This paper has been accepted for publication in IEEE Communications Magazine. Please refer to the final paper version once published.

Bluetooth Low Energy Mesh Networks: a Standards Perspective

Seyed Mahdi Darroudi[1], Carles Gomez[1], Jon Crowcroft[2]
[1]Universitat Politècnica de Catalunya, [2]University of Cambridge
sm.darroudi@entel.upc.edu, carlesgo@entel.upc.edu, jon.crowcroft@cl.cam.ac.uk

***Abstract*—Bluetooth Low Energy (BLE) mesh networking is an emerging technology domain that promises an important role in the Internet of Things (IoT). Significant market opportunities for BLE mesh networking have motivated the recent development of two different BLE mesh networking standards: Bluetooth Mesh and 6BLEMesh, produced by the Bluetooth SIG and the IETF, respectively. These two standards follow different technical approaches. In this paper, we present the main features of Bluetooth Mesh and 6BLEMesh, and investigate their performance characteristics and trade-offs.**

I. Introduction

Bluetooth Low Energy (BLE) is a low-power, short-range wireless technology that was first specified by the Bluetooth Special Interest Group (SIG) in 2010, as part of Bluetooth 4.0 [1]. Since then, BLE has become a fundamental technology for the Internet of Things (IoT). In fact, BLE is suitable for resource-constrained devices (e.g. battery-operated small sensors and actuators), which are typical in the IoT. However, in contrast with other IoT technologies, BLE also has widespread presence in consumer electronics devices such as smartphones. This distinctive feature of BLE facilitates interaction between a user and surrounding BLE devices, since the smartphone may naturally become a user interface and/or a gateway in IoT scenarios.

For simplicity, BLE was originally designed to only enable star topology networks. However, this feature would limit BLE applicability in crucial IoT application domains wherein a star topology network cannot ensure coverage for all intended devices. For example, many relevant wireless technologies in the smart home space, such as ZigBee, Z-Wave or Thread, support the mesh topology [2, 3]. Furthermore, the mesh topology provides path diversity, and thus intrinsic robustness, which allows to better face radio propagation impairments, interference and device failures.

In order to offer greater flexibility, subsequent Bluetooth specification updates removed the network topology constraints of Bluetooth 4.0 for BLE. However, such specification updates did not provide mechanisms to enable end-to-end data delivery in a BLE mesh network. In order to address this problem, a plethora of proprietary and academic BLE mesh network solutions have been recently created [4]. Nevertheless, such solutions do not offer interoperability among products of different manufacturers and developers.

In order to overcome the network topology limitations of BLE, while offering a standardized approach, the Bluetooth SIG published in 2017 the Bluetooth Mesh suite of specifications [5]. On the other hand, the IETF is currently standardizing functionality for enabling IPv6-based BLE Mesh Networks (6BLEMesh), by following a different technical approach [6]. Given the potential of BLE mesh networking, it is fundamental to understand the features and limitations of the two main types of standards-based BLE mesh network solutions. This paper overviews, compares and discusses both Bluetooth Mesh and 6BLEMesh.

The remainder of this article is organized as follows. Section 2 introduces BLE fundamentals. Sections 3 and 4 overview Bluetooth Mesh and 6BLEMesh, respectively. Section 5 comparatively discusses and evaluates these two solutions, while section 6 relates their design goals and performance. Section 7 concludes the article.

II. Bluetooth Low Energy

BLE defines a protocol stack (Fig. 1.a). At the Physical layer, BLE operates over 40 frequency channels in the 2.4 GHz band. These channels are organized into 3 advertising channels and 37 data channels. The Physical layer bit rate is 1 Mbit/s in Bluetooth 4.x. Further bit rates (from 125 kbit/s to 2 Mbit/s) were introduced in Bluetooth 5.0 [7].

In BLE, there exist two approaches for communication between neighboring nodes. The first one is based on using advertising channels, which are defined for broadcasting purposes only. The second one requires two nodes to establish a Link layer connection. Once connected, the nodes become a master, which manages the connection, and a slave. A Link layer connection allows bidirectional data exchange opportunities between two connected devices every *connInterval* over data channels. In this case, the full BLE protocol stack is used. The Logical Link Control and Adaptation Protocol (L2CAP) layer supports upper layer data unit fragmentation and reassembly, and flow control. The Attribute protocol (ATT) layer and the Generic Attribute profile (GATT) layer define functionality for communication between a server (e.g. a temperature sensor) and a client (e.g. a device that collects temperature readings).

III. Bluetooth Mesh

This section overviews Bluetooth Mesh. First, the Bluetooth Mesh protocol stack is presented. Next, each Bluetooth Mesh protocol stack layer is described.

