smoother functioning in dense scenarios. As the number of configuration points in a standard mesh network is limited, this paper describes a novel set of standard compliant Quality of Service (QoS) extensions for BLE mesh networks. The resulting QoS features enable better traffic management in the mesh network, providing sufficient redundancy to achieve reliability whilst avoiding unnecessary packet flooding to reduce collisions, as well as the prioritization of certain traffic flows and the ability to control end-to-end latencies. The QoS-based system has been implemented and validated in a small-scale BLE mesh network and compared against a setup without any QoS support. The assessment in a small-scale test setup confirms that applying our QoS features can enhance these types of non-scheduled and random access networks in a significant way.

Keywords: BLE mesh; QoS; end-to-end latency; traffic priority; reliability; network configuration and management

Citation: Basu, S.S.; Baert, M.; Hoebeke, J. QoS Enabled Heterogeneous BLE Mesh Networks. *J. Sens. Actuator Netw.* **2021**, *10*, 24. https://doi.org/10.3390/jsan10020024

Academic Editor: Atis Elsts

Received: 16 February 2021
Accepted: 24 March 2021
Published: 28 March 2021

1. Introduction

Wireless communication plays a significant role in various Internet of Things (IoT) applications [1,2]. Selecting an appropriate wireless communication technology that fits the requirements of the IoT application is important. In recent years, smart lighting and control is being extensively considered by various building owners and operators for increased energy efficiency and building management flexibility. Among all the available wireless communication technologies, Bluetooth Low Energy (BLE) mesh emerges as a viable option to drive the momentum for mesh-based IoT lighting. This is mainly due to the features that BLE mesh networks provide, which align well with the requirements of smart lighting. Smart lighting mesh networks come with the immediate benefit of saving power and money, but they also lend themselves as an enabler of multiple services and applications (e.g., Heating, Ventilation and Air Conditioning (HVAC)) beyond illumination. BLE mesh networks thus not only pave the way for smart lighting, but also a multitude of mesh-based IoT applications, leading to the concept of heterogeneous BLE mesh networks.

Originally, BLE was designed as a short-range wireless technology with ultra low power consumption, which is a requirement for many IoT applications. The technology has a decent data rate and throughput to meet the requirements of the vast majority of IoT applications. Functioning in the license-free spectrum of 2.4 GHz, it also benefits from independent management and operator-free costs. However, as with every other technology, BLE has limitations, the most important being the limited range and coverage. A typical BLE application is supposed to have a reach of a few tens of meters. Another limitation of BLE is the connected-oriented approach, leading to a star topology consisting

of a single central and multiple peripherals. This not only has the problem of a single point of failure, but limits the number of devices that can be connected to the master device. This is a hindrance for applications such as smart lighting which require flexible deployment and operation across an entire building without range limitations, as well as the possibility of forming a scalable network backbone that supports many-to-many communication patterns. These shortcomings have led to the idea of adding meshing capabilities to BLE, where one combines the advantages of traditional BLE with new properties such as increased scalability, an extended range and the interconnection of devices in a multi-point architecture with relaying capabilities.

Since its inception in 2017, BLE mesh is a mesh networking standard that operates using the advertising and flooding approach of the BLE radio. It is essentially an m:n network where each node is able to communicate with every other node in range. However, nodes that are not within direct range can be reached over multiple hops using the relay feature, where relay nodes flood the message throughout the entire network. Despite the fact that flooding is managed through the use of a Time-to-live (TTL) value and message caching, the redundant traffic leads to increased network loads and possibly congestion. Effective spectrum utilization is necessary for reducing collisions and network traffic redundancy should be used wisely to limit its adverse effects.

Thus, BLE mesh networks open up a new horizon with added features and capabilities, but require proper network management to achieve a decent network performance. In the current BLE mesh specification, the degrees of freedom to configure the mesh network are limited. In addition, these features focus on homogeneous BLE mesh networks, without the ability to differentiate between different applications encountered in a heterogeneous mesh setting. To address this problem, this paper describes a novel set of standard compliant Quality of Service (QoS) extensions for heterogeneous BLE mesh networks. These extensions enable more flexible and application-aware management of BLE mesh networks, which can significantly contribute to an improved overall network performance. The proposed design was implemented and validated using a small-scale BLE mesh network. For each of the proposed QoS extensions, its effect was evaluated and compared to a network without that QoS feature. To our knowledge, this is the first paper that designs, implements and validates QoS extensions for heterogeneous BLE mesh networks.

The paper is organized as follows. Section 2 provides a brief overview of mesh networks and the working principle of BLE mesh with an example scenario. This is followed by an overview of related work in Section 3. Next, Section 4 highlights the areas where BLE mesh networks can be further improved, leading to the identification of newly added QoS features, which are discussed in Section 5. Section 6 describes the QoS functionality and how it was integrated with Nordic Semiconductors' implementation of BLE mesh. In Section 7, the experiments that analyzed the potential benefits of proposed QoS features are described. Lastly, Section 8 concludes the paper with further possibilities of future work in the area of QoS for heterogeneous BLE mesh networks.

2. BLE Meshing

BLE mesh provides the ability for a many-to-many communication network, for which it finds its usage in a wide number of mesh based IoT solutions. This section first provides an example scenario of a heterogeneous BLE mesh network, followed by the underlying BLE mesh concepts defined in the standard.

2.1. Heterogeneous BLE Mesh Example Scenario

Smart lighting and control is being increasingly applied in homes, buildings and work-spaces. Smart lighting applications need to be prompt in action with a decent coverage without the need for a large bandwidth, essentially focusing on the user's feel and experience and the building's increase in energy efficiency. Rooms in a building should be able to detect user occupancy and switch off the lights when rooms are not in use. This requires occupancy sensors to be installed in the rooms which at regular intervals sense the

occupancy of the room and transmit the information to a controller system that can control the lights in the room. In addition to that, it should be possible to adjust the brightness, color and hue of a series of lights in a room for a better user comfort and experience. This requires communication between the sensors, controllers and the actuators (lights in this case) in the building in an efficient way. The controller can be a local one within the room itself or a centralized controller for the entire building at a further location. It is highly expected that some types of sensors, controllers or actuator devices will run on small batteries and are expected to last for a decent time or even be battery-less and running on harvested energy. In addition, they should be easy to install with a plug-and-play approach so that dynamic setups can be possible with the lighting system. In essence, the notion of a highly configurable lighting system with easy installation and barrier-less communication is what is necessary. The lighting system itself can serve as a backbone for other systems in the same area, such as the HVAC system of the building, giving rise to a multitude of application scenarios for a smart building.

This is illustrated in Figure 1, which shows a smart home with a BLE mesh network for smart lighting. The home has a bedroom, hall, kitchen, toilet and a lobby and there are lights, switches, occupancy sensors and controllers in different areas of the room. Each room has its own local switch to control the lights (e.g., the bedroom), but there is also a pair of master switches (switch SwB2 in the bedroom and SwH3 in the hall) to control all lights of the entire home. Sensors (e.g., SnB1 in the bedroom) send data to controllers (e.g., CB1 in the bedroom) on the state of occupancy of the room. Lights can be turned on or off with the switches, but, based on the data from the sensors, the controllers can also control the lights. A switch can control a set of lights (e.g., switch SwB1 and lights LB1 and LB2 in the bedroom), but a light can also be controlled by a set of switches (e.g., light LLH1 at the junction of the lobby and the hall being controlled by the switch SwL1 in the lobby and switch SwH1 in the hall), giving the notion of a many-to-many communication. The whole system forms a network with a unique network identifier, where all involved nodes can communicate with each other by relaying data in the network. For example, the master switch in the hall (SwH3) is used to control all lights in the entire home, but might not be in direct range of the extreme corner lights of the home in the bedroom (LB1) or the lobby (LL1). Therefore, the data will be forwarded via the controller in the bedroom (CB1), as the controller is in reach of the light LB1. Other systems, such as a HVAC system (e.g., the fan HB1 in the bedroom), can be integrated in the same home and use the smart lighting mesh network that is already in place as a backbone network. Installation, setup or removal of additional devices in the same home can be done at run time, without disturbing the functioning network, and requires minimal effort.

2.2. Underlying Characteristics of a Mesh Network

A mesh network is defined as a network of nodes where each node can communicate with any number of nodes in its surroundings, either directly in a single hop or via multiple hops with data relaying at the intermediate nodes. Moreover, due to the relaying characteristic of the network, there can be multiple redundant communication paths between a source and a destination. Hence, a mesh network is essentially a multi-hop and multi-path network with many-to-many connectivity and also called as m:n network. A data packet from a source to the destination can be relayed by multiple nodes in the network via multiple paths, and hence there is the possibility of receiving the same data packet multiple times at the destination. Although this multi-path redundancy aspect could increase the chances of packet reception, care has to be taken for packet de-duplication. Therefore, in a mesh network, multi-hop features the extension of range with the possibility of routing mechanisms, and multi-path provides path redundancy and better reliability and fail-over mechanisms. Multi-path essentially removes single point of failure, so that, when a path breaks down, it is possible for data packets to reach the destination via other possible paths providing path redundancy. Moreover, these type of networks are self-healing, i.e., when a link in the path breaks down, data can be forwarded via other paths, but, when the

link comes back up, the path can be restored. Nodes can even dynamically join and leave a network providing flexible network communication architectures. With self-discoverability features, nodes can even discover each other in vicinity and know the abilities (e.g., quality of neighboring links) of one another providing the sufficient intelligence in the network to choose the best possible path. Mesh networks can be deployed with different technologies: WiFi, ZigBee, Z-Wave, Thread, BLE, etc. This paper focuses on mesh networks using BLE as the underlying technology hence known as BLE mesh.

