Hindawi
Journal of Computer Networks and Communications
Volume 2019, Article ID 6595282, 14 pages
https://doi.org/10.1155/2019/6595282

*Research Article*

# Embedded Parallelism Enabling Ultralow-Power Zigbee Voice Communications

**A. Meliones**,[1,2] **E. Touloupis**,[2,3] **and J. Perello**[4]

[1]*Department of Digital Systems, University of Piraeus, Piraeus, Greece*
[2]*InAccess Networks, Athens, Greece*
[3]*European Patent Office, The Hague, Netherlands*
[4]*Ateknea Solutions, Carrer de Víctor Pradera, 45, 08940 Cornellà, Barcelona, Spain*

Correspondence should be addressed to A. Meliones; meliones@unipi.gr

Received 2 November 2018; Accepted 30 December 2018; Published 5 February 2019

Academic Editor: Zhiyong Xu

Copyright © 2019 A. Meliones et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Short-range wireless technologies are known to transmit voice, audio, image, and video messages in real time. Energy consumption and transmission reach are critical in such networks, especially for portable and power autonomous devices. The purpose of the Voice over Zigbee technology is to provide a competitive offering that excels in these performance aspects. Due to the CSMA-CA mechanism implemented in the 802.15.4 layer, a well-designed strategy must be considered in Zigbee to create a robust, reliable, and full-duplex conversation. In past efforts, we proved that the radio channel of Zigbee has enough bandwidth to support a full-duplex conversation with narrow-band voice codecs. Our embedded implementation of the Speex voice codec targeted the development of a low-cost, ultralow-power, long-range wireless headset using Zigbee technology to transmit voice in full-duplex mode for use with leading PC VoIP programs. Furthermore, we presented the real environment performance evaluation and power consumption tests involving the developed headset prototype. Talk time is comparable to Bluetooth including at the power budget, the codec processing, and analogue audio interface, but its deep-sleep lifetime more than doubles the Bluetooth performance. This was one of the very few successful efforts to port a voice codec on an ultralow-power DSP for use with power sensitive Zigbee applications, which is highly cited in the literature, proving additionally that using an open-source codec can deliver similar voice quality, reducing the total system cost. The current paper elaborates on the embedded parallelism of the Speex implementation and the exploitation of the DSP architectural parallelism which critically enabled the Voice over Zigbee application on the ultralow-power DSP platform. Another significant contribution of this work is towards understanding and resolving the challenges faced when trying to achieve good quality transmission of media over constrained devices and networks. The way to new ultralow-power voice-related Zigbee and constrained network applications is open.

## 1. Introduction

Zigbee is a wireless standards-based technology that addresses the needs of sensory network applications, enabling broad-based deployment of complex wireless networks with low-cost and low-power solutions that run for years on inexpensive primary batteries. The raw data rate of this technology at 2.4 GHz, 250 kbps is low compared with other wireless technologies, but sufficient to transmit voice using narrow-band voice codes.

Prior to our work, few approaches can be found in the literature that examine the possibility of 802.15.4/Zigbee networks supporting voice transmission, either through simulation or based on already deployed networks (e.g., [1–4]). The general conclusion drawn from previous work is that voice can be carried over Zigbee; however, there exist several restrictions in terms of available bandwidth and network topology.

The Eurostars Z-Phone project goal was to develop a low-cost, ultralow-power, long-range, ergonomic wireless headset using Zigbee Technology to transmit voice in full-duplex mode for use with leading PC VoIP programs (e.g., Skype and Messenger), in addition to a USB-Zigbee bridge module and a PC driver [5]. The initial objectives were to achieve 2x communication autonomy and 4x distance compared with Bluetooth technology. DECT and WiFi

consume much more than Bluetooth and cannot be easily integrated into small devices. This increase in autonomy has also an impact on the lifetime of the rechargeable battery and consequently on the lifetime of the headset, since batteries in wireless headsets are not replaceable.

Due to the limited available bandwidth, voice must be compressed before being transmitted. The selection of the most suitable voice codec and its implementation is challenging, considering the requirements of Z-Phone. The paper summarizes the results of our previous work regarding the design, implementation, and validation of the Z-Phone architecture and the headset system and presents the details of the embedded parallelism which made the application feasible in the constrained environment. Important future work items are presented in the conclusions section.

## 2. VoZ Technology

Various wireless communication methods are known to transmit voice, audio, and/or video messages in real time. Examples of these methods are used in the Bluetooth technologies based on the IEEE 802.15.1 protocol, WiFi based on the IEEE 802.11 a/b/g protocol, and DECT. The purpose of the VoZ (Voice over Zigbee) technology is to overcome the competitive technologies regarding energy consumption, a critical issue for portable and/or power autonomous devices, and transmission reach. Due to the CSMA-CA mechanism implemented in the 802.15.4 layer [6], several strategies must be considered in Zigbee to create a robust, reliable, and "full-duplex" conversation. Such strategies include setting up a partially asynchronous communication network and not using confirmation messages in the communication transaction. Robust and good quality conversations have been implemented over Zigbee testing several voice codecs such as G.729A (8 kbps) or G.723.1 (5.3/6.3 kbps). As shown below, the radio channel of Zigbee has enough bandwidth to support a full-duplex conversation with narrow-band voice codecs.

A deeper study of the Zigbee link necessitates the determination of the Zigbee stack time. Stack time is the time that the Zigbee stack firmware running on the microprocessor takes from capturing a Zigbee packet in the air to delivering it to the application layer. A set of transmissions were carried out between Zigbee nodes, with one node sending a message and another resending it instantaneously to calculate the time required for the Zigbee stack to process a packet (Figure 1). Measurements resolution was 1 ms.

The difference between actual time and time stamp (ticks) can be calculated by the following expression:

$$\text{Ticks} = 2 \times \left(t_{\text{stx}} + t_{\text{p}} + t_{\text{srx}}\right). \tag{1}$$

To simplify the calculation, we can suppose that

$$t_{\text{s}} = t_{\text{stx}} = t_{\text{srx}}. \tag{2}$$

The final expression would be

$$t_{\text{s}} = \frac{\text{Ticks}/\left(2 - t_{\text{p}}\right)}{2}. \tag{3}$$

A Zigbee sniffer was used to find out the real header length transmitted. In our trials, using the stack from EmberZNet PRO 3.11 with security features disabled over the xip EM250 [7], the header length resulted to be 30 bytes (4 PHY, 10 MAC, 8 NWK, and 8 APL) [8]. The PHY header length was variable between 3 and 5 bytes. Table 1 shows the test results obtained in function of the bytes of information or payload. The tick value has been obtained as the mean from one thousand measurements.

Table 1 depicts the communication test results including the minimum time between voice frames required to create a full-duplex conversation in function of the payload. To extract the stack time, the Zigbee 2.4 GHz data rate of 250 kbps is taken into account in the calculation of the propagation time. This time depends on the total length of the packet to be sent. In all cases, the measurements follow a normal distribution as shown in Table 2. Figure 2 shows the similarity between both shapes—ticks measurements distribution and normal distribution—in this diagram for 80 bytes.

According to Table 2 and the normal distribution probability formula, we can send and receive an 80-byte packet payload (plus headers) every 22 ms with a probability error near to zero, as shown in the following calculation. That is to say, we could create a conversation using a voice codec of 29 kbps ($80 * 8$ bits/0.022 ms), in this case:

$$\begin{aligned}
\Pr\left(t > 22\ \text{ms}\right) &= 1 - \Pr\left(t \leq 22\ \text{ms}\right) \\
&= 1 - \frac{1}{2}\left(1 + \text{erf}\left(\frac{22\ \text{ms} - \text{mean}}{\text{deviation} \times \sqrt{2}}\right)\right) \\
&= 1 - \frac{1}{2}\left(1 + \text{erf}\left(\frac{22\ \text{ms} - 16.81\ \text{ms}}{1.1\ \text{ms} \times \sqrt{2}}\right)\right) \\
&= 1 - \frac{1}{2}\left(1 + \text{erf}\left(3.3363\right)\right) \\
&= 1 - \frac{1}{2}\left(1 + 0.99999762\right) \approx 0.
\end{aligned} \tag{4}$$

*2.1. Progress beyond The State-of-The-Art.* This section presents a comparison of our work in the Z-Phone project with previous relevant work. The most similar research to the Z-Phone project was the one named "Voice Communications over Zigbee Networks," performed by AT&T Labs Research [1]. In the Zigbee voice communication transmission, the main issue to solve is to avoid the collisions between packets. If two packets are transmitted at the same time, the CSMA/CA used in Zigbee adds a random delay to both packets before resending them. This way the packets are sent too late for a real-time voice communication. The difference between our solution and the one proposed by AT&T is the method used to avoid these collisions. We use a ping-pong method that only allows a node to transmit a packet when the other nodes are not sending any

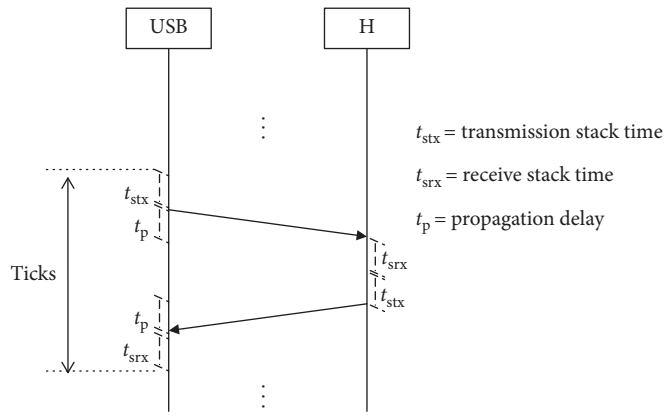



Figure 1: Test communication diagram.

