A Bluetooth Low Energy real-time protocol for industrial wireless mesh networks

Gaetano Patti, Luca Leonardi, Lucia Lo Bello
Department of Electrical, Electronics and Computer Engineering
University of Catania
Catania, Italy
e-mail: {gaetano.patti, lobello}@unict.it

***Abstract*—Low energy consumption and low cost are some of the primary issues that have to be addressed in Industrial Wireless Sensor Networks (IWSN). Such networks are being increasingly used to acquire sensory data that need to be processed in real-time. The Bluetooth Low Energy (BLE) protocol is an attractive solution for implementing low-cost IWSN with reduced energy consumption and high flexibility. However, the current BLE standard does not provide real-time support for data packets and is limited to a star topology. This paper presents a real-time protocol for industrial wireless mesh networks that is developed on top of the BLE and overcomes these limitations. The paper describes the protocol and provides analysis and experimental results.**

Keywords—Bluetooth Low Energy; Industrial wireless mesh network; Industrial wireless sensor network; Real-time.

1. Introduction

Bluetooth Low Energy (BLE) is a short-range wireless transmission technology intended for low-power low-cost and low-complexity communications [1] that offers interesting properties for Industrial Wireless Sensor Networks (IWSNs). BLE significantly reduces the power consumption of the communication stack, thus guaranteeing a longer lifetime than other wireless protocols. However, BLE is not suited for supporting real-time traffic, since message delays cannot be precisely bounded. Furthermore, BLE also suffers from other limitations (e.g., on the distance between nodes and on the network topology) that make it difficult to build a BLE mesh network featuring multiple hops. These limitations reduce the extent to which the coverage of a BLE network can be expanded. To overcome these limitations while maintaining the advantages of low-energy and low-cost communications this paper proposes the Real-Time BLE (RT-BLE), a real-time protocol for industrial wireless mesh networks developed on top of BLE. The proposed protocol exploits a Time Division Multiple Access (TDMA) approach with an optimized transmission allocation that provides data packets with real-time support, while maintaining a good trade-off between the maximum guaranteed delay and the throughput.

The paper provides a twofold contribution. *a)* A configuration method for the BLE standard that guarantees bounded message latencies with star topologies; *b)* a protocol working on top of BLE that allows for meshed topologies while maintaining bounded latencies.

The paper is organized as follows. Section II overviews related work, while Section III describes BLE and the proposed configuration method to achieve bounded latencies on star topologies. Section IV presents the RT-BLE protocol design, while Section V provides the protocol analysis. Section VI presents some experimental results. Finally, Section VII gives our conclusions and hints for future work.

2. Related Work

Many papers investigated the case for Bluetooth-based industrial communications, also proposing improvements and new features [2]. Recently, BLE raised some interest in the scientific community. The paper [3] analytically addresses the maximum peer-to-peer throughput and the minimum turnaround time of a BLE packet, pointing out some limitations of the results achieved by off-the-shelf transceivers compared with the analytical results. Another analytical model for calculating the maximum throughput of BLE was presented in [4], while the work in [5] analyzed and used a simulation tool to estimate the speed and latency of BLE communications. The BLE energy consumption was addressed in [6] and [7], and the results in [6] show that the energy consumption of a BLE transceiver is 2.5 times lower than that of a transceiver based on the IEEE 802.15.4 protocol. The results obtained in all the above mentioned works show that it is worthwhile to investigate novel extensions of BLE. In particular, it is interesting to find a way to improve the BLE performance on real devices through configurations specifically optimized for industrial environments.

New transceivers allowing for the co-existence of multiple communication stacks make it possible to develop dual protocols. In [8] an approach to use BLE in a dual-protocol environment to support real-time traffic is proposed. However, the approach in [8] provides real-time behavior only for communications between nodes that implement the specific dual protocol addressed. Conversely, the RT-BLE protocol proposed in this paper works on standard BLE devices.

The BLE standard is limited to a star topology. In order to increase the communication coverage, it is possible to use a network with a tree topology, as it provides for multi-hop relaying. The work in [9] proposes an approach to implement a scalable tree-based network using BLE network that aims at expanding the BLE network coverage, thus increasing its scalability. However, the approach in [9] does not provide data packets with real-time support, as it is intended for Internet of Things (IoT) applications. Conversely, the RT-BLE proposed in this work increases the BLE network scalability while supporting real-time traffic.

