applied system innovation

Article

RESEMBLE: A Real-Time Stack for Synchronized Mesh Mobile Bluetooth Low Energy Networks

Luca Leonardi, Lucia Lo Bello * and Gaetano Patti

Department of Electrical, Electronic and Computer Engineering, University of Catania, Viale A. Doria 6,
95125 Catania, Italy
* Correspondence: lucia.lobello@unict.it

Abstract: Bluetooth Low Energy (BLE) is a wireless technology for low-power, low-cost and low-complexity short-range communications. On top of the BLE stack, the Bluetooth Mesh profile can be adopted to handle large networks with mesh topologies. BLE is a promising candidate for the implemention of Industrial Wireless Sensor Networks (IWSNs), thanks to its wide diffusion (e.g., on smartphones and tablets) and the lower cost of the devices compared to other wireless industrial communication technologies. However, neither the BLE nor the Bluetooth Mesh specifications can provide real-time messages with bounded delays. To overcome this limitation, this work proposes RESEMBLE, a real-time stack developed on top of BLE that is able to realize low-cost IWSNs over mesh topologies. RESEMBLE offers support to both real-time and non-real-time communications on the same network. Moreover, RESEMBLE provides clock synchronization, thus allowing for Time Division Multiple Access (TDMA) transmissions. The clock synchronization provided by RESEMBLE can be also exploited by the upper layers' industrial applications to implement time-coordinated actions.

Keywords: Bluetooth Low Energy; mesh networks; real-time networks

check for updates

Citation: Leonardi, L.; Lo Bello, L.; Patti, G. RESEMBLE: A Real-Time Stack for Synchronized Mesh Mobile Bluetooth Low Energy Networks. *Appl. Syst. Innov.* **2023**, *6*, 19. https://doi.org/10.3390/asi6010019

Academic Editor: Christos Douligeris

Received: 6 January 2023
Revised: 21 January 2023
Accepted: 24 January 2023
Published: 26 January 2023

1. Introduction and Motivation

Thanks to Industry 4.0, factories have significantly evolved in recent years. New devices and tools, such as smart sensors, tablets, advanced displays and monitors with effective user interfaces and virtual assistants, are gradually replacing traditional devices and conventional interfaces [1–4]. Moreover, novel functions, such as the monitoring of production levels, control and predictive maintenance of the plant basic facilities and digitalized management of the supply chain, are gaining ground. In this context, low-cost communication systems are one of the enabling technologies for modern industrial applications, as they make it possible to monitor all phases of the production chain [5]. However, industrial communications pose several challenging requirements, such as predictability (i.e., the ability to provide guaranteed upper bounds on the delivery times of time-critical messages), mobility support (i.e., connectivity for mobile nodes) and high reliability (i.e., low error probability and low packet loss). To meet these requirements, industrial communication systems adopt specifically devised protocols that are typically quite expensive and not used beyond the scope of industrial scenarios.

This consideration is the motivation behind this work, which presents a real-time stack developed on top of Bluetooth Low Energy (BLE). The proposed approach, called RESEMBLE, enables the use of BLE, i.e., a low-cost, widespread consumer communication technology, in the industrial field. What is new in RESEMBLE is the ability to support real-time communications and node mobility in BLE networks with mesh topologies. The current Bluetooth Mesh specifications do not include support for real-time traffic; therefore, Bluetooth Mesh is not suitable for Industrial Wireless Sensor Networks (IWSNs). To solve this limitation and enable the BLE technology to be adopted in IWSNs, RESEMBLE introduces the ability to provide guaranteed bounded delays to real-time messages. Moreover, RESEMBLE offers connectivity to mobile nodes.

RESEMBLE is a low-cost software solution that can be implemented on common mobile devices (e.g., smartphones and tablets) without hardware modifications.

The rest of this paper is organized as follows. Section 2 presents and discusses related work. Section 3 provides an overview of the Bluetooth Mesh stack and discusses its limitations for real-time communications. Sections 4 and 5 describe RESEMBLE's design and implementation, respectively. Section 6 presents an experimental assessment of RESEMBLE on commercial-off-the-shelf (COTS) devices and discusses the synchronization accuracy and end-to-end delays that were obtained. Finally, Section 7 gives conclusions and hints for future works.

2. Related Work

Many works in the literature investigated the adoption of BLE for a wide range of applications, such as home automation [6,7], Internet of Things (IoT) [8–10], passive human sensing [11], healthcare and wearable devices [12,13]. Several works also investigated the case for BLE in industrial communications [14–18]. For instance, the work in [16] addressed BLE's suitability for industrial automation communications, while the work in [18] focused on Industrial IoT applications. However, the limited transmission range represents an issue for BLE's use in large industrial networks. To solve this issue, the work in [19] proposed extending the coverage of BLE clusters by leveraging LoRa technology. Another approach to solve this issue is to enable multihop transmissions over BLE. To this aim, the work in [20] proposed a connection-oriented approach for BLE networks that supports tree topologies. However, the approach does not support mobility and real-time communications, which are typical requirements for industrial applications. A solution, aiming to achieve real-time communications over BLE networks with mesh topologies, is proposed in [21]. The proposed protocol, called MRT-BLE, allows for real-time transmissions and obtains maximum delays from hundreds to thousands of milliseconds. However, MRT-BLE does not provide mobility support, as it works in a connection-oriented way.

To increase the BLE coverage, the Bluetooth SIG defined the Bluetooth Mesh Profile [22], a stack operating on top of BLE that supports node mobility and mesh topologies. The work in [23] evaluated the end-to-end delays and the network-joining phase duration of Bluetooth Mesh (called provisioning) and proposed a lightweight provisioning protocol to speed up the network-joining time of the nodes. The work in [24] evaluated Bluetooth Mesh in a mobile robotic system for logistics and showed its suitability for this kind of applications. However, Bluetooth Mesh lacks the support for real-time communications, as it cannot guarantee bounded delays. An way to deal with Quality-of-Service (QoS) in Bluetooth Mesh is presented in [25], where a transmission backoff algorithm allows for the message priorities to be differentiated. However, this approach is not able to provide bounded delays.

In industrial environments, both the timeliness of the communication and their robustness to noise and interference [26] are important to avoid retransmissions that may affect the end-to-end delays. For this reason, Bluetooth Mesh was assessed in [27] and the results demonstrated that, in dense networks, Bluetooth Mesh obtains a high packet error rate, mainly due to collisions. The adoption of dynamic traffic smoothing techniques inspired by those proposed in [28,29] can help reduce the collision probability.

As will be discussed in the following sections, the RESEMBLE stack that is proposed here fills a gap, as it introduces both real-time communications and node mobility over mesh topologies to BLE. Moreover, RESEMBLE can be implemented on COTS devices with no hardware modifications.

3. Overview of the Bluetooth Mesh Architecture

The Bluetooth Mesh Profile specifications [22] define a layered architecture that operates on top of the BLE stack. The layers of this architecture are shown in Figure 1.

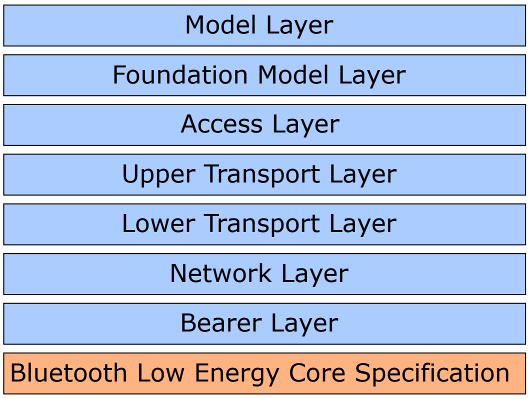

Figure 1. Bluetooth Mesh Profile stack.

