# Dual-Protocol Support for Bluetooth LE Devices

Mauro Marinoni, Gianluca Franchino, Daniel Cesarini, Alessandro Biondi, Pasquale Buonocunto, Giorgio Buttazzo

*{name.surname}@sssup.it*
*Scuola Superiore Sant'Anna, Pisa, Italy*

***Abstract*—Low energy consumption is one of the primary issues that have to be addressed in body area networks to prevent frequent battery recharges in the nodes. 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 personal area networks with reduced energy consumption, also because it is supported by consumer devices such as tablets and smart phones; however, it cannot guarantee a bounded delay for managing real-time traffic.**

**This paper overcomes such a limitation by presenting a bandwidth sharing mechanism that allows partitioning the available network bandwidth between the BLE and another user-defined protocol built on top of the raw radio transceiver. Experimental results are also reported to characterize the timing behavior of the dual protocol on a specific platform.**

## I. Introduction

The increasing availability of computing and communication devices connected to the Internet (as smart phones) has pushed the development of sensor networks for acquiring data from the environment, using the device as a sink for collecting, processing, and storing the data, and as a gateway for sending them to a remote repository. For example, in a body area network (BAN), different types of sensor nodes can be used to monitor various physiological parameters, like ECG, blood pressure, oxygen saturation, and temperature. These data need to be analyzed in real-time on the mobile device to detect critical situations and possibly send an alarm signal over the Internet to notify the proper medical center. In other applications (e.g., telerehabilitation), multiple inertial sensors can be used to monitor the position and orientation of specific limbs [1]. In this case, real-time data integration is required to reconstruct the posture of the patient or the action performed during the execution of exercises, for evaluating the exercise performance and providing a proper feedback to the patient. To reduce the error in the reconstructed position, it is crucial that all the values coming from the different sensors are associated with a common time reference. Real-time clocks used in sensor nodes, however, are subject to small variations that in the long term produce large differences on the relative time values. To reduce such time differences, nodes must be synchronized using a proper synchronization protocol, which exploits internode real-time communication to adjust local clocks rates.

Another crucial problem that has to be addressed in body sensor networks is energy consumption. In fact, since nodes are powered by batteries, they have to be managed to reduce energy consumption as much as possible, to avoid frequent annoying recharges. Low-power communication protocols, as BLE [2] or ANT+ [3], can be used to guarantee a longer lifetime, with the advantage that they are supported by most mobile devices, as smart phones and tablets. Although energy efficient, however, these protocols are not suited for supporting real-time traffic, since message delays cannot be precisely bounded. On the other hand, other protocols proposed for managing real-time traffic in wireless sensor networks (see Section II) are not supported by commercial mobile devices.

To overcome this limitation and achieve both advantages of a real-time communication among nodes and a standard low-energy connectivity with mobile devices, this paper proposes a bandwidth sharing mechanism that allows BLE to use only a fraction of the available network bandwidth, reserving the remaining part to another custom protocol that can be used for real-time messages and node synchronization. The proposed approach exploits the capability provided by some new radio transceivers to switch between a commercial protocol and the transmission of raw data, which can be used to implement a real-time custom protocol. From our knowledge, this is the first approach in which the BLE is used in a dual-protocol setting to support real-time traffic. The dual-protocol solutions that exist today have been developed by the industry mainly for reducing cost, space, and power consumption. For instance, Broadcom [4] proposed a technology that combines Bluetooth and BLE for supporting both protocols with one radio, while Nordic [5] integrated two previous solutions to obtain a device combining BLE with ANT+.

The rest of the paper is organized as follows. Section II briefly presents the state of the art on the key aspects addressed in the paper. Section III illustrates the system architecture and the related model. Section IV describes the analysis methodology for guaranteeing the real-time traffic. Section V presents some experimental results carried out on the system to better characterize its behavior. Section VI states our conclusions and future work.

