Computer Communications 89–90 (2016) 51–59

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

BlueVoice: Voice communications over Bluetooth Low Energy in the Internet of Things scenario

M. Gentili[a], R. Sannino[a], M. Petracca[b,c,*]

[a] STMicroelectronics, Agrate Brianza, Italy
[b] Scuola Superiore Sant'Anna di Pisa, Pisa, Italy
[c] Consorzio Nazionale Interuniversitario per le Telecomunicazioni, Pisa, Italy

ARTICLE INFO

Article history:
Available online 16 March 2016

Keywords:
Internet of Things
Bluetooth Low Energy
Speech services

ABSTRACT

Bluetooth Low Energy (Bluetooth LE) is a key technology in the envisioned Internet of Things (IoT) scenario. In fact, its extremely low-power characteristics make it one of the most suitable solutions to enable wireless communications among battery powered IoT objects ubiquitously deployed in the field with the aim of building smart environments. Although Bluetooth LE specification targets a specific set of applications mainly devoted to monitoring purposes, innovative solutions can lead to the adoption of such technology in different applications, such as multimedia streaming, allowing IoT objects to exploit new functionalities. In this direction this article presents BlueVoice, an application targeted to Bluetooth LE devices to enable speech streaming services.

In the article BlueVoice is presented by first detailing the services set extension needed to support the new envisioned multimedia service, then a description of application choices is given, followed by an evaluation of its performance in real IoT objects. Thanks to the selected speech encoding technique, connection design choices and packetization strategies, BlueVoice application requires a communication bandwidth of 64.3 kbps to transmit audio at 16 kHz in ADPCM format. BlueVoice performance has been evaluated in terms of power consumption, memory and processing requirements, showing feasibility of the developed solution in resource constrained devices, thus confirming the correct choices in the application design. The set of performance information obtained show that BlueVoice is a viable solution to enable speech communications in ubiquitous wireless IoT nodes based on the Bluetooth LE technology.

1. Introduction

In the last three decades Internet has experienced a rapid growth, penetrating almost every aspect of daily life to improve lifestyle of millions of connected people. The future Internet expansion will connect billions or trillions of uniquely identifiable wireless "Things," able to communicate among themselves to perform advanced and ubiquitous tasks while interacting with humans and with the surrounding environment. In such a vision, "Things" can be sensors, actuators, appliances, toys, and in general, any other virtual or physical entity capable of being identified. This new envisioned Internet evolution is referred as the Internet of Things (IoT).

In the IoT vision, all devices are connected in a global network through wireless interfaces by using standard protocol solutions (i.e., the Internet Protocol). To connect "Things" together, a plethora of radio technologies exists, though low-power communication solutions must be considered the most suitable when IoT objects are autonomous battery powered devices deployed in the field (e.g., wireless sensors for ubiquitous temperature monitoring applications). In this respect the IEEE 802.15.4 [1] and Bluetooth Low Energy (Bluetooth LE) [2] technologies can be considered two of the most effective solutions for the IoT, and their inclusion in the Internet world has been either completely defined, as in the case of the 802.15.4 [3], or it is in an early stage of definition as in the case of Bluetooth LE [4]. Nowadays, both solutions are mainly adopted in the IoT scenario for monitoring applications in which all devices are organized in a low-power Wireless Sensor Network (WSN). For instance in [5] and [6] 802.15.4 is used for climate and access monitoring, while in [7] and [8] Bluetooth LE is used for life parameter monitoring purposes. Along with classical monitoring services, new advanced applications, mainly based on 802.15.4,

* Corresponding author at: Scuola Superiore Sant'Anna di Pisa, Pisa, Italy. Tel.: +390505492280.
E-mail addresses: maurizio.gentili@st.com (M. Gentili), roberto.sannino@st.com (R. Sannino), matteo.petracca@sssup.it (M. Petracca).

http://dx.doi.org/10.1016/j.comcom.2016.03.004
0140-3664/

started to be investigated in the last few years. In [9] and [10], for instance, multimedia communications, both voice and video, have been proposed and analyzed. On the contrary, the use of Bluetooth LE for multimedia data transmission is still at an early stage, and the lack of available solutions mainly depends on the original set of applications thought for this technology (e.g., healthcare, fitness).

In the article we address such limitations by proposing BlueVoice, an application targeted to Bluetooth LE devices to enable voice streaming services. Starting from an overview of the Bluetooth LE technology, we first detail the defined service set extension necessary to support the new envisioned application. Then, the application design is presented before discussing the performance obtained in terms of power consumption, memory footprint, processing requirements, and communication delay measured in real hardware platforms.

The rest of this article is structured as follows: in Section 2 related works on voice communications over low-power wireless sensor networks based on both 802.15.4 and Bluetooth LE are presented. Section 3 details the Bluetooth LE working principle by describing the whole communication stack, and introducing the concept of profile. In Section 4, the BlueVoice application is presented by detailing its Bluetooth LE profile and design principles, while its implementation and achieved performance are presented in Section 5. Conclusions are drawn in Section 6.