A. Protocol stack

Bluetooth Mesh offers networking services and support for applications by means of a protocol stack (Fig. 1.b-1.c). This protocol stack comprises the Bearer layer, the Network layer, the Lower Transport layer, the Upper Transport layer, the Access layer, the Foundation Model layer, and the Model layer.

B. Bearer layer

The Bearer layer uses BLE (i.e. as defined in Bluetooth 4.x or 5.x specifications) as the means to carry Bluetooth Mesh messages. For simplicity, Bluetooth Mesh relies typically on advertising-based bearers (Fig. 1.b). Nevertheless, for nodes that do not operate as advertisers, the GATT bearer is also supported in Bluetooth Mesh by means of a special proxy node role, which relays messages from advertising-based bearers to GATT-based bearers (Fig. 1.c), and vice versa. Hereinafter, we assume that when a Bluetooth Mesh node transmits a message, the latter is sent via the 3 advertising channels.

C. Network layer

The Network layer offers end-to-end transmission of upper layer data units over a Bluetooth Mesh network, by means of a controlled flooding mechanism. Message forwarding is carried out by Relay nodes. In order to limit the message overhead of the flooding mechanism, two techniques are applied. First, a Time To Live (TTL) field is included in each message header and it is decremented each time a message is relayed, thus limiting the maximum number of hops for a message. A sender can tailor the TTL of a message to the hopwise distance between itself and its destination. Secondly, each time a Relay node receives a new data message, the node stores it in a cache. If a copy of the same message is received subsequently, that copy is discarded by the node.

The Network layer also provides security services, as it encrypts and authenticates all messages in a Bluetooth Mesh network. Furthermore, relevant Network layer header fields, including the source address, are obfuscated in order to avoid privacy threats, such as node tracking.

D. Lower Transport layer

The Lower Transport layer provides efficient segmentation and reassembly for upper layer data units that cannot be carried by a single Bearer layer data unit. When segmentation is used, the receiving peer endpoint transmits a *block* acknowledgment, which reports whether the segments of a message have been received or not. The sender selectively retransmits any missing segments.

E. Upper Transport layer

The Upper Transport layer offers three main services. The first one is securing application data by means of encryption and authentication. Such security functionality is separate from the Network layer one, as a measure to further protect the network from eavesdroppers. A different application-level key will be used for each application, and will only be available to the communicating endpoints.

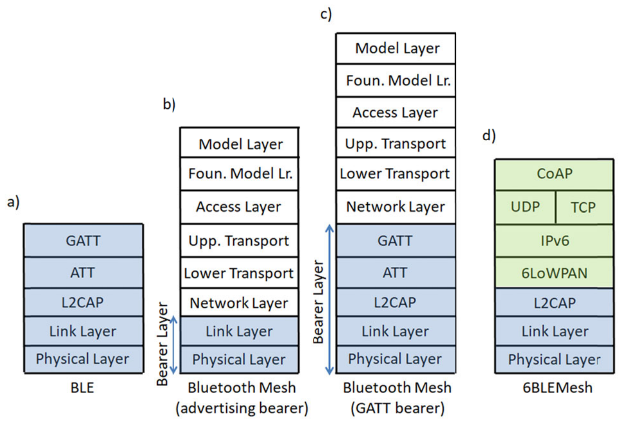

Fig. 1. Protocol stacks: a) BLE, b) Bluetooth Mesh (advertising bearer), c) Bluetooth Mesh (GATT bearer), d) 6BLEMesh.

The second Upper Transport layer service is support for energy-constrained devices called Low Power nodes (LPNs). This service is based on a concept called friendship, whereby a so-called Friend node allows a neighboring LPN to operate at reduced duty cycles to save energy. The Friend node stores messages intended for the LPN while the latter is in sleep state to conserve energy. Every *PollTimeout* interval, the LPN wakes up, polls its Friend node, listens during *ReceiveWindow* milliseconds for potentially incoming messages, and returns to sleep state.

The third service provided by the Upper Transport layer is periodic transmission of network-wide Heartbeat messages, which allows receiving nodes to learn the hopwise distance traversed by such messages and optimize the scope of data message flooding.

F. Access layer

The Access layer defines a format for application data that indicates how such data need to be handled, and supports end-to-end reliability. This layer defines whether a message is acknowledged or unacknowledged. In the first case, an automatic repeat request mechanism is supported.

G. Foundation Model layer and Model layer

In Bluetooth Mesh, applications are based on a client-server architecture, where servers support resources called states (e.g. on/off variables and their values) and clients operate on such states by using messages (e.g. to toggle a physical switch associated with an on/off variable). A set of states, messages and associated behaviors related with a specific purpose is called a model.