Table 1: Communication test results in function of payload.

| Payload (bytes) | 10 | 20 | 40 | 60 | 80 |
|---|---|---|---|---|---|
| Mean ticks (ms) | 11.05 | 11.80 | 13.60 | 15.08 | 16.81 |
| Max ticks (ms) | 17.00 | 18.00 | 18.00 | 21.00 | 22.00 |
| Propagation time (ms) | 1.28 | 1.6 | 2.24 | 2.88 | 3.52 |
| Max stack time (ms) | **3.61** | **3.7** | **3.38** | **3.81** | **3.74** |
| Mean stack time (ms) | **2.12** | **2.15** | **2.18** | **2.33** | **2.44** |

Table 2: Distribution of 10000 tick results in function of payload.

| Payload | Tick measures distribution (ms) | | | | | | | | | | | | | | Mean | Deviation |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | | |
| 10 | 820 | 2312 | 3413 | 2433 | 1011 | 7 | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 11.05 | 1.10 |
| 20 | 72 | 1171 | 2629 | 34297 | 2066 | 551 | 4 | 3 | 5 | 2 | 0 | 0 | 0 | 0 | 11.80 | 1.10 |
| 40 | 0 | 0 | 184 | 1449 | 2979 | 3297 | 1733 | 356 | 1 | 1 | 0 | 0 | 0 | 0 | 13.60 | 1.10 |
| 60 | 0 | 0 | 0 | 0 | 767 | 2269 | 3510 | 2382 | 1040 | 29 | 1 | 1 | 1 | 0 | 15.08 | 1.10 |
| 80 | 0 | 0 | 0 | 0 | 0 | 64 | 1170 | 2666 | 3408 | 2076 | 611 | 1 | 0 | 4 | 16.81 | 1.10 |

information (similar to a synchronization method used in the 1-channel serial data buses), while the AT&T system uses the solution proposed in the VoIP protocol, which includes a TDMA mechanism access to the channel. At the end, the work presented in [1] will achieve similar features, with limited 1-2 hops voice communication and 8 meters free obstacle transmission distance. In our own work, we have used the G.729a codec to test and implement our ping-pong method, as it is the most standard for voice compression in embedded systems. AT&T system has used the same codec in their work. In the Z-Phone project, we moved further and proved that, using an open-source codec can deliver similar voice quality, reducing the total cost of the system. Moreover, we have presented engineering details of efficient codec implementation and validation in a restricted ultralow-power-embedded environment [7, 9].

The works presented in [3, 4] have a different approximation as per voice transmission. Both share the voice transmission with various data transmission, which increases significantly the packet collisions and applies ineffectively a solution which is similar to our ping-pong method. In these cases, they use a time-slotted communication in the MAC layer to avoid the collisions, but a strict synchronization method (beaconed frames or AM radio synchronization packets) needs to be integrated in the proposed mechanism to work correctly. The codecs used in these works are similar to G.729a or Speex, with a bit rate of 4.4 to 8 kbps in one case and 8 to 16 kbps in the other, respectively. Furthermore, the work presented in [2] is only a simulation to test if a Zigbee network could have enough throughputs to manage a 16 kbps voice codec, but the authors do not provide any information about how to solve the synchronization and collision problem.

In conclusion, the focus of our work is on the challenges faced when trying to achieve high-quality voice transmission over constrained devices and networks, which is also what sets apart this work from the previous publications referenced. The aim is to present in detail the above procedure using Z-Phone's requirements as a case study for this problem. Furthermore, our porting of the Speex open-source voice codec on an ultralow-power DSP for use with power sensitive Zigbee and constrained network applications is one of the very few successful efforts in this area.

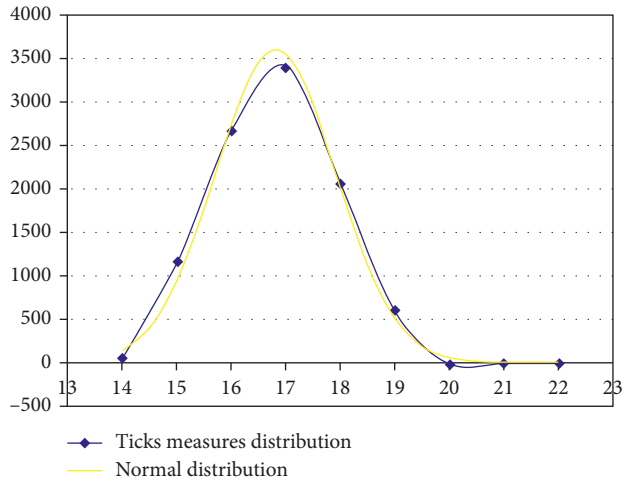



Figure 2: Graphical representation of ticks measurements distribution for 80 bytes payload.

## 3. Related Works

Quite many research works in the literature acknowledge and build upon our contribution and examine or advance voice transmission over low-power WSN, involving multiple hops as well, and propose new interesting applications. An indicative nonexhaustive list of follow-up efforts is presented in this section. Sunder and Kushwaha [10] investigate the different factors like bit rate, algorithmic delay, and implementation and robustness of codec's perceptual quality to noise. These factors play a vital role in choosing a speech codec and evaluating its performance. Raju and Sharma [11] compare Speex and Opus speech codecs for wireless sensor network- (WSN-) based speech/voice transmission between two wireless nodes. Opus can achieve better quality at a wider range of sample frequencies. The ultimate goal with Opus codec is to integrate the best features of both speech and audio codecs. Gentili et al. [12] present a solution for streaming speech over Bluetooth Low Energy (BLE). The application requires a communication bandwidth of 64.3 kbps to transmit audio at 16 kHz in ADPCM format. 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 a Bluetooth LE controller. Performance evaluation reported in this article shows the feasibility of the solution for the IoT context in terms of power consumption, processing requirements, and memory footprint. A power consumption of 10 mW has been measured for the network module during audio streaming. Prabahar [13] explains how to make high-performance WSN nodes by incorporating a DSP chip with the WSN node and how to implement a suitable speech processing algorithm on it. The design of hardware is based on an ultralow-power DSP chip to improve the battery operating time of the WSN node. The proposed DSP-enhanced WSN node employs G723.1 speech compression algorithm, which provides low bit rate of 5.3 kbps with acceptable quality. Based on the power consumption calculation, suitable Li-ion battery pack with smaller size is selected for the application. The high-performance WSN nodes can be incorporated in voice over Zigbee and WSN acoustic applications, such as WSN-based remote area surveillance applications and communication applications in mines.

Chang et al. [14] designed and implemented the Speex voice codec at 8, 11, and 15 Kbps bit rates on an embedded system with Xbee WSN nodes using multihopping Zigbee communication, presenting a performance analysis in terms of distance-signal strength dependence, network delay, and PESQ-MOS voice quality. Meiqin et al. [15] achieved wireless real-time voice communication based on TI-Zstack protocol and AMBE-1000 and CC2530 modules. The proposed wireless voice communication system consists of five nodes, of which one is coordinator, one is end device, and the other three are routers. The system can be used in underground coal-mine equipment, earthquake disaster relief, fire fight rescue, etc. The longest end-to-end communication distance is 15 m. When using power amplifier (increasing the energy consumption) and routers, distance can be longer. The max hop is two due to each router adding noise to the signal. Yang et al.[16] provide a voice communication system which adopts CML's CMX618 for voice data collection and forward error correction coding, TI's CC2530 for Zigbee network formation, and voice data transmission. The authors assess the system for Point-to-Point (P2P) and End-to-End (E2E), using a network of one voice base station and eight router nodes, configurations. The results, stressing the voice quality in mesh Zigbee with multiple hops (for mine monitoring), show that the proposed system ensures voice quality in complex Zigbee network with higher integration and lower power consumption. Fu et al. [17] present the detailed development of a proprietary low-power WSN platform with star topology and 3 voice communication modes: P2P, Peer-Central-Peer, and voice conference, using the CVSD 15.625 Kbps voice codec. Zigbee communications are based on the 8-bit RF SOC CC2430/CC2530. Extensive analysis demonstrates that the proposed system is a low-power, low-speed, and high-performance WSN platform. It consists of up to 16 audio sensor nodes, 64 typical parameter monitoring nodes, and 1 central node for network establishment and management. The audio channel capacity is 3 real-time two-way voice communications or audio conference including all audio nodes at the same time, and the voice delay is less than 40 ms. The communication distance between audio nodes is longer than 70 meters indoors and 120 meters outdoors. The authors suggest possible applications in emergency voice communication, audio/sound sensor networks, and health monitoring systems.