2. Related works

Even if low-power wireless sensor networks have traditionally focused on applications such as environmental monitoring, in which sensor data are periodically sent at low data-rate (a few bytes every second or even longer), multimedia streaming applications over low-power links have gained much attention in recent years. In particular, there are several works focused on voice streaming application in the WSN scenario, addressing the constraints in terms of communication bandwidth, computational power and energy budget that severely affects the actual streaming capabilities of low-power wireless sensor devices. In [9], 42 embedded sensors nodes based on 802.15.4 communication standard were placed in a coal mine in order to address emergency situations. The system was designed to provide a 2-ways voice channel and, in order to meet voice streaming real-time requirements, a hardware-based global time synchronization mechanism was used to schedule voice packets. Moreover, in order to face the low data-rate of the wireless link, the analog signal generated by the embedded microphone was acquired at 4 kHz, converted in digital format and then encoded(through the onboard 8 bit microcontroller) with the ADPCM (Adaptive Differential Pulse Code Modulation) algorithm. In [11] the authors implemented an 802.15.4 network for high quality secure audio transmission in a point-to-point communication. In this system, a voice signal is acquired at 8 kHz, with 8 bit per sample, and encoded with a dedicated speech vocoder (resulting in an data rate of 8 kbps) while security is added via an 128 bit based advanced encryption system. In [12] a voice quality evaluation methodology is introduced and the performance of a pure 802.15.4 network for voice transmissions are investigated using a network simulator. Voice over Sensor Network (VoSN) based on 802.15.4 are discussed in [13], where methods to improve the quality of multiple full duplex voice communication are deeply described. Real environment performance evaluations are presented in [14,15], where several metrics of multi-hop communication such as throughput, jitter, latency and packet loss have been measured and analyzed, while in [16] a real system evaluation based on users experience is performed. A real-time analysis of a 802.15.4 wireless link performed on a real system, exploiting a novel compression mechanism is presented in [17]. Simulations and experimental results shown in these works have demonstrated that the achievable throughput is not enough for high quality audio streams, but it suffices for most common voice streaming applications exploiting speech compression algorithms. In fact, from the analysis of these works it clearly emerges that speech compression before the transmission over the wireless channel is a common solution to address the low data-rate available, even though a trade-off among compression ratio, speech quality and compression algorithm complexity is required. Speex codec was selected among other possible options for the Z-Phone project, presented in [18]. The implemented headset prototype (working at 8 kbps and 11 kbps) proves that the 802.15.4 radio channel has enough bandwidth to support a full-duplex conversation with such a narrow band voice codec. Speex is compared with another advanced compression algorithm, named Opus, in [19] where listening results show that Opus has high speech quality performance, while no comments are reported in terms of computational power requirements. A multistage coding algorithm is proposed in [20], while a Linear Predictive Coding (LPC) based approach is introduced and evaluated in simulation in [21]. Finally, a perception approach for data selection and voice protection, together with a cooperative transmission mechanism for multi-hop communication is presented in [22,23] and [24].

Although different solutions have been investigated in literature for low-power wireless networks based on 802.15.4 technology, only a few research activities focused on voice communications over a Bluetooth LE link have been proposed yet. In [25], IPv6 over BLE is proposed to enable a real-time acoustic data streaming service. A pc-based simulation demonstrates a maximum audio bitrate of about 16kbps due to the overhead introduced by this implementation approach. Research interest on BLE for audio streaming applications in low power contexts is confirmed by the work described in [26], where the introduced low latency audio codec is claimed to be suitable for a BLE audio streaming service. The BlueVoice application envisioned in this article is the first real hardware device with proven feasibility and performance measurements that enables voice services over Bluetooth LE wireless links.

3. Bluetooth LE overview

Bluetooth Low Energy, also known as Bluetooth Smart, was included in the Bluetooth Version 4.0 Core Specification in 2010. Even though the Bluetooth LE design keeps similarities with classic Bluetooth [27], it has been mainly developed for ultra low-power performance. Healthcare, fitness, smart homes are only few possible applications in which Bluetooth LE technology can be successfully used to connect battery powered devices.

The whole Bluetooth LE protocol stack, reported in Fig. 1, is divided in two main parts: the Controller and the Host. Applications make use of the services provided by protocols belonging to the Host layer of the stack. In particular, the Host part is composed of five layers: Logical Link Control and Adaptation Protocol (L2CAP), Attribute Protocol (ATT), Generic Attribute Profile (GATT), Security Manager Protocol (SM), and Generic Access Profile (GAP). The Controller part is instead divided into two layers: the Physical Layer (PHY) and the Link Layer (LL). The Host-Controller Interface (HCI) depicted in Fig. 1 enables communications between the Controller and the Host.

The *Physical Layer* is in charge of managing bits modulation to send and receive data through the wireless channel. In this respect, Bluetooth LE defines 40 channels, 3 dedicated to advertising and 37 for data communications, all channels are in the 2.4 GHz ISM (Industrial, Scientific, and Medical) band. In order to reduce interference, a Frequency Hopping Spread Spectrum (FHSS) solution is used. The maximum data rate at physical layer is 1 Mb/s, and the typical communication range is in the order of few tens of meters.

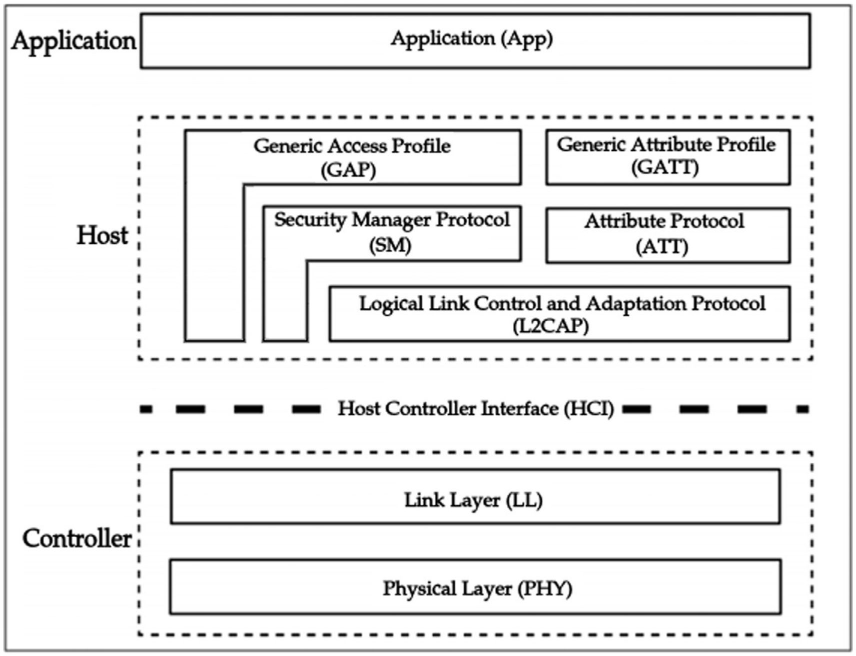

Fig. 1. Bluetooth LE protocol stack.

The *Link Layer* specifies the functionality for bidirectional communications between two devices. Two possible roles are defined for Bluetooth LE nodes: Master and Slave. Master nodes (e.g., laptop, smartphone) usually scan for other devices, while a Slave node sends data (e.g., body sensor devices) whenever it is necessary. A Slave is usually in sleep mode, and it periodically wakes up to be discovered by a Master. Along with communication functionality, the link layer provides error and flow control capabilities. Errors inside data packets are detected through a cyclic redundancy check code. The successful reception of a data packet is notified through an acknowledgment.

Above the link layer, the *Logical Link Control and Adaptation Protocol* provides two main functionalities. The main role of the protocol is to provide multiplexing capabilities encapsulating upper-layers multiple protocols data into the standard Bluetooth LE packet format. Moreover, the L2CAP layer may perform segmentation, retransmission and detection of duplicate packets according to the L2CAP mode in use.