The two highest layers of the BLE mesh protocol stack are the Foundation Model layer and the Model layer, both of which define models. The Foundation Model layer provides support for managing a BLE mesh network. The Model layer offers a framework for smart home applications (e.g. scenes, lighting, etc.), as well as generic device and sensor functionality.

IV. 6BLEMESH: IPV6-BASED BLE MESH NETWORKS

The previous section overviewed the Bluetooth SIG's

Bluetooth Mesh standard. Another major standards-based solution for BLE mesh networking, 6BLEMesh, is currently being developed by the IETF IPv6 over Networks of Resource-Constrained Nodes (6Lo) working group [6]. In order to enable IPv6-based BLE mesh networks, 6BLEMesh extends RFC 7668. The latter specifies IPv6 over star-topology BLE networks by leveraging IPv6 over Low Power Wireless Personal Area Network (6LoWPAN) [8].

This section presents 6BLEMesh. First, the background concepts of 6LoWPAN and IPv6 over star-topology BLE networks are introduced. Subsequently, the main 6BLEMesh features are described.

A. 6LoWPAN

6LoWPAN is an adaptation layer that was originally designed to efficiently enable IPv6 over IEEE 802.15.4 networks [9]. Like BLE networks, IEEE 802.15.4 networks typically comprise resource-constrained devices, and offer relatively low bit rates. IEEE 802.15.4 networks are fundamentally different from the resource-rich networking environments assumed for IPv6 when it was created. In fact, an adaptation layer between the IPv6 layer and the IEEE 802.15.4 layer is required to comply with IPv6 requirements, and for efficiency.

6LoWPAN comprises three fundamental mechanisms: i) compression of IPv6 and UDP headers, ii) optimized IPv6 Neighbor Discovery (ND), and iii) fragmentation functionality. The first two mechanisms allow energy- and bandwidth-frugal operation. 6LoWPAN header compression exploits intra-packet redundancy and an expectation of typically used header field values. 6LoWPAN-optimized IPv6 ND reduces use of multicast and allows energy conservation intervals by enforcing interactions initiated by energy-constrained devices. 6LoWPAN fragmentation supports the transmission of 1280-byte packets (as required for IPv6) over the smaller maximum frame payload size of IEEE 802.15.4, of ~100 bytes.

IEEE 802.15.4 supports the mesh network topology. Accordingly, 6LoWPAN defines three node roles for such topology: i) 6LoWPAN Border Router (6LBR) for routers at the edge of the 6LoWPAN network, ii) 6LoWPAN Router (6LR) for routers internal to the 6LoWPAN network, and iii) 6LoWPAN Node (6LN) for non-routing devices. A 6LBR often supports several network interfaces, typically including one that offers Internet connectivity. A 6LBR also manages the configuration of a 6LoWPAN network. 6LRs typically support only one network interface and enable the connectivity between 6LNs and the 6LBR. Since 6LBRs and 6LRs need to generally be ready to receive (and forward) data packets, they often require mains power. 6LNs are typically simple devices that run on limited energy sources.

B. IPv6 over star topology BLE networks

RFC 7668 enables IPv6 over star topology BLE networks by modifying 6LoWPAN for BLE [8]. RFC 7668 uses Link layer connections over data channels for communication between neighboring BLE devices. When an IPv6 packet needs to be sent, 6LoWPAN-based header compression is applied; then, the packet is handled by the BLE L2CAP layer.

RFC 7668 simplifies and optimizes 6LoWPAN for BLE star topology networks in different ways. First, in star topology networks, the 6LR role does not exist, leading to a network comprising only a central 6LBR directly connected to a set of neighboring 6LNs. In such scenario, a routing protocol is not needed. Second, in a star topology, a 6LN can omit the source address from the packets it transmits, since the 6LBR can unambiguously infer that the packet sender is that 6LN; likewise, the 6LBR can omit the destination address from the packets it sends to a 6LN. Finally, 6LoWPAN fragmentation is not needed over BLE, since L2CAP provides native fragmentation functionality.

C. 6BLEMesh features

In order to enable IPv6-based BLE mesh networks, 6BLEMesh inherits features from RFC 7668, while defining new functionality.

As in RFC 7668, 6BLEMesh uses Link layer connections over data channels between neighboring devices. This approach is different from the Bluetooth Mesh one, where advertising channels are used as the main packet bearers.

To enable mesh topology operation, 6BLEMesh restores the 6LR role and requires an IPv6-based routing protocol. The IPv6 Routing Protocol for Low-power and lossy networks (RPL) is the main candidate routing protocol for 6BLEMesh, since it is the routing protocol standardized by the IETF for IoT environments [10]. Nevertheless, other routing protocols have been selected for some IP-based IoT protocol stacks. For example, Thread uses a routing protocol based on the Routing Information Protocol [3].