At the bottom layer of the protocol stack, there is the BLE Core Specification (lower layer in Figure 1). The Bluetooth Mesh architecture operates on top of this layer.

The Bearer Layer interfaces BLE with the layers of the mesh network architecture. The specifications provide two kinds of bearers:

- Advertising Bearer. This uses the advertisement, i.e., messages that are periodically broadcast.
- GATT Bearer. This operates on top of the Generic Attribute Profile (GATT) Layer of BLE. The GATT defines a connection-oriented application protocol for BLE. The GATT Bearer was defined to maintain the backward compatibility with the devices that support connection-oriented transmissions only.

The Network Layer is in charge of the encryption/decryption, authentication and forwarding to the upper layers of the received messages and their retransmission to the other nodes. The routing algorithm is based on flooding, i.e., a node that receives a message addressed to another node and retransmits the message. In order to avoid loops and unlimited retransmissions, the Network Layer adopts two mechanisms. The first mechanism is based on a cache that stores the received messages, while the second is based on the message time-to-live (TTL), i.e., the number of hops that have been traversed. The cache mechanism avoids forwarding messages that have already been received by storing a copy of each message in a cache maintained at the Network Layer. When a new message is received, the Network Layer inspects the cache to check whether the message has already been received. If this is the case, the new message is dropped; otherwise, it is stored in the cache and then processed. Moreover, every Network Layer message maintains a TTL field to limit the number of times the message can be forwarded.

The Lower Transport Layer segments the messages received from the upper layer to allow for their transmission on the network and reassembles the segments received from the Network Layer. To increase the communication reliability, this layer adopts a segment acknowledgement and retransmission mechanism.

The Upper Transport Layer is mainly in charge of encrypting and decrypting the messages received from (or addressed to) the upper layer.

The Access Layer interfaces the Upper Transport Layer with the Foundation Model Layer and the Model Layer for the transmission of data messages. The Access Layer supports both reliable (through an acknowledgement & retransmission mechanism) and non-reliable message transmissions and provides a "Publish and Subscribe" communication mechanism.

The Foundation Model Layer defines the states, the messages and the models to configure and manage the mesh network, such as the configuration of the communication models for the Model Layer, the management of the health status of the network nodes or service, etc.

The Model Layer defines the communication models to standardize the operations of specific user scenarios, such as light management, sensor data acquisition, etc.

3.1. Limits of Bluetooth Mesh in Supporting Real-Time Communications

Bluetooth Mesh operates on top of the BLE stack and transmits messages in a connectionless way. Thanks to this property, Bluetooth Mesh is able to support mobility on mesh topologies. However, the BLE stack is not able to offer predictable communications on mesh networks, as there are several sources of non-determinism. The following subsections describe the BLE transmission mechanisms used in Bluetooth Mesh and discuss why these mechanisms are not able to support real-time traffic over mesh networks.

3.1.1. Bluetooth Low Energy v.4.2

The BLE 4.2 [30] specifications for connectionless message transmissions define the Broadcaster/Observer communication paradigm, in which the Broadcaster transmits the advertisements (ADVs) containing data and the Observer receives them (the Broadcaster and Observer roles can alternate). For the ADV transmission, the specifications allow for the adoption of only 3 of the 40 channels available in BLE. The ADVs are transmitted within a time interval called an advertisement event, as depicted in Figure 2. Within an advertisement event, multiple ADVs can be transmitted, each one being transmitted on a different channel (starting with the first channel reserved for the ADVs). The time interval between two consecutive ADVs within the same advertisement event has to be lower than or equal to 10 ms.

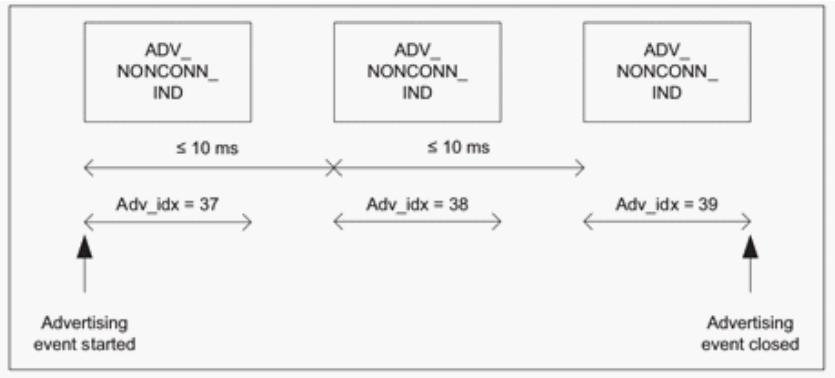

Figure 2. Advertisement event.

The advertisement event periodically repeats with period $T_{advEvent}$. This period is the sum of a constant time and a pseudo-random time, as shown in (1)

$$T_{advEvent} = advInterval + advDelay \tag{1}$$

where *advInterval* is a constant in a range between 20 ms (100 ms in the case of non-connectable ADVs) and 10.24 s and varies in steps of 0.625 ms. Instead, *advDelay* is a pseudo-random time between 0 and 10 ms that is calculated at every advertisement event. As far as the Observer role is concerned, a reception window is defined, called a *scanWindow*, which periodically repeats with period *scanInterval*. Both the *scanWindow* and the *scanInterval* can be configured in a range from 0 and 10.24 s, with *scanWindow* lower than or equal to *scanInterval*.

Using the Broadcaster/Observer paradigm described above, there is no synchronization between the transmissions of nodes. This entails a high probability of message collisions, as the ADV transmissions from two or multiple nodes may overlap. Looking at the example in Figure 3, there are two Broadcasters (B1 and B2) that transmit ADVs (for the sake of simplicity, we assume that a single message can be transmitted within each advertisement event). The Observer (O1) will not receive any message during the first two advertisement events of both the Broadcasters, as the messages collide. At the third advertisement event, the Observer will receive both messages.

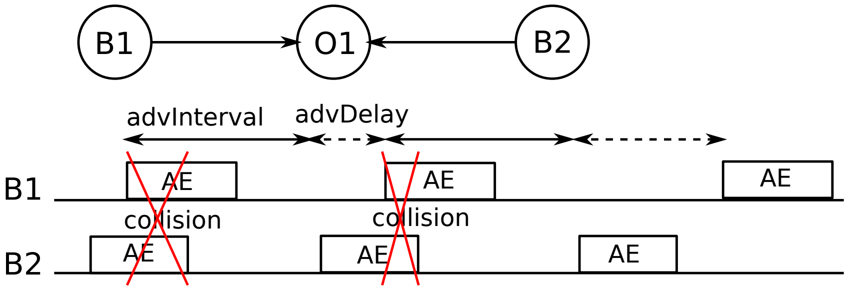

Figure 3. Advertisement collision example.

The example presented in Figure 3 shows that the ADV transmission mechanism defined in the BLE specifications is a source of non-determinism. In fact, the only component that avoids the indefinite repetition of collisions is a random interval (i.e., advDelay), and this leads to unpredictability, as the nodes that caused a collision may cause collisions again. Moreover, as there are only three advertisement channels (fixed by the standard) and their use cannot be controlled for each message (as they are used in sequence within an advertisement event), the collision probability remains high. The BLE specifications, in the case of ADV transmissions, do not provide any acknowledgement mechanism, so the error management is delegated to the upper layers.