3. Achieving bounded latencies on BLE networks

A. Overview on BLE

In BLE once a connection is established, two Link Layer (LL) roles for the devices are defined, i.e., master and slave. The master coordinates the medium access adopting a TDMA-based polling mechanism, in which it periodically polls the slaves. In BLE [1] the time is divided into units called *Connection Events* (CE). Frequency hopping is realized on each CE, i.e., in each CE a different channel is used. The connection event starts with the transmission of a data packet from the master to a slave. The start time of a CE is called an Anchor Point (AP). The start times of connection events are regularly spaced, with a configurable interval called a Connection Interval (CI). During a connection event, the master and one slave transmit and receive packets alternately. The connection event is considered to be open as long as the devices continue to transmit packets. If none of the devices has data to transmit, the slave switches to the sleep mode until the next AP. At this point, the master starts the communication with another slave and the process repeats until all the slaves have transmitted. At that point, the master also goes to sleep until the next connection event starts. This way, the master basically splits the connection interval into as many connection events as the number of connections. Consecutive packet transmissions are separated by an Interframe Space (IFS) whose duration is 150 µs. Fig. 1 shows an example in which there are two slaves (S1 and S2) and one master (M). The bottom line of Fig. 1 indicates the connection events for the slave S1 and S2, which are periodically repeated with a period equal to CI. The top line of Fig. 1 presents the details of each connection event. At the anchor point of S1 (AP_{S1}) the master transmits a packet to S1, thus starting the polling and the alternating transmission sequence (M->S1, S1->M, etc.). At the end of the CE_{S1}, the master starts the connection event for the S2 (CE_{S2}), which is periodically repeated with a period equal to CI. While the CI is the same for all the slaves connected to the same master, the relevant anchor points are shifted, as the master polls one slave at a time. The BLE specification also defines a parameter, called *Connection Slave Latency*, which specifies the number of consecutive CEs that a slave has to skip while remaining in sleep mode.

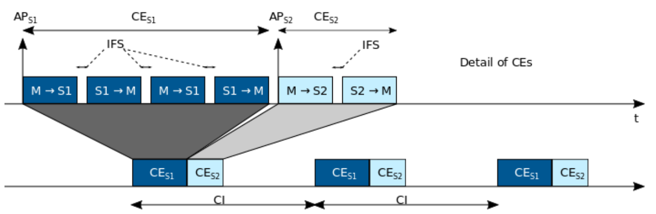

Fig. 1 Example of the BLE medium access mechanism.

The slave shall always send a reply to the master for any packet received, valid or not [1]. The ack for the previous packet is transmitted within the header of the reply to the master.

Several BLE implementations [12] require a *guard band* (GB) between the connection events to cope with synchronization accuracy. Moreover they provide configurable parameters to set the maximum size of each connection event.

B. Configuring BLE to obtain bounded latencies

This section explains how to set the main configuration parameters in order to enforce bounded message latencies over BLE networks with a star topology. A mathematical formulation is provided and discussed.

The maximum packet length (L) that can be transmitted at the physical layer is 47 bytes [1], thus the time needed for transmitting it at 1 Mbps, here called T_L, is equal to 376 µs. Hence, the duration of the connection event j ($T_s^{\,j}$) that can embed the transmission of M_j maximum-length data packets can be calculated as

$$T_S^j = 2M_jT_L + (2M_j - 1)IFS. \qquad (1)$$

Equation (1) takes into account the time for M_j master/slave – slave/master transmissions within the j-th CE plus the IFS intervals between the transmissions. Taking into account Eq. (1), the maximum time that the master takes for handling S connection events (T_{trx}) (S is the number of connections) can be calculated as

$$T_{trx} = (S-1)GB + \sum_{j=0}^{S} T_S^j. \qquad (2)$$

where (S-1) indicates the number of GB intervals between the connection events. The standard specifies that the connection interval can be configured as a multiple of 1.25ms in the range from 7.5ms to 4s [1]. Hence, we have to calculate the minimum multiple of 1.25ms that is higher than or equal to T_{trx} so that the minimum connection interval (CI_{min}) for S connection events can be calculated as

$$CI_{min} = max\left(\left\lceil \frac{T_{trx} + IFS}{1.25ms} \right\rceil 1.25ms, 7.5ms\right). \qquad (3)$$

The BLE protocol with a suitable configuration (i.e., with bounded connection events) allows for bounded delays when the network has a star topology, so a master can rule the timing and synchronization of all the network nodes. Some issues raise in the networks in which the slaves are connected to multiple masters (i.e., a slave holds connections with multiple BLE networks). In fact, as there are multiple masters and each of them provides synchronization to the associated slaves, it may happen that the CEs assigned to the same slave by different masters overlap. This can happen, for instance, in the topology that is shown in Fig. 2.