## II. State of the art

This section briefly presents the state of the art on the different aspects addressed in the paper. A lot of effort has been devoted to the development of communication protocols for BANs, focusing on throughput, communication range, energy consumption, security support, code complexity, and stack requirements.

Classic Bluetooth was developed with the aim of replacing cables on desktop computers in connecting devices such as printers, mobile phones, etc. It covers distances from $1$ m to $100$ m, depending on the radio class used in the implementation, with a throughput of $0.7 - 2.1$ Mbit/s. The maximum number of slaves is 7 and the peak current consumption is approximately $30$ mA. During the last years, it has been

This work has been partially supported by Telecom Italia.

extended into Bluetooth Low Energy (BLE), having a lower application throughput of $0.27\ \mathrm{Mbit/s}$, but supporting a higher number of concurrent communication slaves with lower peak power consumption. The main difference with respect to the classic Bluetooth is the communication paradigm, which has a lower overhead for establishing and maintaining connections with devices. With respect to Classic Bluetooth (Basis Rate, BR, and Enhanced Data Rate, EDR), the BLE has a much smaller memory requirement in the microcontroller stack [6].

On a different side, to guarantee a timely communication at the MAC layer in wireless networks, several solutions have been proposed in the literature, as wireless fieldbus protocols [7], [8]. Some other authors proposed to control the channel access by a real-time scheduling algorithm to prevent conflicts in transmission and guarantee message deadlines [9], [10]. Other authors proposed solutions for guaranteeing real-time communication and energy efficiency on IEEE 802.15.4 networks [11], [12].

The development of dual protocols was made possible by new transceivers allowing the co-existence of multiple communication stacks. Examples are the multimode IEEE 802.15.6/Bluetooth Low-Energy WBAN transceiver for biotelemetry applications proposed in [13] and the WiZi-Cloud, an application-transparent dual ZigBee-WiFi radio [14]. Another example is represented by the Nordic nRF51822 [15] device, that supports multiple protocols, such as ANT+ and BLE, other than providing support for custom protocols.

Other industrial projects proposed solutions to integrate multiple protocols in the same transmission device. Zehavi et al. [16] developed a wireless transceiver capable of a joint signal transmit/receive section and a number of signal up/down conversion sections to transmit and receive signals in accordance with two alternate protocols. In some of the developed devices, the transceiver is provided with a processor programmed to implement a time sharing schedule to facilitate the coordinated control and performance of transmit and receive operations. Protocols can be selected among Bluetooth, 802.11x, and Home RF. The wireless device can also be used as a master or gateway of two wireless networks.

Bridgelall [17] proposed a time sharing mechanism that allows alternating Bluetooth BR with 802.11. Time frames are reserved for each protocol by exploiting the 802.11 Request-To-Send mechanism to accommodate for Bluetooth transmission windows. Finally, Desai et al. [4] proposed a dual-mode BLE device that identifies idle intervals within Bluetooth BR/EDR traffic communication and uses the dual-mode BLE device to concurrently perform various BLE activities. For example, advertising packet transmissions may be concurrently performed inside identified idle intervals within the Bluetooth BR/EDR traffic communication.

All the industrial solutions presented above, however, mainly aim at integrating standard protocols to improve interoperability, without addressing real-time issues.

## III. System Architecture and Model

The network considered in this paper consists of $n$ nodes, $N_1, N_2, \ldots, N_n$, using the BLE protocol [2] to deliver best-effort traffic and a real-time custom protocol to deliver messages subject to timing constraints. BLE is used by each node to communicate with a single external device, working as a master of the BLE network (e.g., an Android tablet). This configuration is the simplest BLE network that can be formed, and it is referred to as Piconet.

