Half-duplex Voice Communications over Bluetooth Low Energy

Maurizio Gentili, Roberto Sannino, and Matteo Petracca

***Abstract*—Bluetooth Low Energy is widely considered an enabling 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. Although Bluetooth Low Energy specifications are nowadays targeted to a specific set of applications, innovative solutions can lead to the adoption of such technology in different applications, such as multimedia streaming. In this direction this article presents BlueVoice, a speech streaming application for Bluetooth Low Energy devices.**

In the article BlueVoice is presented by first detailing the services set extension needed to support the voice streaming service, then the application is described by evaluating its performance in real devices. Thanks to the application design choices the application fully support voice streaming services while avoiding energy wasting.

I. Introduction

In the last decades Internet has experienced a rapid growth, penetrating in almost every aspect of daily life. The future Internet expansion will connect billions or trillions of uniquely identifiable wireless ”Things” able to intercat with humans and with the surrounding environment to perform advanced tasks. 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” among them 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. In this respect the Bluetooth Low Energy (Bluetooth LE) [1] technology can be considered one of the most effective solutions for the IoT, and its inclusion in the Internet world is in progress [2].

Nowadays Bluetooth LE solutions are mainly adopted in the IoT scenario for life parameter monitoring purposes. Along with classical monitoring services, new advanced applications, mainly based on other technoogies, started to be investigated in the last few years. In [3], for instance, voice communications have been proposed and analysed in IEEE802.15.4-based networks [4]. In this respect 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 envisioned for this technology (e.g., healthcare, fitness). In the article we address such limitation by describing the BlueVoice application, able to support voice streaming services over Bluetooth LE devices. 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 to show real performance in the STMicrolectronics STM32 Nucleo L476 board.

The rest of this article is structured as follows. Section II details the Bluetooth LE working principle by describing the whole communication stack, and introducing the concept of profile. In Section III the BlueVoice application is presented by detailing its Bluetooth LE profile, design principles, implementation, and achieved performance. Conclusions are drawn in Section IV.

II. Bluetooth LE overview

Bluetooth Low Energy has been included into the Bluetooth Version 4.0 Core Specification in 2010. Even though the Bluetooth LE design keeps similarities with classic Bluetooth, 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 Figure 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

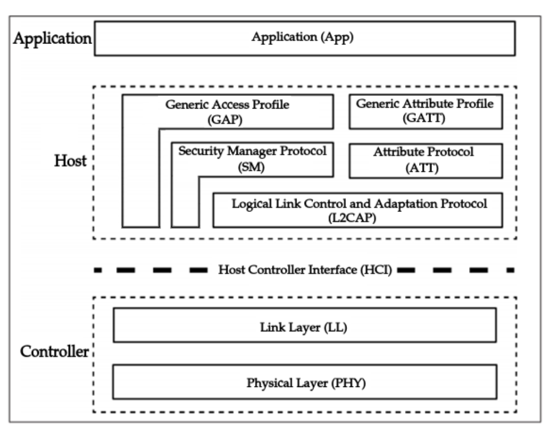

Fig. 1. Bluetooth LE protocol stack

M. Gentili and R. Sannino are with AST Audio/Sensors Platforms R&D and Audio SW Ecosystem, STMicroelectronics, Agrate Brianza, Italy (e-mail: [maurizio.gentili|roberto.sannino]@st.com).

M. Petracca is with Scuola Superiore Sant'Anna di Pisa and National Inter-University Consortium for Telecommunications, Pisa, Italy (e-mail: matteo.petracca@sssup.it) .

(SM), and Generic Access Profile (GAP). The Controller part is instead divided in two layers: the Physical Layer (PHY) and the Link Layer (LL). The Host-Controller Interface (HCI) depicted in Figure 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. The maximum data rate is 1 Mb/s, and the typical communication range is in the order of few tens of meters.

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.

Above the link layer, the *Logical Link Control and Adaptation Protocol* provides two main functionality. The main role of the protocol is to provide multiplexing capabilities encapsulating upper-layers multiple protocols data into the standard Bluetooth LE packet format.

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 to 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 to the storing services). Once a new application is built on top 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 as *profile*. Standard profiles guarantee interoperability among devices of different vendors.

III. BlueVoice application

In the following the BlueVoice application is presented starting from the definition of a voice communication Bluetooth LE profile, then the application design is described by considering the communication roles of involved devices, audio processing and compression choices, packetization issues and bandwidth requirements. Two configurations are proposed, different in terms of audio acquisition, and therefore power consumption, thus addressing applications with different constraints. In the last part of the section a BlueVoice implementation in real hardware devices is presented before comparing and discussing the measured performance in terms of power consumption, memory footprint, processing requirements and Automatic Speech Recognition (ASR) performance.