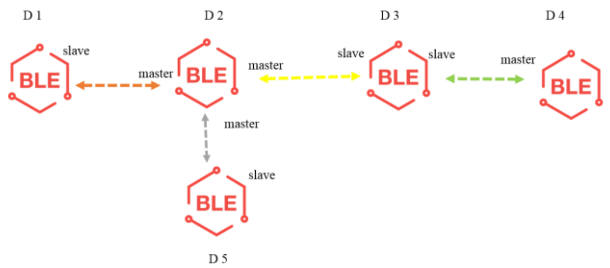

Fig. 2 Example of topology with a potential for CE overlap.

In fact, the node D3 may experience a CE overlap problem, as such a node is connected to two different masters, i.e., the nodes D2 and D4. In the worst case, the two masters transmit a packet to the node D3 at the same time, thus determining a CE total or partial overlap. In particular, one of the four cases depicted in Fig. 3 may occur.

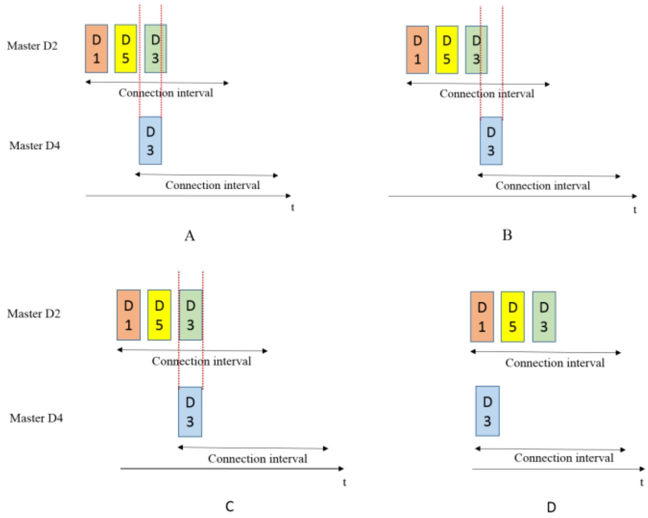

Fig. 3 Possible overlap of the connection events.

In Fig. 3A and 3B the CE overlap is partial, while in Fig. 3C the overlap is total. In Fig. 3D no overlap occurs.

Section IV proposes a solution to this problem that enables real-time communications over BLE and also offers support for mesh network topologies.

4. Protocol Design

The RT-BLE protocol proposed in this paper enables BLE to achieve bounded message delays, thus providing support for real-time communications, and also introduces multi-hop data transmissions, thus allowing for the creation of meshed networks. The main idea to achieve this result is to create multiple networks (here called sub-networks), each one coordinated by a master. The sub-networks in turn share one or multiple slaves that act as “bridges” between the sub-networks. Fig. 4 shows an example of network topology in which there are three sub-networks, coordinated respectively by nodes 2, 4, and 6.

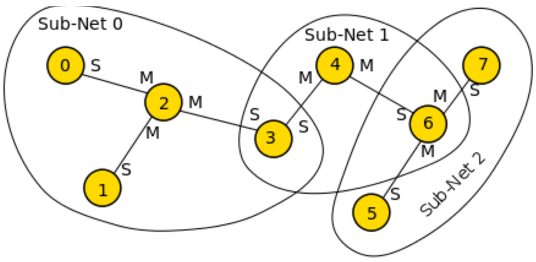

Fig. 4 Example of RT-BLE network topology.

In Fig. 4, the node 3 is a slave of both the node 2 (i.e., the master of the sub-network 0) and the node 4 (i.e., the master of the sub-network 1). The node 6 is the master for the nodes 5 and 7, while it is a slave of the node 4. In the example of Fig. 4, the node 3 has to communicate with two masters that are not synchronized with each other. Consequently, due to the potential for CE overlap, the transmissions to node 3 cannot be guaranteed.