Chang et al.[18] develop an end-to-end rescue communication voice gateway to provide a stable voice transmission over Bluetooth and Zigbee networks for mountain climbers. To implement the device, it adopts Speex with submode 4 and 11 kbps data rate to provide the best tradeoff between speech quality and computational complexity, based on our work. The performance analysis, in terms of end-to-end throughput, packet loss rate, jitter, and delay, shows that the proposed implementation can efficiently

support voice transmission over wireless sensor networks. A mine wireless voice communication system based on Zigbee is presented in [19] adopting CC2530 as RF sending-receiving unit of voice communication node and AMBE voice codec (selected as an amateur radio speech codec) to realize voice message two-way wireless communication over the Zigbee protocol. Zigbee is also examined in the framework of space mission operations [20]. WPANs are used in space to convey information over relatively short distances among the participant receivers. Unlike WLANs, connections effected via WPANs involve little or no infrastructure. This allows small, power efficient, inexpensive solutions to be implemented for a wide range of devices that can be duty-cycled aggressively to a low-power sleep state. Report [20] acknowledges that Zigbee has not been as widely adopted as expected, due in large part to the difficulty of the 802.15.4 MAC in enabling reliable transport in the face of difficult networking environments. In this framework, the authors of the present work were contacted by engineers of the Institute of Space Techniques and Technologies of the Republic of Kazakhstan, who were working in the field of transmitting voice over Zigbee technology, to discuss important details of our VoZ speech codec implementation described in [9].

The European EAR-IT project addresses real-life experimentations of intelligent acoustic for supporting high societal value applications in a large-scale smart environment. For instance, a city emergency center can request on-demand acoustic data samples for surveillance purposes and management of emergencies. Pham et al. [21] and Pham and Cousin [22] present experimentations on streaming encoded acoustic on the SmartSantander large-scale test-bed comprising more than 2000 IoT nodes (ATmega1281 microcontroller-based WaspMote nodes and TelosB CM5000 and CM3000 motes). An audio board was built around a 16-bit Microchip dsPIC33EP512 microcontroller offering enough processing power to encode the audio data in real-time to produce an optimized 8 kbps encoded Speex audio stream (Speex encoding library is provided by Microchip). Audio boards used the TelosB nodes as host boards due to inability of multihop transmission in WaspMotes. Multihop transmission used both WaspMote and TelosB motes as relay nodes. These works further highlight the main sources of delays and show how multihop streaming of acoustic data can be achieved by carefully taking into account these performance limitations with appropriate audio aggregation techniques.

Koucheryavy et al. [23] acknowledge the positive experience of transfer of voice data over the Zigbee protocol and examines research issues of Public Flying Ubiquitous Sensor Networks (FUSN-P) and Flying Ad Hoc Networks (FANETs) involving terrestrial segments and aerial segments composed by Unmanned Aerial Vehicles (UAVs) and drones. It presents a model network for full-scale experiment and solutions for the Internet of Things. The voice and video transmission from terrestrial segment to UAV-P using different protocols (Zigbee, 6LoWPAN, and RPL) is a critical investigation task because often it may be the only chance to pass the necessary information to the area of terrestrial sensor fields. Other important tasks involve the development of clustering algorithms for terrestrial and flying segments, route optimization for data collection, as well as the consideration of UAV-P's as a queue system and FUSN-P as a queue network, respectively. Kirichek et al. [24] examine the efficiency of using Zigbee in Flying Ubiquitous Sensor Networks for transferring voice and image data. The authors conclude that Zigbee networks, praised for their low cost, high autonomy, simple creation, and survivability, are useful for transmitting multimedia information only where requirements to the quality of voice and image transmission rate are low.

## 4. Z-Phone Architecture

The software/firmware architecture of Z-Phone used with a PC is illustrated in Figure 3. It consists of a wireless headset and a USB dongle providing Zigbee connectivity to the PC. In addition, the Z-Phone driver installed on the PC provides a software interface to any hosted VoIP application. The Z-Phone driver and the headset's DSP contain one instance of a voice codec. Voice packets arriving from the Internet are handled by the VoIP application (e.g., Skype). The decoded voice data are buffered for the Z-Phone driver to re-encode them using the voice encoder. The encoded voice data are passed through the USB interface to the dongle where they are handled by the Zigbee stack and are transmitted over the air to the Z-Phone headset. After passing the Zigbee stack at that end, the data are transferred to the headset's DSP engine where they are decoded and reproduced as voice through the speaker. The opposite path is followed for the upstream voice data from the headset's microphone to the Z-Phone driver and then from the VoIP Application to the Internet.

Several factors affect the useful bandwidth that is used for carrying the encoded voice frames. Most importantly the fact that voice is transmitted in full-duplex mode reduces the available bandwidth in half, since the two nodes do not transmit data concurrently. In addition, it was decided not to include many voice frames in one packet in order to reduce latency and also reduce the impact of lost packets on voice quality. The tradeoff for this decision is the considerable overhead induced by the headers. Since small packets are regularly transmitted, stack processing and transmission times also affect the useful bandwidth. In Section 2, we proved that, in the worst-case scenario (maximum transmission times), the respective data rates should not exceed 29 Kbps. Several measurements under these conditions show that data rates up to 38 Kbps are feasible.

## 5. System Implementation and Validation

*5.1. Codec Selection.* In Z-Phone, we defined a set of requirements to provide a guideline in the selection of the most appropriate codec. A fixed-point implementation had to be considered since the DSP platform of the headset was a low-end, low-power DSP without FPU support. Bandwidth, processing, and memory limitations of the end system favored the selection of a low bit-rate codec. However, voice quality is an important factor for Z-Phone; hence, the

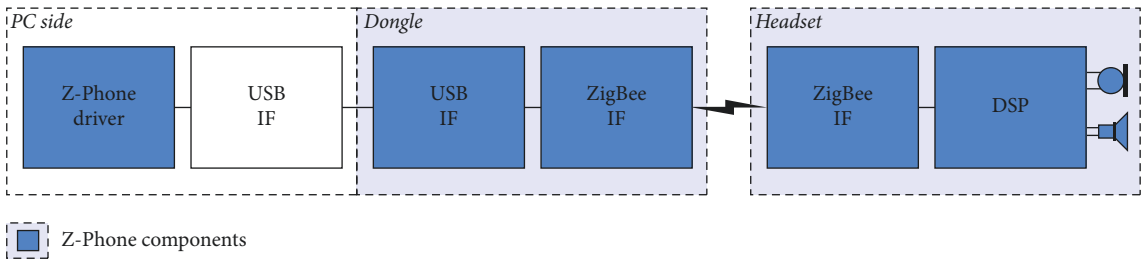



Figure 3: Z-Phone software/firmware architecture.

tradeoff for choosing a low bit-rate codec should be carefully evaluated. Furthermore, voice quality is affected by the algorithmic delay of a speech codec (time required to gather the samples that form one speech frame), being a significant part of the transmission delay between the sender and the receiver in a VoIP call (see [9] for a comprehensive analysis on algorithmic delay). Another aspect of speech codecs that affects voice quality in network conditions with increased packet loss rates is their packet loss concealment mechanism.

Most of the voice codecs evaluated had a fixed mode of operation in terms of bit rate and consequently voice quality. The only codecs that offer flexibility in this matter, which would allow fine tuning when integrated in the Z-Phone system, are AMR, SILK, and Speex with the first being royalty-based. A royalty-based codec, although used often for proof-of-concept work (e.g., [1]), clearly hinders the market perspective of the product envisioned by Z-Phone, by increasing its cost. In addition, the Z-Phone system does not require the use of specific codecs for issues such as interoperability. For that reason, the standard G.729 choice was abandoned. SILK and iLBC presented many attractive features but at the time of performing this work they were not available as open source. Therefore, the Speex codec was selected for this project, since it can offer superior voice quality to the vast majority of equivalent codecs when operating at 15 or 18 kbps [9].