Although the BLE protocol is quite flexible in terms of network formation, it is not able to provide support for timing guarantees and requires a multi-hop path for a communication between two non-master nodes. Moreover, not all devices provide an implementation of the full standard to reduce the footprint and power consumption. Typically, the companies producing the transceivers do not provide the code for the network stack, but only a library in binary format (e.g., Nordic SoftDevice [15]) that works as a black box. For this reason, in this paper, the BLE implementation is considered as a black box and the analysis is developed without making a priori assumptions on the BLE behavior, but rather using profiling to determine the parameters that mostly affect the real-time performance, as reported in the experimental section.

To share the channel between the two protocols, a bandwidth reservation mechanism is used to assign the real-time protocol a slot of $Q$ time units every $P$ time units. Thus, $P$ denotes the *period* of the communication protocol, while $Q$ denotes the total *budget* reserved in every period to real-time traffic.

To enable bandwidth reservation, some radio devices (as the Nordic, used in this paper) provide a mechanism for using the radio device in raw mode in specific time slots, interleaved with the BLE traffic. Such time slots, however, are provided with a variable delay $\delta$, which is necessary to complete the ongoing operation and release the radio device leaving the BLE protocol in a consistent state. To perform a safe guarantee test, the analysis proposed in Section IV is performed by considering an upper bound $\delta_{max}$ of the interval required to switch from BLE to the real-time protocol. Figure 1 shows an example of access sequence to the radio device by the two protocols.

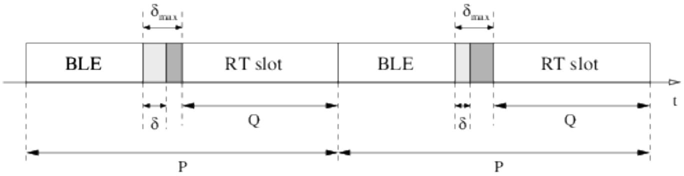

Figure 1. Example of the protocols access sequence to the radio.

To correctly boot the system in dual-protocols mode, the nodes perform the following initialization procedure: (i) nodes start using the radio in raw mode and self-organize in a cluster by executing a network formation procedure [18]; (ii) a synchronization protocol is executed to align the internal time references of the nodes, still exploiting the transceivers in raw mode (we assume that clocks are synchronized by using one of the existing techniques [19]); (iii) a common time instant $t_0$ at which the first real-time slot is requested by all nodes is defined as $t_0 = t + T_{ini}$, where $t$ is the current time and $T_{ini}$ is an interval sufficient for carrying out the operation described in the next two phases; (iv) the raw communication is disabled and the BLE protocol is initialized to allow the formation of the Piconet; (v) nodes start sending their sensory data to the external device using the BLE protocol; (vi) at time $t_0$, each node issues the first request for getting a real-time slot of length $Q$ and such a request is then repeated every period $P$.

## IV. Analysis

This section presents the analysis for guaranteeing the real-time traffic upon the proposed dual-protocol framework. As described in Section III, bandwidth partitioning is enforced by a reservation mechanism that provides a *budget* $Q$ every *period* $P$ for accessing the radio device in real-time. Hence, the bandwidth reserved to the real-time traffic is

$$U^{\mathrm{RT}} = \frac{Q}{P}. \tag{1}$$

When node $N_i$ issues a request for the raw mode at time $t$, it is guaranteed that the BLE will release the radio before time $t + \delta_{max}$. Note that, in the best case, the radio can be immediately released at time $t$; in this case, $\delta_{max}$ units of time are wasted, because, although the radio is available for the raw mode for node $N_i$, no real-time traffic can be guaranteed for all the nodes until $t + \delta_{max}$. The ratio

$$U^{w} = \frac{\delta_{max}}{P} \tag{2}$$

represents a wasted bandwidth because it cannot be guaranteed for the BLE, nor for the raw mode. As a consequence, the minimum bandwidth that can be guanranteed for the BLE traffic is

$$U^{\mathrm{BLE}} = 1 - U^{\mathrm{RT}} - U^{w} = 1 - \frac{Q + \delta_{max}}{P}. \tag{3}$$

Since the wasted bandwidth is proportional to $\delta_{max}$, characterizing such a delay is crucial for evaluating the performance of the dual protocol. For this reason, a few experiments aimed at measuring the delay $\delta$ are presented in the next section, under different conditions.