3.1.2. Bluetooth Low Energy v.5.3

The BLE v.5.3 specifications [31] introduce several innovations in the transmission of ADVs while maintaining the interoperability with the previous specifications. The most important innovation is the ability to use all 37 BLE data channels for the transmission of auxiliary messages within the same transmission event. The specifications provide two kinds of channels for the transmission of ADVs, i.e., the primary and the secondary channels. The primary channels are the three channels dedicated to the ADV that were already used in the previous version of BLE, while the secondary channels are the remaining 37 channels that, in the previous versions of BLE, could not be used for transmitting ADVs.

An advertisement event always starts with the transmission of an ADV on one of the three primary channels, and this limits the possibility of exploiting channel diversity to avoid collisions. This limit can be partially overcome defining suitable mechanisms to choose the channel for the auxiliary messages, but a mechanism to avoid collisions on the primary channels still needs to be defined. The considerations of the determinism for BLE v.4.2 are also valid for BLE v.5.3. However, the BLE v.5.3 specifications define several mechanisms and multiple transmission methods for the ADVs. This increases the protocol complexity but, at the same time, opens new ways to design medium access mechanisms that support real-time communications. For example, the chance to use all 37 channels for the transmission of the ADVs, combined with the ability to set the channel to be used, enables the adoption of multichannel communication mechanisms that allow for multiple nodes to transmit at the same time on different channels.

4. RESEMBLE Design

The findings of Section 3 show that the main limitations to supporting real-time communications derive from the lack of synchronization between the nodes in the BLE layer. In particular, the adoption of ADVs is the main source of non-determinism and unpredictabilities in the communication delay. However, connectionless transmissions provide very low start-up times for links, as there is no need to establish connections. For this reason, the adoption of a connectionless approach may be a good solution.

To enable the adoption of Bluetooth Mesh networks for industrial communications, the following goals have to be achieved:

- Goal 1. Providing a medium access protocol that allows for predictable message transmissions with bounded delays and no collisions. The first step to achieve this goal is to define a synchronization mechanism for the node communication. To this end, a novel medium access mechanism, operating over BLE v.4.2 and BLE v.5.3, is required.
- Goal 2. Introducing message priority in order to adopt real-time scheduling algorithms.
- Goal 3. Enabling the separation between real-time and non-real-time traffic through the establishment of two "logical" channels, a real-time channel and a non-real-time one.

The solution proposed in this work to achieve real-time communication over Bluetooth Mesh networks is based on the design of a novel Bearer Layer (RESEMBLE Bearer Layer in Figure 4) that is able to provide a TDMA-based medium access mechanism while maintaining compliance with the Bluetooth Mesh specifications, i.e., using non-connectable undirected advertising events.

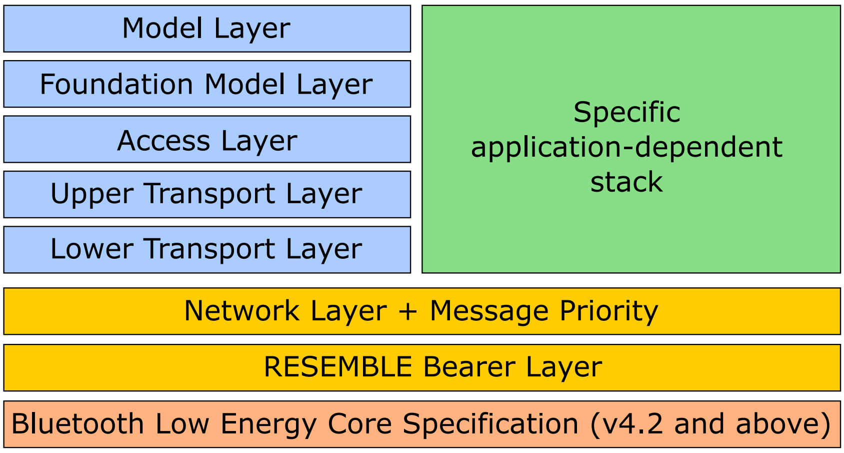
Model Layer
Foundation Model Layer
Access Layer
Upper Transport Layer
Lower Transport Layer
Specific application-dependent stack
Network Layer + Message Priority
RESEMBLE Bearer Layer
Bluetooth Low Energy Core Specification (v4.2 and above)

Figure 4. RESEMBLE stack.

The network stack is made up of two layers. On top of the stack, there is an optimized version of the Network Layer, in which the flooding mechanism (already defined in the Bluetooth Mesh specifications [22]) is optimized to support message priority. Below the Network Layer, the RESEMBLE Bearer Layer guarantees bounded latencies and collision-free transmissions. The layers that are on top of the Network Layer in the Bluetooth Mesh standard are not addressed in this design for two reasons:

1. Industrial networks generally have a very minimal stack, as each layer introduces delays and, in some cases, such delays can be unpredictable.
2. The functionalities provided by the upper layers are application-dependent.

The different design steps are as follows:

1. Enabling TDMA-based communications. This requires a clock synchronization mechanism, as the nodes have to know the exact time at which they can transmit.
2. Providing a medium-access mechanism to allow unsynchronized nodes to transmit/receive both synchronization and provisioning messages without interfering with TDMA transmissions.
3. Defining a clock synchronization mechanism that is suitable for Bluetooth Mesh communications. To achieve clock synchronization, nodes have to exchange synchronization messages with each other, but without interfering with the TDMA transmissions of the synchronized nodes.
4. Introducing message priority to cope with the different temporal requirements of the messages.

These steps are outlined in the following subsections. A detailed description of the transmission mechanism is not possible at this time, as there is a patent pending (International patent pending No. PCT/IT2022/050003); therefore, this is left for future work. Section 5 addresses how to implement RESEMBLE on COTS devices and the results in Section 6 show its effectiveness.

4.1. How to Enable Unsynchronized Node Transmissions

To enable TDMA-based communications, the nodes have to be synchronized, so that each node knows when it can transmit. However, to be synchronized, the nodes need to transmit messages without interfering with regular transmissions. In the case a node wants to join the network, its transmissions have to be regulated so that they do not interfere with the transmissions of the other nodes.

To achieve this, in RESEMBLE, time is divided into time intervals, called superframes (Figure 5), which are, in turn, partitioned in timeslots. Each network node is statically assigned one or multiple timeslots to transmit a single message using the ADV event. In the remaining time of the superframe, the node remains in scanning state to receive the messages transmitted from the other nodes. With this approach, each node knows the exact time at which it can transmit a message and is also aware of the transmission times of the other nodes. One or multiple timeslots within the superframe are marked as shared timeslots (SH in Figure 5) and are to transmit non-real-time messages. The transmission within shared timeslots is regulated by a random backoff algorithm. Before starting the transmission, a node with an outgoing non-real-time message has to wait a random number of shared timeslots, thus decreasing the collision probability.