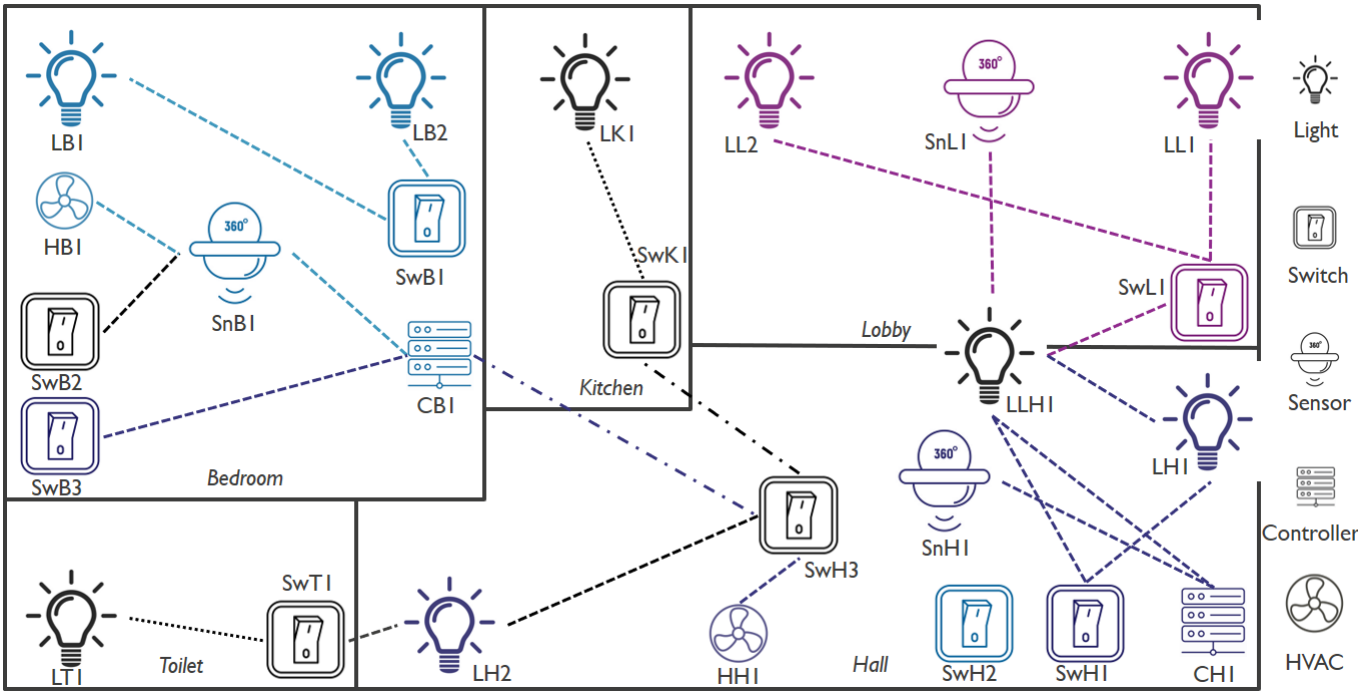

Figure 1. An example scenario of a heterogeneous BLE mesh network, where smart lighting mesh nodes act as a backbone for a HVAC system.

2.3. Primer on BLE Mesh Technology

Figure 2 shows an example of a mesh network. Nodes A, B, C, H, I, J, K, L, M and N are end nodes while D, E, F and G are relay nodes. Some of the nodes can have more optional features. For example, nodes M and N are low power nodes in BLE mesh supporting friendship feature with friend node D. The friendship feature reduces power consumption in low power nodes, aiding power-constrained devices to still participate in a mesh network without 100% duty cycling. Moreover, some nodes in BLE mesh can also implement the proxy feature which helps stand-alone BLE devices without meshing capabilities to participate in a mesh network. In the example shown above, a communication from node A to node B is one hop. Node A transmits the data to node B directly, but node B will not relay the packet further since it is not a relay node. If however node A wants to send a data packet to node I, which is not within direct reach, a multi-hop communication is needed. Node A has to send the packet to node F and then node F relays the packet to node I. If however the link between node F and node I is broken, the path can be modified as A, F, G, I in sequence. However, it is possible that both the paths are taken simultaneously and I receives two identical copies of the packet, in which case it drops the second packet. This is how a BLE mesh network functions in the fundamental level.

Figure 3 shows the BLE mesh stack from the standard specification. As BLE mesh runs on top of BLE technology, the standard protocol layers above (in light blue) denote the protocol stack specific to BLE mesh, while the two bottom layers (in dark blue) denote the BLE technology on which BLE mesh depends. Data flows across layers starting from

the application layer down to the physical layer at the sender's side and reverse up on the receiver's side. Each of the layers are briefly described below.

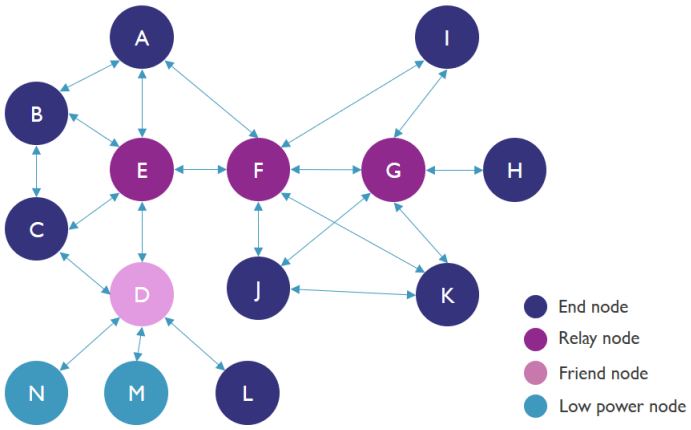

Figure 2. An example of a BLE mesh network.

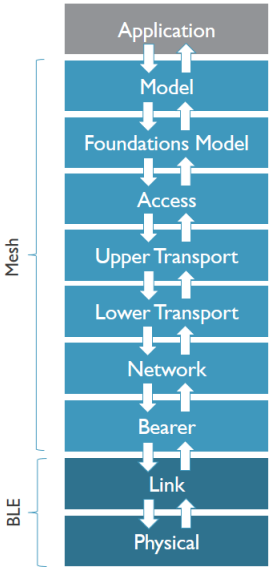

Figure 3. BLE mesh stack.

- **Physical layer:** The physical layer is responsible for translating the digital symbols to analog ones and vice versa. It is the lowest layer of the stack and forms a bridge between the link layer and the Bluetooth radio.
- **Link layer:** The link layer depends on the services of the physical layer below and provides services to the bearer layer above. It is mainly concerned with advertising, scanning, creating and maintaining connections in BLE.
- **Bearer layer:** The bearer layer defines how the different BLE mesh packets are handled. There are two types of bearers in BLE mesh, namely the advertisement bearer and the Generic Attribute Profile (GATT) bearer. The advertisement bearer takes care of handling packets using the advertisement mode of BLE, while the GATT bearer is used by proxy nodes to setup a connection with a non-mesh device in order to allow it to participate in a mesh network. The bearer layer uses the services of the link layer below and provides services to the network layer above.
- **Network layer:** The network layer is concerned with handling network level PDUs, particularly dealing with packet addresses, network level authentication and encryption using the network key and how packets are relayed within the network. It interfaces with the lower transport layer above and uses the services of the bearer layer below.

- **Lower Transport layer:** The lower transport layer deals with segmentation and reassembly (SAR) of packets and provides acknowledged or unacknowledged transport of messages to the peer device on the other end.
- **Upper Transport layer:** The upper transport layer deals with authentication and encryption of access layer messages using application or device key and defines transport control messages and procedures to handle friendship, heartbeat functionality, etc.
- **Access layer:** The access layer ensures that messages are transmitted and received in the right context of a model and its related application keys.
- **Foundation Model and Model layer:** These two layers define standard ways to handle application layer data with the use of models, leading to a uniform, interoperable communication in a high-level context. Models define state and messages that govern change of state from one to the other immediately or over a certain period of time. Foundation models are the mandatory ones involving client or server models according to the roles of the nodes, namely the health client/server model and the configuration client/server model. Models can also be extended from a root model.
- **Application layer:** The application layer determines the top layer and denotes any application of specific interest (e.g., turning lights on/off in a smart-lighting application). It uses the models below (e.g., generic on-off model) to participate in a mesh network.

The communication mechanism of BLE mesh follows a publish–subscribe model and fosters easy addition and removal of a node to/from a network. Nodes can publish and subscribe to unicast, group and virtual addresses. A unicast address denotes the address of an element of a node, while group and virtual addresses denote a group of elements within a set of nodes. Nodes subscribe to addresses which they are interested to receive messages from. For a single transaction of a message sent and received, the sender publishes the message to an address (unicast, group or virtual) and the receiver if subscribed to this address processes the data. With the previous example of the smart lighting in a home shown above, the publish–subscribe mechanism is shown in Figure 4.

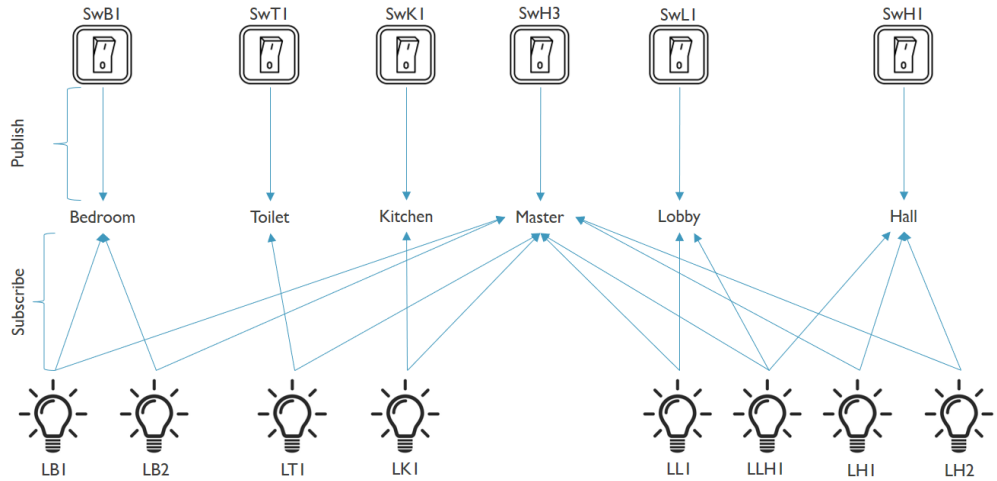

Figure 4. Publish–subscribe mechanism of communication in BLE mesh.