In 6BLEMesh, the RFC 7668 header compression optimization which allows omitting a full address can only be applied in links between 6LNs and their routers. For example, in a link between routers A and B, if router A receives an IPv6 packet from router B without a source address, router A cannot determine whether the source of the packet was router B or a previous node. Still, applying such header compression when possible is useful since 6LNs are likely to be energy-constrained devices.

D. Upper layers

At the application and transport layers of the 6BLEMesh protocol stack, any IP-based set of protocols can be used. However, the Constrained Application Protocol (CoAP), atop UDP or TCP, appears to be a suitable choice [11]. CoAP is a lightweight application-layer protocol specifically designed for the IoT. CoAP is based on the REST architecture, like HTTP, and it allows the communication between CoAP and HTTP endpoints through a translation proxy. While CoAP was originally designed over UDP, use of CoAP over TCP has been recently specified, as it is required in some environments to traverse middleboxes such as firewalls [12]. Fig. 1.d illustrates a protocol stack for 6BLEMesh, which includes IoT-specific upper layer protocols such as CoAP.

V. Bluetooth Mesh vs 6BLEMesh: a Comparative Discussion

This section compares Bluetooth Mesh and 6BLEMesh, in terms of the following performance metrics and features: protocol encapsulation overhead, latency, energy consumption, message transmission count, link corruption robustness, variable topology robustness, and Internet connectivity.

A. Protocol encapsulation overhead

We define the protocol encapsulation overhead of either Bluetooth Mesh or 6BLEMesh as the total header and footer overhead added by all protocol stack layers to a user data payload before transmission. We assume that the user data payload fits into a single Physical layer data unit.

The minimum protocol encapsulation overhead of Bluetooth Mesh (29 bytes) is slightly greater than that of 6BLEMesh (25 bytes). Only the lowest protocol stack layer (which contributes 8 bytes to the protocol encapsulation overhead) is shared by both BLE mesh network approaches. In 6BLEMesh, header compression is crucial to produce a 7-byte compressed IPv6/UDP header (in contrast with a 48-byte uncompressed one). In Bluetooth Mesh, the 9-byte Network layer header is the greatest contributor to protocol encapsulation overhead. This header includes 4 bytes used for security purposes, such as identifying the keys used to protect a message, and prevent replay attacks.

B. Latency

We now study the latency of packet transmission over a multihop path in Bluetooth Mesh and in 6BLEMesh. This performance parameter is particularly critical for applications where a human expects a quick reaction to an action (e.g. turning on a lightbulb after pressing a button on a remote control). Such applications are typical in smart home, a major target domain for BLE mesh networking. In these scenarios, interaction is often considered real-time when latency is below 500 ms [13].

Fig. 2 depicts the whole range of average per-hop latency values for Bluetooth Mesh and 6BLEMesh, assuming negligible processing time and ideal channel conditions. As it can be seen, both solutions offer flexibility for determining latency performance. The relevant involved parameters and mechanisms are described next.

In Bluetooth Mesh, each hop contributes at least the time required to transmit a packet via the advertising channels, denoted *T*. This time depends on how BLE is implemented or configured, and may fall between 1 ms and 20 ms.

In Bluetooth Mesh, when the next hop is a LPN, the latter will only be able to receive data packets after polling its Friend node, which happens every *PollTimeout*. Therefore, in this case, the last hop will contribute a random, uniformly distributed, additional delay of up to *PollTimeout*. The minimum and maximum possible values for this parameter are 1 second and 4 days, respectively. Remarkably, the minimum

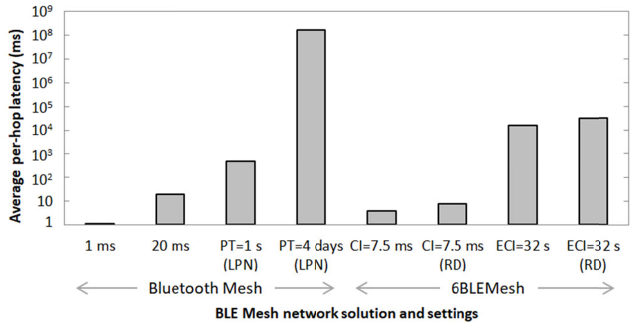

Fig. 2. Theoretical average per-hop latency of Bluetooth Mesh and 6BLEMesh, for their whole ranges of main settings. PT, CI, ECI and RD stand for *PollTimeout*, *connInterval*, equivalent *connInterval* and Route Discovery, respectively.

PollTimeout value does not ensure real-time interaction when the destination node is a LPN.