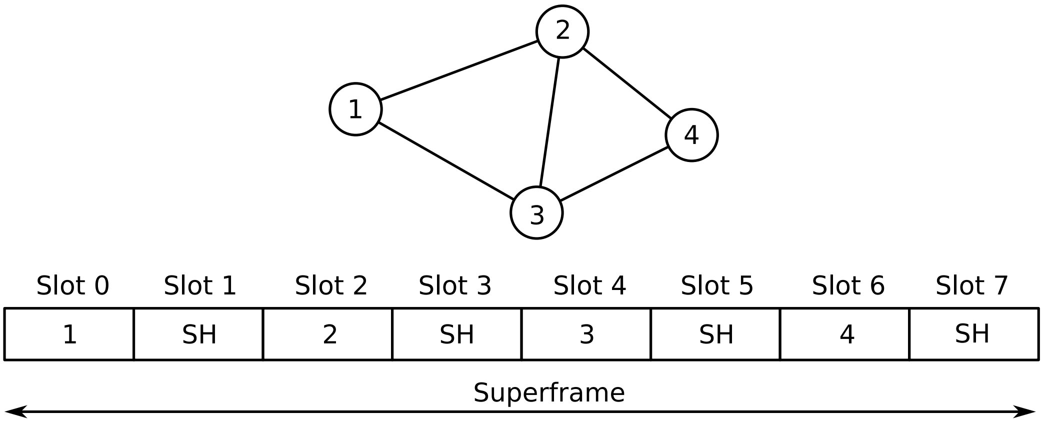

Figure 5. Example of RESEMBLE superframe.

The proposed approach works in two phases:

1. Synchronization and provisioning.
2. Online phase. Nodes are synchronized; therefore, they can transmit.

The phase in which a node joins the network is called provisioning. Here, the provisioning procedure is the one specified in the Mesh BLE standard [22] with no modifications. The specifications [22] define "unprovisioned" as a node that has not joined the network and "provisioned" as a node that joined the network. The provisioning phase is managed by a single node, called the Provisioner. During the synchronization and provisioning phase, a new node willing to join the network exchanges messages with the Provisioner node (i.e., the node that manages the joining phase of new nodes in the network) to run the provisioning procedure to join the network. In the proposed approach, the Bearer Layer transmits provisioning messages in the non-real-time channel. This way, no modifications are needed to the provisioning procedure, as the Bearer acts in a transparent way. As clock synchronization is required, in addition to the provisioning procedure, the clock synchronization procedure has to be executed in this phase. In our design, clock synchronization is achieved with a centralized approach, in which a node with timing master functionalities (for instance, the Provisioner) provides the reference clock for all network nodes. Note that clock synchronization is used during the online phase to schedule real-time transmissions. Once synchronized, the nodes enter the online state, in which they communicate on the real-time channel.

4.2. Clock Synchronization

The synchronization protocol is inspired by the concept of media-dependent layer of the IEEE 802.1AS [32] for wireless communications. Synchronization messages are always transmitted in the non-real-time channel. Clock synchronization is achieved on a link-by-link basis, exchanging synchronization messages. The Timing Master synchronizes its neighbor nodes that, in turn, synchronize their neighbor nodes that have a higher hop count from the Timing Master.

4.3. Delay Bound Calculation for Real-Time Transmissions

This subsection defines the end-to-end delay (e2eDelay) of a message and describes the calculation of the end-to-end delay bound of a flow in RESEMBLE.

The message end-to-end delay is defined as the difference between the message generation time at the source and the message reception time at the destination, measured at the Network Layer. The e2eDelay is the sum of two components, i.e., the message queuing time and the message transmission time, at each hop in the path, from the source to the destination. In this analysis, we assume that the message transmission time is equal to the timeslot duration (T_{ts}), as the latter is configured to allow for the complete message transmission. As a result, in the worst-case e2eDelay analysis presented here, the e2eDelay bound, i.e., the upper bound of the delay that a message of the flow may experience, is given by the following equation

$$e2eDelay\_bound_i = h \times T_{ts} + \sum_{k=1}^{h} T_{q_i}^{(k)} \tag{2}$$

where h is the number of hops a message of the i-th flow has to traverse to reach the destination, T_{ts} is the timeslot duration and $T_{q_i}^{(k)}$ is the maximum queuing delay of a message belonging to the i-th flow at the k-th hop.

To determine the worst case $T_{q_i}^{(k)}$ at each node traversed by a message of the i-th flow, first, both the resource availability and the worst case for the resource request are calculated. In RESEMBLE, the resource availability for the k-th node is given by the number of timeslots that the node n_k can use to transmit, while the resource request is the number of timeslots needed to transmit the packets generated or forwarded by the node n_k. Based on these definitions, the analysis can be performed by applying the worst-case response time (WCRT) analysis that was proposed in [33] and extended with the "busy period" approach in [34–36]. The analysis has already been applied to TDMA communications in [37,38].

At the Network Layer, RESEMBLE adopts the flooding mechanism; therefore, the messages are transmitted over multiple paths. For this reason, the *e2eDelay_bound* is calculated considering all the possible paths for a flow from the source to the destination.

5. RESEMBLE Implementation

As described in Section 4, the main component of RESEMBLE is the Bearer Layer, which replaces the Advertising Bearer defined in the Bluetooth Mesh specifications [22]. This component is able to provide:

- A TDMA-based medium access mechanism for the transmission of real-time messages over BLE.
- A medium access mechanism to allow for both unsynchronized and synchronized nodes to transmit/receive non-real-time messages, such as synchronization and provisioning messages, without interfering with the real-time transmissions.

- A centralized clock synchronization protocol suitable for Mesh BLE communications.
- Support to message priority, to cope with the different temporal requirements of the messages.

The details of the RESEMBLE software components are depicted in Figure 6.

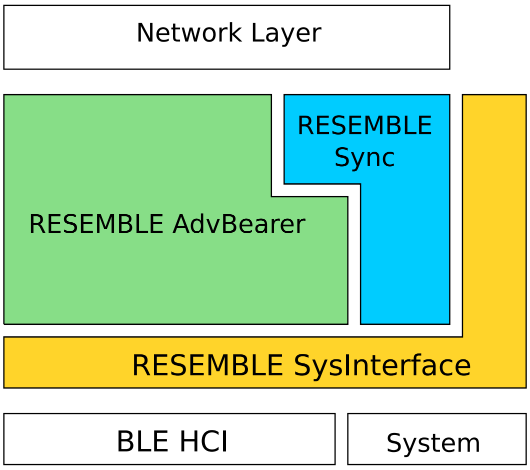

Figure 6. RESEMBLE software components.

The RESEMBLE Bearer consists of three main modules:

- **The System Interface.** This module interfaces the AdvBearer module with the BLE Host Controller Interface (HCI) and System. Moreover, it maintains the clock that provides the network timings and is synchronized with the reference clock maintained by a node (the timing master) in the network. This component is platform-dependent, i.e., its implementation depends on the platform on which it runs.
- **The AdvBearer module.** This module implements the transmission mechanism that handles both real-time and non-real-time communications.
- **The Sync module.** This module implements the clock synchronization protocol. It is strictly coupled with the AdvBearer module.

6. Experimental Assessment

This section presents an experimental performance assessment of RESEMBLE aiming to both evaluating the clock synchronization accuracy at different hop distances and assessing the maximum end-to-end delays in a mesh topology. The assessment also demonstrates the feasibility of RESEMBLE on COTS devices.

The devices used for the experiment are the STMicroelectronics NUCLEO-L476RG [39] microcontrollers, equipped with the BLE controller provided in the X-NUCLEO-IDB05A2 [40] expansion board. The BLE controller is compliant with the BLE v4.2 [30] specifications.