The *Security Manager Protocol* and the *Generic Access Profile* provide security and management services respectively. In particular, such protocol stack components specify how security keys are generated and exchanged for a secure communication between peers (SM), as well as how devices interact with each other at lower level (GAP).

When developing new applications, two essential components are the *Attribute Protocol* and the *Generic Attribute Profile*. The ATT protocol is a stateless client/server protocol: without considering the device role at lower level, Master or Slave, each device can be a server, a client or both. The client requests data from a server, and a server sends data to a client. The data are stored into the server as *attributes*; each attribute contains data managed by the GATT, and it is identified by an Universal Unique Identifier (UUID). The ATT protocol creates a communication between a server attribute and a client by using a dedicated L2CAP channel. The GATT adds a data abstraction model on top of ATT, it is in charge of discovering the data stored in the ATT protocol in order to exchange *characteristics* between devices. Each Bluetooth LE device contains a set of possible attributes (storing services) and characteristics (properties associated with the storing services). Once a new application is built on top of the Bluetooth LE protocol stack both attributes and characteristics must be specified. The whole set of characteristics, attributes and low-level specifications for a certain application is referred to as a *profile*. Standard profiles guarantee interoperability among devices of different vendors.

4. BlueVoice application

In this section, the BlueVoice application is presented starting from the definition of a voice communication Bluetooth LE profile, and then the application design is described by considering the communication roles of the involved devices, audio processing and compression choices, packetization issues and bandwidth requirements.

4.1. Service definition

Since the audio streaming use case is not part of the set of profiles specified by the standard, in order to enable voice streaming services, the BlueVoice application defines a so-called “vendor specific profile”, named BlueVoice Service (BVS), on top of the Bluetooth LE protocol stack, thus specifying the way voice data are exchanged between server and client devices.

As previously described, ATT protocol is used by GATT as transport protocol to exchange data between devices: the smallest entities defined by ATT, the *attributes*, are addressable pieces of information (uniquely identified by a UUID) that may contain user data or meta-information regarding the architecture of the attributes themselves such as permissions, encryption, and authorization properties. GATT server attributes, instead, are organized as a sequence of services, each one starting with a service declaration attribute including one, or more, *characteristic* with possible descriptors. Each characteristic is an exposed attribute. Besides standard profile UUIDs, proprietary and vendor-specific UUIDs can be used in custom implementations to develop new services with their own characteristics, as in the case of the BlueVoice application. A BVS profile exported by a server node exposes to clients how data are organized in terms of type and format, as well as

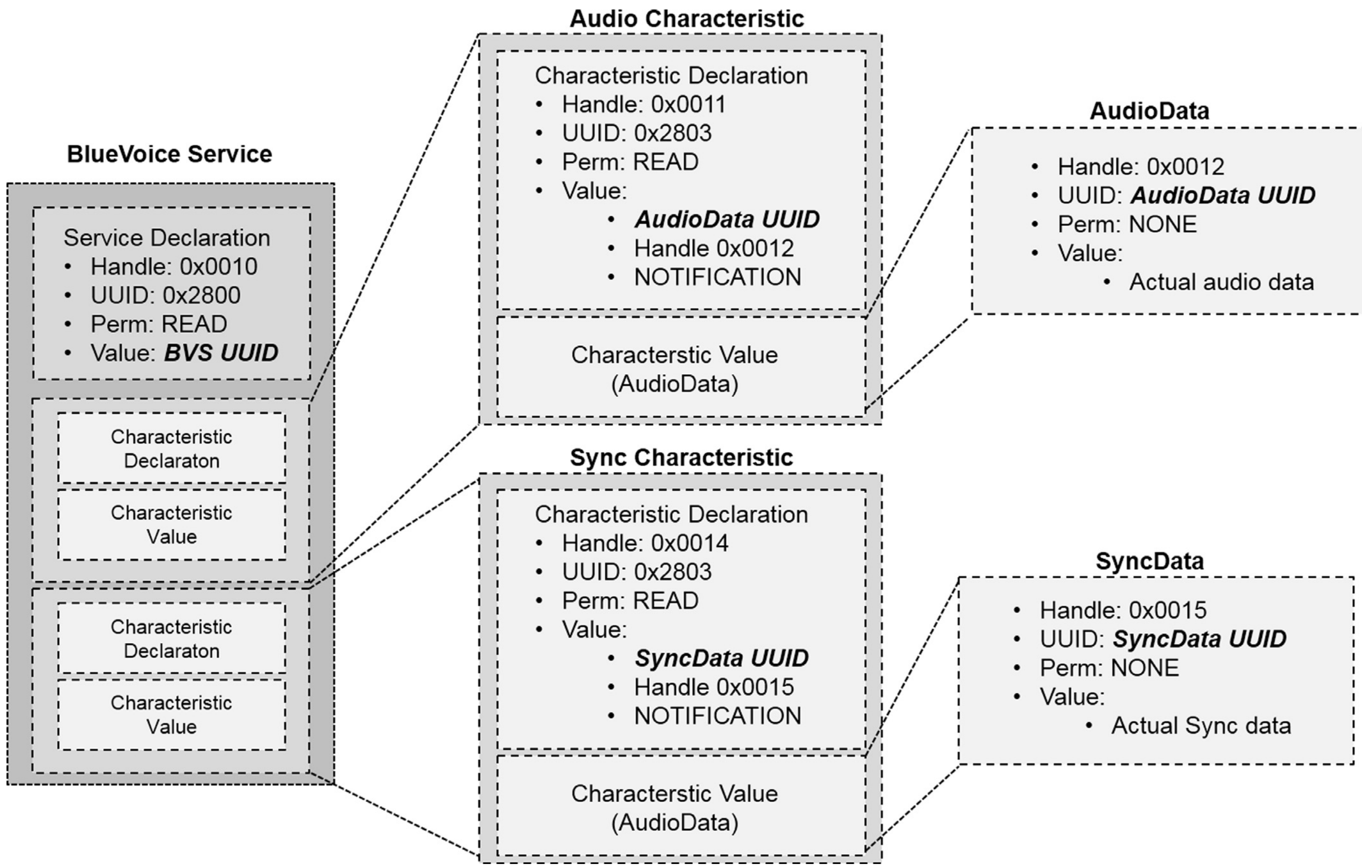

Fig. 2. BlueVoice service (BVS) definition.