BLE has two modes of operation namely the advertising mode and the connected mode. BLE mesh utilizes the advertising mode of BLE only as it is much simpler, random and follows a best effort delivery approach. The sender transmits the data on the advertisement channels 37, 38 and 39 in succession while the receiver listens on either of the channels, switching from one to the other in succession as well. This way, it is expected that the receiver picks up at least one of the three packets sent by the sender. This advertising phenomenon of course has negative impacts on the reliability aspects, but that is mitigated with the adoption of acknowledgment and retransmission mechanisms at various layers of the protocol stack. Acknowledgments can happen either at the application layer or the

lower transport layer for segmented messages. If a message is transmitted and the acknowledgment does not arrive within a certain time interval, the message is retransmitted ensuring message reliability. Moreover, the same message can be transmitted multiple times irrespective of whether it is an acknowledged transmission or not, increasing the chances of message reception at the receiver. Further, a message from a source to the destination is relayed by multiple nodes in the network which increases the reliability rate as the same message can be received on the receiver multiple times due to multiple paths. From the specifications of BLE mesh [3], each node in the network with the relay feature supported and enabled should relay a message.

Figure 5 shows the relay and queue operations of a relay node participating in a BLE mesh network. Once a packet is received, by default a relay node relays it as per the specification. The packet is enqueued in the relay buffer as it is (at the network layer perspective) with the source, destination and application key identifier (AKI) information, and a timer is scheduled with a random backoff delay. Once the timer expires, the packet is dequeued from the buffer and transmitted on the advertisement channels 37, 38 and 39 in succession, and the repetition counter is incremented by 1. Again, the timer is reset with the backoff delay and on expiry the same transmission process repeats until the repetition counter reaches the maximum value and the next packet from the buffer is fetched. This process continues until the buffer is empty. As part of the QoS study which is the objective of this paper, the backoff delay, the number of repetitions, the transmission channel settings and the enqueuing and dequeuing process are some of the parameters (shown in color) that are made configurable to achieve better network efficiency, which is discussed below.

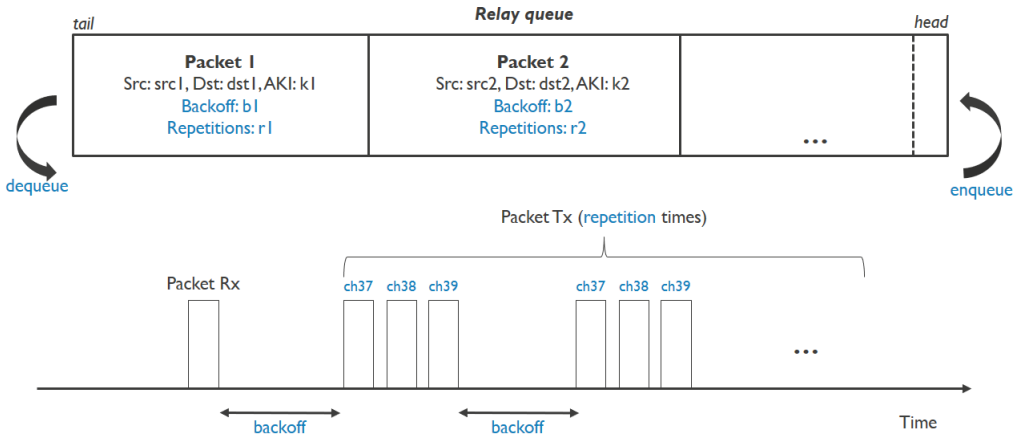

Figure 5. Relay and queue operations in BLE mesh.

The relay mechanism results in message flooding, which is handled with the approach of managed flooding with the use of TTL and message caching to avoid infinite message loops in the network. Each message has a TTL value (with a maximum value of 127) which is decremented on each hop so that the message relaying stops when the TTL value reaches 1. In addition, each message has a sequence number and messages with the same sequence number when received from the same source determines duplicate packets due to redundancy in transmission or path. These duplicate messages are dropped and only one copy of the message is cached in each of the nodes in the path, until the cache becomes full to replace the least recently cached packet. However, the entire mechanism has no routing principles in the relay nodes and it works as an open advertising scheme to reduce the complexity of the nodes. This is decent for a sparse network, but for a very dense network the large number of retransmissions and message flooding can lead to an unnecessary network congestion reducing the performance of the network as a whole. Hence, there needs to be a more efficient management scheme to ensure a smooth functioning of the network with better efficiency, leading to the concept of the application of QoS features in BLE mesh networks. Moreover, BLE mesh networks has the optional relay, proxy and friendship features but has no QoS features in place. The addition of an optional QoS

feature will aid a BLE mesh network with more capabilities in terms of overall network performance and individual node efficiency.

3. Related Work

Baert et al. [4] presented an overview of the working principle of BLE mesh networks and gave further insights on the latencies for these type of networks. The authors of ref. [5] showed the importance of having the correct configuration of the parameters in a BLE mesh network for better reliability, latency and scalability of these networks. Their study is based on a simulation and closely studies the channel and timing characteristics of these networks, whereas our study performs a similar analysis on a real test setup, but also includes novel QoS features. The authors of ref. [6] evaluated the performance of a BLE mesh network in a real testbed and showed how the PDR and message reliability is affected with networks becoming denser. ref. [7] also stated the importance of configuring the various parameters across layers to increase the efficiency and reliability of a BLE mesh network and outlines possible standard ways of improvements. They also identified various limitations of the chipsets used and the overhead of the protocol stack. The authors of ref. [8] illustrated a design of a multi-hop real-time protocol over BLE mesh to consider bounds for time critical applications. In ref. [9], various ways for efficient relay selection in a distributed fashion in a BLE mesh network are illustrated. The authors of ref. [10] presented the main features of BLE mesh and 6BLEMesh and investigated their performance characteristics and tradeoffs. In ref. [11], the capabilities of BLE mesh networks are studied, and the authors indicated that BLE mesh is a promising technology for mesh applications, and further research in the domain is necessary to exploit its full potential. Lastly, the authors of ref. [12] studied an effective QoS differentiation scheme for IEEE 802.16 WiMAX mesh networks, identifying the key parameters that influence the performance in different services.

In view of these works, it is clear that BLE mesh networks have a wide number of open challenges, and, with increasing network densities, these networks degrade in performance. Both simulations and real world experiments pertain to the fact that BLE mesh is a novel technology that offers interesting possibilities to support the communication needs of IoT applications. The flooding approach that is used by this technology has limited complexity but comes with limitations. There is a need for robust network configuration and management to use the technology to its fullest potential. This motivates us in the direction of our proposed approach of introducing QoS mechanisms to BLE mesh networks, and we show how this can bring a difference in network performance. Compared to previous works, our work is, to the best of our knowledge, the first paper that addresses these challenges through the incorporation of standard-compliant QoS features in the BLE mesh stack, along with better and more flexible management capabilities for increasing the performance of BLE mesh networks.

4. QoS in BLE Mesh: Challenges and Opportunities

BLE mesh networks have a number of specific characteristics which impose challenges regarding the design of standard-compliant QoS mechanisms and limit the possible solution space. This is further outlined in the following subsections.

4.1. m:n Network

As the name implies, a BLE mesh network is an m:n network or a many-to-many network, where each node has the possibility of communicating with every other node in range, in a unicast, multicast or broadcast way. Moreover, it is a multi-hop network, with multiple possible paths that enable a sender to communicate with one or more receivers.

4.2. Advertising and Packet Flooding

BLE meshing utilizes the advertisement mode of BLE for packet transmissions. Each message is advertised three times, on channels 37, 38 and 39 consecutively, and subsequent messages are transmitted at advertisement intervals, thereby using a random backoff within each

advertisement window. Upon reception of a message by a relay node, the message is relayed in the network leading to message duplication and flooding. This results in a cascading effect of the number of messages generated in the network for a single message transmission by a source. As networks become denser, this can become a serious issue with a linear growth in the number of messages transmitted.

4.3. Distributed, No Access Points

A BLE mesh network is entirely distributed in nature without any central access point. Despite the advantage of not having a single point of failure, the distributed nature implies that every node is on its own regarding how it communicates with other nodes applying its own intelligence on when to access the spectrum, setting the interval between packet transmissions, buffer sizes, channel settings, etc.

4.4. Connectionless

BLE mesh networks are connectionless networks that use the advertisement mode of the BLE technology. Hence, the channel access mechanism is entirely random with no co-ordination between the sender and the receiver. As explained above, the entire procedure relies on the principle that, to receive a message, the receiver has to scan on one of the three channels that the sender uses to transmit the data. The scan window must be sufficiently larger than the total time of the transmission of the three consecutive transmissions from the sender. This approach results in a best effort chance to receive the packets without any guarantees on successful reception.

4.5. No Scheduling, Entirely Random

BLE mesh networks are asynchronous and have no scheduling mechanisms between the sender and the receiver. There is no organized way of accessing the spectrum, which may lead to collisions and thus packet losses. In the case of retransmissions, additional latency is introduced. With the network becoming denser, the chances of collision and interference become higher, resulting in a lower performing network.

4.6. Solution Space

The above discussion reveals that, due to the design of BLE mesh networks, there are some inherent challenges to provide applications with QoS. Further, the complexity to do this increases as the network becomes denser.

To overcome these issues, QoS mechanisms are required. However, to remain compliant with the BLE mesh specification, the working principles of the BLE mesh network cannot be changed. Consequently, the introduction of QoS features should focus on obtaining a better performance of the network by: (i) extending the current set of configuration parameters within the degrees of freedom offered by the BLE mesh specification; and (ii) applying these configuration parameters in an application-dependent way in order to allow more fine-grained configurations. Once such mechanisms are in place, a system that collects monitoring data from the network and correctly configures the nodes, is expected to result in a better functioning of the entire distributed system.

This approach will not only enhance the performance of the network in dense scenarios, but will also offer the possibility to enhance certain network characteristics such as latency, jitter, reliability, traffic priority, reduced collision and higher probability of packet reception. Once the application requirements and network characteristics are known, they can be utilized to generate the proper configuration settings for the individual nodes. This way it becomes possible to achieve better network resources sharing and overall performance. For example, if there are multiple relays between a source and a destination with all the relays relaying the message, configuring offset values for their transmission times will result in lower collisions on the relayed messages, which results in a better managed mesh network.

Based on these observations, a set of QoS features with configurable parameters was designed, which is described in the following section.

5. Supported QoS Features

Figure 6 summarizes the QoS features that were defined to better support QoS in heterogeneous BLE mesh networks. In the following subsections, each of these features is described in more detail.