The assessed metrics are:

- The clock skew (S), measured as the clock difference between two generic nodes (N_x, N_y) at the time t, i.e.,

$$S(N_x, N_y) = ClockValN_x(t) - ClockValN_y(t) \tag{3}$$

  where $ClockValN_x(t)$ is the clock value of the Node N_x at time t.
- The Maximum Absolute Clock Skew (MaxAbsSkew), calculated as

$$MaxAbsSkew(N_x, N_y) = max(|S(N_x, N_y)|) \tag{4}$$

- The end-to-end delay ($e2eDelay$), measured above the Network Layer as the difference between the message generation time at the source and the message reception time at the destination.

$$e2eDelay = RxTime - GenTime \tag{5}$$

In the following, Section 6.1 describes the experiment aiming to evaluate the clock synchronization accuracy of the RESEMBLE synchronization protocol, while Section 6.2 discusses the experiment aiming to evaluate the obtained *e*2*eDelays*.

6.1. Clock Synchronization Accuracy Assessment

The assessed scenario, shown in Figure 7, consists of four stationary nodes connected in a line topology. The parameters are presented in Table 1.

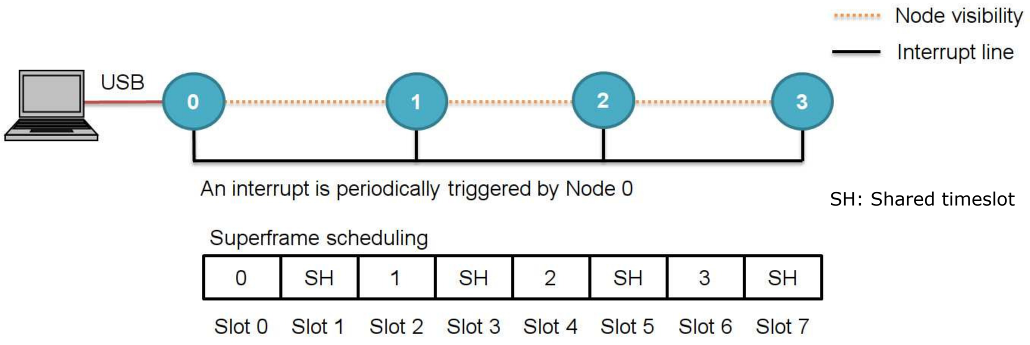

Figure 7. Scenario for clock synchronization accuracy assessment.

Table 1. Testbed parameters.

Parameter	Value
Node count	4
Timeslot count	8
Synchronization period	10 s
Clock sampling period	5 s
Experiment duration	30 min
Result sample count	360

To realize the line topology, the node visibility is artificially limited using a whitelist filter in the BLE interface. The filter drops the transmissions arriving from the nodes that are not in the whitelist.

The Node 0 acts as the timing master for the other nodes. When the network nodes are in the online state, the Node 0 periodically triggers an interruption on the Nodes 1, 2, and 3. When the interruption arrives to Nodes 1, 2, and 3, each node captures its timestamp and sends it to Node 0 through a RESEMBLE message. Messages are forwarded to the Node 0 according to the flooding mechanism defined in the Mesh BLE specifications. The Node 0 collects all timestamps, calculates the clock skew and transmits it to a computer connected through a USB port to the Node 0.

Figure 8 shows the clock skew distribution between Node 1 (N1) and Node 0 (N0).

Node 1 is one hop away from Node 0. Among the measured clock skew values, the most recurring ones are between −2 and 2 ms, while only in one sample the clock skew (N1–N0) reached −3 ms.

Figure 9 shows the clock skew distribution between Node 2 (N2) and Node 0 (N0).

The clock skew at the distance of two hops from N0 increases, as Node 2 is synchronized by Node 1. Most of the time, the clock skew is between −3 and 3 ms, while clock skew (from N2 to N0) reached −3 ms in only two samples.

Figure 10 shows the clock skew distribution between Node 3 (N3) and Node 0 (N0).

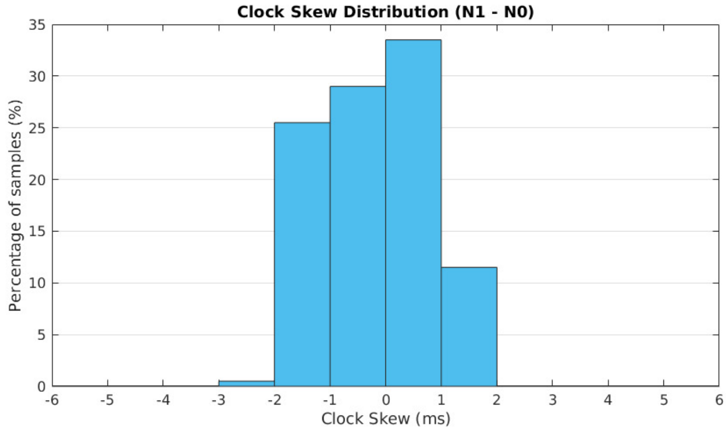

Figure 8. Clock skew distribution (from N1 to N0).

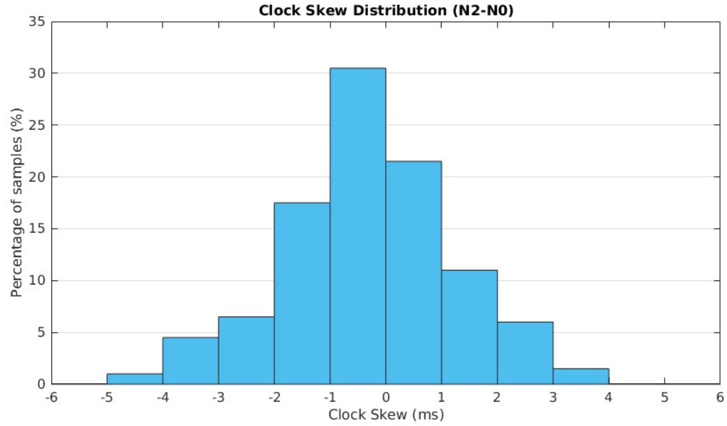

Figure 9. Clock skew distribution (from N2 to N0).

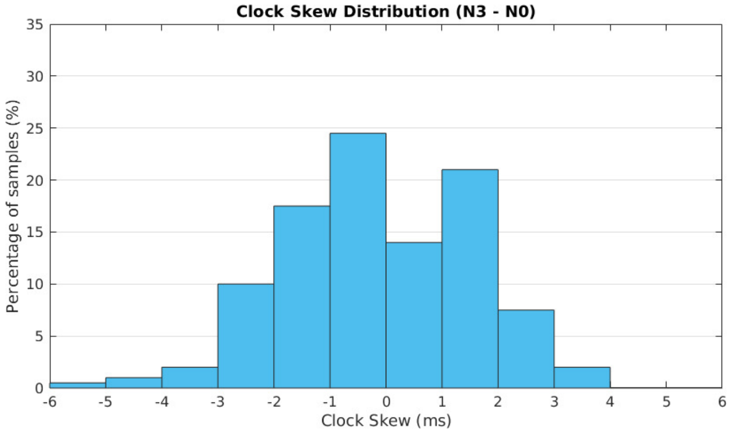

Figure 10. Clock skew distribution (from N3 to N0).

As was expected, the clock skew at a distance of three hops from Node 0 increases and presents the highest variability. In fact, most times, the clock skew is between −3 and 3 ms, while in one sample, the clock skew (from N3 to N0) reached −3 ms.