## V. Experimental Results

This section discusses some implementation details on a real platform and presents some experiments carried out to characterize the delay introduced by the BLE protocol to switch the radio in raw mode.

### A. Experimental setup

Experimental data have been derived on a network of wireless nodes built around the nRF51822 System on Chip (SoC) device produced by Nordic, combining a low-power ARM Cortex-M0 core with a 2.4 GHz radio transceiver. The core is endowed with 256kB of flash memory and 16kB of RAM memory. The transceiver supports the BLE communication stack plus a raw operation mode that it is used to run a custom real-time protocol based on Weighted Round-Robin. Nodes exchange data among each others using the custom protocol and send data to a master unit (consisting of an Android tablet) using the BLE protocol.

The software in the wireless nodes has been developed in C language on top of the Erika Enterprise real-time kernel [20]. The BLE stack is provided by Nordic as a binary library named SoftDevice. The access to the raw radio mode is performed through an asynchronous API, passing the interval length $Q$ and the delay $\delta_{req}$ by which the slot has to be granted. If the slot cannot be granted within $\delta_{req}$, because it is not compatible with the timing of the BLE protocol, the API returns an error code. Otherwise, it returns a success code and a slot start signal is sent to the application through a software interrupt.

In order to have a guaranteed bandwidth $U^{\mathrm{RT}}$ to support real-time traffic, each slot request must succeed. This is obtained by specifying a $\delta_{req}$ value equal to the upper bound $\delta_{max}$ compatible with a correct behavior of the SoftDevice, that is $\delta_{req} = \delta_{max}$.

### B. Experiments for evaluating the delay

A set of experiments has been carried out to characterize the $\delta$ parameter. Timing measurements were performed to profile the length of the interval between the instant $TS\_REQ$ when a time slot is requested using the API and the instant $INT\_TS\_START$ when the signal notifies that the radio can start transmitting using the custom protocol. In a first experiment, a sequence of requests has been issued by a periodic task, measuring the delay $\delta$ of each request. This experiment was repeated for different values of the period, namely, 10 ms, 20 ms, 35 ms, 50 ms, and 100 ms.

To evaluate the effect of the BLE communication on the delay, the experiment has been done in two load configurations: without data traffic (i.e., transmitting only the message needed for the advertisement phase) and with data traffic (i.e., sending packets of 4 bytes of payload at a frequency of 50 Hz). For each configuration, the delays have been acquired for an interval of 120 seconds. Figure 2 reports the delay distribution for the experiment performed with a period $P$ of 100 ms and BLE data traffic. The distributions produced with the other configurations are very similar and are not shown here for lack of space.

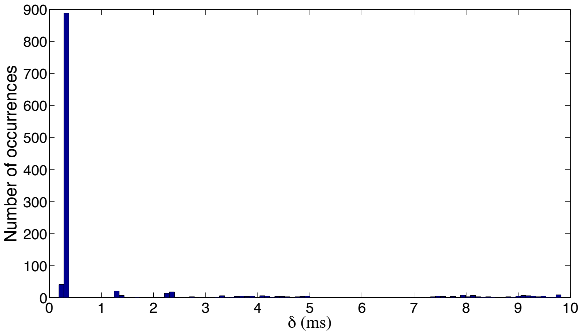

Figure 2. Delay distribution with a request period of 100 ms.

As it can be easily seen, the delay has a huge variability, mainly due to the different internal states in which the SoftDevice can be when the request is received. In particular, most of the requests are grated in about 0.4 ms, which is the time needed to switch the context when there are no ongoing BLE activities. However, the maximum measured delay is significantly larger and close to 10 ms.