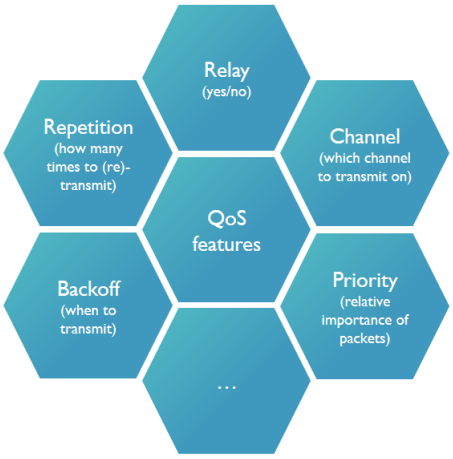

Figure 6. QoS features.

5.1. Relay Choice

The standard BLE mesh specification states that all nodes that support the relay feature and that have that feature enabled should relay messages within the network. Relaying messages provides redundancy, but at the same time increases the network load. Hence, one should carefully select which nodes should act as relays. Moreover, depending on the requirements of the specific application, a different relaying behavior might be desirable. Therefore, this feature was extended with the capability to choose whether to relay a message or not based on application-related information available in the incoming network packet. More precisely, the set of source and destination pairs and the AKI are used to make the relaying decision. When a message is destined from source S to destination D with AKI I, the node will check the presence of a relaying rule (configurable at run time) for that particular combination of source S, destination D and AKI I. If there is a matching rule, then the corresponding relay decision is applied, otherwise the default behavior of that node (always/never relaying the message) is applied. Having the flexibility of enabling or disabling the relaying functionality in a more fine-grained decision helps in reducing the congestion within networks and promoting efficient paths for message forwarding, whilst considering particular requirements of an application.

5.2. Repeat Transmission Count

Whenever a packet is allocated in the memory buffer for transmission, either at a source or a relay, it is transmitted a fixed number of times, as indicated by the repeat transmission count. This can be an issue with unacknowledged traffic where packets might get lost, thereby affecting the reliability of the application. Multiple (re-)transmissions might be of help in these scenarios, increasing the chances of packets being received in a lossy or low signal strength environment. To achieve fine-grained settings of this parameters, i.e., at node level and per application, the retransmission parameter was modified to let the node choose the number of times a particular message will be transmitted. Again, the value of the repeat count is stored for a corresponding source, destination and AKI value. It is used when a match applies or defaults to the standard value when no match is found. The adoption of this feature brings the possibility of increased reliability during times when a receiver is very far away from the sender and the signal strength is poor, as the

chances of receiving the message increase with multiple transmissions. Applications that require high reliability can also benefit from this feature. Of course, an excessive amount of repetitions may contribute to increased network congestion, so this trade-off needs to be taken into consideration. Making this parameter configurable empowers the node with more possibilities of fine-tuning the network.

5.3. Backoff Times

When a message is scheduled for transmission either at a source or a relay, it is taken from the queue for transmission after a certain back-off time, as shown in Figure 5. This back-off time is a random value within the advertising interval to reduce the chances of collision within the network. Such randomization can handle a certain degree of network load, after which the network performance starts to degrade. The drawback of randomization is that latencies become more unpredictable. From a QoS point of view, this is not desirable for prioritized traffic that needs to meet certain latency bounds.

Therefore, to have more configuration flexibility, one should be able to configure different backoff times at the individual nodes, to not only set a maximum backoff value but also a minimum backoff value and to adjust the backoff timings to the specific application.

With such an extension, higher priority packets can be configured with lower backoff times and vice versa. Using this feature also helps to predict and improve end-to-end delays, which might be important for real-time or time-critical applications. The backoff time configuration allows a node to have a range of values between a minimum and maximum and randomize the value within the range. This way one still retains the randomization strategy, but with the added advantage of reducing collisions, handling priority traffic and guaranteeing end-to-end latencies.

5.4. Channel Configuration

By default, data are transmitted on advertisement channels 37, 38 and 39 in succession. This works pretty well in sparse network environments. However, as the network becomes denser, with multiple senders transmitting at the same time on the same channel, collisions and interference increase resulting in a lower network performance. To reduce this effect, one has to take care that transmissions happen in different channels as long as possible. The channel configuration QoS feature allows a node to configure the order of channels to be used. Appropriate channel configuration in nodes within a network can lead to an environment with a lower amount of collisions.

5.5. Priority Traffic Handling

The standard treats all messages the same and follows a first-in-first-out (FIFO) approach in handling messages in the queue. This does not allow prioritizing certain application traffic. Hence, a feature was added to include a priority level for each message and the FIFO approach was changed to a priority-based approach. The priority value is configured and stored in the node as a rule along with the source, destination and AKI value to which it applies. When the timer to start the next message transmission expires in the node (either a source or a relay), the queued packet that has the highest priority in the buffer is transmitted. However, this does not preempt a packet already in transmission with possibly multiple repeat counts. The addition of priority levels and priority queuing helps in handling priority traffic in an efficient way. Assigning an equal priority value to all messages boils down to the default FIFO mechanism. In addition, support was provided to enqueue packets with higher priority when the buffer is full, by dropping packets that have the lowest priority. This way, it is ensured that higher priority packets are always enqueued and never dropped.

5.6. Combining QoS Features

Each of the proposed QoS features can be configured independently. However, they can also be mixed and matched to achieve the desired behavior for a specific application

flow. The observed effect can either be at the cost of other applications or developing an overall network performance.

6. QoS Implementation

This section describes the complete QoS architecture, along with its implementation, which implemented the QoS features introduced in the previous section. For the implementation, we used the BLE mesh stack of Nordic Semiconductor [13], along with the nRF52840 DK hardware [14].

6.1. Architecture

The architecture of the BLE mesh protocol stack with the novel QoS features is shown in Figure 7. The QoS features mentioned in Section 5 are managed by a QoS manager which consists of a table that defines the rules (described in detail in the following subsection). This QoS manager can be consulted from various layers (network and bearer layer) in the mesh stack in order to fetch the values of the corresponding parameters to be used while transmitting or relaying a packet. The network layer extracts the values for backoff, repetition, priority and relay (for relay nodes), whereas the bearer layer extracts the channel configuration that is used for sending the packet. The table entries are configured by means of a configuration interface, and they can be dynamically modified at run time. Hence, the QoS manager does the bookkeeping of the parameters. Based on a certain traffic flow and requirements of the application or the network, these parameters are configured across all nodes participating in the network. The intelligence of the QoS functionality relies on the nodes themselves and does not require any modifications to the packets being exchanged. As such, our design remains backwards compatible and can coexist with other devices and applications that use the standard BLE mesh functionalities. The QoS Manager layer is entirely optional and can be easily enabled or disabled. The use of this layer leads to slightly more processing delays but can lead to a more flexible, configurable and better performing mesh network, as described in Section 7.

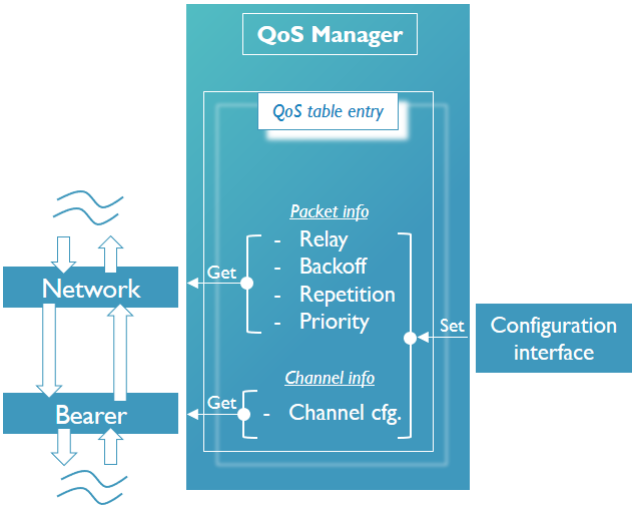

Figure 7. BLE mesh stack with added QoS features.

6.2. QoS Table

The QoS features are implemented as part of the QoS manager by means of a table that stores the configuration data for each node individually. The format of the table entries is shown in Figure 8 and consists of the following parameters.

Src	Dest	AID	Direction	Relay	QoS	Backoff	Repetition count	Channel

Figure 8. QoS table entry format.

- **Source Address:** A 16-bit network unicast source address.
- **Destination address:** A 16-bit network destination address (unicast, group or virtual).
- **Application Key Identifier (AKI):** A 6-bit Application Key ID denoting a specific application.
- **Direction:** A boolean indicating the validity of the entry for the packet direction marked with the source and destination fields in the table. It can be either unidirectional or bidirectional and leads to a more compact QoS table and easier management.
- **Relay flag:** A boolean indicating whether to relay the packet or not.
- **QoS flag:** A boolean indicating whether QoS features are to be applied or not.
- **Backoff time:** The backoff time applied while transmitting or relaying the packet. The backoff time is implemented as a combination of a minimum and maximum backoff time (each consisting of 2 bytes), and the resulting backoff time is calculated as a random value between the minimum and maximum.
- **Repetition count:** The number of times a packet should be (re-)transmitted at the network layer (1-byte representation).
- **Channel:** Two bits indicate the channel configuration for the transmission, resulting in four possible strategies: default (37, 38 and 39), randomized, fixed predefined order or based on the receive channel.

The QoS table is part of each node participating in the mesh network. The parameters can be configured either statically in each node during compile time or by the use of a QoS configuration model dynamically at run time via a configuration interface. The way the table is parsed is illustrated in Figure 9. It consists of two passes. The first pass only considers uni-directional matches. If no match is found, a second pass is performed. In the example, the communication from source 1 to destination 2 is uni-directional and has a different configuration for each direction. The communication from source 1 to destination 2 is of high priority with a low backoff time, while the communication in the other direction is of low priority with a higher backoff time. In the case of communication from source 3 to destination 1, although there is a match with the bi-directional rule, this rule will not be applied as in the first pass the latter rule from source 3 to destination 1 with low priority will be matched and the second pass will be aborted. This is just to show the working principle as an example, so that the configurations are made in a consistent fashion. The inclusion of this direction feature enables better management of the table for a large number of nodes.