In 6BLEMesh, the time required to deliver a packet from one node to its next hop (regardless of the role of the latter) is a uniformly distributed random variable up to *connInterval*. The minimum value allowed for this parameter is 7.5 ms, therefore real-time communication is possible in 6BLEMesh even when the destination node is a 6LN. The maximum *connInterval* setting is 4 s, although a BLE slave is allowed to skip a number of consecutive communication opportunities, leading to a maximum equivalent *connInterval* value of 32 s.

In 6BLEMesh, route discovery may be an additional contributor to end-to-end packet latency. If a reactive routing protocol is used, route discovery delay can be estimated as twice the one-way end-to-end delay. Proactive routing does not add a route discovery delay.

C. Energy Consumption

We now study the energy consumption performance of Bluetooth Mesh and 6BLEMesh. We determine the theoretical lifetime of a battery-powered Bluetooth Mesh LPN and a 6BLEMesh 6LN, based on current consumption measurements carried out on an nRF51 DK hardware platform, and a battery capacity of 235 mAh. We assume that the battery-powered device transmits a data message periodically. The data message payload size is 8 bytes (therefore, it fits into a single Physical Layer data unit) and the transmit power is 0 dBm.

As shown in Fig. 3, the achievable device lifetime depends strongly on the parameter settings of the link maintenance mechanisms of Bluetooth Mesh (i.e. *PollTimeout* and *ReceiveWindow*) and 6BLEMesh (i.e. *connInterval*). The maximum and minimum values for such parameters are considered in Fig. 3, in order to determine the full range of device lifetime results. The data message interval (DI) only influences device lifetime for relatively infrequent link maintenance interactions. In such conditions, device lifetime increases asymptotically with DI. The maximum achievable device lifetime is 644 days.

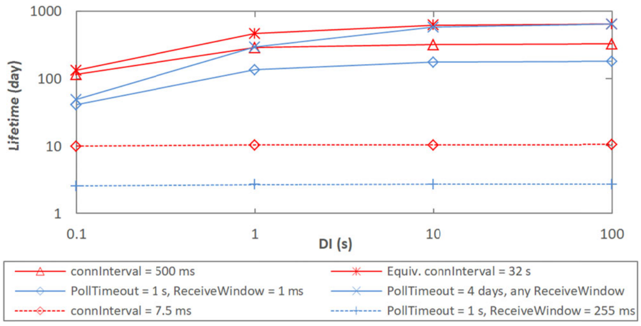

Fig. 3. Lifetime of a battery-operated device for Bluetooth Mesh and 6BLEMesh, as a function of DI.

Reducing communication latency (by means of either decreasing *PollTimeout* or *connInterval*) decreases device lifetime. It is possible to achieve a 325-day device lifetime with 6BLEMesh, while keeping single-hop latency below the real-time threshold of 500 ms (i.e. for *connInterval* ≤ 500 ms). In contrast, in Bluetooth Mesh, the lowest assured latency is 1 s (for *PollTimeout* = 1 s), whereas the greatest device lifetime in this case is only 181 days. Remarkably, the operations carried out by a device in 6BLEMesh every *connInterval* (which include one receive and one transmit interval) consume less than 25% of the energy consumed in a poll action in Bluetooth Mesh (which includes three transmit intervals and one longer receive interval). Therefore, an energy-constrained device consumes less energy in 6BLEMesh than in Bluetooth Mesh for a given latency target.

D. Message transmission count

We now evaluate by simulation the message count (i.e. total number of message transmissions) of Bluetooth Mesh and 6BLEMesh, for a range of network sizes, node densities and protocol parameters (Fig. 4). We assume connected, steady-state static networks without message losses, where each node sends a data message every DI to a randomly chosen destination node. We also assume that, in 6BLEMesh, shortest path routes are found and maintained by the Point-to-Point extension of RPL [14].

Fig. 4 illustrates the average value of the total message transmission count in the whole network per time unit. Each individual result has been obtained over 100 different topologies where nodes are randomly distributed over a square area. The details of each scenario are shown in Table I. The coverage area of a node is 400 m^2.

The message transmission count in a BLE mesh network comprises two main components: i) data traffic, and ii) network maintenance traffic. In Bluetooth Mesh, the latter corresponds to Heartbeat and polling messages, whereas in 6BLEMesh it comprises link maintenance and routing messages. We next analyze how data and network maintenance traffic contribute to the total message count for both BLE mesh networking approaches.

Bluetooth Mesh presents a greater number of data message transmissions than 6BLEMesh, since Bluetooth mesh uses (controlled) flooding, whereas the latter uses single-path routing. Furthermore, in Bluetooth Mesh, each message transmission is performed thrice (i.e. once per advertising channel).