A straightforward improvement can be achieved by synchronizing the requests with the activities of the BLE stack exploiting the Radio Notification feature provided by the SoftDevice [15]. However, this approach leads to a moderated improvement (i.e., maximum delay of 8 ms), as shown in Figure 3. This limited reduction is due to the transmission of advertisement packets with a period of 30 ms and a length of multiple data packets.

To obtain a significant reduction we must also consider the protocol activities.

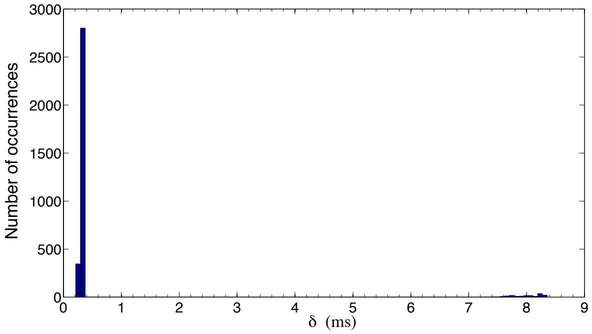

Figure 3. Delay distribution including advertisement packets.

Figure 4 illustrates the delay distribution obtained by avoiding issuing requests before the transmission of an advertisement packet. In this way, the maximum delay stays below 0.35 ms. However, such a solution does not allow providing real-time slots in a periodic fashion with period independent of the BLE traffic and would requires more advanced analysis techniques based on bandwidth supply and request functions.

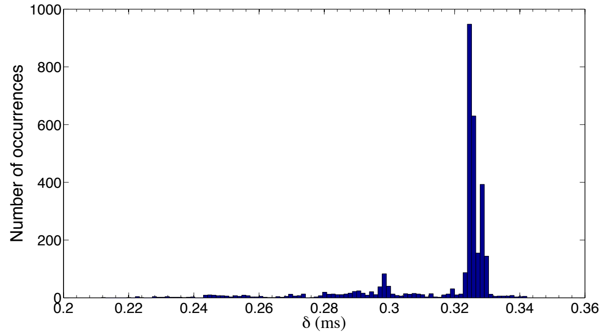

Figure 4. Delay distribution interleaving slots requests with BLE traffic.

These experiments show that requesting the real-time slots at a proper time and with a safe delay $\delta_{max}$ allows achieving a considerable performance improvement for the real-time application. For example, the application described in [1] generates a BLE traffic of 4 bytes packets at 50 Hz. Two different configurations have been used for testing the real-time reservation mechanism: the first one assigns a budget $Q = 128ms$ every period $P = 160ms$, while the second one sets a budget $Q = 50ms$ every period $P = 66ms$. In both configurations no packet loss has been observed.

## VI. Conclusions

This paper presented a bandwidth reservation mechanism for partitioning the radio transceiver between two protocols: the Bluetooth LE and a real-time custom protocol. The advantage of this approach is to enable real-time communication among nodes in a body area network, while preserving a standard connectivity with mobile devices, as tablets and smart phones. The real-time bandwidth partition can be used for internode communication subject to timing constraints and for node synchronization, necessary to trigger the slots allocated to the two protocols and reduce the timing error associated with the data acquired from different nodes. The preliminary experimental results show the effectiveness of the proposed approach, which has been implemented on a body area network of inertial nodes [1] for reconstructing limb movements during telerehabilitation exercises.

## References

[1] P. Buonocunto and M. Marinoni, “Tracking limbs motion using a wireless network of inertial measurement units,” in *Proc. of the 9th IEEE Int. Symp. on Industrial Embedded Systems (SIES 2014)*, Pisa, Italy, June 18-20, 2014.

[2] *BlueTooth Specification Version 4.2*, https://www.bluetooth.org/en-us/specification/adopted-specifications, 2014.

[3] *ANT Message Protocol and Usage*, https://www.thisisant.com/resources/Fant-message-protocol-and-usage.

[4] P. Desai, C. Kwan, A. Polo, S. Qiao, C. Tian, Y. Zhuang, G. Xie, L. Caliaperoumal, and B. Ibrahim, “Method and system for a dual-mode bluetooth low energy device,” May 27 2014, uS Patent 8,737,917.