	Src	Dst	Priority	Back-off	...	Dirn	
Pass 1 (uni)	1	2	High	Low		Uni	
	1	3	Medium	Medium		Bi	Pass 2 (bi)
	2	1	Low	High		Uni	
	3	1	High	Low		Uni	

Figure 9. QoS table parsing rule.

6.3. QoS Logic

The operation of the QoS-enabled BLE mesh network is similar to the standard BLE mesh network, but now with the addition of some extra logic at each of the nodes that implement the novel QoS mechanisms. Nodes with and without QoS features can coexist in the same network. Figure 10 shows the flowchart of the working principle at each of the nodes that implements the QoS features. The blocks in blue denote the standard BLE mesh principles, while the ones in purple denote the added QoS logic and data flow.

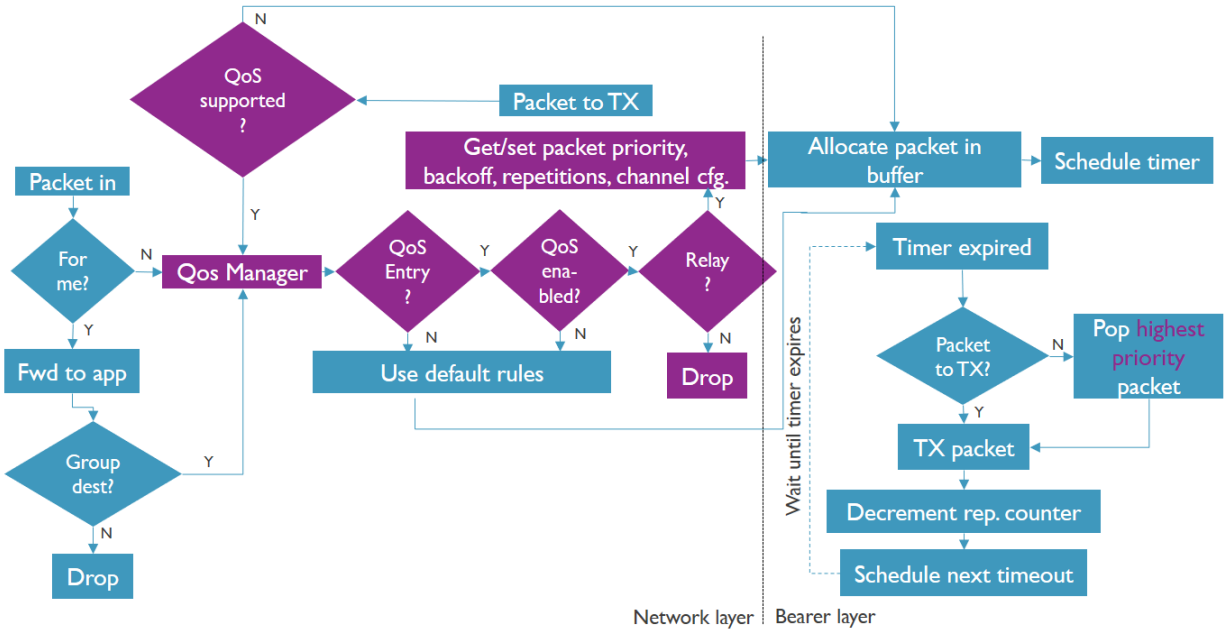

Figure 10. Flowchart for QoS enabled packet flow.

A node can be idle, processing, transmitting or receiving at any instant in time, and it changes from one state to the other over time. When a node has data to send, i.e., as a source node, it processes the packet that needs to be transmitted, turns on the radio, transmits the packet and returns to the idle state. When transmitting the packet, it switches between the channels with some delay in between, as it has to transmit on three channels successively. When not transmitting, a node is in the receive state scanning on a particular channel for the scan window period, and switching between channels at a certain interval known as the scan interval. Once it receives a packet, it forwards the packet for further processing and gets back to its scanning state. After the packet processing is complete, it decides whether the packet is meant for itself or needs to be relayed as well. If so, it consults the QoS manager to retrieve the parameters to be used for relaying the packet. The QoS manager performs a lookup in the table to find a match for the source address, the destination address and the Application Key Identifier (AKI) of the incoming packet. If no QoS entry is found, the QoS manager will instruct to proceed with the default relay rules. If a match is found, the QoS parameters of the entry will be retrieved, including whether the packet needs to be relayed or not, the packet priority, the minimum and maximum backoff time, the repetition count, the channel settings, etc. Next, the node sets the configuration as per these parameters, the packet is allocated to the buffer and scheduled for transmission using a timer. The same process of consulting the QoS manager takes place when the node is a source node for a packet transmission. The packet is transmitted upon the expiry of the timer, when it checks the buffer for a packet transmission. On each transmission of the packet, the packet counter is reduced by one and the timer re-scheduled as per the configured backoff delay. The packet is removed from the buffer when the repetition count becomes 0 and the next packet to transmit is popped from the buffer. With the QoS features enabled, the highest priority packet from the buffer is scheduled for transmission, compared to the First-in-first-out (FIFO) approach used in the standard BLE mesh stack. The priority approach also applies when a packet needs to be enqueued in the buffer. In the event the buffer is full and a high priority packet arrives, the lowest priority packet is removed from the buffer, thereby making space for the high priority packet.

7. Evaluation

To evaluate the potential performance gains that can be obtained by using the proposed QoS features for BLE mesh networks, experiments were performed on real hardware. First,

a set of experiments was performed on a modular basis based on the QoS feature under study and then some experiments were performed where multiple features were combined to see the overall outcome in a more complex environment.

7.1. Relay Choice

Figure 11 shows an example topology of a BLE mesh network. It consists of two different applications denoted by the two colors at the nodes. In the first application, client C1 communicates with servers S1 and S3, while, in the second application, client C2 communicates with servers S2 and S4. The nodes form a bus topology equidistant from one another and each lying at the marginal reach of its neighboring nodes. The objective of this experiment was to see how the application of QoS leads to the reduction of the total number of messages relayed in the network, without breaking the application functionality. All nodes support the relay feature. The set of messages relayed with and without using the QoS relay feature are shown in the diagram. It shows that the functioning of the network remains intact, while the number of relayed messages is reduced.

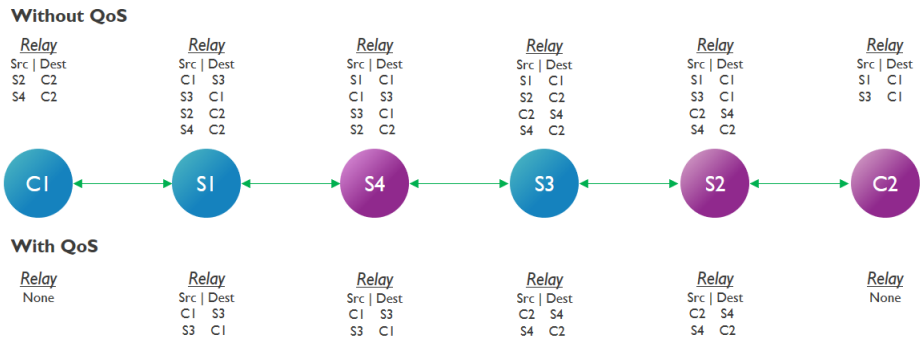

Figure 11. Customized relay feature for QoS.

The native implementation without QoS, thus without the capability of enabling/disabling the relaying functionality for specific application flows, relays all packets. Looking more closely, it is clear that, in the case an application is bound to a specific region of the network, it is not needed to relay the traffic across the entire network. For instance, S1 does not need to relay packets belonging to the purple application. Using the designed QoS relaying mechanism, relaying can be enabled or disabled on a per application basis. Reducing the amount of relaying reduces the network load, keeps the spectrum clean and increases the performance of the network. In this small network with only six nodes, with perfect network operation and optimal functionality without unwanted congestion and packet loss, the number of relayed messages as seen from the experiment is reduced by 75% from 16 to 4 in the acknowledged case, considering all sets of client initiated communication in both the applications with no message repetitions. Considering an even larger and denser network with higher repetition count for each message, valuable network resources can be saved to a large extent. Hence, the usage of intelligent, application-aware relaying in a mesh network turns out to be a critical factor determining the efficiency of a network.

7.2. Repeat Transmission Count

Message repetition increases the chances of message reception in an unacknowledged communication environment. The repetition feature can be achieved from either the application layer or the network layer of the protocol stack, each with their own advantages and disadvantages. Thus, it is important to have the feature configurable at both the layers.

The first experiment shows the variable repetition values configured as part of the QoS against the constant values from the native implementation. It consists of four relay nodes S1, S2, S3 and S4 which are relay enabled and a random message is inserted into the network. The blue bars in Figure 12 show Nordic's implementation without a configurable repetition count and the purple bars indicate the variable repetition count added at the network layer using QoS.

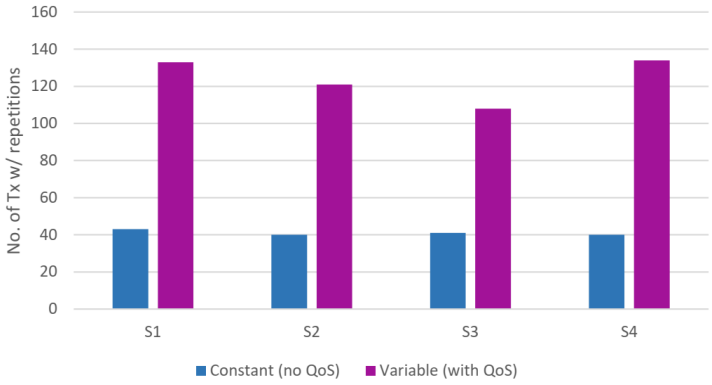

Figure 12. Repeat count feature of QoS.

Message repetitions are important, however, they can be handled by either the application layer or the network layer, each having its own benefits. Application layer message repetition requires more time for message transmission and more space in the message buffer against network layer repetitions. However, application layer repetition enhances end-to-end message reliability against network layer, which is on a per hop basis. The following experiment studied the differences between them both from a latency and memory perspective. It studied a client sending a message with multiple repetitions to a server. First, the repetitions were performed at the application layer and then at the network layer. The time taken to complete the message transmission was measured, and the results are shown in Figure 13.