TABLE I
SCENARIOS WITH DIFFERENT NETWORK SIZES AND DENSITIES

Scenario Size	Area (m^2)	Node Degree	Scenario Name	Number of Nodes
Small	25·25	7	S7	11
		11	S11	18
		15	S15	24
Medium	50·50	7	M7	44
		11	M11	69
		15	M15	94
Large	100·100	7	L7	175
		11	L11	275
		15	L15	375
Very large	150·150	7	VL7	394
		11	VL11	619
		15	VL15	844

Regarding network maintenance traffic, in Bluetooth Mesh each node sends Heartbeat messages (which are forwarded network-wide) periodically. Therefore, the total rate of Heartbeat message transmissions is a function of N^2, where N denotes the number of network nodes. In contrast, in 6BLEMesh, each node sends one message (which is not relayed) per connected neighbor every *connInterval*. Routing traffic is negligible in comparison, as in steady state and with default P2P-RPL settings, the time between consecutive routing protocol messages sent by a node to a neighbor is in the order of hours. Therefore, the 6BLEMesh network maintenance message rate depends on the number of network links, N_{links}, which is smaller than N^2.

The described features of data and network maintenance traffic yield a message transmission count of Bluetooth Mesh that scales worse with network size and node density (i.e. a greater slope in Fig. 4) than 6BLEMesh. However, if *connInterval* is set to low values (e.g. to achieve low end-to-end latency), the message count in 6BLEMesh may be greater than that of Bluetooth Mesh, for the same DI. This occurs mainly in networks with smaller size or node density.

E. Link corruption robustness

Links in a BLE mesh network are prone to suffering bit errors, due to phenomena such as radio signal fading or interference, among others. Bluetooth Mesh and 6BLEMesh support different mechanisms intended to tackle this problem.

Bluetooth Mesh's flooding offers path diversity to each packet transmission. In contrast, 6BLEMesh typically uses single-path routing for unicast communication. Both Bluetooth Mesh and 6BLEMesh support frequency diversity in different ways. In Bluetooth Mesh, each message is typically sent via the 3 advertising channels in parallel. In 6BLEMesh, Link layer retries are performed (if needed) over a frequency channel that is updated every *connInterval*, as long as the Link layer connection remains open. The maximum number of consecutive Link layer retries in a Link layer connection, denoted R, is configurable.

In order to illustrate the performance of Bluetooth Mesh and 6BLEMesh in the presence of bit errors, Fig. 5 depicts the

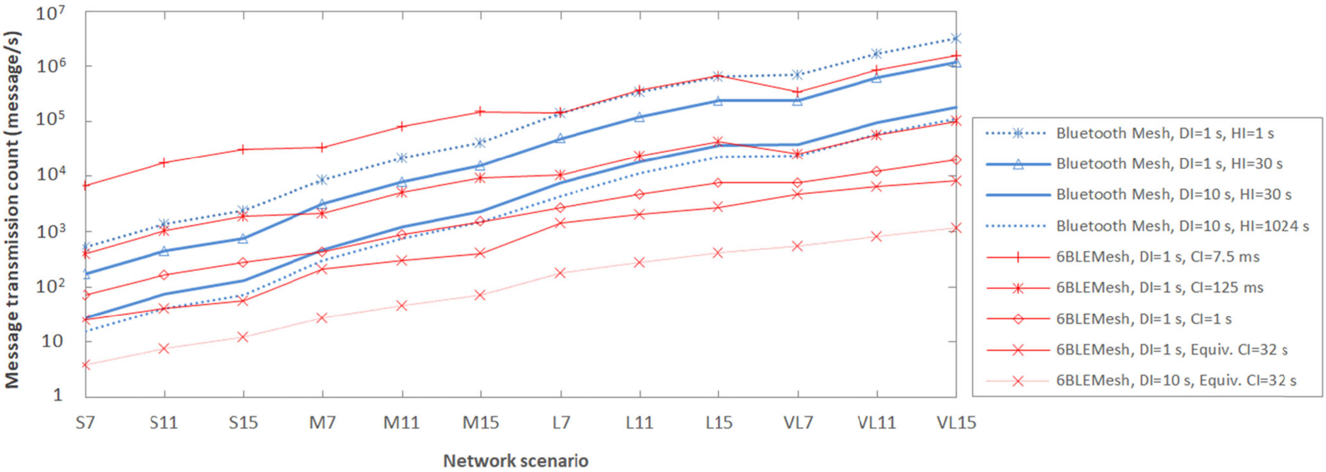

Fig. 4. Message count of Bluetooth Mesh and 6BLEMesh, for the scenarios shown in Table I, and for various parameter settings. CI and HI stand for *connInterval* and Heartbeat interval, respectively.

end-to-end packet delivery probability of both approaches for a network comprising M independent end-to-end paths of equal characteristics between a source and a destination, of N end-to-end uncorrelated hops each, and for a link delivery probability p of 0.6. As shown in Fig. 5, Bluetooth Mesh requires path diversity in order to achieve high packet delivery performance, especially for long end-to-end paths. In contrast, 6BLEMesh approaches ideal packet delivery probability, as long as R is set to a high enough value (e.g. a 99% packet delivery probability is achieved for a 10-hop path for $R \geq 7$ and p=0.6), at the expense of latency increase.