Figure 11 shows the MaxAbsSkew between all the network nodes.

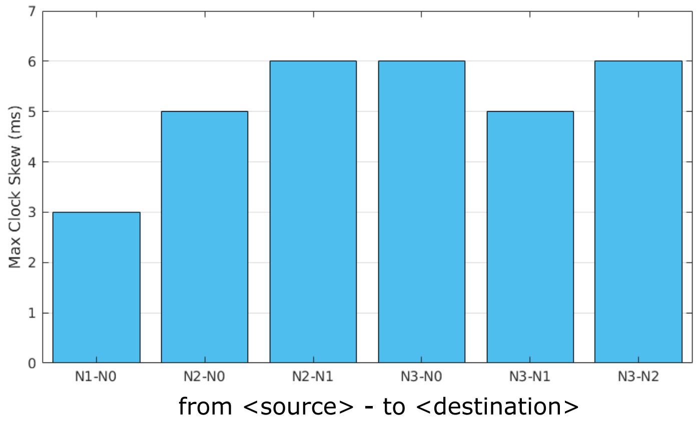

Figure 11. Maximum absolute clock skew.

In these experiments, the MaxAbsSkew never exceeded 6 ms.

6.2. End-to-End Delay Assessment

The assessed scenario, which is shown in Figure 12, consists of four stationary nodes connected in a mesh topology. To realize the mesh topology, the node visibility is artificially limited using a whitelist filter in the BLE interface. The filter drops the transmissions arriving from the nodes that are not in the whitelist.

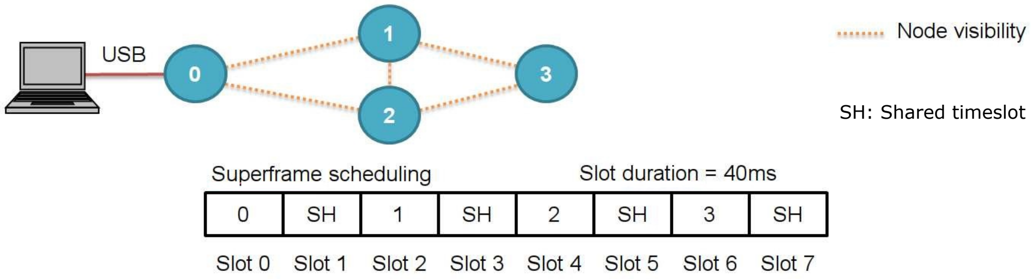

Figure 12. Scenario for end-to-end delay assessment.

Each node generates a periodic message with period 2 s. The Nodes 1, 2, and 3 transmit the messages to Node 0. The Node 0 transmits its messages to Node 3. The flooding algorithm defined in Mesh RT BLE is used. In this scenario all messages are assigned the same priority. As the nodes are synchronized and the MaxAbsSkew in this scenario is 6 ms, the results have a ±6 ms synchronization accuracy.

The parameters are presented in Table 2.

Table 2. Testbed parameters.

Parameter	**Value**
Node count	4
Timeslot count	8
Timeslot duration (T_{ts})	40 ms
Max. hop count	2
Transmission period	2 s
Message size	31 bytes
Experiment duration	30 min
Result sample count	900

The obtained e2eDelays are compared with the analytical maximum e2eDelay of 1280 ms obtained when applying the worst-case response-time analysis for TDMA-based networks, as explained in Section 4.3. A stochastic response-time analysis [41,42] to obtain less pessimistic results will be addressed in future work.

Figure 13 shows the e2eDelay distribution for the flows generated by Nodes 1 and 2. In all the graphs, the vertical red line indicates the analytical e2eDelay bound.

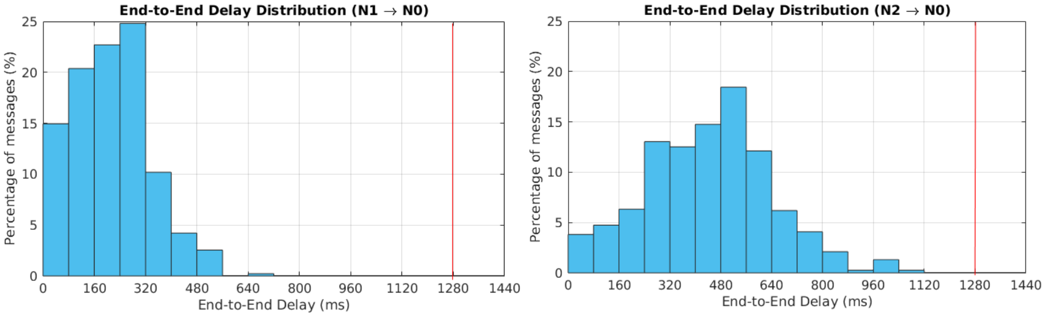

Figure 13. End-to-end delay distributions for the flows generated by the Nodes 1 and 2.

The messages transmitted by Node 1 and Node 2 are always received before the analytically obtained maximum e2eDelay (1280 ms), which is an upper bound. The e2eDelay difference between Node 1 and Node 2 is because the application is not synchronized with the Bearer Layer.

Figure 14 shows the e2eDelay distribution for the flows generated by the Nodes 3 and 0.

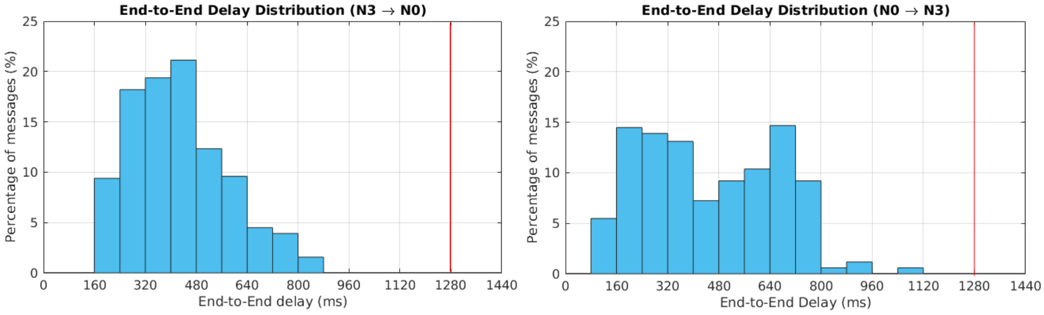

Figure 14. End-to-end delay distributions for the flows generated by the Nodes 0 and 3.

In this case, the e2eDelay of the messages transmitted by the Node 3 are lower than the theoretical maximum e2eDelay (i.e., 1280 ms). In this case, the messages follow a two-hop path; therefore, the e2eDelays are higher than those of the messages transmitted by Nodes 1 and 2. The same holds for the messages transmitted by Node 0 delivered to Node 3.

6.3. Discussion

As shown in this section, RESEMBLE enables several features, i.e., both real-time and non-real-time communications in the same network, node mobility and clock synchronization.

RESEMBLE does not provide retransmission mechanisms at the Bearer Layer and delegates this functionality to the upper layers, i.e., the ones on top of the Network Layer. In fact, such a functionality is strictly dependent on the Application Layer adopted for each specific industrial application. For instance, CANOpen, an Application Layer adopted by many industrial protocols (EtherCAT, CAN bus, etc.), implements its own retransmission mechanisms, while protocols based on the publisher/subscribers model allow the so-called “fire and forget” transmissions, which do not require retransmissions. However, in RESEMBLE, retransmissions are possible.