[5] PRWEB, “Nordic semiconductor and ant wireless are developing nrf51 series multi-protocol combo solution for seamless and concurrent ant+ and bluetooth low energy connectivity to smartphones, tablets and pcs,” PRWEB, Tech. Rep., 2012. [Online]. Available: http://www.prweb.com/releases/2012/9/prweb9898604.htm

[6] E. Mackensen, M. Lai, and T. M. Wendt, “Performance analysis of an bluetooth low energy sensor system,” in *Proc. of the 1st IEEE International Symposium on Wireless Systems (IDAACS-SWS 2012)*, Offenburg, Germany, 2012, pp. 62–66.

[7] A. Kutlu, H. Ekiz, and E. Powner, “Performance analysis of mac protocols for wireless control area network,” in *Parallel Architectures, Algorithms, and Networks, 1996. Proceedings., Second International Symposium on*. IEEE, 1996, pp. 494–499.

[8] A. Willig, “Polling-based mac protocols for improving real-time performance in a wireless profibus,” *Industrial Electronics, IEEE Transactions on*, vol. 50, no. 4, pp. 806–817, 2003.

[9] T. L. Crenshaw, A. Tirumala, S. Hoke, and M. Caccamo, “A Robust Implicit Access Protocol for Real-Time Wireless Collaboration,” in *Proc. of the 17th IEEE Euromicro Conference on Real-Time Systems (ECRTS 2005)*, Palma de Mallorca, Spain, Jul. 2005.

[10] J. Wu, J.-C. Liu, and W. Zhao, “Utilization-Bound Based Schedulability Analysis of Weighted Round Robin Schedulers,” in *Proc. of 28th IEEE Real-Time Systems Symposium*, Tucson, AZ, USA, Dec. 2007.

[11] A. Koubaa, M. Alves, and E. Tovar, “Energy/Delay Trade-off of the GTS Allocation Mechanism in IEEE 802.15.4 for Wireless Sensor Network,” *International Journal of Communication Systems*, vol. 20, no. 7, pp. 791–808, July 2007.

[12] E. Toscano and L. Lo Bello, “Multiplechannel Superframe Scheduling for IEEE 802.15.4 Industrial Wireless Sensor Networks,” *IEEE Trans. on Industiral Informatics.*, vol. 8, no. 2, May 2012.

[13] A. C. W. Wong, M. Dawkins, G. Devita, N. Kasparidis, A. Katsiamis, O. King, F. Lauria, J. Schiff, and A. J. Burdett, “A 1v 5ma multimode ieee 802.15.6/bluetooth low-energy wban transceiver for biotelemetry applications,” *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, 2013.

[14] T. Jin, G. Noubir, and B. Sheng, “Wizi-cloud: Application-transparent dual zigbee-wifi radios for low power internet access,” in *Proc. of the INFOCOM, 2011*, Shangai, China, Apr. 2011, pp. 1593–1601.

[15] *NORDIC Semiconductor. nRF51822 Product Specification v3.1*, http://www.nordicsemi.com.

[16] E. Zehavi and R. Nevo, “Wireless apparatus having a transceiver equipped to support multiple wireless communication protocols,” 2006, US Patent 6,990,082.

[17] R. Bridgelall, “Dual mode wireless data communications,” 2005, US Patent 6,895,255.

[18] P. Kumarawadu, D. Dechene, M. Luccini, and A. Sauer, “Algorithms for node clustering in wireless sensor networks: A survey,” in *Proc. of the 4th Inter. Conference on Information and Automation for Sustainability (ICIAFS 2008)*, Sri Lanka, Dec. 2008.

[19] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, “Clock synchronization for wireless sensor networks: a survey,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 281–323, 2005.

[20] *Erika Enterprise RTOS kernel*, http://erika.tuxfamily.org/drupal/.