*5.2. Codec Implementation.* Available Speex source code ports to popular DSP platforms, such as ARMv4, ARMv5E, the Blackfin architecture, the TI C5xx family, Freescale's DSP56852, and the Microchip dsPIC family are not suitable for Z-Phone that targeted an ultralow-power current consumption for both the DSP and the audio interface support circuitry. The DSP selected for Z-Phone was ONSemi's BelaSigna 300 [25]. A very important advantage of BelaSigna 300 over other similar DSPs (e.g., microchip's dsPIC family) is the integrated analogue audio interface that minimizes the need for external components, which could prove to be a huge benefit in housing and its associated costs. It offers high-performance audio processing capabilities and flexible programming while satisfying form-factor (size of a rice seed) and low-power constraints. The main CFX DSP core is user programmable using 24-bit fixed-point, dual-MAC, and dual-Harvard architecture. It is able to perform two MACs, two memory operations, and two pointer upgrades in one clock cycle. BelaSigna 300 further includes a configurable accelerator that is controlled by the main DSP core.

Codec implementation involved the porting of the subset of Speex encoder and decoder functionality that was necessary for our application. The question that was raised was whether a very small IC that operates at a low clock frequency which is expected to present performance and memory constraints such as BelaSigna 300 would be able to handle the processing power and memory requirements of Speex. In [26], a port of Speex operating at 8 kbps on an ARM-based microcontroller working at 72 MHz and the resulting resource requirements are presented; however, the target platform was not directly comparable to BelaSigna 300. On the other hand, the results of the porting of Speex on dsPIC [27], using only narrowband encoder and decoder and only one mode of operation (8 kbps), indicated that BelaSigna could support at least part of Speex functionality.

We have presented detailed information regarding the codec implementation in [7, 9], which we do not reiterate here. We decided to implement submodes 3 and 4 (8 and 11 kbps) as these seem to present the best tradeoff between speech quality and performance requirements. They also share the same functions for LSP quantization and adaptive codebook search meaning that supporting both submodes does not require significant additional effort.

During the codec implementation, it was found that it was not always possible to achieve bit-exact results (exact same output with a reference implementation for a specific input). The two factors that affected bit-exactness were the differences in arithmetic and in rounding between BelaSigna 300 and Speex source code. The memory requirements of the partial Speex port to BelaSigna 300 are presented in Table 3.

*5.3. Speech Quality.* Speex delivers good quality both in 8 kbps and in 11 kbps mode [28]. In order to evaluate the outcome of the voice codec implementation, a series of tests on the speech quality was performed using ITU-T Perceptual Evaluation of Speech Quality test methodology for automated assessment of the speech quality as experienced by a user of a telephony system [29]. PESQ takes as input the original signal which in this case was a wav file containing 51 seconds of different male speakers and the degraded signal which was a wav file that resulted from encoding and decoding the original wav file.

Figure 4 depicts the PESQ-MOS obtained for the two Speex submodes that are of interest for Z-Phone, for different complexities. Each submode was tested in three

Table 3: Memory requirements of Speex (submodes 3 and 4) in Belasigna 300.

| Memory structure | Data width | Available | Used by Speex |
|---|---|---|---|
| Program memory | 32 bit | 12288 | 6988 |
| X data memory | 24 bit | 6144 | 5257 |
| Y data memory | 24 bit | 2048 | 619 |

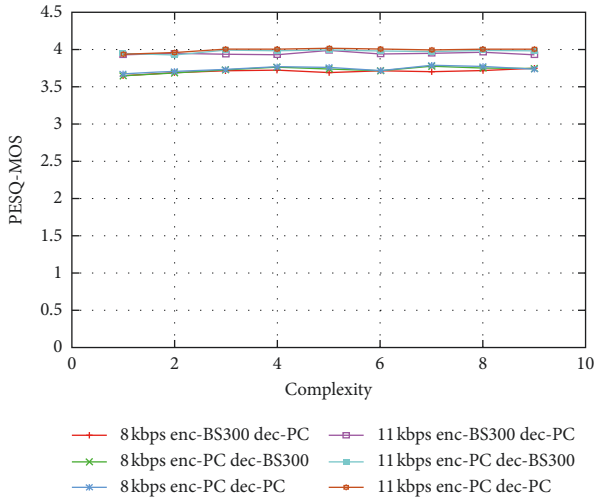



Figure 4: Speech quality of Speex implementation in BelaSigna 300.

scenarios, including encoding and decoding on the PC using the original Speex library as well as Belasigna transmit encoding and receive decoding. It must be noted that this test is not used as an evaluation of the speech quality that Speex is able to deliver, but instead as an indication for the development process of the correctness of the porting. It also gives an indication of the impact of the non-bit-exact implementation. As it can be seen on the graph, the Speex port to BelaSigna 300 follows closely the original Speex implementation in terms of speech quality with no significant deviations observed. It can also be seen that the value of the "complexity" configuration parameter does not have a significant impact on voice quality. Considering the tradeoff between performance requirements and speech quality, a value of 1 to 2 for the "complexity" parameter of Speex seems to be sufficient.

*5.4. Performance.* Speex uses a 20 ms speech frame. Therefore, it must be ensured that one instance of the encoder, one instance of the decoder, and the remaining tasks, such as data transfer and mixing, are executed in less than 20 ms. The decoder execution time does not change significantly for different submodes and values of complexity in the encoding process, and it was measured to be around 1.5 ms. The remaining tasks require roughly 0.5 ms, mostly due to data transfers.

Figure 5 depicts the execution time of all the DSP's tasks for the two submodes and for different values of the complexity parameter. The execution time exceeds the 20 ms

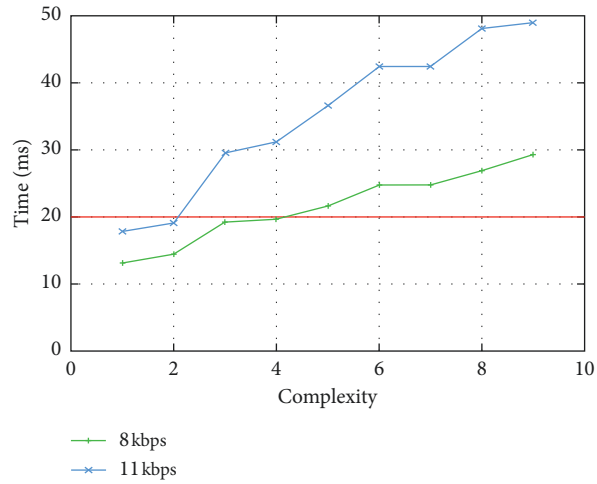



Figure 5: Execution time of the DSP tasks.

limit for values of complexity higher than 2 for 11 kbps and higher than 4 for 8 kbps. However, since the speech quality is not significantly improved with higher values of complexity, the performance of the implementation is acceptable for this application.

*5.5. Transmission Tests.* Three different distances were applied between the Z-Phone headset and the USB dongle to check the transmission capability of the system. To test the link, 256 packets were sent to the USB dongle, with 60 msec delay after each package. The transmission power was 3 dBm. The Link Quality Index and the received signal strength were recorded for every received packet (Figures 6 and 7). 255 packets arrived from distances 1.5 m and 5 m (99.61%). Another packet was lost when distance between the devices increased to 8 m (successful reception 99.22%). As expected, the results show that as the distance between the Headset and the Dongle increased, the quality of the link decreased. However, there was no big difference detected between the link quality results even if it measured at 5 m or 8 m distances.

*5.6. Power Consumption.* This section summarizes the power consumption performance of the Z-Phone system and a comparison with a Bluetooth reference system which is based on the Infineon PMB 8753 BlueMoon UniCellular single-chip Bluetooth v2.0 + EDR solution [30]. The major energy consumption elements of the Z-Phone system are the Belasigna 300 audio DSP and the EM250 single-chip Zigbee solution with an integrated Zigbee transceiver. The system further comprises (i) a small power management unit targeted for low-power consumer handheld end equipment, which contains the battery charger and a step-down DCDC converter and (ii) a programmable low-power clock generator (10 $\mu$A max in power-down mode), which provides the external clock required by Belasigna 300 WLCSP package at 40 MHz working frequency for optimized codec performance on the restricted platform. The Bluetooth chipset is clocked through the internal crystal oscillator