A. Service definition

Since 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. Moreover, considering an half-duplex communication specific design choice are needed, presented in detail later on in this section.

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 authorisation properties. GATT server attributes, instead, are organised 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. Considering a monodirectional audio streaming asymmetric system, BVS profile exported by a server node exposes to clients how data are organised in terms of type and format, as well as how they can be accessed. BVS service is defined with the following attributes, shown in Figure 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 UUIDs (BVS UUID) is declared. BVS includes two characteristics, named *Audio and Sync Characteristics*. In 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 UUIDs (AudioData and SyncData UUID) containing actual audio data and side information synchronization values respectively. Audio and Sync Characteristics declaration 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.

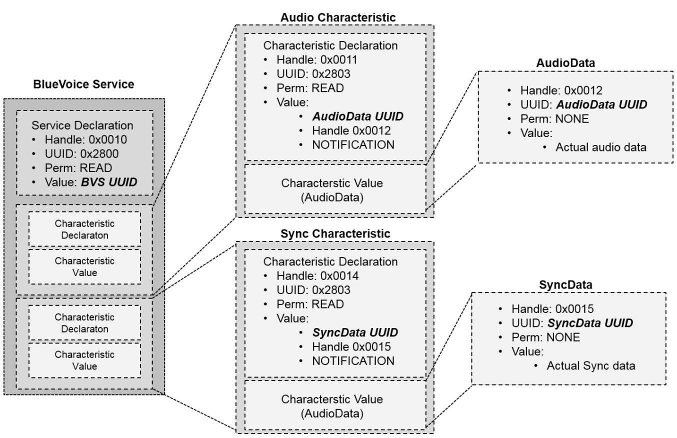

Fig. 2. BlueVoice service (BVS) definition

B. 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 Bluetooth LE specification, communications can be either broadcast or connection-based. The BlueVoice application relies, at LL level, on a connection-based communication paradigm, providing a permanent point-to-point link between two devices, playing two separate roles: Central and Peripheral. The Central (Master) device generally supports complex functions with respect to Peripheral (Slave) devices. 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. Such role assignment is consistent with the asymmetric design concept of Bluetooth LE where to the device having greater energy constraints is given less operations to perform: a portable device equipped with a small battery is typically a Slave. Nevertheless, it must be noted that, according to the specification [1], data can be sent independently by either device at each connection event and the roles do not impose restrictions in data throughput or priority. Considering an half-duplex scenario in which the BlueVoice application runs on autonomous battery powered wireless sensor devices both equipped with microphones (and eventually scalar sensors, as in the typical IoT scenario for ubiquitous monitoring applications), role assignment is therefore decoupled from transmitter or receiver functionality.

On top of LL, the GATT layer defines Client and Server roles of interacting devices that are independent by 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. Considering a monodirectional audio streaming asymmetric system, the only device that has voice informations is the one provided with a microphone that has therefore to be considered the Server of the communication. The other device acts as Client, sending request to the Server and accepting server initiated updates containing audio data. In a bidirectional system, where voice signal travels in either direction, the architecture is symmetric: both the Central and the Peripheral modules are provided with a microphone and may act as Servers, exporting audio data organized in attributes. At the same time, they can also act as Clients, sending requests and accepting updates from the other module.

In both directions, 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 module goes in advertising mode and starts sending advertising packets at low frequency. Vice versa, the Master 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, asynchronous notifications packets containing audio data are periodically sent from Server to Client, according to the selected direction: Peripheral-to-Central or Central-to-Peripheral. Figure 3 shows BlueVoice role assignment at the GATT layer.

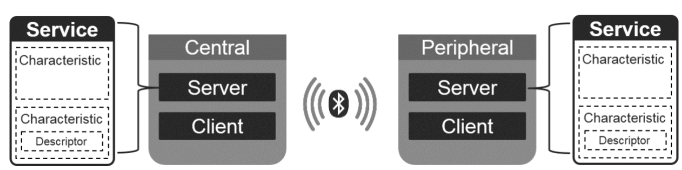

Fig. 3. BlueVoice profile role assignment

2) Audio processing.

Audio processing in the BlueVoice application is designed to achieve a target audio sampling frequency of 8kHz or 16 kHz at receiver side, selectable according to the application. In fact, 8kHz sampling rate could be an interesting option in scenarios where low-power constraints are more stringent and a lower audio quality is acceptable. Example of this applications are the ones where audio is meant to be used as input for ASR services and not for human listening.