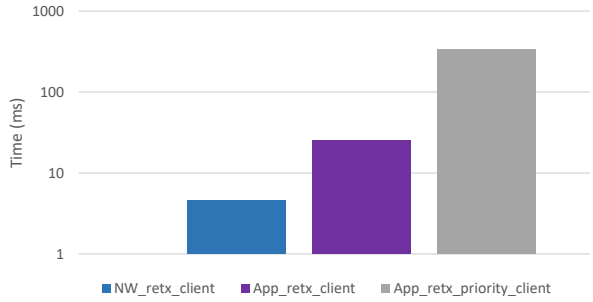

Figure 13. Application layer retransmission vs Network layer retransmission.

The results depict that network layer retransmissions are faster than application layer retransmissions (comparing the first blue bar against the other two). This is expected as application layer retransmissions incur a higher overhead as it happens higher in the protocol stack and has to pass the packet information to the network layer multiple times. On the other hand, with network layer retransmissions, the application layer hands over the packet information to the network layer only once and the network layer takes care of the retransmissions without involving the application layer, thereby reducing the inter-layer communication delay. Moreover, each re-transmitted packet from the application layer has a unique sequence number and hence each unique packet reserves separate memory in the buffer adding more delay and consuming more space, while network layer retransmissions only allocate memory for a single packet with a retransmission counter saving both time and space. In addition, the third bar in grey shows the time taken to send a priority packet (discussed below) from the application layer, which shows even an increased time due to finding the packet in the buffer and re-arranging the buffer. Hence, it shows that application layer repetitions have a higher overhead both in terms of time and space, and network layer repetitions are more useful and lead to an increased network performance.

The next experiment was performed to see the effects of repetition on the reliability of message reception at the receiver, in an environment having different packet error rates. Figure 14 shows the topology of a real BLE mesh network consisting of five nodes having

various link qualities. The numbers against each of the links denote the packet delivery rates (PDRs) in percentage as performed from real experiments in a lab setup. The links which are bi-directional denote consistent PDR in both directions, whereas the scenarios where the links have different PDRs in different directions are shown as unidirectional.

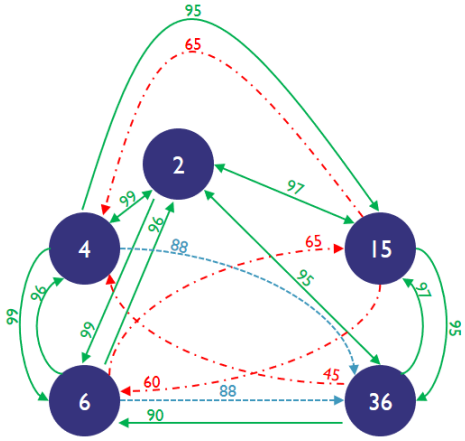

Figure 14. Network topology and PDR representation of real lab setup.

Based on this real-life data, a similar local network was emulated to see the effects of repetition on message reliability. Each node transmits a broadcast message 100 times (application layer) with variable repetition values (network layer) and the messages are checked (application layer) to verify whether they have been received at the receiver's side. The PDR values range from 90% to 45% in different links and the repetition counter is incremented gradually starting with no repetitions.

Figure 15 shows the result where the number of lost packets can be reduced by increasing the repetition count according to the observed PDR values. In addition, for lower PDR values, a higher number of repetitions shows an increase in message reliability, which is the expected outcome.

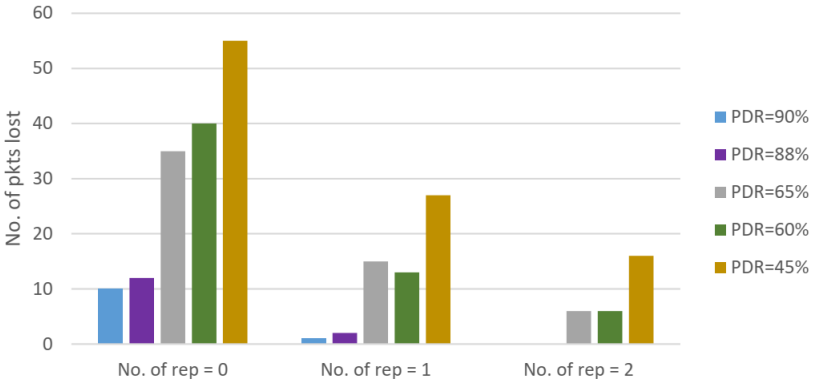

Figure 15. Effect of repetition on packet reception.

Although message repetitions increase reliability, they also come with a price, as, with an increased number of repetitions, the latency of successive message transmissions from a node is also expected to increase. Therefore, the next experiment studies the effect of repetitions on the latency of buffered packets at a node. Four consecutive packets P1, P2, P3 and P4 are sent at the same instant from a source node to a destination node with a single hop distance. The priorities of all the packets are the same, so that the packets are transmitted in the order they are buffered. All packets have their repetition counts set to the same value. The first packet P1 is delayed by 1 s prior to its transmission, followed by the rest of the transmissions of P1 as well as the other remaining packets in succession with a backoff time of 1 ms. The experiment was repeated with variable repetition counts from

4 to 254. The packet reception time at the receiver was noted and the latency of each of the packets was measured from the instant they were queued in the buffer at the sender for transmission. As observed from the outcome in Figure 16, all packets received at the receiver maintain the same order P1, P2, P3 and P4, and each one of them has a higher latency than its previous ones for each of the repetition count values, due to the buffering at the sender's side. The latency of packet P1 is verified to be constant at 1 s, since it is the first packet in the buffer, followed by increased latencies for the other packets. However, it should be noted that, as packets are repeated, the same packet is received multiple times at the receiver. The first successfully received packet is cached at the receiver and the rest are dropped. It might happen that the first packet received at the receiver is the nth packet of the retransmitted packets. As can be seen, the latency of each of the packets increases with the increase in the number of repetitions. With an increase in the backoff time, the latency will increase even more. Hence, the use of repetition should be configured in an optimal way in a BLE mesh network.

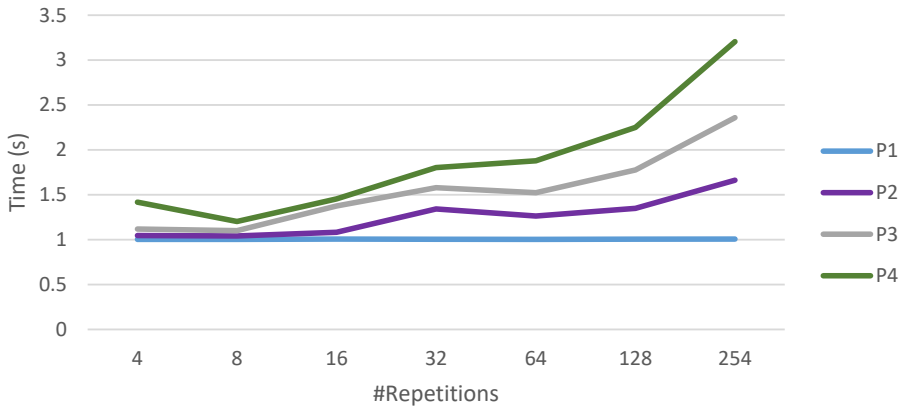

Figure 16. Effect of repetition on packet latency.

Message repetitions also have a price to pay as the network load increases with increasing message repetitions and therefore chances of collision and interference also increases. This is shown in Sections 7.3 and 7.4, as well as how it was mitigated with the application of variable backoff and channel configuration mechanisms.

7.3. Channel Configuration

BLE mesh network use the advertisement mode of BLE by transmitting successively on channels 37, 38 and 39 for every packet transmission. The following, specifically crafted, experiment emulated a dense network scenario to understand how channel configuration might reduce the packet losses due to collision. The experiment was a continuation of the previous experiment with repetitions for packet transmissions, but with changing channel configurations. The topology of the experiment is shown in Figure 17, where a client needs to send a packet to a server. However, the client is not in the direct range of the server and uses a relay in between to reach the server in two hops. The relay is configured with infinite message repetitions, so that it floods the network with relay packets. Next, one more relay is added. The server is configured with a scan interval of 5 s and a scan window of 1 s. The total number of packets received during this scan window is measured for different network configurations regarding the number of relays and variable channel configurations.

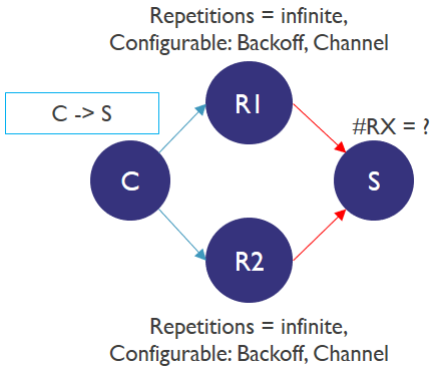

Figure 17. Topology for stress test.

Figure 18 shows the result of the experiment with a constant backoff time of 3 ms for each of the relay packets sent. For a single relay with and without channel randomization, an average of 325 packets (with repetitions) are received at the receiver. The default implementation with two relays results in a wide variation in the number of packets received. The variation depends on the state of the two relays with regard to the channels they receive the data on, as the devices are started at arbitrary times. However, inclusion of channel randomization for two relays has a lower variation, a higher minimum and a lower maximum. This is expected as randomization will nullify the two extremes, one where the two relays receive the packets on the same channel (and hence will lead to collision while transmitting on the default channels) and the other where they receive on different channels (leading to no collisions while transmitting, as the same, very small, backoff is complemented by an offset of the time difference of their reception). This specific example illustrates how the use of the channel configuration mechanism might be beneficial in situations where simultaneous packet transmissions are expected. To further address this issue, the other two channel configuration mechanisms (static and dynamic, as discussed in Section 6.2) are being explored in further research work under study.

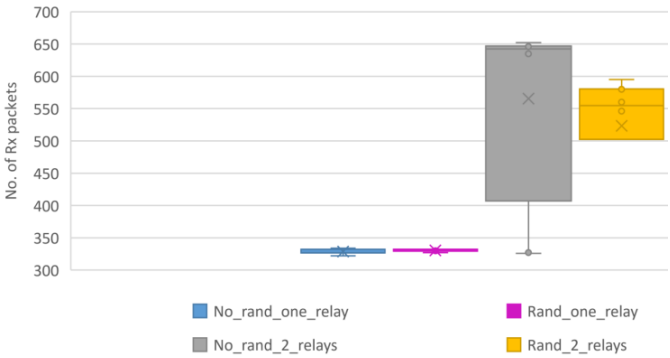

Figure 18. Effect of channel randomization on packet reception in a dense network scenario.