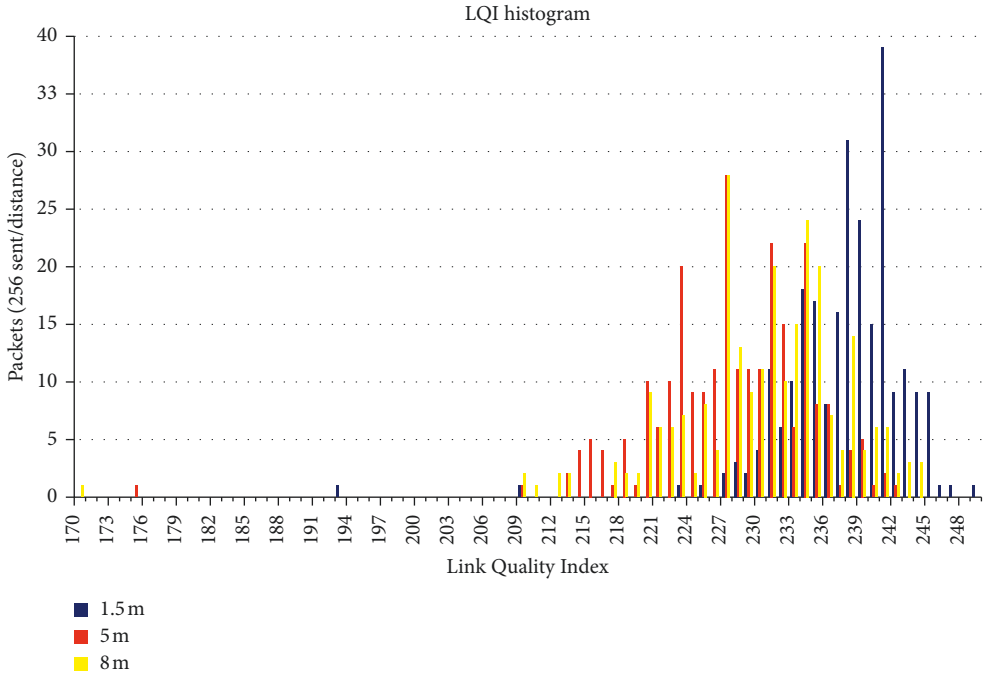



Figure 6: Transmission test: Link Quality Index.

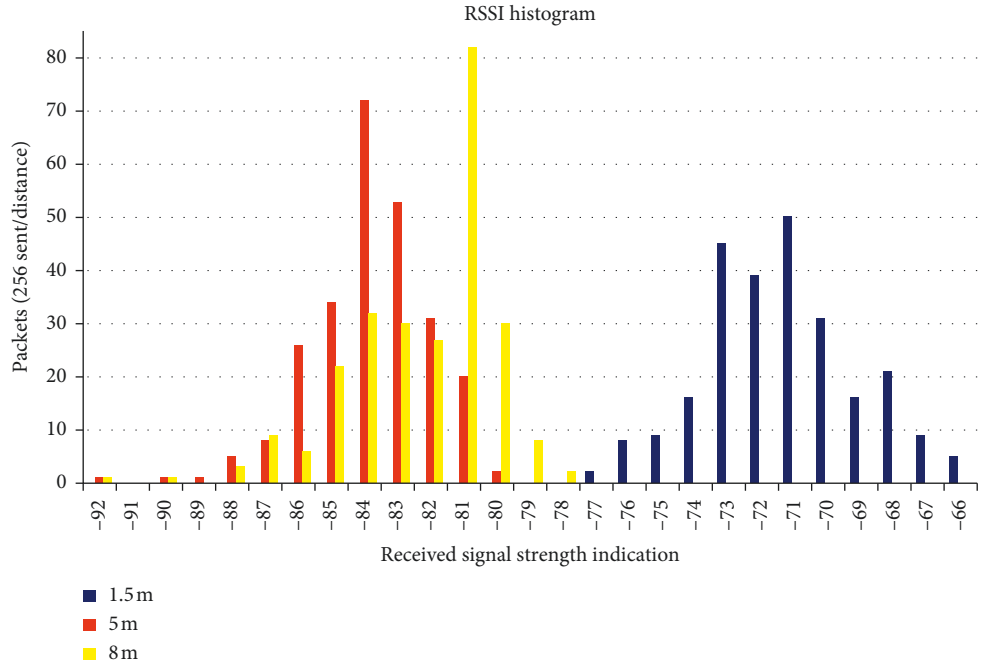



Figure 7: Transmission test: RSSI histogram.

which generates a 26 MHz reference clock. Both systems are powered by a 3.7 V-150 mAh battery.

Table 4 summarizes the power consumption figures. The typical Z-Phone talk time is approximately 7.51 hours. In deep-sleep mode Z-Phone can demonstrate an impressive lifetime reaching up to 2362 h (98.42 days) before battery charging is necessary. As expected, Z-Phone takes full advantage of the exceptionally ultralow-power consumption of the Zigbee transceiver in deep-sleep mode (1 $\mu$A). The Bluetooth reference system talk time is reaching 7.88 h over an eSCO link carrying EV3 voice packets and 8.23 h over a SCO link; however, at almost the same power budget, Z-Phone integrates the DSP voice encoding/decoding and analogue microphone/audio interface. In our low-power optimized Bluetooth reference system design, the standby time can reach up to 175.91 h (mode not available in

Table 4: Power consumption: Z-Phone vs Bluetooth.

| Time (h) | Z-Phone | Bluetooth | | |
|---|---|---|---|---|
| | | SCO/HV3 | eSCO/EV3 | ACL/DH1 |
| Talk | 7.51 | 8.23 | 7.88 | 4.63 |
| Standby | n/a | 175.91 | 175.91 | — |
| Deep-sleep | 2362 | 1144 | 1144 | 1144 |

Z-Phone), while the deep-sleep lifetime (similar to the "almost ready" function available in some Bluetooth headset products) can reach 1144 h (48 days). A methodology for measurement of average current consumption as a useful tool to select a chipset/module and help the engineer understand a system/product early in the development phase is presented in [31]. We further measured the efficiency of our Bluetooth reference system for carrying voice over packet-oriented ACL links, which has been shown to achieve better TCP connection performance at a slight increase of voice delay in a piconet topology sharing bandwidth between voice and data links [32]. Even if Bluetooth headsets do not use ACL links for voice communication, in fact this comparison with Z-Phone over a better comparable network sublayer demonstrates the power efficiency of our Zigbee implementation as opposed to Bluetooth in terms of bandwidth use.

Zigbee demonstrates more than 2x deep-sleep lifetime against Bluetooth between successive battery charges, consequently extending battery life. Excluding the almost zero current drawn by the Zigbee transceiver and the power management unit and analogue input and output circuitry which can be considered common between the two systems, it turns out that Z-Phone's Belasigna 300 DSP and low-power clock generator in total require approximately half of the ultralow-power mode current of the Bluetooth chipset. Current consumption values are taken with: LEDs and external EEPROMs disconnected, 150 Ohm speaker impedance, no RF retransmissions, RF TX power set to 0 dBm (class 3 devices), and microphone bias set to minimum current level. More details regarding the Z-Phone DSP and clock generator current consumption can be found in [30].

*5.7. Z-Phone Headset Unit.* The headset unit plays an important role in the Z-Phone system: it not only serves as a voice transcoder combined with a Zigbee transceiver but also provides a basic optical and interacting interface for the users and battery charging options via standard microUSB socket. The audio DSP is wired to the EM250 Zigbee transceiver via I2C communication bus and eases the interfacing of the microphone and the speaker by providing a built-in analogue front-end and Class D output stage. Mixing the different call tones/status with voice is also the task of the DSP.

The unit is powered by a standard 3.7 V-150 mAh Li-Ion coin battery. Charging it and supplying the different voltage levels for the DSP and the Zigbee chip is the role of a multioutput DC/DC converter and battery charger unit. The battery can be easily charged by plugging the unit to any USB host device. The user can read the battery status from the optical indicators located on the headset. To eliminate glitches, artefacts in the voice and to handle the demand of the two different voltage levels and the two different frequencies of the clock signals, a serially configurable low-power 2-channel PLL clock generator that offers low jitter, individually settable power supply and frequency for both clock outputs is also incorporated in the design.

A standard user interface was created for the user to learn how to use the buttons of the headset: power on/off, hook on/hang up, pairing, and volume up/down. The same applies to the optical indicator. The dedicated LEDs indicate different battery and call statuses. The housing of the headset is made with injection moulding using a nonconducting thermoplastic material, ABS, which is impact resistant and mechanically robust.

*5.8. Z-Phone Driver.* The USB dongle supports USB Audio and USB HID standards, so it is compatible with the major operating systems. The memory resident application running under Windows environment provides a bridge between selected applications (e.g., Skype and X-Lite) and the headset by handling call events, such as picking up an incoming call or terminating a call. It also contains features such as call logging and an address book including photos, etc. Without this application, the headset can be used in a similar way to the ordinary headphones.

## 6. Embedded Parallelism

*6.1. Belasigna 300 System Architecture.* The BelaSigna 300 system is an asymmetric dual-core architecture, mixed-signal system-on-chip designed specifically for audio processing. The BelaSigna 300 system is centered around two processing cores: the CFX DSP and the HEAR Configurable accelerator. The CFX DSP Core is used to configure the system and coordinate the flow of signal data progressing through the system. The CFX DSP can also be used for custom signal processing applications that cannot be handled by the HEAR Core. The HEAR Core is a microcode configurable signal processing engine that works with the CFX DSP Core to execute a variety of signal processing functions.