To deploy a meshed network that avoids the CE overlap, the nodes have to be configured according to the following rules:

1. A node not acting as a master can establish a connection with up to two masters.
2. A node (A) acting as a master can establish a connection with at most another master (B). In this connection, the node A shall play the slave role.

Note that, according to the BLE specifications [1], a device can act both as a master and a slave for different connections and can also establish connections with several masters. Several devices, such as [12], support different roles at the same time.

In order to enable a slave node to be on the right sub-network at the right time, the connection intervals of the different sub-networks have to be the same. To this aim and to avoid CE overlap, the connection interval for each sub-network has to be calculated, using Eq. (3). Then the CI for any sub-network, henceforward called CI* is chosen equal to the maximum of the CI_{min} values among all the sub-networks.

The solution proposed in this paper to solve the problem of CE overlap and to enable real-time communications on BLE-based meshed networks is explained through the case-study shown in Fig. 5, in which two master nodes are connected with four slave nodes each.

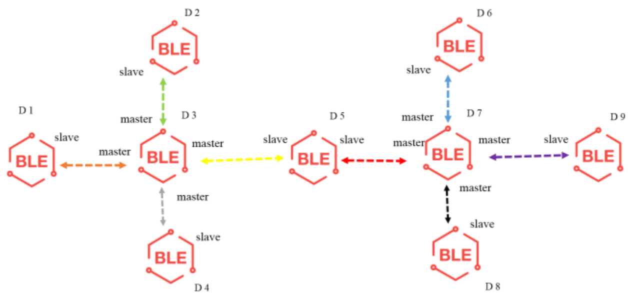

Fig. 5 Case-study topology for the RT-BLE approach.

The node D5 is a slave connected with the nodes D3 and D7, therefore the connection events of the node D5 may overlap. The solution here proposed is to alternate the connection of the node D5 with a master at a time. This means that in a given connection interval the node D5 will disable the connection with the node D3, while for the next one it will enable the connection with the node D3 and disable the connection with the node D7. This can be easily realized, as at the higher levels of the BLE stack a mechanism to enable/disable a connection setting the Client Characteristic Configuration Descriptor (CCCD) is provided. This is a two-bit field that works like a real switch. It allows one of the two nodes to enable or disable a connection, notifying both the nodes about the new status of the connection. The slave D5 in Fig. 5 can modify these bits at any time and each master, before sending data, has to check that the connection is enabled. If this is not the case, the master inserts the packet in a queue and transmits it in the next connection event. This way the CE overlap is avoided.

Fig. 6 shows the timing of the masters D3 and D7. The semi-transparent bigger rectangles represent the intervals during which the node D5 disables the connection with one of the two masters.

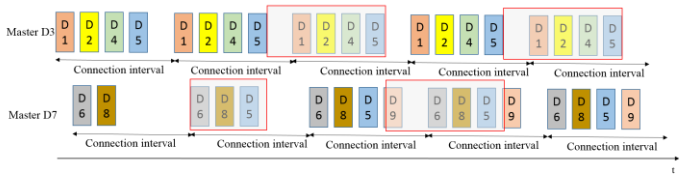

Fig. 6 The solution to the overlapping CEs problem.

Fig. 6 shows that the node D5 is initially connected as a slave to the node D3. Subsequently the master D7, which in the initial phase has only two connections (with the nodes D6 and D8, respectively), establishes a connection with the node D5 (and later with the node D9). The master D7 allocates one CE for the connection with the node D5, which is a shared slave with two connections. Once the connection with the node D7 is established, it is immediately disabled (semi-transparent rectangle). As soon as the node D5 transmissions with the node D3 finish, the node D5 disables the connection with the node D3 for a time equal to the connection interval and enables the connection with the node D7, for a connection interval. This mechanism repeats over time.

In the proposed approach the network is configured offline. Dynamic configuration will be investigated in future works.