F. *Variable topology robustness*

A BLE mesh network exhibits variable topology for several reasons, including node mobility or node failure. If a path being used in 6BLEMesh for end-to-end communication fails, an alternative path (if any) is only used after detection of the problem. In addition, some routing protocols may need to reactively discover an alternative path. In consequence, in 6BLEMesh, a topology change prevents end-to-end connectivity during significant time, typically in the order of at least several seconds. Instead, the flooding, multipath approach in Bluetooth Mesh allows continuous end-to-end packet delivery, as long as an alternative path between the two communicating endpoints exists.

G. *Internet connectivity*

While 6BLEMesh naturally supports IPv6-based Internet connectivity, the Bluetooth Mesh standard does not. Therefore, connectivity of Bluetooth Mesh devices with the Internet requires a protocol translation gateway between the Bluetooth Mesh network and the Internet. The protocol translation gateway transforms message formats received on one interface to those used on the other one, and vice versa. This solution is feasible, and even desirable in some cases for the sake of privacy for the Bluetooth Mesh network. However, it limits application development scalability (since applications on the Bluetooth Mesh network side need to be designed specifically for Bluetooth Mesh, and cannot be used over other technologies), it encumbers protocol consistency at both sides of the protocol translation gateway, and it precludes use of well-known IP-based tools and protocols for end-to-end connectivity, security and management.

VI. Design Goals and Performance

The different characteristics and performance of Bluetooth Mesh and 6BLEMesh are due to their respective design goals.

The main application domain for Bluetooth Mesh is smart home. In this domain, small network diameter (often up to 4 hops [2]) and good path diversity are expected. In such conditions, Bluetooth Mesh's flooding performs reasonably well in terms of message transmission count and link corruption robustness, while avoiding connectivity gaps due to topology changes. In contrast, 6BLEMesh was not created for a particular application area. 6BLEMesh follows a generic approach based on unicast routing on top of typically persistent Link layer connections. Thus, in 6BLEMesh, message transmission count and link corruption resiliency scale better with network size and density than Bluetooth Mesh, at the penalty of connectivity gaps after route failures.

Finally, note that intrinsic Internet connectivity support was not considered for Bluetooth Mesh, whereas it was a fundamental goal for 6BLEMesh leading to the IPv6-centric design of the latter.

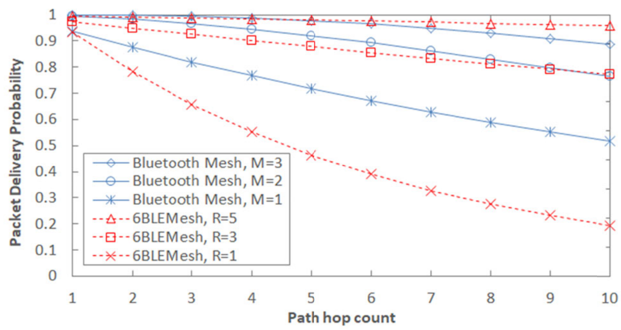

Fig. 5. Packet delivery probability for Bluetooth Mesh and 6BLEMesh for a network comprising M independent N-hop paths between two endpoints and R retries, for p=0.6.

VII. Conclusions

Bluetooth Mesh and 6BLEMesh offer fundamentally different BLE mesh networking solutions. Their performance depends significantly on their parameter configuration. Nevertheless, the following conclusions can be obtained. Bluetooth Mesh exhibits slightly greater protocol encapsulation overhead than 6BLEmesh. Both Bluetooth Mesh and 6BLEMesh offer flexibility to configure per-hop latency. For a given latency target, 6BLEMesh offers lower energy consumption. In terms of message transmission count, both solutions may offer relatively similar performance for small networks; however, 6BLEMesh scales better with network size and density. 6BLEMesh approaches ideal packet delivery probability in the presence of bit errors for most parameter settings (at the expense of latency increase), whereas Bluetooth Mesh requires path diversity to achieve similar performance. Bluetooth Mesh does not suffer the connectivity gaps experimented by 6BLEMesh due to topology changes. Finally, 6BLEMesh naturally supports IP-based Internet connectivity, whereas Bluetooth Mesh requires a protocol translation gateway.