In RESEMBLE flooding allows for the same message to be transmitted over multiple paths, but, differently from Bluetooth Mesh [22], real-time messages are transmitted in their dedicated timeslots, thus avoiding collisions.

As far as the energy consumption is concerned, compared to the standard BLE Mesh [22] protocol, the adoption of RESEMBLE does not make a significant difference. In fact, the network nodes, when they are not in transmitting mode, stay in receiving mode, with a duty cycle close to 100%, as in [22], in order to listen for both real-time and non-real-time incoming messages. Moreover, as, thanks to the TDMA scheme in RESEMBLE, the transmission instants of each node are known a priori, any node with low energy consumption requirements can go to sleep when it does not have to transmit or receive real-time messages and wake up when needed. This paves the way for the applyication of energy-saving policies, which will be investigated in future work.

TDMA-based transmission schemes suffer from limited scalability. In RESEMBLE, two main factors affect scalability, i.e., the length of the superframe and the overall network workload. The first factor that affects scalability is the superframe length, which depends on the number of network nodes. In RESEMBLE, each node requires at least one timeslot to transmit real-time messages. This entails a longer superframe and, consequently, higher delays. The second factor is the workload generated by the application and, in particular, the number of flows per node and the flow periods. In general, the higher the workload, the longer the queuing time, i.e., the more time a message spends in the transmission queue. A possible solution to this problem is clustering, as it allows for shorter superframes and limits the message flooding within the cluster, as in [43].

7. Conclusions

This paper proposed RESEMBLE, a stack for industrial wireless mesh networks developed on top of BLE that guarantees delay bounds for real-time communications and offers support for node mobility. Moreover, RESEMBLE allows for different traffic classes (i.e., real-time and non-real-time) to be managed on the same BLE network and also provides clock synchronization. The experimental results show that RESEMBLE obtains a 6 ms clock synchronization accuracy over three hops and bounded delays in the order of seconds. These values prove that RESEMBLE is suitable for several industrial applications, such as process control, logistics, etc. RESEMBLE is a software component that can be implemented with no hardware modifications on COTS devices, such as smartphones, tablets and IoT nodes.

Future work will address an improvement in the flooding algorithm run at the network layer to reduce the energy consumption [44], network overhead and delays. Moreover, extensive measurements in a real industrial scenario will be performed.

8. Patents

There is an international patent pending for this work: No. PCT/IT2022/050003. Italian patent deposit no. No. 102021000000773.

Author Contributions: Conceptualization, L.L.B. and G.P.; methodology, L.L., L.L.B. and G.P.; investigation, L.L., L.L.B. and G.P.; writing—original draft preparation, L.L., L.L.B. and G.P.; writing—review and editing, L.L., L.L.B. and G.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the Project "Soluzioni innovative per la connettività wired e wireless nell'industria 4.0 a supporto dell'efficienza e resilienza della fabbrica", within the action IV.4—"Dottorati e contratti di ricerca su tematiche dell'innovazione del nuovo Asse IV del PON Ricerca e Innovazione 2014-2020 Istruzione e ricerca per il recupero—REACT-EU".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data underlying this article will be shared on reasonable request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

ADV	Advertisement
BLE	Bluetooth Low Energy
COTS	Commercial-off-the-shelf
e2eDelay	end-to-end delay
FIFO	First-In First-Out
GATT	Generic Attribute Profile
HCI	Host Controller Interface
IoT	Internet of Things
IWSN	Industrial Wireless Sensor Network
MaxAbsSkew	Maximum Absolute Clock Skew
PDU	Protocol Data Unit
QoS	Quality-of-Service
RT	Real-time
S	Clock skew
SH	Shared
TDMA	Time Division Multiple Access
TTL	Time-to-live
WCRT	Worst-case response time

References

1. Yang Lam, T.C.; Ling Yew, S.S.; Keoh, S.L. Bluetooth Mesh Networking: An Enabler of Smart Factory Connectivity and Management. In Proceedings of the 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Matsue, Japan, 18–20 September 2019; pp. 1–6. [CrossRef]
2. Iannizzotto, G.; Lo Bello, L.; Nucita, A.; Grasso, G.M. A Vision and Speech Enabled, Customizable, Virtual Assistant for Smart Environments. In Proceedings of the 2018 11th International Conference on Human System Interaction (HSI), Gdansk, Poland, 4–6 July 2018; pp. 50–56. [CrossRef]
3. Sabban, A. Active Compact Wearable Body Area Networks for Wireless Communication, Medical and IoT Applications. *Appl. Syst. Innov.* **2018**, *1*, 46. [CrossRef]
4. Iannizzotto, G.; Lo Bello, L.; Patti, G. Personal Protection Equipment detection system for embedded devices based on DNN and Fuzzy Logic. *Expert Syst. Appl.* **2021**, *184*, 115447. [CrossRef]
5. Collotta, M.; Pau, G.; Talty, T.; Tonguz, O.K. Bluetooth 5: A Concrete Step Forward toward the IoT. *IEEE Commun. Mag.* **2018**, *56*, 125–131. [CrossRef]
6. Pau, G.; Arena, F.; Collotta, M.; Kong, X. A practical approach based on Bluetooth Low Energy and Neural Networks for indoor localization and targeted devices' identification by smartphones. *Entertain. Comput.* **2022**, *43*, 100512. [CrossRef]
7. Pandya, S.; Ghayvat, H.; Kotecha, K.; Awais, M.; Akbarzadeh, S.; Gope, P.; Mukhopadhyay, S.C.; Chen, W. Smart Home Anti-Theft System: A Novel Approach for Near Real-Time Monitoring and Smart Home Security for Wellness Protocol. *Appl. Syst. Innov.* **2018**, *1*, 42. [CrossRef]
8. Kandris, D.; Nakas, C.; Vomvas, D.; Koulouras, G. Applications of Wireless Sensor Networks: An Up-to-Date Survey. *Appl. Syst. Innov.* **2020**, *3*, 14. [CrossRef]
9. Raza, S.; Misra, P.; He, Z.; Voigt, T. Bluetooth smart: An enabling technology for the Internet of Things. In Proceedings of the 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Lyon, France, 19–21 October 2015; pp. 155–162.
10. Pau, G.; Collotta, M.; Maniscalco, V. Bluetooth 5 energy management through a fuzzy-pso solution for mobile devices of internet of things. *Energies* **2017**, *10*, 992. [CrossRef]
11. Iannizzotto, G.; Milici, M.; Nucita, A.; Lo Bello, L. A Perspective on Passive Human Sensing with Bluetooth. *Sensors* **2022**, *22*, 3523. [CrossRef]
12. Bonaiuto, V.; Boatto, P.; Lanotte, N.; Romagnoli, C.; Annino, G. A Multiprotocol Wireless Sensor Network for High Performance Sport Applications. *Appl. Syst. Innov.* **2018**, *1*, 52. [CrossRef]
13. Zhang, T.; Lu, J.; Hu, F.; Hao, Q. Bluetooth low energy for wearable sensor-based healthcare systems. In Proceedings of the Healthcare Innovation Conference (HIC), Seattle, WA, USA, 8–10 October 2014; pp. 251–254.
14. Díez, V.; Arriola, A.; Val, I.; Velez, M. Reliability Evaluation of Bluetooth Low Energy for Industry 4.0. In Proceedings of the 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Zaragoza, Spain, 10–13 September 2019; pp. 1148–1154. [CrossRef]
15. Ghori, M.R.; Wan, T.C.; Sodhy, G.C. Bluetooth Low Energy Mesh Networks: Survey of Communication and Security Protocols. *Sensors* **2020**, *20*, 3590. [CrossRef]