5. Timing Analysis and Analytical Assessment

This section presents a timing analysis to determine the end-to-end worst case latencies. We consider a network made up of *N* nodes. Each node generates multiple messages belonging to different flows (*f*). Messages are periodically transmitted with a period P_f. The routing path (R_f) for each flow is fixed and configured offline. To improve fault-tolerance, each node has to maintain a routing table with a backup path for each flow (this way the analysis can be repeated for each path). Each flow is thus characterized by the pair (P_f, R_f), where $R_f=(n_0^f, \ldots, n_h^f)$ is the vector of the nodes (*n*) that a message of the flow *f* has to traverse to reach the destination (n_h), while *h* is the number of hops.

For the sake of simplicity, here it is assumed that each node has assigned one CE within the connection interval. CEs are sized so as to provide room in every connection interval for all the messages that are transmitted and/or forwarded by the node. This way, all the messages in the transmission queue of the node are transmitted within one CE.

According to the approach presented in Sect. IV, a node has the opportunity to transmit its messages every two connection intervals and connection intervals have the same length by design o by configuration, as it was explained in Sect. IV. Therefore, the maximum time a node has to wait to transmit its messages (*WT*) is equal to

$$WT = 2\ CI^* \qquad (4)$$

where CI^* is the maximum of the CI_{min} values among all the sub-networks. To calculate CI_{min}, according to Eq. (1), (2) and (3), the maximum number of messages (M_j) that the node *j* has to transmit is calculated as

$$M_j = \sum_{f:\, j \in R_f} 1, \qquad (5)$$

i.e., the number of flows that are generated by the node j plus those that have node *j* in their routing path (*R*). Finally, the end-to-end worst case latency for the flow *f* is calculated as the product of WT and the number of hops (*h*) the flow *f* has to traverse to reach the destination, i.e.,

$$RT_f = WT \cdot h^{(f)}\ . \qquad (6)$$

A. Analytical results.

This subsection presents an analytical evaluation of the RT-BLE protocol with aim of providing some quantitative performance indicators. The first computed metrics is the minimum connection interval, calculated using Eq. (3). The evaluation was performed varying the number of nodes (*N*) and the number of messages transmitted from each node (*M*). For the analysis, it was assumed that one CE is assigned to each node (i.e., S = N). Results are shown in Fig. 7.

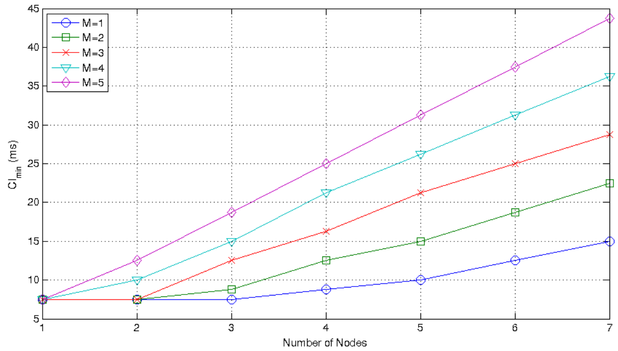

Fig. 7 Minimum connection intervals (CI_{min}) varying the number of nodes (*N*) and the number of messages transmitted from each node (*M*).

Figure 7 shows that the minimum connection interval grows linearly with the number of nodes. In particular, in all the evaluated cases, the CI_{min} value is lower than 50ms. Such a value is close to and comparable with the network duty cycle of other IWSN protocols (e.g., those based on the IEEE 802.15.4 with its industrial flavors [10], [11]).

The second computed metrics is the worst case end-to-end latency, calculated as in Eq. (6). In the considered scenario, each node periodically transmits one packet to one sink node. Nodes are connected in a daisy-chain topology (Fig. 8). Such a topology represents the worst case for our protocol, as increasing the number of nodes, the number of flows for all the nodes in the chain also increases. The results, shown in Fig. 9, are obtained varying the number of hops (h) and the number of chains connected to the sink node (each chain is a network segment, with a daisy-chain topology, connected to the sink node as shown in Fig. 8).

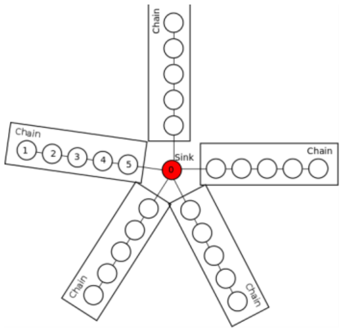

Fig. 8 Assessed network made up of 5 chains of 5 nodes each.