7.4. Backoff Times

The backoff times play an important role as one of the QoS parameters of a node in a BLE mesh network, particularly with respect to latency. This experiment tries to see how a proper configuration of the backoff time leads to a more managed mesh network with better control over the end-to-end latencies to provide a guaranteed network service. The first experiment setup is a simple scenario where a source sends a packet to a destination over a single hop and the latency and jitter are measured. The experiment is carried out with different backoff times. Figure 19 shows the one hop average latency. The first box shows the latency without the application of QoS, which varies between 0 and 20 ms. The second box shows the application of QoS with the backoff time configured to vary between 10 and 100 ms, giving more flexibility for traffic that is not time-critical. In the third case, the backoff time is configured to be between 3 and 10 ms, giving less room for variation

which might be needed for time-critical applications. As per the need of the application, the backoff times can thus be squeezed or expanded. Figure 20 shows the results for the packet jitter for the same setup. It is observed that with the application of configurable backoff times, the jitter can be better controlled. The jitter for the case where backoff time is squeezed between 3 and 10 ms is considerably low, which is beneficial for applications that need real time streaming of data without buffering at the receiver side.

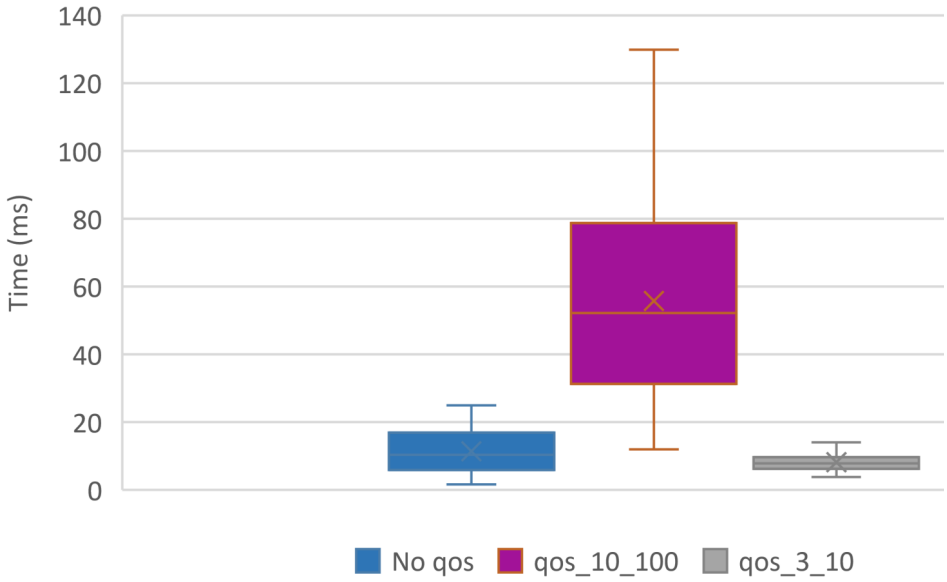

Figure 19. Single hop latency.

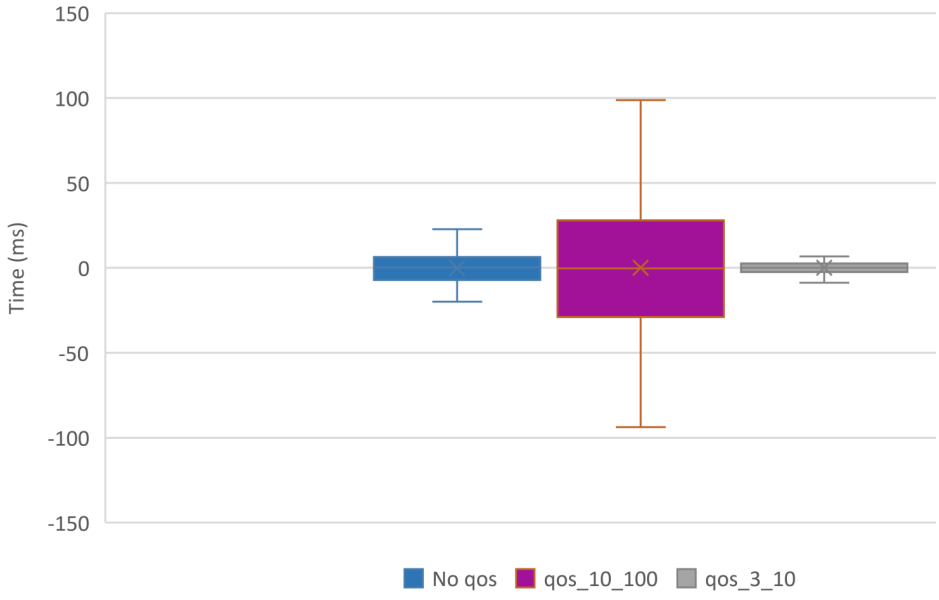

Figure 20. Single hop jitter.

The next experiment extended the previous experiment with three hops by including two relays R in between the client C and server S, as described in Figure 21. The target of this experiment was to achieve a guaranteed end-to-end latency between C and S, with the results shown in Figure 22. In the first case, QoS was not applied leading to an end-to-end latency between 10 and 80 ms. However, next achieving an end-to-end latency of 15 ms was attempted, by configuring the backoff time to be 5 ms for each hop. The achieved latency is around 20 ms, due to processing delays, and shows very little variation. Lastly,

an end-to-end latency of 3 ms was targeted by configuring the backoff latency to be 1 ms for each hop, resulting in an achieved value of 10 ms due to processing delays. In addition, the jitter values were calculated, which are shown in Figure 23. The jitter values rise sharply when extending the hop count from one to three. However, the same can be reduced from 0.4 to 0.1 ms with the configuration of lower backoff times resulting in a decrease of the jitter value by 75%.

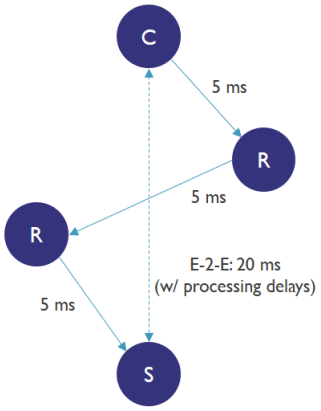

Figure 21. Topology for a three-hop network.

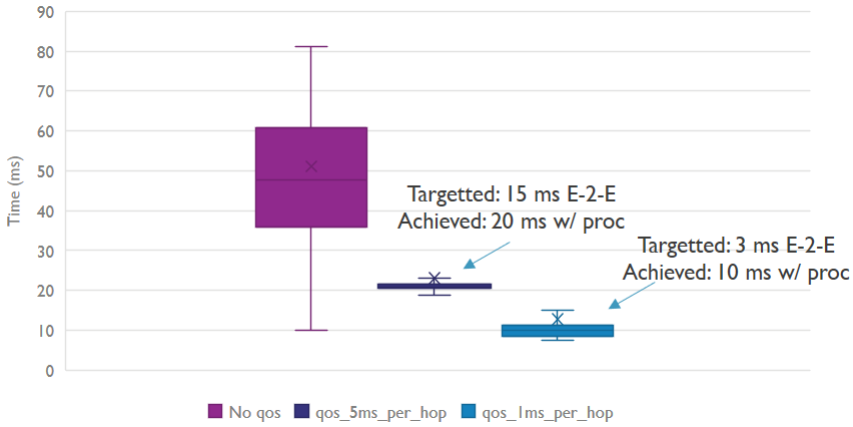

Figure 22. Targeted latency for three hops.

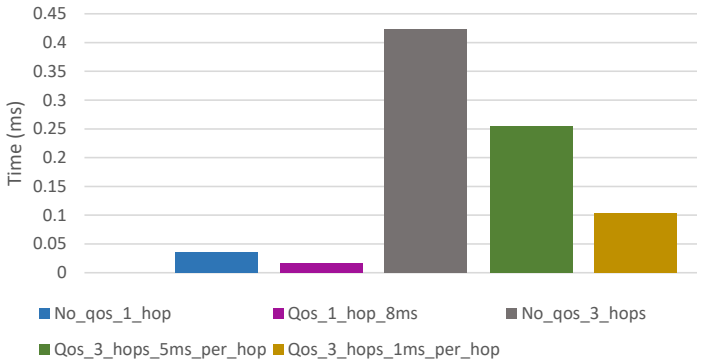

Figure 23. Jitter vs backoff.

The backoff times also play an important role in the packet delivery ratio (PDR) in a BLE mesh network, as shown in the next experiment. The network topology is shown in Figure 24.

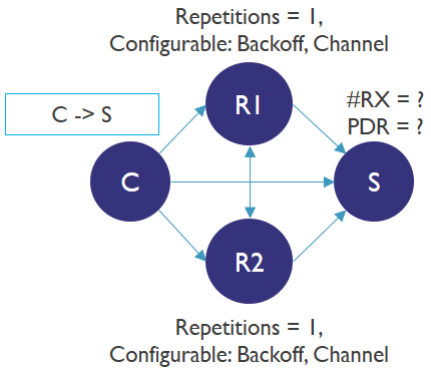

Figure 24. Network topology to test effect of variable backoff on PDR.

It consists of a client and a server and two relays in between. All the nodes are within the reach of one another and are configured with a repetition count of 1. The client sends packets at intervals of 10 ms to the server, with no relays enabled, with either one or both of them enabled. The client sends packets using the default channel configuration of 37, 38 and 39 in sequence. The server scans successively on channels 37, 38 and 39 with a scan window of 2 s, and the resulting PDR is measured at the end of one scan round in all the three channels for a total time of 6 s. The objective was to see how the network reacts to congestion when increasing the number of packet transmissions due to an increased number of relays. The relays are configured first with no QoS features and then with variable backoff times for each of them. As seen from the result in Figure 25, the application of the backoff time as a QoS feature leads to an increased PDR.