how they can be accessed. The BVS service is defined with the following attributes, shown in Fig. 2 for clarity:

- Service declaration (Handle 0x0010)
  - UUID: standard 16-bit UUID for a primary service declaration (0x2800).
  - Permission: read.
  - Value: proprietary 128-bit *BVS UUID*.
- Characteristic declaration (Handle 0x0011)
  - UUID: standard 16-bit UUID for a characteristic declaration (0x2803).
  - Permission: read.
  - Value: proprietary 128-bit *Audio UUID*, notification only, Handle: 0x012.
- Characteristic data (Handle 0x0012)
  - UUID: proprietary 128-bit *Audio UUID*.
  - Permission: none.
  - Value: actual audio data.
- Characteristic declaration (Handle 0x0014)
  - UUID: standard 16-bit UUID for a characteristic declaration (0x2803).
  - Permission: read.
  - Value: proprietary 128-bit *Sync UUID*, notification only, Handle: 0x0015.
- Characteristic data (Handle 0x0015)
  - UUID: proprietary 128-bit *Sync UUID*.
  - Permission: none.
  - Value: actual synchronization data.

According to the standard, the primary *Service Declaration* is the first attribute of the service, and its value field contains the definition of the UUID that the declaration introduces. For the BlueVoice application, a 128-bit proprietary UUID (BVS UUID) is declared. BVS includes two characteristics, named *Audio and Sync Characteristics*. In the Bluetooth LE specification, each characteristic is composed of at least two attributes: the characteristic declaration defining its properties in form of metadata, and the characteristic value containing actual characteristic data. In the case of BlueVoice, both Audio and Sync Characteristics include a single attribute defined by a 128-bit proprietary UUID (AudioData and SyncData UUID) containing actual audio data and side information synchronization values respectively. Audio and Sync Characteristics declarations define *AudioData and SyncData* attributes as ”notification only” with no read and write permission from the client, meaning that audio and sync data are only exchanged in form of notifications, thus without response, from Server to Client. Consistently with the hierarchical architecture of Bluetooth LE services, other characteristics may be added in future releases of the BlueVoice application as part of the BVS, such as the possibility for a client to configure certain parameters on the server side (i.e., volume, speech compression standard, enabling/disabling voice processing).

4.2. Application design

The BlueVoice application design is detailed in this section in terms of (i) Bluetooth LE communication choices, and (ii) audio processing.

(1) Bluetooth LE communication.

According to the Bluetooth LE specification, communications can be either broadcast or connection-based. The BlueVoice application relies, at the LL level, on a connection-based communication paradigm, providing a permanent point-to-point link between two devices, one transmitter (Tx module) and one receiver (Rx module). Connections involve two separate roles: Central and Peripheral. The Central (Master) device generally supports complex

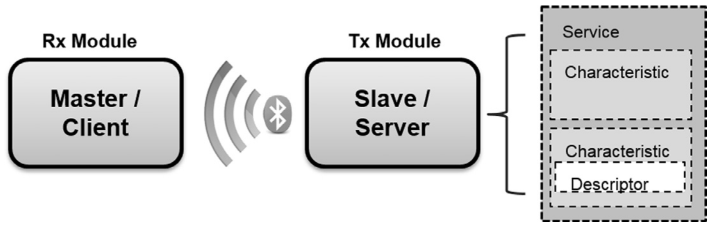

Fig. 3. BlueVoice profile role assignment.

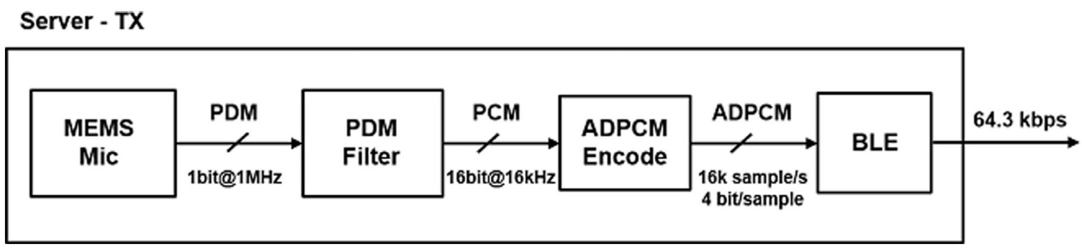

Fig. 4. BlueVoice transmission chain.

functions with respect to Peripheral (Slave) devices. The Central device is the initiator of the connection, handles adaptive frequency hopping, sets encryption and manages communication timing, defining how data are exchanged between the two devices. Considering a scenario in which the BlueVoice application runs on autonomous battery powered wireless sensor device equipped with both microphone and scalar sensors (i.e., the typical IoT scenario for ubiquitous monitoring applications), the Slave role is assigned to the Tx module, and the Master role to the Rx module. Such role assignment is consistent with the asymmetric design concept of Bluetooth LE where to the device having greater energy constraints has fewer operations to perform: a portable device equipped with a small battery is typically a Slave.

On top of the LL, the GATT layer defines Client and Server roles of interacting devices that are independent from Master and Slave roles described above. A Server is the device which has information, while a Client is the device asking or receiving information updates. In the BlueVoice application, the Server role is therefore assigned to the Peripheral-Slave Tx module with the microphones, while the Client role is assigned to the Central-Master Rx module. Voice data streaming is based on periodic Server to Client notifications which do not require a request or response from the receiving device. At power up the Slave-Server Tx module goes into advertising mode and starts sending advertising packets at low frequency. Vice versa, the Master-Client Rx module goes in discovery mode, looking for other devices, and as soon as it detects the presence of a Slave device, by receiving an advertisement packet, it sends a connection request. After the connection establishment phase, server updates are sent to the Client as asynchronous notification packets containing audio data. Fig. 3 shows BlueVoice role assignment at the GATT layer.

(2) Audio processing.