Fig. 9 refers to the case of one subnet and five nodes, where the Node 1 transmits one message to the Node 2, the Node 2 transmits two messages to the Node 3, and so on. Fig.9 shows that the maximum latency is lower than 100ms. In the case of 5 chains (i.e., 5 network segments with 5 nodes each) the maximum latency is lower than 350ms. Such a result proves that the RT-BLE approach makes BLE suitable for industrial WSN, as the maximum latencies achieved are comparable with those required for industrial communications [14].

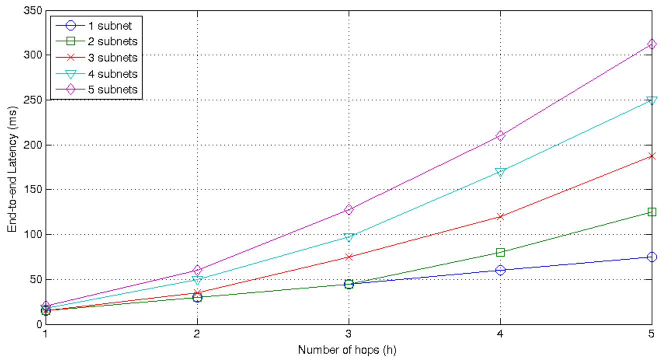

Fig. 9 Worst case end-to-end latency results.

6. Experimental Results

This section presents some experiments carried out to characterize the feasibility of the RT-BLE approach on real devices and to measure the delay experienced by real-time data exchanges.

The testbed was devised so as to obtain the latency bound on a single hop, which represents the most important condition for the feasibility of the proposed approach.

Measurements were performed using the X-NUCLEO-IDB05A1 devices produced by STMicroelectronics. These devices are equipped with SPBTLE-RF BlueNRG-MS, i.e., a communication module compliant with the Bluetooth Specification v4.1[1].

The testbed was composed of 4 devices, as shown in Fig. 10. Device A was the master for the devices B, C and D (slaves). All the slave nodes periodically generated one packet with a period equal to the CI. Only the mechanisms of RT-BLE for single-hop real-time communications were implemented in this phase, to prove that, with a suitable configuration of the CE lengths and of the connection interval, the network provides bounded latencies.

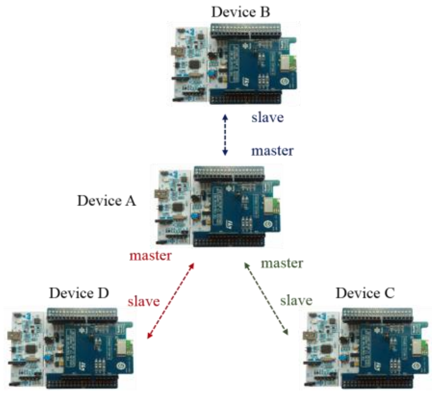

Fig. 10 Implemented topology.

In the implemented topology, which is shown in Fig. 10, the connection interval was set to 20ms, as the Blue-NRG transceiver communicates with the MCU through the SPI port and this entails higher latencies.

The end-to-end delay is the time interval between the sending time of the application packet at the transmitter and the receiving time of the packet at the receiver. The results relevant to the packet end-to-end delay for a connection are shown in Fig. 11.

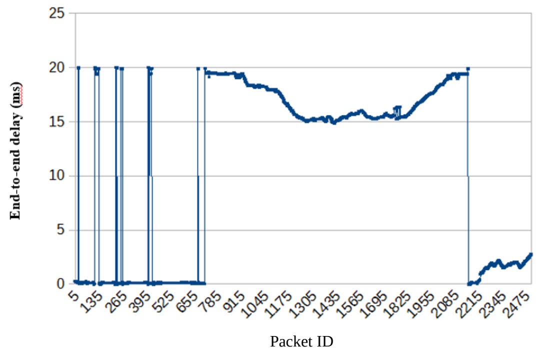

Fig. 11 End-to-end delay.

As it was expected, the maximum end-to-end delay that was measured is 20ms, so it is equal to the connection interval. Thanks to the configuration methodology presented in Sect. IIIB, the end-to-end delay for a packet is always lower than the connection interval configured, as all the messages generated from a node are transmitted in the same connection event and all the connection events are scheduled within a single CI. This result provides an evidence of the latency bound on a single hop, which represents the most important condition for the feasibility of the proposed approach.