The CFX and HEAR cores have a few local and shared memories available to them. The two cores process data brought into system memory by the Input/Output Controller (IOC). The IOC can handle inputs from either an analogue input stage or PCM interface input and can handle outputs to a digital direct-drive output stage or PCM interface output. The FIFO controller simplifies access to data by the CFX DSP, the HEAR, and the IOC by providing configurable hardware-based FIFO buffers. Figure 8 illustrates the Belasigna 300 system architecture.

*6.2. Use of System-Level Parallelism on Speex Implementation.* Although BelaSigna 300 includes a powerful coprocessor (HEAR), we decided to refrain from extensive use of it due to the following two factors:

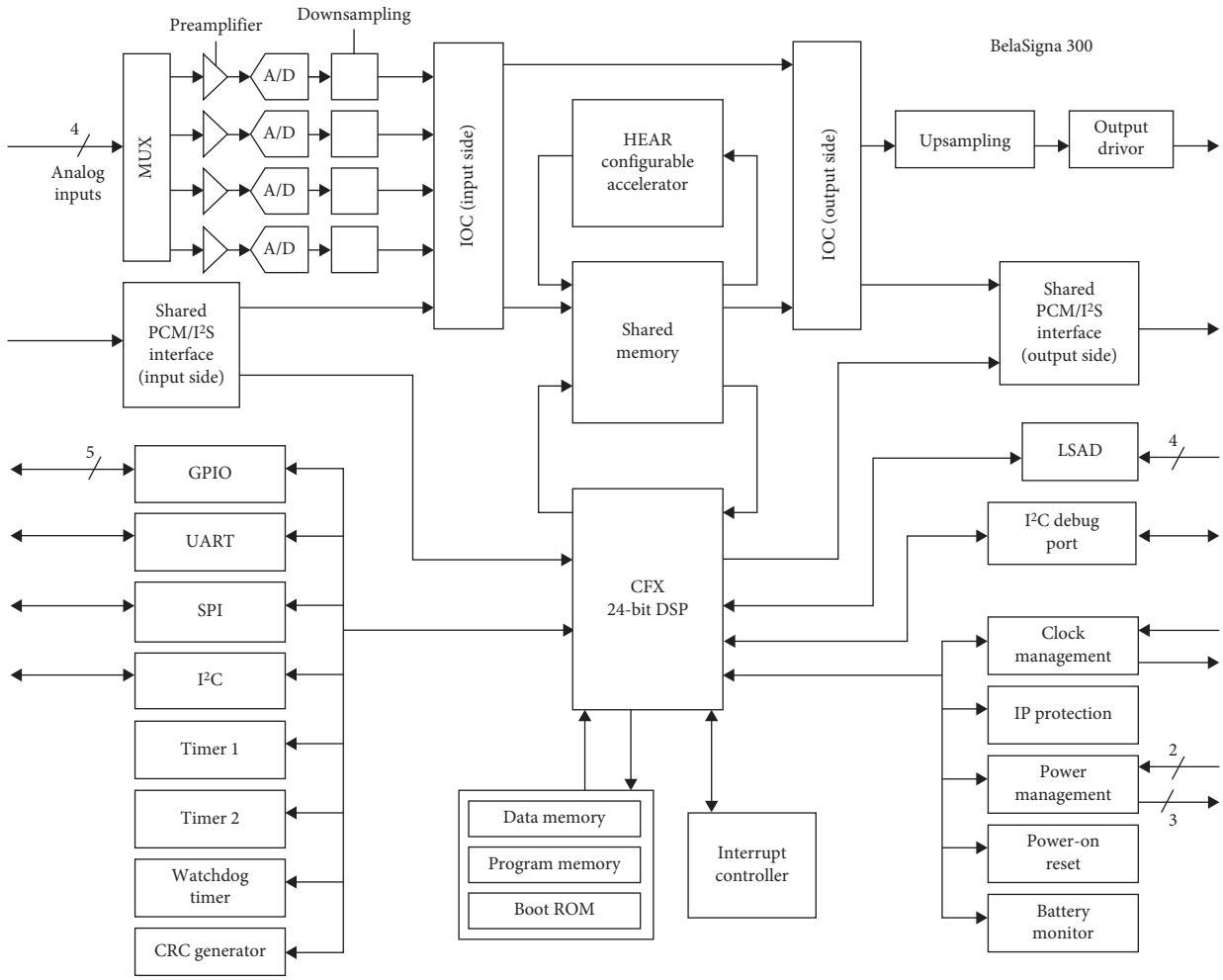



Figure 8: Belasigna 300 system architecture.

(i) The differences of its rounding mechanism compared to the rounding needed by Speex and the fact that Speex uses operations between operands with different accuracy would possibly require adaptation of the algorithm (e.g., recalculating filter coefficients with different accuracies).

(ii) Its dedicated memory for input and output data would require frequent data transfers. This would have a negative impact on performance and development time since this approach could prove to be error-prone.

Considering the development time requirements of the project, we decided to use the HEAR coprocessor only for a few basic vector operations.

All processing is performed within 20 ms, which is the frame duration used by Speex. At any given time, 4 frames are being processed: one frame being captured by the input stage, one frame being encoded, one frame being decoded, and one decoded frame being reproduced by the output stage. Encoded frames are exchanged between the DSP engine and the Zigbee interface module through the I2C controller. On system level, the following tasks are performed in parallel within a 20 ms time slot:

(1) Analogue-to-digital conversion and storing of the samples on input FIFO buffer by the IOC input side.

(2) Reading frame from input FIFO, frame encoding, frame decoding, and possible tone generation by the DSP. This is followed by mixing the two outputs of the previous step and storing the final output on the output FIFO buffer, using the HEAR coprocessor.

(3) Digital-to-analogue conversion of the previously decoded frame that is stored in the output FIFO by the IOC output side.

(4) Encoded frame exchange with the Zigbee interface module through the I2C interface.

*6.3. Instruction-Level Parallelism.* The Speex encoder and decoder are fully implemented on the CFX DSP. Extensive instruction-level parallelism is possible, in a very compact format. The DSP can execute up to four computation operations in parallel with two data transfers [33]. This is achieved by the different execution units that utilize the multiple bus and memory architecture and the dual data path (X and Y) as it can be seen in Figure 9. The execution units are as follows:

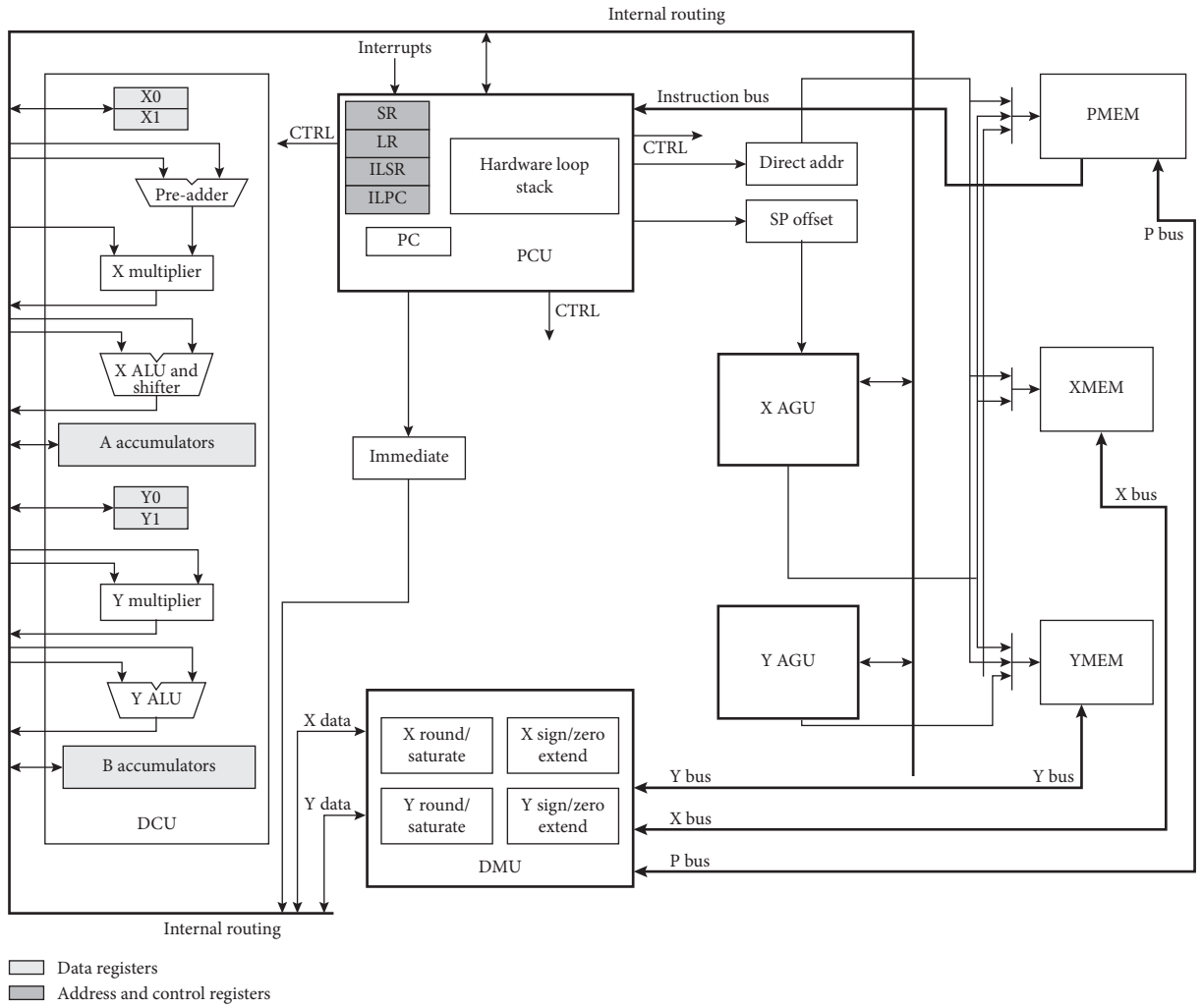



Figure 9: CFX DSP core architecture.

(i) The Data Computation Unit (DCU) that contains 2 ALUs and 2 multipliers for the two data paths.

(ii) The Data Movement Unit (DMU) that handles data transfers between registers and memory.

(iii) The X and Y Address Generation Units (AGUs) that provide addresses for indirect memory accesses, generating up to two memory addresses and updating up to two address registers simultaneously. The X-AGU is used to access X, Y, and P memory while Y-AGU is used to access Y and P memory.

(iv) The Program Control Unit (PCU) that fetches, decodes, and executes instructions in a 3-stage pipeline.

On instruction level, CFX has three types of instructions.

*Long Instructions*. These instructions perform only one operation in one clock cycle.

*Arithmetic-Move Instructions*. These instructions contain two parts: an arithmetic part that is executed by the DCU and a data movement part that is executed by the DMU and AGUs. Syntactically, the two parts are separated by "|||" indicating that they are executed in parallel, so an abstract Arithmetic-Move instruction is written like this,

<arithmetic part> ||| <move part>

The arithmetic and move parts can be further separated into dual arithmetic and dual move instructions operating on the X and Y data path, respectively, which means that an arithmetic-move instruction can execute up to 4 arithmetic and move operations in one cycle, having a form like this,

<X arith op> !! <Y arith op> ||| <X move part> !! <Y move part>

The move instructions can additionally perform within one cycle a load/store operation and pointer register arithmetic. This means that arithmetic-move instructions can execute in parallel up to 6 operations. An example of such an instruction is the following:

and a0, a0, b0 !! sub b1, b1, a1 ||| load x0, x[xp1++] !! load y0, y[yp2++]

*Move-Move Instructions*. These instructions are a way of compacting two move operations into one instruction,

providing instruction memory space optimization. The two operations are not performed in parallel but serially. Their syntax is like this,

<first move part> >>> <second move part>

The Z-Phone Belasigna 300 Speex implementation contains 4975 CFX assembly instructions. The distribution among the different types is shown in Table 5.

Arithmetic-move instructions can perform from 1 up to 6 operations per cycle. Practically, it is very challenging if not impossible to use this feature at its full potential. Our Speex implementation has used arithmetic-move instructions varying from 1 up to 5 operations per cycle. Obviously, the performance gain is more significant for instructions (and their operations) inside a loop. Table 6 presents the distribution of the arithmetic-move instructions used in the Speex implementation, based on their loop level and also their operations per cycle count.

As mentioned before, the move-move instructions consist of two move instructions that execute in 2 clock cycles. However, each of these move instructions may perform one additional pointer register arithmetic operation, improving this way their efficiency. The distribution of move-move instructions based on their loop level and the number of operations per 2 clock cycles is shown in Table 7.

Because only of the implemented instruction-level parallelism the codec execution time (measured in clock cycles) is reduced by one-third on average in all test cases using several different test speech samples and profiling data, as opposed to the total number of individual Speex Belasigna 300 operations and equal number of clock cycles in a serial execution. This benefit is added to the speed-up achieved already through system-level parallelism. Performance benefit through instruction-level parallelism can increase considerably, if our Speex Belasigna CFX assembly code is carefully re-engineered to exploit better the CFX parallelism potential. The adoption of parallel programming principles for task decomposition involving the identification of tasks that can be done concurrently (linear, iterative, and recursive task decomposition), data structures (input/output/intermediate) or parts of them that can be managed at the same time, and the dependencies that impose ordering constrains (synchronization) and data sharing can help maximize the speed-up of code execution through maximizing concurrency and minimizing parallelization overheads. Illustrating the importance of instruction-level parallelism, if it had not been implemented in the codec port, the execution time plots of Figure 5 would have raised along the $y$-axis above the threshold line (marginally for complexity value 1 in the 8 kbps mode), jeopardizing the feasibility of the application. Besides the gains through instruction-level parallelism, the codec execution time can be further reduced through effective use of the HEAR co-processor/accelerator.

Table 5: Distribution of Belasigna 300 Speex instructions.

| Single | Arithmetic-move | Move-move |
|---|---|---|
| 1899 | 2968 | 108 |

## 7. Conclusions and Future Work

The Zigbee standard is designed to enable the deployment of low-cost, low-power wireless sensor and control networks based on the IEEE 802.15.4 physical radio standard. Despite the low data rates of Zigbee, its use for transmission of voice is feasible. In this paper, we prove that robust and good quality conversations can be implemented over Zigbee. In fact, the radio channel of Zigbee has enough bandwidth to support a full-duplex conversation with narrow-band voice codecs.

We have discussed various aspects of the selection process for a voice codec for high-quality voice transmission over Zigbee. The initial "obvious" choice for a good quality codec around 8 kbps was a G.729 flavour. When we realized the cost implications of such a choice on a commercial product, we looked into the open-source community and discovered a wide selection of high-quality solutions. The results of the porting effort are presented showing that it is feasible to use a narrow-band codec in a real-environment embedded application with strict low-power requirements and bandwidth limitations such as a wireless headset based on Zigbee. We managed to achieve real ultralow-power operation, while including all the necessary analogue interfaces which proved to be also a huge benefit. Finally, we briefly presented the Z-Phone headset unit and the results of the transmission range and power consumption tests involving the developed embedded system. This is one of the very few expert engineering efforts known in the literature achieving a successful porting of a (royalty-free) speech codec on an ultralow-power DSP enabling voice communications in restricted ultralow-power devices and networks.

Future work will focus on significant project extensions concerning response time and robustness in front of interference regarding the use of the system as a headset for voice communications, as well as towards the migration of the developed technology in other applications, such as Voice Directed Warehousing (VDW) and in the promising and challenging domain of Wireless Sensor Network (WSN) applications.

Regarding the first point, response time, we intend to demonstrate that voice communication over Zigbee can be held using a repeater node. This would provide a coverage range significantly bigger when compared with Bluetooth headsets and double the current achievement.

Regarding the second point, robustness in front of interferences, we intend to take into account that many users will be using the system, even simultaneously, and the environment (e.g., an office) will be changing. The objective of this task will be to develop a method for effective use of the available channels and ability to react to interferences that might appear during a voice conversation while ensuring that the number of simultaneous conversations that can be held in the same space (office) is high enough to be used in a realistic environment. The number of available Zigbee channels, 11, will be sufficient for supporting a sufficient number of users.