16. Grover, M.; Pardeshi, S.K.; Singh, N.; Kumar, S. Bluetooth low energy for industrial automation. In Proceedings of the 2015 2nd International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, 26–27 February 2015; pp. 512–515. [CrossRef]
17. Szyc, K.; Nikodem, M.; Zdunek, M. Bluetooth low energy indoor localization for large industrial areas and limited infrastructure. *Ad Hoc Netw.* **2023**, *139*, 103024. [CrossRef]
18. Rondon, R.; Gidlund, M.; Landernas, K. Evaluating Bluetooth Low Energy Suitability for Time-Critical Industrial IoT Applications. *Int. J. Wirel. Inf. Netw.* **2017**, *24*, 278–290. [CrossRef]
19. Leonardi, L.; Lo Bello, L.; Patti, G. LoRa support for long-range real-time inter-cluster communications over Bluetooth Low Energy industrial networks. *Comput. Commun.* **2022**, *192*, 57–65. [CrossRef]
20. Wang, S.D.; Chiang, K.J. BLE Tree Networks for Sensor Devices in Internet of Things. In Proceedings of the 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Orlando, FL, USA, 6–10 November 2017; pp. 1304–1309. [CrossRef]
21. Leonardi, L.; Patti, G.; Lo Bello, L. Multi-Hop Real-Time Communications Over Bluetooth Low Energy Industrial Wireless Mesh Networks. *IEEE Access* **2018**, *6*, 26505–26519. [CrossRef]
22. Bluetooth Specification: Mesh Profile v1.0.1. 2019. Available online: https://www.bluetooth.com/specifications/specs/mesh-model-1-0-1/ (accessed on 5 January 2023).
23. Hortelano, D.; Olivares, T.; Ruiz, M. Providing interoperability in Bluetooth mesh with an improved provisioning protocol. *Wirel. Netw.* **2021**, *27*, 1011–1033. [CrossRef]
24. Aijaz, A. Infrastructure-less Wireless Connectivity for Mobile Robotic Systems in Logistics: Why Bluetooth Mesh Networking is Important? In Proceedings of the 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vasteras, Sweden, 7–10 September 2021; pp. 1–8. [CrossRef]
25. Basu, S.S.; Baert, M.; Hoebeke, J. QoS Enabled Heterogeneous BLE Mesh Networks. *J. Sens. Actuator Netw.* **2021**, *10*, 24. [CrossRef]
26. Toscano, E.; Lo Bello, L. Cross-channel interference in IEEE 802.15. 4 networks. In Proceedings of the 2008 IEEE International Workshop on Factory Communication Systems (WFCS 2008), Dresden, Germany, 21–23 May 2008; pp. 139–148. [CrossRef]
27. Rondón, R.; Mahmood, A.; Grimaldi, S.; Gidlund, M. Understanding the Performance of Bluetooth Mesh: Reliability, Delay, and Scalability Analysis. *IEEE Internet Things J.* **2020**, *7*, 2089–2101. [CrossRef]
28. Kweon, S.K.; Shin, K. Achieving real-time communication over Ethernet with adaptive traffic smoothing. In Proceedings of the Sixth IEEE Real-Time Technology and Applications Symposium, RTAS 2000, Washington, DC, USA, 31 May–2 June 2000; pp. 90–100. [CrossRef]
29. Carpenzano, A.; Caponetto, R.; Lo Bello, L.; Mirabella, O. Fuzzy traffic smoothing: An approach for real-time communication over Ethernet networks. In Proceedings of the 4th IEEE International Workshop on Factory Communication Systems (WFCS 2002), Vasteras, Sweden, 28–30 August 2002; pp. 241–248. [CrossRef]
30. *Bluetooth Specification v4.2*; Bluetooth Core Specification. Bluetooth SIG: Kirkland, WA, USA, 2014.
31. *Bluetooth Specification v5.3*; Bluetooth Core Specification. Bluetooth SIG: Kirkland, WA, USA, 2021.
32. *IEEE Std. 802.1AS-2011*; IEEE Standard for Local and Metropolitan Area Networks—Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks. IEEE: Piscataway, NJ, USA, 2011.
33. Joseph, M.; Pandya, P.K. Finding Response Times in a Real-Time System. *Comput. J.* **1986**, *29*, 390–395. [CrossRef]
34. Lehoczky, J. Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Deadlines. In Proceedings of the IEEE Real-Time Systems Symposium (RTSS), Lake Buena Vista, FL, USA, 5–7 December 1990; pp. 201–209.
35. Lo Bello, L.; Ashjaei, M.; Patti, G.; Behnam, M. Schedulability analysis of Time-Sensitive Networks with scheduled traffic and preemption support. *J. Parallel Distrib. Comput.* **2020**, *144*, 153–171. [CrossRef]
36. Davis, R.I.; Burns, A.; Bril, R.J.; Lukkien, J.J. Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised. *Real-Time Syst.* **2007**, *35*, 239–272. [CrossRef]
37. Patti, G.; Lo Bello, L. A priority-aware multichannel adaptive framework for the IEEE 802.15. 4e-LLDN. *IEEE Trans. Ind. Electron.* **2016**, *63*, 6360–6370. [CrossRef]
38. Almeida, L.; Pedreiras, P. Scheduling within Temporal Partitions: Response-Time Analysis and Server Design. In Proceedings of the 4th ACM International Conference on Embedded Software, Pisa, Italy, 27–29 September 2004; pp. 95–103. [CrossRef]
39. NUCLEO-L476RG: STM32 Nucleo-64 Development Board with STM32L476RG MCU, Supports Arduino and ST Morpho Connectivity. Available online: https://www.st.com/en/evaluation-tools/nucleo-l476rg.html (accessed on 6 January 2023).
40. X-NUCLEO-IDB05A2: Bluetooth Low Energy Expansion Board Based on the BlueNRG-M0 Module for STM32 Nucleo. Available online: https://www.st.com/en/ecosystems/x-nucleo-idb05a2.html (accessed on 6 January 2023).
41. Kaczynski, G.A.; Lo Bello, L.; Nolte, T. Deriving exact stochastic response times of periodic tasks in hybrid priority-driven soft real-time systems. In Proceedings of the 2007 IEEE Conference on Emerging Technologies and Factory Automation (EFTA 2007), Patras, Greece, 25–28 September 2007; pp. 101–110. [CrossRef]
42. Diaz, J.; Lopez, J.; Garcia, M.; Campos, A.; Kim, K.; Lo Bello, L. Pessimism in the stochastic analysis of real-time systems: Concept and applications. In Proceedings of the 25th IEEE International Real-Time Systems Symposium (RTSS 2004), Lisbon, Portugal, 5–8 December 2004; pp. 197–207. [CrossRef]

43. Patti, G.; Leonardi, L.; Lo Bello, L. A Novel MAC Protocol for Low Datarate Cooperative Mobile Robot Teams. *Electronics* **2020**, *9*, 235. [CrossRef]
44. Pradeep, S.; Sharma, Y.K.; Verma, C.; Dalal, S.; Prasad, C. Energy Efficient Routing Protocol in Novel Schemes for Performance Evaluation. *Appl. Syst. Innov.* **2022**, *5*, 101. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.