7. Conclusions

This paper proposed RT-BLE, a protocol for industrial wireless mesh networks developed on top of BLE that provides real-time support for data packets and extends the network scalability and coverage. The paper also presented an analytical evaluation of the connection intervals and of the worst case end-to-end latencies. The results obtained show that RT-BLE allows for latencies that are comparable to those of other IWSN protocols and prove that RT-BLE is suitable for industrial applications. Moreover, a proof-of-concept implementation of the same mechanisms as RT-BLE showed its feasibility on COTS devices.

Further work will investigate a dynamic configuration mechanism combined with load balancing techniques, such as those investigated in [13], to provide more flexibility and to increase reliability and fault-tolerance. Moreover, a full-fledged version of RT-BLE will be developed on COTS devices to assess the protocol in a realistic multi-hop scenario.

References

[1] Bluetooth SIG, “Bluetooth Core Specification Version 4.1”, Dec. 2013.

[2] M. Collotta, L. Lo Bello, O. Mirabella, "An innovative frequency hopping management mechanism for Bluetooth-based industrial networks", IEEE International Symposium on Industrial Embedded Systems (SIES), Trento, Italy, IEEE, pp. 45–50, July 2010.

[3] K. Mikhaylov, N. Plevritakis, J. Tervonen, “Performance Analysis and Comparison of Bluetooth Low Energy with IEEE 802.15.4 and SimpliciTI”, Journal of Sensor and Actuator Networks 2013, pp. 589-613.

[4] C. Gomez, I. Demirkol, J. Paradells, “Modeling the Maximum Throughput of Bluetooth Low Energy in an Error-Prone Link”. IEEE Commun. Lett. 2011, 15, 1187–1189.

[5] C. Gomez, J. Oller, J. Paradells, “Overview and Evaluation of Bluetooth Low Energy: An Emerging Low-power Wireless Technology”, Sensors 2012, 12, 11734–11753.

[6] S. Kamath, J. Lindh, “Measuring Bluetooth® Low Energy Power Consumption”, AN092S (WRA347a), Texas Instruments, Inc.: Dallas, TX, USA, 2012; pp. 1–24.

[7] M. Siekkinen, M. Hiienkari, J.K. Nurminen, J. Nieminen, “How Low Energy is Bluetooth Low Energy? Comparative Measurements with ZigBee / 802.15.4”, in Proceedings of IEEE Wireless Communications and Networking Conference Workshops (WCNCW’12), Paris, France, 1 April 2012; pp. 232–237.

[8] M. Marinoni, G. Franchino, D. Cesarini, A. Biondi, P. Buonocunto, G. Buttazzo, “Dual-protocol support for Bluetooth LE device”, Proc. 13th International Conference on Industrial Informatics, Cambridge, pp. 919-922, July 2015.

[9] J. Kim, S. Kang, J. Park, “Bluetooth-based Tree Topology Network for Wireless Industrial Application”, Proc. 15th International Conference on Control, Automation and System, Busan, South Korea, October 2015, pp. 1305-1308.

[10] G. Alderisi, G. Patti, O. Mirabella, L. Lo Bello, “Simulative assessments of the IEEE 802.15.4e DSME and TSCH in realistic process automation scenarios”, Proc. 13th International Conference on Industrial Informatics, Cambridge, pp. 948-955, July 2015.

[11] E. Toscano, L. Lo Bello, "A multichannel approach to avoid beacon collisions in IEEE 802.15. 4 cluster-tree industrial networks", IEEE Int. Conf. of Emerging Technologies and Factory Automation (ETFA), pp. 1–9, Mallorca, Spain, Sept. 2009.

[12] “BlueNRG, BlueNRG-MS stacks programming guidelines”, online available. http://www.st.com/web/en/resource/technical/document/programming_manual/DM00141271.pdf

[13] M. Collotta, L. Lo Bello, E. Toscano, O. Mirabella, "Dynamic load balancing techniques for flexible wireless industrial networks", Annual Conference on IEEE Industrial Electronics Society (IECON), Phoenix, AZ, USA, IEEE, pp. 1329–1334, Nov. 2010.

[14] A. Flammini, P. Ferrari, D. Marioli, E. Sisinni, A. Taroni, “Wired and wireless sensor networks for industrial applications.” (2009) Microelectronics Journal, 40 (9), pp. 1322-1336.