Audio processing in the BlueVoice application is designed to achieve a target audio sampling frequency of 16 kHz at the receiver side. The audio signal transmitted over the Bluetooth LE link is compressed via an Adaptive Differential Pulse Code Modulation algorithm in order to fit into the available data rate, and at the same time the compression minimizes radio transmission time as well as power consumption. A fully digital solution has been designed by means of digital MEMS microphones, which are suitable for IoT wireless sensor devices thanks to their interesting features in terms of dimensions and audio quality. Fig. 4 shows the whole speech processing chain: the 1 bit Pulse Density Modulation (PDM) signal at 1 MHz generated by the digital MEMS microphone is acquired and converted to 16 bit Pulse Code Modulation (PCM) samples at 16 kHz, which are in turn compressed in 4 bit ADPCM samples, at a frequency of 16 ksamples/s, ready to be transmitted. In addition, a set of side information synchronization data is sent at a lower frequency. The resulting bandwidth is equal to 64 kbps of audio data plus 300 bps of sync information, for a total of 64.3 kbps. In the following, the blocks introduced above are described.

The analog signal generated by the MEMS microphone capacitive sensing element is amplified, sampled at high rate and then processed by the built-in sigma-delta modulator, which combines the operations of quantization and noise shaping to give in output a single bit at a high sampling rate in PDM format. PCM conversion is chosen as an intermediate step between PDM and compressed audio data that will be afterwards sent over the wireless channel. In order to convert the PDM stream into PCM data, a decimation filter, followed by two individually configurable filters (low pass and high pass), is used. The output of this process block is a stream of 16 bit samples in the PCM format.

The ADPCM encoding block compresses PCM samples in order to both save transmission bandwidth and reduce energy consumption because of a lower number of transmitted packets. As previously stated, ADPCM is a compression algorithm for lossy waveform coding which predicts the current signal value from previous values and transmits only the difference between the real and the predicted value quantized by using an adaptive quantization step. The ADPCM compression has been selected, among other possible compression standards, because it is based on a waveform coding approach, thus being more suitable to be used in sensor network devices (usually based on microcontrollers), with respect to more highly complex solutions based on a vocoder approach. In the BlueVoice application, each 16 bit PCM sample is encoded into 4 bit ADPCM data so that the required application transmission bandwidth is 64 kbps, which is compatible with Bluetooth LE streaming capabilities.

As previously reported, the whole bandwidth required by the BlueVoice application is 64.3 kbps because BlueVoice improves the communication robustness by adding additional information when data are sent through the channel. Voice data at 64 kbps is sent in form of 4 packets of 20 bytes each every 10 ms (the selected connection interval) , thus using for each data packet the maximum available payload. Transmitter side information is sent at a lower frequency in the form of an additional packet of 6 bytes every 16

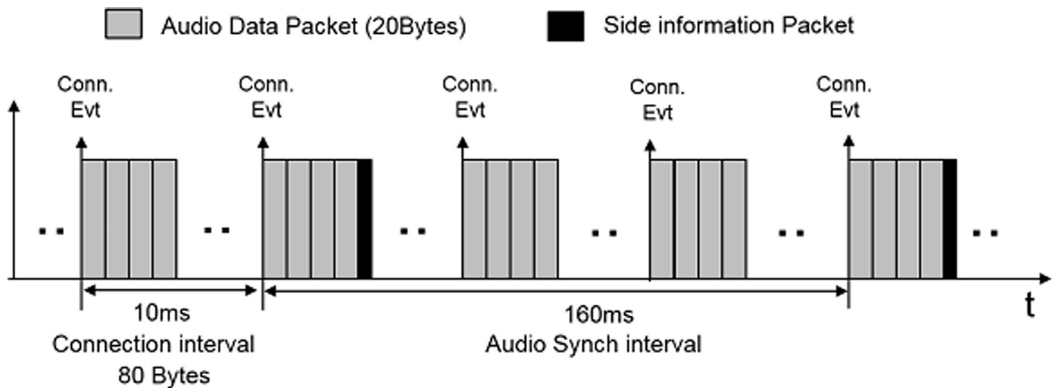

Fig. 5. BlueVoice packetization schema.

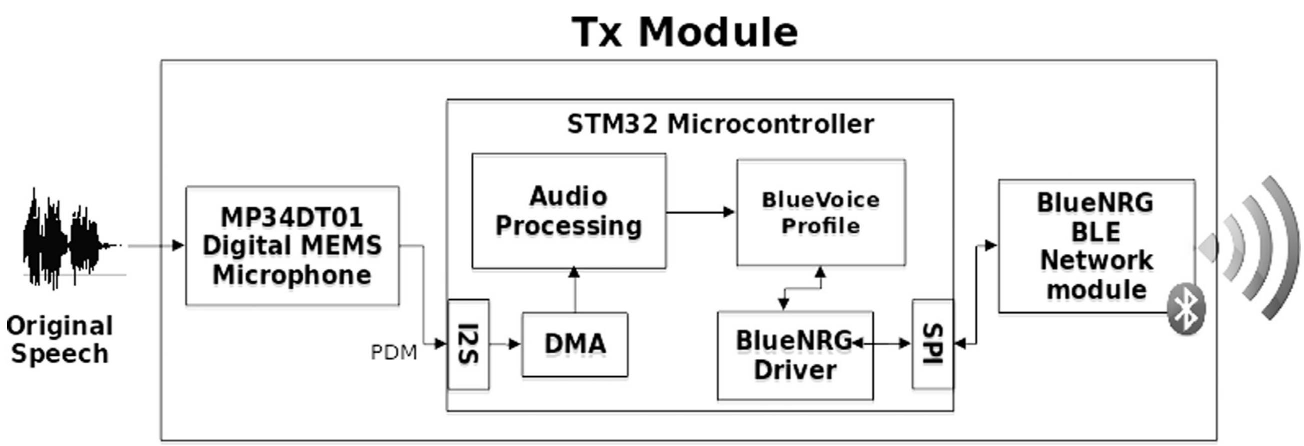

Fig. 6. Tx module architecture.

connection intervals. The whole data packetization policy on top of the Bluetooth LE protocol stack is summarized in Fig. 5: 4 voice data packets of 20 bytes are sent at each connection interval of 10 ms via Audio characteristics, while transmitter side information is sent in an additional packet every 160 ms via Sync characteristic.

5. BlueVoice implementation and performance

In this section, the BlueVoice implementation in real hardware devices is presented before discussing measured performance in terms of power consumption, memory footprint, processing requirements, and communication delay.