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, while at the same time minimising 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 wireless sensor devices thanks to their interesting features in terms of dimensions and audio quality. Figure 4 shows the whole speech processing chain for a sampling rate of 16 kHz: 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. For 8kHz sampling rate, the resulting ADPCM sample rate is 8 ksamples/s, resulting in a bandwidth of 31.3 kbps includind side information. In the following, the blocks introduced above are deeply described.

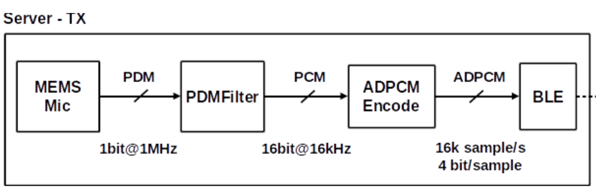

Fig. 4. BlueVoice transmission chain in 16kHz configuration

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 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. According to the selected sampling frequency, a different configuration of the decimation filter is used, thus obtaining 16-bit PCM data samples.

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 consists in predicting the current signal value from previous values, and transmitting 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 higher complexity 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 equal to 32 or 64 kbps depending on the sampling frequency, which is compatible with Bluetooth LE streaming capabilities.

As previously reported, the whole bandwidth required by the BlueVoice application is greater than the theoretical value of 32 or 64 kbps. This is because BlueVoice improves the communication robustness by adding additional information when data are sent through the channel. A connection interval of 10 ms is used for 16 kHz configuration, while a connection interval of 20 ms is used for 8 kHz use case. In fact, having a less amount of data to be transferred allows to increase the connection interval and therefore to save energy. Voice data packets are sent in form of packets of 20 bytes, so as to use for each data packet the maximum available payload.

Therefore, in the 16 kHz configuration voice data is sent in form of 4 packets every 10 ms, while in the 8 kHz configuration voice data are sent in form of 4 packets each 20 ms resulting in 64 kbps and 32 kbps of data respectively. Transmitter side information are sent at a lower frequency in form of an additional packet of 6 bytes every 160 ms, corresponding to 16 or 8 connection intervals. The whole data packetization policy on top of the Bluetooth LE protocol stack is summarised in Figure 5: 4 voice data packets of 20 bytes are sent at each connection interval of 10 or 20 ms via Audio characteristics, while transmitter side information are sent in an additional packet every 160 ms via Sync characteristic.

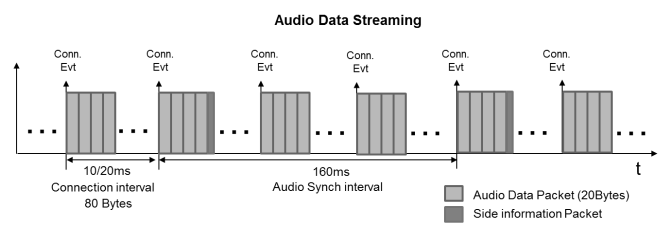

Fig. 5. BlueVoice packetization schema

C. 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 III-B, has been implemented in real hardware devices. The selected platform is the STMicrolectronics STM32 Nucleo L476 board [5], an open development platform based on STM32L476, a 80 MHz, 32-bit ARM Cortex-M4 microcontroller. Despite the system embeds a very powerful microcontroller with respect to usual platforms used in wireless sensor network applications, the STM32 Nucleo has been chosen because 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 specialised expansion boards, thus allowing exploration, prototyping and validation of new ideas. In particular, STM32L4 microcontroller was selected because of its advanced low-power functionalities and for the on-board Digital Filter for

Sigma-Delta Modulators (DFSDM) peripheral, implementing PDM to PCM conversion filter referred by Figure 4. These features make it perfectly suitable for BlueVoice application. Demonstrating an half-duplex communication channel, BlueVoice Central and Peripheral modules have a symmetrical hardware configuration based on a STM32Nucleo expanded with a Bluetooth LE connectivity board and with a microphone expansion board. BLE connectivity board is based on STMicroelectronics BlueNRG [6], 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, 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. Speech is acquired by means of an additional microphone expansion board, based on STMicroelectronics MP34DT01 [7], 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.

Figure 6 shows one module architecture schema: STM32 microcontroller is configured to acquire PDM samples from the microphone via DFSDM peripheral connected to a DMA peripheral, while communication with BlueNRG component is performed via a Serial Peripheral Interface (SPI) by means of a set of specific Application Programming Interfaces (APIs). Module architecture is symmetric for Central and Peripheral device and includes a USB Audio interface to provide reconstructed audio to a PC. Figure 7 shows the actual prototypes.