Acknowledgment

Carles Gomez has been partly funded through Jose Castillejo grant CAS18/00170 and project TEC2016-79988-P. He contributed to this work partly as a visiting scholar at the Computer Laboratory of the University of Cambridge.

References

[1] D. Hortelano et al., "From Sensor Networks to Internet of Things. Bluetooth Low Energy, a Standard for this Evolution", Sensors, Vol. 17, No. 2, Feb. 2017, Art. No. 372.

[2] G. Ferrari et al., “Wireless Sensor Networks: Performance Analysis in Indoor Scenarios”, EURASIP Journal on Wireless Communications and Networking, Vol. 2007, Jan. 2007, Art. No. 081864.

[3] H.S. Kim et al., “Thread/OpenThread: a Compromise in Low-Power Wireless Multihop Network Architecture for the Internet of Things”, IEEE Comm. Mag., Vol. 57, No. 7, Jul. 2019, pp. 55-61.

[4] G. Patti et al., “A Bluetooth Low Energy Real-Time Protocol for Industrial Wireless Mesh Networks”, in *Proc. IECON 2016*, Firenze, Italy, 2016, pp. 4627-4632.

[5] M. Baert et al., “The Bluetooth Mesh Standard: An Overview and Experimental Evaluation”, Sensors, Vol. 18, No. 8, Jul. 2018, Art. No. 2409.

[6] C. Gomez et al., “IPv6 Mesh over BLUETOOTH(R) Low Energy using IPSP”, IETF Internet Draft, Sep. 2019 (Work in progress). https://tools.ietf.org/html/draft-ietf-6lo-blemesh-07 (Accessed on Jan. 19, 2020.)

[7] M. Collotta et al., “Bluetooth 5: a Concrete Step Forward Toward the IoT”, IEEE Comm. Mag., Vol. 56, No. 7, Jul. 2018, pp. 125-131.

[8] M. Spörk et al., “BLEach: Exploiting the Full Potential of IPv6 over BLEin Constrained Embedded IoT Devices”, in Proc. ACM SenSys’17, Delft, Netherlands, 2017, Art. No. 2.

[9] J. Hui, D.E. Culler, "Extending IP to Low-Power, Wireless Personal Area Networks", IEEE Internet Computing, Vol. 12, No. 4, July 2008, pp. 37-45.

[10] J.G. Ko et al., "Connecting Low-Power and Lossy Networks to the Internet." IEEE Comm. Mag., Vol. 49, No. 4, Apr. 2011, pp. 96-101.

[11] C. Bormann et al., "CoAP: an Application Protocol for Billions of Tiny Internet Nodes", IEEE Internet Computing, Vol. 16, No. 2, Mar./Apr. 2012, pp. 62-67.

[12] C. Bormann et al., “CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets”, RFC 8323, Feb. 2018. https://tools.ietf.org/html/rfc8323 (Accessed on Jan. 19, 2020.)

[13] A. Brandt et al., “Home Automation Routing Requirements in Low-Power and Lossy Networks, RFC 5826, Apr. 2010. https://tools.ietf.org/html/rfc5826 (Accessed on Jan. 19, 2020.)

[14] M. Goyal et al., “Reactive Discovery of Point-to-Point Routes in Low-Power and Lossy Networks”, RFC 6997, Aug. 2013. https://tools.ietf.org/html/rfc5826 (Accessed on Jan. 19, 2020.)

List of Acronyms

6BLEMesh: IPv6-based BLE Mesh Networks
6LBR: 6LoWPAN Border Router
6LN: 6LoWPAN Node
6LoWPAN: IPv6 over Low Power Wireless Personal Area Network
6LR: 6LoWPAN Router
ATT: Attribute protocol
BLE: Bluetooth Low Energy
CoAP: Constrained Application Protocol
DI: Data message interval
GATT: Generic Attribute profile
IoT: Internet of Things
L2CAP: Logical Link Control and Adaptation Protocol
LPN: Low Power node
ND: Neighbor Discovery
RPL: Routing Protocol for Low-power and lossy networks
TTL: Time To Live

Biographies

Seyed Mahdi Darroudi is a PhD student at UPC. He has held several academia and industry positions. His main research interests comprise wireless technologies and the IoT.

Carles Gomez is an associate professor at UPC. He is a coauthor of 100+ technical contributions, and a chair of the IETF 6Lo working group.

Jon Crowcroft is the Marconi Prof. of Communications Systems in the Computer Lab., University of Cambridge. He has worked on Internet protocols for over 30 years.