5.1. Implementation in real hardware devices

To evaluate the BlueVoice feasibility in real low-complexity wireless sensor network platforms supporting Bluetooth LE communications, the whole application, as described in Section 4.2, has been implemented in real hardware devices. The selected platform is the STMicrolectronics STM32 Nucleo F401 board [28], an open development platform based on STM32F01, a 168 MHz, 32-bit ARM Cortex-M4 microcontroller. Despite the fact that the system embeds a very powerful microcontroller with respect to the usual platforms used in wireless sensor network applications, the STM32 Nucleo F401 has been chosen because of its flexibility and versatility. In fact, the board is provided with a set of connectivity supports and expansions headers that make it easy to expand its functionality with a set of specialized expansion boards, thus allowing exploration, prototyping and validation of new ideas in the IoT scenario. BlueVoice Tx and Rx modules, in their common hardware configuration, are based on a STM32Nucleo expanded with a Bluetooth LE connectivity board based on STMicroelectronics BlueNRG [29], a very low power Bluetooth LE single-mode network processor compliant with Bluetooth specification v4.0. BlueNRG can be configured in Master and Slave mode, and it has a maximum current in transmission of 8.2 mA that can be reduced to 1.7 uA when the Bluetooth LE stack is active. In the Tx device, speech is acquired by means of an additional microphone expansion board, based on STMicroelectronics MP34DT01 [30], a digital omnidirectional MEMS microphone with an acoustic overload point of 120 dBSPL, 63 dB signal-to-noise ratio and −26 dBFS sensitivity. The MP34DT01 is built with a capacitive sensing element and an integrated circuit that embeds a sigma-delta modulator and a noise shaping mechanism, providing a PDM output at a frequency that goes from 1 to 3.25 MHz.

Fig. 6 shows the Tx module architecture schema: the STM32 microcontroller is configured to acquire PDM samples from the microphone via an Inter-IC Sound (I2S) bus connected to a DMA peripheral, while communication with the BlueNRG component is performed via a Serial Peripheral Interface (SPI) by means of a set of specific Application Programming Interfaces (APIs). The Rx module architecture is symmetric, except for audio acquisition, and includes a USB Audio interface to provide reconstructed audio to a PC. Fig. 7 shows the actual prototypes.

5.2. Performance

The real system described in Section 5.1 has been used as a laboratory testbed to evaluate the performance of the BlueVoice application in terms of power consumption, memory footprint, processing requirements and communication delay. In particular,

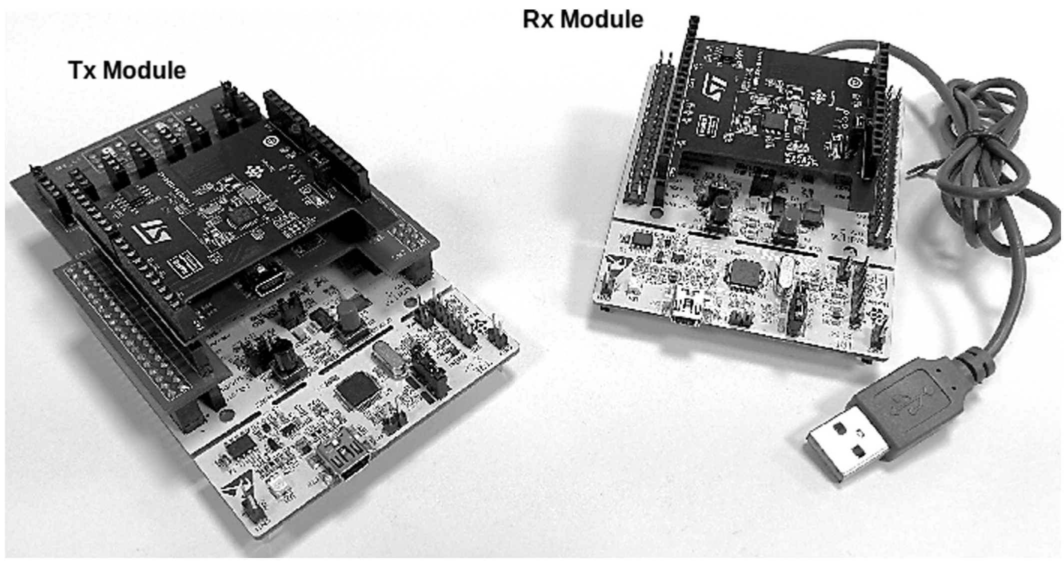

Fig. 7. Tx and Rx prototypes.

Table 1
BlueVoice power consumption of transmitter side (Slave/Server).

Tx module (Slave/Server) power consumption	
State	Power consumption (mW)
Advertising	0.297
Connection	1.855
Streaming	9.570

considering a scenario with a set of tiny wireless microphones modules deployed in the field, and considering the intrinsic asymmetry of Bluetooth LE (where a Slave device has to be designed to be as compact and low power as possible), the performance evaluation reported in this section is focused on power consumption, memory footprint and processing requirements of the Tx (Slave) module, responsible for speech acquisition and streaming. In addition, another reported performance indicator is the communication delay measured at the Rx side: in fact, such a parameter is a key indicator of perceived speech quality in voice communications. According to the ITU G.107 standard [31], the communication delay (composed of encoding, transmission, and decoding delays) is the most important contribution in evaluating perceived voice quality at the receiver side [32]. Lower communication delays results in higher speech quality values in terms of Mean Opinion Score (MOS) values, a common speech quality measure [33].

(1) Power consumption, memory and processing requirements.

As described in the previous section, the BlueVoice application has been implemented in a real hardware device composed of a STMicrolectronics STM32 Nucleo board, acting as host device, and a Bluetooth LE network module acting as controller device. Since radio transmission in wireless sensor nodes is the most demanding aspect in terms of energy consumption [34], we focused the power consumption evaluation on the network module only, disregarding the power consumed by the microcontroller. In fact, it must be stressed that the microcontroller power consumption strictly depends on hardware characteristics and possible low-power configurations, thus being an additive platform-dependent value in the overall computation of consumed power. In Table 1 power consumption values for the communication module are reported for the three main states in which BlueVoice can operate: advertising, connection and streaming. Power consumption in the Bluetooth LE scanning phase is not reported because it is related to the Master module, which is the less critical component of the BlueVoice application in terms of power consumption and which can be theoretically replaced with any other module running the BlueVoice profile. All values reported in Table 1 have been obtained as the average of the consumption values given by a precision multimeter directly measuring the current going into the network module subsystem operating at 3.3 V. Measured values mainly depend on the particular connection parameters and communication bandwidth, and are in line with simulation results given by the current consumption simulator software provided by STMicroelectronics for the BlueNRG component.