D. Performance

The real system described in Section III-C has been used as laboratory testbed to evaluate the performance of the BlueVoice application in terms of power consumption, memory footprint, processing requirements and ASR recognition rate. In particular, 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-Peripheral 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 Slave-Peripheral module in both 8 kHz and 16 kHz configurations. In addition, another reported performance indicator is the ASR performance measured at Rx side: in fact, such parameter can

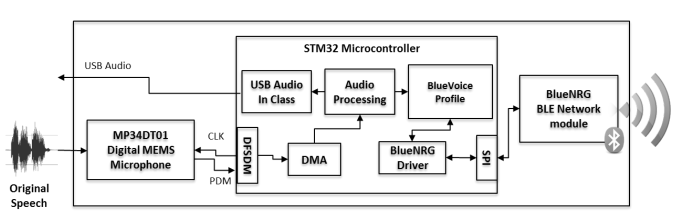

Fig. 6. BlueVoice modules architecture.

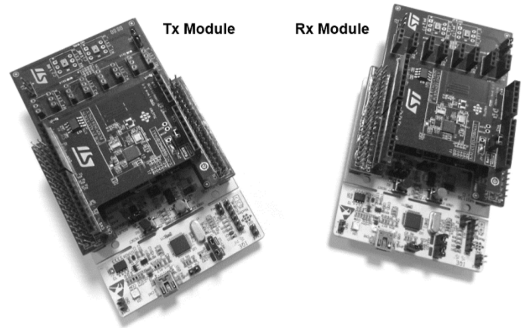

Fig. 7. Tx and Rx prototypes

be a key indicator of speech quality in voice communications and it is fundamental for emerging voice-controlled applications (remote controller, IoT objects).

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 by a STMicrolectronics STM32 Nucleo board, acting as host device, and a Bluetooth LE network module acting as controller device. In Table I power consumption values for both host and controller devices (STM32 and BlueNRG respectively) are reported for the three main states in which BlueVoice Peripheral module can operate: advertising, connection and streaming. Reported data compare 8 kHz and 16 kHz configurations and were measured operating the device at 3.3 V. It must be stressed that microcontroller power consumption strictly depends on hardware characteristics and low-power configurations, thus being an additive platform-dependent value in the overall computation of consumed power. Values reported here have therefore to be considered an indication and may vary according to application requirements.

TABLE I
BLUEVOICE POWER CONSUMPTION

Power consumption						
	8 kHz			**16 kHz**		
State	**STM32 (mW)**	**BlueNRG (mW)**	**Total (mW)**	**STM32 (mW)**	**BlueNRG (mW)**	**Total (mW)**
Advertising	3.20	0.30	3.50	7.92	0.30	8.22
Connection	3.20	0.78	3.98	7.92	1.56	9.48
Streaming	4.13	5.94	10.07	10.6	9.24	19.84

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, considering a Peripheral-to-Central communication, Peripheral node acts as Transmitter (server) while the Central node acts as Receiver (client): periodic notifications are sent from Server to Client. For 8 kHz configuration, during Advertising phase the total average power consumed by Tx Peripheral module (STM32 + BlueNRG) is very low 3.50 mW, while power consumption is equal to 3.98 mW when the connection is established. For 16 kHz configuration, instead, Advertising phase has a power consumption of 8.22 mW while Connection phase requires 9.48 mW. It is important to note that the power consumption in the connection phase is strongly related to the connection interval design choice, which is the main difference in the 8 kHz or 16 kHz configurations (20 ms and 10 ms respectively). In both cases, Connection interval is set to a value that is close to the minimum allowed by the standard (7.5 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 10.07 mW for 8 kHz configuration and 19.84 mW for 16 kHz configuration. Therefore, considering a battery with a capacity of 200 mAh, the ideal lifetime of an IoT node composed of STM32 + BlueNRG in continuous streaming mode would be approximately 65 and 33 hours respectively. Such power consumption values demonstrate suitability of BlueVoice approach for audio streaming over BLE and in particular that 8 kHz configuration allow to reduce power consumption significantly.