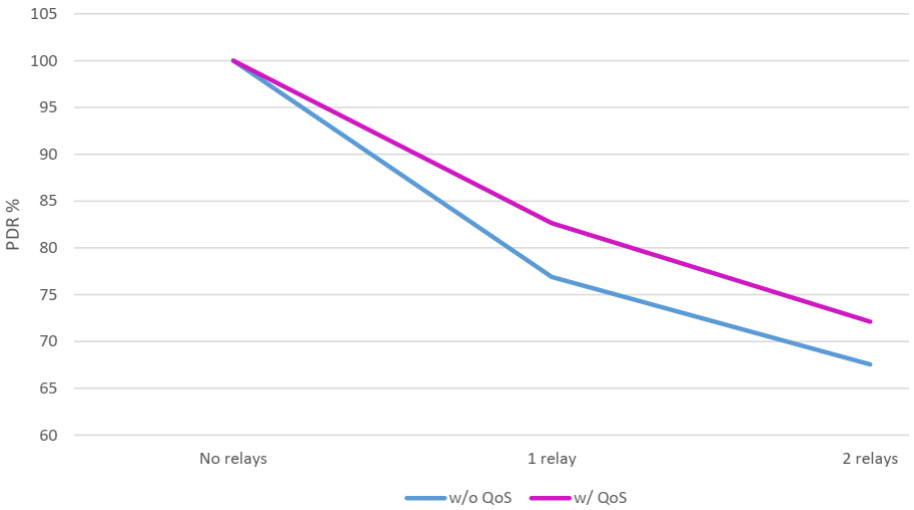

Figure 25. PDR with and without the application of backoff time as a QoS feature.

7.5. Priority Based Traffic Selection for Relaying

The last experiment dealt with handling priority traffic in a BLE mesh network by means of the QoS priority parameter. The topology of the network and the experiment setup is shown in Figure 26. It consists of two clients C1 and C2 and two servers S1 and S2. C1 sends data to S1 and C2 sends data to S2. However, the communication between C1 and S1 has priority over the one between C2 and S2. Both communication paths involve relay R in between, where the packets are queued and relayed. Four consecutive events are fired at an interval of 100 ms, the events being a light switch *on* request from C1 to S1, a light switch *off* request from C2 to S2, a light switch *on* request from C2 to S2 and a light switch *off* request from C2 to S2. However, at R, all events are configured to have a backoff time of 1 s. This ensures that all four events with different priorities are queued at the relay at the same time.

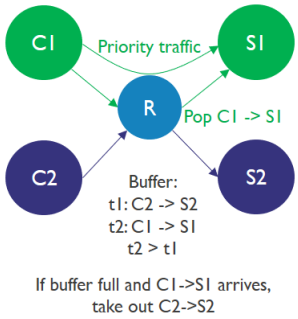

Figure 26. Topology for priority traffic.

The result is shown in Figure 27, with the sequence of events shown with and without using QoS. When QoS is not used, the events follow the same sequence as enqueued in the buffer at the relay. However, with the application of QoS capabilities, it is noted that the sequence of the events change with the communication between C2 and S2 immediately following after the first communication between C1 and S1, as traffic between C1 and S1 is prioritized over traffic between C2 and S2. Since the implementation is non-preemptive, the very first event is always the first one to be executed, with the later events being considered for the priority traffic handling. This way different traffic types with different priority levels can be configured for more fine-grained data flow handling.

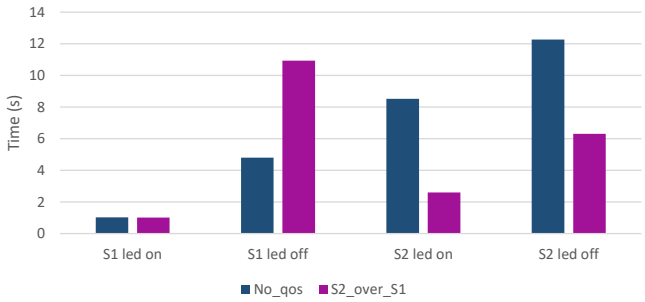

Figure 27. Priority traffic handling.

8. Conclusions and Future Work

IoT applications are witnessing a rapid increase in most home, work-space and industrial areas today. With the proliferation of new wireless communication and networking technologies, it is possible to choose from a wide range of them according to the specific application needs. The technology of mesh networks has some interesting advantages over other types of networks, as they operate in a distributed fashion removing a single point of failure and can span large areas with limited infrastructure. BLE mesh networks are thus a viable option for emerging smart applications. However, the flexibilities gained in these types of networks come at a price of increased complexity. As networks become denser, network performance decreases and an efficient management of them is necessary to ensure a decent QoS. This has been observed in BLE mesh networks, and it will become more pronounced in the heterogeneous BLE mesh networks we envision in the future. Therefore, this paper focuses on including optional, but standard-compliant, QoS features into a BLE mesh network. It is shown how the inclusion of these QoS features can play an effect to the overall performance of the network. The applied features involve configuration of the individual nodes with relay choice, backoff times, repetition counts, channel configuration and priority traffic handling on a per application level. The experiments were performed and assessed against the standard implementation without these features and the results show that the proposed QoS features make it possible to more optimally configure the network, yielding interesting opportunities for advanced management of heterogeneous BLE mesh networks. The present implementation involves the static configuration of the nodes and only considers the potential of the QoS features on a small scale network. As a

next step, we will target the remote management of BLE mesh networks as well as a larger-scale validation of the proposed concepts. This involves collection of run-time statistical data of the network characteristics. Based on the obtained information, the design of an algorithm that automatically generates QoS configuration for the individual nodes will be studied. Such an algorithm should consider, amongst others, the topology, link states, PDR and other application requirements to determine the best possible paths and reduce unnecessary message flooding. These configuration settings will then be distributed to the nodes over the air by the use of a configuration model, following the model approach taken by BLE mesh.

Author Contributions: Conceptualization, J.H.; methodology, S.S.B. and J.H.; software, S.S.B.; validation, S.S.B.; formal analysis, S.S.B., M.B. and J.H.; investigation, S.S.B., M.B. and J.H.; resources, M.B. and J.H.; data curation, S.S.B., M.B. and J.H.; writing—original draft preparation, S.S.B.; writing—review and editing, S.S.B., M.B. and J.H.; visualization, S.S.B., M.B. and J.H.; supervision, J.H.; project administration, J.H.; and funding acquisition, J.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was co-financed by imec and received project support from Flanders Innovation & Entrepreneurship (project No. HBC.2018.0530).

Data Availability Statement: Not applicable.

Acknowledgments: This work was executed within the imec.icon project BLUESS, a research project bringing together academic researchers and industry partners.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fortino, G.; Savaglio, C.; Spezzano, G.; Zhou, M. Internet of Things as System of Systems: A Review of Methodologies, Frameworks, Platforms, and Tools. *IEEE Trans. Syst. Man Cybern. Syst.* **2020**, *51*, 223–236. [CrossRef]
2. Sikder, A.K.; Acar, A.; Aksu, H.; Uluagac, A.S.; Akkaya, K.; Conti, M. IoT-enabled smart lighting systems for smart cities. In Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–10 January 2018; pp. 639–645.
3. Bluetooth SIG. Bluetooth® Mesh Networking Specifications. 2018. Available online: https://www.bluetooth.com/specifications/mesh-specifications/ (accessed on 16 February 2021).
4. Baert, M.; Rossey, J.; Shahid, A.; Hoebeke, J. The Bluetooth mesh standard: An overview and experimental evaluation. *Sensors* **2018**, *18*, 2409. [CrossRef]
5. Rondón, R.; Mahmood, A.; Grimaldi, S.; Gidlund, M. Understanding the Performance of Bluetooth Mesh: Reliability, Delay, and Scalability Analysis. *IEEE Internet Things J.* **2019**, *7*, 2089–2101. [CrossRef]
6. Almon, L.; Álvarez, F.; Kamp, L.; Hollick, M. The King is Dead Long Live the King! Towards Systematic Performance Evaluation of Heterogeneous Bluetooth Mesh Networks in Real World Environments. In Proceedings of the 2019 IEEE 44th Conference on Local Computer Networks (LCN), Osnabrueck, Germany, 14–17 October 2019; pp. 389–397.
7. Pérez-Díaz-De-Cerio, D.; García-Lozano, M.; Bardají, A.V.; Valenzuela, J.L.; Hernández-Solana, A. Bluetooth Mesh Analysis, Issues, and Challenges. *IEEE Access* **2020**, *8*, 53784–53800.
8. Leonardi, L.; Patti, G.; Bello, L.L. Multi-hop real-time communications over bluetooth low energy industrial wireless mesh networks. *IEEE Access* **2018**, *6*, 26505–26519. [CrossRef]
9. Hansen, E.A.; Nielsen, M.H.; Serup, D.E.; Williams, R.J.; Madsen, T.K.; Abildgren, R. On relay selection approaches in Bluetooth mesh networks. In Proceedings of the 2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Moscow, Russia, 5–9 November 2018; pp. 1–5.
10. Darroudi, S.M.; Gomez, C.; Crowcroft, J. Bluetooth Low Energy Mesh Networks: A Standards Perspective. *IEEE Commun. Mag.* **2020**, *58*, 95–101. [CrossRef]
11. Zenker, P.; Krug, S.; Binhack, M.; Seitz, J. Evaluation of BLE Mesh capabilities: A case study based on CSRMesh. In Proceedings of the 2016 Eighth International Conference on Ubiquitous and Future Networks (ICUFN), Vienna, Austria, 5–8 July 2016; pp. 790–795.
12. Hu, H.; Zhang, Y.; Chen, H.H. An effective QoS differentiation scheme for wireless mesh networks. *IEEE Netw.* **2008**, *22*, 66–73.
13. Nordic Semiconductor. nrf5 SDK for Mesh. 2018. Available online: https://www.nordicsemi.com/Software-and-tools/Software/nRF5-SDK-for-Mesh (accessed on 16 February 2021).
14. Nordic Semiconductor. nRF52840 DK: Bluetooth Low Energy, Bluetooth mesh, NFC, Thread and Zigbee Development Kit for the nRF52840 SoC. 2018. Available online: https://www.nordicsemi.com/Software-and-Tools/Development-Kits/nRF52840-DK (accessed on 16 February 2021).