According to the Bluetooth LE standard, before a connection is established between two nodes the Slave device is in Advertising mode, while the Master device is in Scanning mode. As soon as the Master detects the presence of a Slave via an advertising packet, it establishes the connection. In the BlueVoice solution, the Slave is the Tx node, acting as server, while the Master is the Rx node, the client, and periodic notifications are sent from Slave to Master. During the Advertising phase, the average power consumed by the Tx communication module is a very low 0.297 mW, while power consumption is equal to 1.855 mW when the connection is established. It is important to note that the power consumption in the connection phase is strongly related to the connection interval design choice: in the BlueVoice solution, it is set to a value close to the minimum value allowed by the standard (10 ms), so as to have a minimum latency in the transmission. Once the connection is established, the BlueVoice application goes into a Streaming state in which the average power consumption is 9.570 mW. Therefore, considering a battery with a capacity of 200 mAh, the ideal lifetime of an IoT node composed only of the RF part in continuous streaming mode would be approximately 69 h. Such power consumption value demonstrates the suitability of BlueVoice approach for deployment in IoT context, thus highlighting the correct choices in application design.

In addition to the power consumption analysis, the feasibility of BlueVoice has been evaluated by considering its requirements in terms of memory footprint and processing. The BlueVoice application in its actual implementation requires, as reported in Table 2, 22.66 kB of Flash and 7.52 kB of RAM memory, two values which are highly compatible with resource-constrained systems.

Table 2
BlueVoice memory footprint at the transmitter side

Memory footprint	
Memory	Occupation (kB)
Flash	22.66
RAM	7.52

Table 3
BlueVoice encoding phase processing requirements

Processing requirements	
Function	MIPS
PDM to PCM conversion	5.04
ADPCM compression	1.60
Encoding phase	6.64

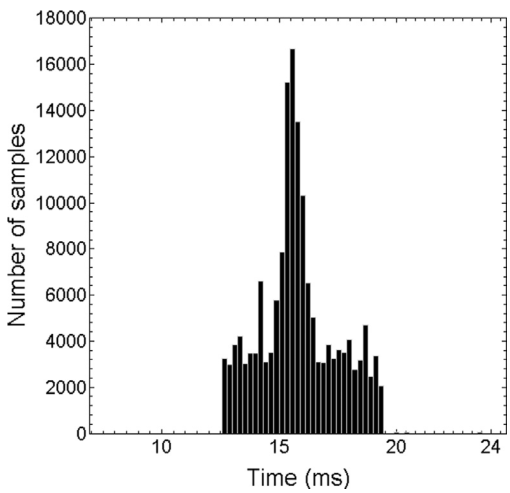

Fig. 8. Communication delay distribution.

The measured BlueVoice processing requirements are reported in Table 3 in terms of million instructions per second (MIPS) by considering both PDM to PCM conversion and ADPCM compression. The overall encoding process requires 6.64 MIPS which, together with the reduced requirements in terms of memory footprint, make BlueVoice a suitable solution to enable speech communications in IoT nodes deployed in the field to fulfill monitoring tasks.

(2) Communication delay.

The BlueVoice feasibility in terms of power consumption, processing requirements and memory footprint does not guarantee to have a speech signal with a sufficient perceived quality at the receiver side. According to the E-model reported in [31] a key role in determining the speech quality is played by the whole communication delay, evaluated by taking into account encoding, transmission and encoding delays. In order to evaluate the performance of the BlueVoice solution, an extensive transmission test has been performed measuring the communication delay at the receiver side for a high number of audio samples. The resulting distribution is reported in Fig. 8. It is important to note that in order to focus the performance evaluation on communication delay only, reported results have been obtained running the test in ideal conditions in terms of Packet Loss Rate (PLR). In fact, since PLR is an indicator depending on several factors which were out of scope of this study (e.g., communication distance, antenna design, sources of interference), test conditions were imposed to have PLR values equal to zero.

The average communication delay is equal to 15.80 ms with a standard deviation of 1.64 ms, while in the worst case, the delay is lower than 29.00 ms. From final results reported in [32], 185.80 ms is the maximum communication delay allowed while still reaching a very satisfactory speech quality in case of a PCM encoding and no data losses. Although the BlueVoice application uses ADPCM speech coding technique, requiring a higher number of instructions with respect to the PCM encoding, in the worst case the communication delay is significantly lower than 185.80 ms. In particular, the maximum measured delay is less than 16% of the maximum allowed communication delay reported in [32], showing that the BlueVoice application is able to reach very high speech quality values at the receiver side.

6. Conclusions

In this article an innovative solution for streaming speech over Bluetooth Low Energy is presented. The so-called BlueVoice application demonstrates the feasibility of multimedia communication in constrained sensor devices acting as IoT nodes. In the article, a vendor specific Bluetooth LE profile for speech communication is defined, and the BlueVoice design is presented by considering communication roles of involved devices, audio processing and compression choices, packetization issues and bandwidth requirements. The BlueVoice application is composed of a Tx node and a Rx node, acting as Bluetooth LE Slave/Server and Master/Client respectively. Periodical notifications are sent from the Server to the Client after the connection is established. On the Tx node, MEMS microphone digital output in PDM format is acquired, converted to PCM and then compressed to ADPCM, resulting in a communication bandwidth of 64 kbps on the Bluetooth LE link. The profile also defines a side-information mechanism, requiring an additional bandwidth of 300 bps, for a total bandwidth of 64.3 kbps required by the application. In order to evaluate the performance of the proposed solution, the BlueVoice application has been implemented in real hardware devices. Particular focus has been devoted to the Tx node, which has been implemented as a fully digital system composed of a MEMS microphone, a microcontroller acting as host and a network module acting as Bluetooth LE controller. Performance evaluation reported in the article shows the feasibility of the solution for the IoT context in terms of power consumption, processing requirements and memory footprint. In particular, a power consumption of only 9.570 mW has been measured for the network module during audio streaming, while completely acceptable memory and processing requirements have been experienced. Moreover, the communication delay has been measured as a key performance indicator. Such a figure of merit results in an average of 15.80 ms and a maximum of 29.00 ms, showing that the BlueVoice application is able to reach very high speech quality values at the receiver side and therefore confirming it as a suitable solution for advanced applications in the IoT scenario.