In addition to power consumption analysis the BlueVoice feasibility has been evaluated by considering its requirements in terms of memory footprint. The BlueVoice application in its actual implementation requires, as reported in Table II, the same amount of Flash (21.85 kB) for both configurations, while the requirements in terms of RAM memory are 13.32 kB for 8 kHz configuration and 7.86 kB for 16 kHz configuration. These values depends on the fact that in order to reduce overhead and power consumption of the solution, audio processing steps (PDM to PCM conversion and ADPCM compression) are performed each 20 ms and 10 ms for 8 kHz and 16 kHz configurations respectively. This results in a bigger amount of data to be stored for 8 kHz configuration between two successive processing steps. In both cases, these values are highly compatible with resource-constrained systems.

TABLE II
BLUEVOICE MEMORY FOOTPRINT AT TRANSMITTER SIDE

Memory footprint		
Memory	**Occupation 8 kHz (kB)**	**Occupation 16 kHz (kB)**
Flash	21.85	21.85
RAM	13.32	7.86

2) ASR performance.

The BlueVoice feasibility in terms of power consumption, processing requirements and memory footprint does not guarantee to have a speech signal with a sufficient quality at the receiver side. To the end of evaluating the performance of the BlueVoice solution, an extensive transmission test has been performed measuring ASR performance at receiver side, leveraging on a web-based ASR service. A number of audio samples containing known english words are recorded in parallel by a 16 kHz USB microphone (taken as reference) and by a 8 kHz and 16 kHz BlueVoice system and then passed to the ASR engine. The results of the test comparing the solutions in terms of recognized words are reported in the Table III.

Reported results confirm that ADPCM compression does not degrade the signal and is therefore suitable for ASR applications: BlueVoice 16 kHz configuration has a performance which is quite close to the one of USB microphone while 8 kHz system shows a minor degradation (18 %), which may be tolerable for applications where low-power requirements are extremely demanding. In fact, such ASR performance is obtained with a system requiring 50% of the power required by the 16 kHz configuration.

TABLE III
BLUEVOICE ASR PERFORMANCE

ASR Performance			
Measure	**USB 16 kHz**	**BlueVoice 8 kHz**	**BlueVoice 16 kHz**
Recognized words (%)	85	67	82

IV. CONCLUSIONS

In this article a solution to stream speech over Bluetooth Low Energy is presented. A vendor specific Bluetooth LE profile for half-duplex speech communication is first defined, then 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 Central node and a Peripheral node, acting as Bluetooth LE Server and Client according to the selected communication direction. Periodical notifications are sent from the Server to Client after the connection is established. Two different configurations are compared: 8 kHz and 16 kHz audio sampling rate. On the node acting as transmitter, MEMS microphone digital output in PDM format is acquired, converted to PCM and then compressed to ADPCM, resulting in a communication bandwidth of 32 or 64 kbps on the Bluetooth LE link respectively. The profile defines also a side-information mechanism at low-frequency, requiring an additional bandwidth but increasing error rejection capabilities. To evaluate the performance of the proposed solution, the BlueVoice application has been implemented in real hardware devices, fully digital system composed of a MEMS microphone, a STM32 microcontroller acting as host and a network module acting as Bluetooth LE controller. Performance evaluation reported in the article shows the feasibility of proposed solutions for low-power speech streaming applications in terms of power consumption, processing requirements and memory footprint. In particular, power consumption value of only 10.07 mW and 19.84 mW have been measured for the sensor device during audio streaming at 8 kHz and 16 kHz configuration, while completely acceptable memory and processing requirements have been experienced. Moreover, ASR performance has been measured as audio quality indicator. Such a figure of merit results in a percentage of recognized words equal to 67% for 8 kHz configuration and to 82% for 16 kHz configuration compared to 85% obtained by a 16 kHz USB microphone, showing that the BlueVoice application is able to reach very high speech quality values at receiver side and to combine low power performance.

References

[1] "Specification of the Bluetooth System," *Covered Core Package Version: 4.0*, June 2010.

[2] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, and C. Gomez, "Transmission of IPv6 Packets over BLUETOOTH(R) Low Energy," Internet-Draft, February 2015.

[3] R. Mangharam, A. Rowe, R. Rajkumar, and R. Suzuki, "Voice over sensor networks," in *IEEE Real-Time Systems Symposium*, December 2006, pp. 291–302.

[4] "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*, October 2003.

[5] "STMicroelectronics STM32 MCU Nucleo," www.st.com/stm32nucleo.

[6] "STMicroelectronics BlueNRG, Bluetooth® low energy wireless network processor," www.st.com/web/catalog/sense_power/FM1968/CL1976/SC1898/PF258646.

[7] "STMicroelectronics MP34DT01, MEMS audio sensor omnidirectional stereo digital microphone," www.st.com/web/en/catalog/sense_power/FM125/SC1564/PF250940.