References

[1] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPAN), The Institute of Electrical and Electronics Engineers, 2003.
[2] Specification of the Bluetooth System, Covered Core Package Version: 4.0, 2010.
[3] N. Kushalnagar, G. Montenegro, C. Schumacher, IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals, 2007(RFC 4919).
[4] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, C. Gomez, Transmission of IPv6 Packets over BLUETOOTH(R) Low Energy, 2015(Internet-Draft).
[5] K. Martinez, R. Ong, J. Hart, Glacsweb: a sensor network for hostile environments, in: IEEE Sensor and Ad Hoc Communications and Networks Conference, 2004, pp. 81–87.

[6] M. Petracca, S. Bocchino, A. Azzarà, R. Pelliccia, M. Ghibaudi, P. Pagano, WSN and RFID integration in the IoT scenario: an advanced safety system for industrial plants, J. Commun. Softw. Syst. 9 (1) (2013) 104–113.
[7] L. Guo-cheng, Y. Hong-yang, Design and implementation of a Bluetooth 4.0-based heart rate monitor system on ios platform, in: International Conference on Communications, Circuits and Systems, 2, 2013, pp. 112–115.
[8] P. Young-Jin, C. Hui-Sup, Transmission of ECG data with the patch-type ECG sensor system using Bluetooth low energy, in: International Conference on ICT Convergence, 2013, pp. 289–294.
[9] R. Mangharam, A. Rowe, R. Rajkumar, R. Suzuki, Voice over sensor networks, in: IEEE Real-Time Systems Symposium, 2006, pp. 291–302.
[10] C. Salvadori, M. Petracca, S. Madeo, S. Bocchino, P. Pagano, Video streaming applications in wireless camera networks: a change detection based approach targeted to 6LoWPAN, J. Syst. Archit. 59 (10) (2013) 859–869.
[11] L.Y. Hua, F.F. Teng, Delivering high quality, secure speech communication through low data rate 802.15.4 WPAN, in: IEEE International Conference on Telecommunications and Malaysia International Conference on Communications, 2007, pp. 758–763.
[12] W. Chonggang, K. Sohraby, R. Jana, J. Lusheng, M. Daneshmand, Voice communications over zigbee networks, IEEE Commun. Mag. 46 (1) (2008) 121–127.
[13] H. Lee, M.J.S. Cho, The method for improving sound quality of IEEE 802.15.4 PHY based voice communication with TDMA/TDD MAC protocol, in: Information and Communication Technologies, 2013, pp. 37–40.
[14] D. Brunelli, M. Maggiorotti, L. Benini, F. Bellifemine, Analysis of audio streaming capability of zigbee networks, in: Wireless Sensor Networks, in: Lecture Notes in Computer Science, 4913, Springer, Berlin, Heidelberg, 2008, pp. 189–204.
[15] D. Brunelli, L. Teodorani, Improving audio streaming over multi-hop zigbee networks, in: IEEE Symposium on Computers and Communications, 2008, pp. 31–36.
[16] M. Liu, Y. Wang, Z. Fan, S. Zhang, Voice communication based on zigbee wireless sensor networks, in: Chinese Control Conference, 2014, pp. 389–394.
[17] T. Facchinetti, M. Ghibaudi, E. Goldoni, A. Savioli, Real-time voice streaming over IEEE 802.15.4, in: IEEE Symposium on Computers and Communications, 2010, pp. 985–990.
[18] A. Meliones, E. Touloupis, J. Perello, A. Serrat, Z-phone: design and implementation of embedded voice over zigbee applications, in: IEEE Symposium on Computers and Communication, 2014, pp. 1–7.
[19] K. Raju, A. Sharma, Comparison of two speech communication codecs for transmitting voice/speech over zigbee, in: International Conference on Signal Processing and Integrated Networks (SPIN), 2015, pp. 685–690.
[20] Z. Chen, S. Kang, E. Choi, J. Huh, Multistage waveform coding for voice communication over zigbee networks, in: International Conference on Consumer Electronics, 2009, pp. 1–2.
[21] C. Scavongelli, R. Bonfigli, D. Donati, E. Montali, M. Conti, Lpc real-time speech transmission over 802.15.4, in: Workshop on Intelligent Solutions in Embedded Systems, 2013, pp. 1–6.
[22] M. Petracca, J. De Martin, G. Litovsky, M. Tacca, A. Fumagalli, Low-complexity perceptual packet marking for speech transmission over tiny mote device, in: IEEE International Conference on Multimedia and Expo, 2009a, pp. 806–809.
[23] M. Petracca, G. Litovsky, A. Rinotti, M. Tacca, J. De Martin, A. Fumagalli, Perceptual based voice multi-hop transmission over wireless sensor networks, in: IEEE Symposium on Computers and Communications, 2009b, pp. 19–24.
[24] M. Petracca, M. Ghibaudi, R. Pelliccia, P. Pagano, Perceptual voice communications in IEEE802.15.4 networks for the emergency management support, in: Baltic Congress on Future Internet Communications, 2012, pp. 121–126.
[25] J. Yim, S. Kim, N.K. Kim, Y.B. Ko, Ipv6 based real-time acoustic data streaming service over Bluetooth low energy, in: IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), 2015, pp. 269–273.
[26] H. Moon, N. Lee, H. Kim, S. Lee, Low latency audio coder design for Bluetooth and Bluetooth low energy, in: IEEE International Conference on Consumer Electronics (ICCE), 2015, pp. 138–141.
[27] C. Bisdikian, An overview of the Bluetooth wireless technology, IEEE Commun. Mag. 39 (12) (2001) 86–94.
[28] STMicroelectronics STM32 MCU Nucleo, www.st.com/stm32nucleo.
[29] STMicroelectronics BlueNRG, Bluetooth® Low Energy Wireless Network Processor, www.st.com/web/catalog/sense_power/FM1968/CL1976/SC1898/PF258646.
[30] STMicroelectronics MP34DT01, MEMS Audio Sensor Omnidirectional Stereo Digital Microphone, www.st.com/web/en/catalog/sense_power/FM125/SC1564/PF250940.
[31] International Communication Union, G.107: The E-model: A Computational Model for Use in Transmission Planning, 2014.
[32] K. Perlicki, Simple analysis of the impact of packet loss and delay on voice transmission quality, J. Telecommun. Inf. Technol. 2 (2002) 53–56.
[33] International Communication Union, P.800.1: Mean Opinion Score (MOS) Terminology, 2006.
[34] V. Raghunathan, C. Schurgers, S. Park, M. Srivastava, Energy-aware wireless microsensor networks, IEEE Signal Process. Mag. 19 (2) (2002) 40–50.