Table 6: Distribution of Speex arithmetic-move instructions based on loop level and number of operations per cycle (1–5).

| | Loop level 0 | | | | | Loop level 1 | | | | | Loop level 2 | | | | | Loop level 3 | | | | | Loop level 4 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1115 | | | | | 1294 | | | | | 441 | | | | | 114 | | | | | 4 | | | | |
| Ops per cycle | **1** | **2** | **3** | **4** | **5** | **1** | **2** | **3** | **4** | **5** | **1** | **2** | **3** | **4** | **5** | **1** | **2** | **3** | **4** | **5** | **1** | **2** | **3** | **4** | **5** |
| | 724 | 287 | 91 | 9 | 4 | 705 | 371 | 198 | 12 | 8 | 212 | 147 | 70 | 10 | 2 | 70 | 33 | 7 | 4 | 0 | 2 | 1 | 1 | 0 | 0 |

Table 7: Distribution of Speex move-move instructions based on loop level and number of operations per 2 clock cycles.

| Loop level 0 | | | Loop level 1 | | | Loop level 2 | | | Loop level 3 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 42 | | | 39 | | | 18 | | | 9 | | |
| 2 ops per 2 cycles | 3 ops per 2 cycles | 4 ops per 2 cycles | 2 ops per 2 cycles | 3 ops per 2 cycles | 4 ops per 2 cycles | 2 ops per 2 cycles | 3 ops per 2 cycles | 4 ops per 2 cycles | 2 ops per 2 cycles | 3 ops per 2 cycles | 4 ops per 2 cycles |
| 0 | 2 | 40 | 5 | 6 | 28 | 2 | 3 | 13 | 1 | 0 | 8 |

Regarding the third point, the integration of the Z-Phone system in VDW makes it necessary considering a different use scenario in which, although the core of the headset for VoIP system is valid, some additional characteristics must be taken into account, such as audio quality, dynamic routing procedures, conversation profile, channel sharing to increase number of users, and delay tolerated. In the broader WSN/VoSN (Voice over Sensor Network) scenario, voice streaming will need to address 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. Several metrics of multihop communication such as throughput, jitter, latency, and packet loss will have to be measured and analyzed.

Last but not least, a variable bit-rate codec implementation will be tested, using the highly-scalable novel open codec Opus (bit rates from 6 Kbps to 510 Kbps, sampling rates from 8 KHz to 48 KHz and frame sizes between 2.5 ms and 20 ms). Opus combines technology from Speex and Skype's SILK and is the first open-source codec which has become an IETF standard, unlike other open-source codecs, such as Speex targeting speech and Vorbis, which is aimed at comparing music and audio in general. Towards this end, the objective will be to perform a deep study of the different possible configurations of Opus, to implement it in a DSP system and to effectively improve the current MOS score for the platform developed in the Z-Phone project.

## Data Availability

Reproducing the findings presented in the paper by a third party would require the on-hand availability of the Zigbee system code, besides the target platform, including the implementation of the VoZ transmission mechanism as well as the Speex assembly code implementation on the Belasigna 300 DSP. The private organisations involved in the Z-Phone project, namely, Ateknea Solutions and inAccess Networks, would not allow disclosing the system code and depositing it in a publicly available data repository.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work has been partially supported by the EU Eurostars Z-Phone research project under grant agreement EU-08-4302. The publication of this paper has been partly supported by the University of Piraeus Research Center. The authors would like to especially thank Mr. Albert Serrat, member of the Ateknea project team, for his concentrated know-how and technical contribution towards the development of the VoZ transmission mechanism.

## References

[1] C. Wang, K. Sohraby, R. Jana, L. Ji, and M. Daneshmand, "Voice communications over zigbee networks," *IEEE Communications Magazine*, vol. 46, no. 1, pp. 121–127, 2008.

[2] E. Choi, Y. Hur, J. Huh, Y. Nam, D. Yoo, and W. Choi, "Simulation and implementation of voice-over-IEEE 802.15.4 LR-WPAN," in *Proceedings of the International Conference on Consumer Electronics (ICCE 2008)*, Hoi An City, Vietnam, January 2008.

[3] L. Y. Hua and F. F. Teng, "Delivering high quality, secure speech communication through low data rate 802.15.4 WPAN," in *Proceedings of the IEEE International Conference on Telecommunications and Malaysia International Conference on Communications (ICT-MICC 2007)*, Malaysia, 2007.

[4] R. Mangharam, A. Rowe, R. Rajkumar, and R. Suzuki, "Voice over sensor networks," in *Proceedings of the 27th IEEE International Real-Time Systems Symposium*, Rio de Janeiro, Brazil, December 2006.

[5] *Eurostars Z-Phone Project: Novel, Low-Power Consumption VoIP Headset Based on ZigBee Technology*, https://www.eurostars-eureka.eu/project/id/4302.

[6] IEEE 802.15.4-2006, IEEE standard for information technology—local and metropolitan area networks—specific requirements, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs), 2006, https://standards.ieee.org/standard/802_15_4-2006.html.

[7] E. Touloupis, A. Meliones, and S. Apostolacos, "Implementation and evaluation of a voice codec for ZigBee," in *Proceedings of the 16th IEEE International Symposium on Computers and Communications*, Kerkyra, Greece, June-July 2011.

[8] Zigbee Specification, September 2012, Zigbee Alliance, http://www.zigbee.org/wp-content/uploads/2014/11/docs-05-3474-20-0csg-zigbee-specification.pdf.

[9] E. Touloupis, A. Meliones, and S. Apostolacos, "Speech codecs for high-quality voice over ZigBee applications: evaluation and implementation challenges," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 122–128, 2012.

[10] D. S. Sunder and R. K. Kushwaha, "Evaluation of narrow band speech codecs for ubiquitous speech collection and analysis systems," in *Proceedings of 2015 International Conference on Industrial Instrumentation and Control (ICIC)*, Pune, India, May 2015.

[11] K. S. Raju and A. Sharma, "Comparison of two speech communication codecs for transmitting voice/speech over Zigbee," in *Proceedings of 2015 International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 685–690, Noida, India, February 2015.

[12] M. Gentili, R. Sannino, and M. Petracca, "Bluevoice: voice communications over bluetooth low energy in the internet of things scenario," *Computer Communications*, vol. 89-90, pp. 51–59, 2016.

[13] A. Prabahar, "Development of high performance wireless sensor node for Acoustic applications," in *Proceedings of 2013 IEEE International Conference on Green High Performance Computing (ICGHPC)*, Piscataway, NJ, USA, March 2013.

[14] L.-H. Chang, C.-H. Chang, H. F. Chang, and T.-H. Lee, "Implementation and evaluation of multi-hopping voice transmission over ZigBee networks, advanced technologies, embedded and multimedia for human-centric computing," *Lecture Notes in Electrical Engineering*, vol. 260, pp. 1195–1204, 2014.

[15] L. Meiqin, W. Yuxuan, F. Zhen, and Z. Senlin, "Voice communication based on ZigBee wireless sensor networks," in *Proceedings of 2014 33rd Chinese Control Conference (CCC)*, Nanjing, China, July 2014.

[16] Y. Yang, W. Li, H. Li, and J. Zhu, "HD2UB: a voice communication system for underground mine monitoring," in *Proceedings of CENet2015*, Shanghai, China, September 2015.

[17] Y. Fu, Q. Guo, and C. Chen, "A-LNT: A wireless sensor network platform for low-power real-time voice communications," *Journal of Electrical and Computer Engineering*, vol. 2014, Article ID 394376, 19 pages, 2014.

[18] L.-H. Chang, C.-C. Chen, and T.-H. Lee, "Voice transmission over wireless sensor networks," in *Proceedings of International Conference on Ubiquitous Computing and Multimedia Applications*, vol. 7, pp. 158–163, Bali, Indonesia, June 2012.

[19] D. H. Wang, Q. Q. Zhang, and Y. F. Sun, "Design of wireless voice communication system in underground coal mine based on ZigBee," *Applied Mechanics and Materials*, vol. 548-549, pp. 1402–1406, 2014.

[20] NASA, "Report concerning space data segment standards: wireless network communications overview for space missions operations," CCSDS 880.0-G-0.169, Green Book, Consultive Committee for Space Data Systems, NASA, 2009.

[21] C. Pham, P. Cousin, and A. Carer, "Real-time on-demand multi-hop audio streaming with low-resource sensor motes," in *Proceedings of 2014 IEEE 39th Conference on Local Computer Networks Workshops (LCN Workshops)*, Edmonton, Canada, September 2014.

[22] C. Pham and P. Cousin, "Benchmarking low-resource device test-beds for real-time acoustic data," in *Testbeds and Research Infrastructure: Development of Networks and Communities, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 137, pp. 97–106, 2011.

[23] A. Koucheryavy, A. Vladyko, and R. Kirichek, "State of the art and research challenges for public flying ubiquitous sensor networks," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems, Lecture Notes in Computer Science*, vol. 9247, pp. 299–308, 2015.

[24] R. Kirichek, M. Makolkina, J. Sene, and V. Takhtuev, "Estimation quality parameters of transferring image and voice data over Zigbee in transparent mode," in *Procceedings of 18th International Conference, DCCN 2015*, Moscow, Russia, October 2015.

[25] ON Semiconductor, *BelaSigna 300 Data Sheet*, Phoenix, AZ, USA, 2008.

[26] ST Microelectronics, Application Note AN2812, *Vocoder Demonstration Using a Speex Audio Codec on STM32F101xx and STM32F103xx Microcontrollers*, ST Microelectronics, Geneva, Switzerland, 2008.

[27] Microchip, *dsPIC® DSC Speex Speech Encoding/Decoding Library User's Guide*, Chandler, AZ, USA, 2008.

[28] A. Ramo and H. Toukomaa, "On comparing speech quality of various narrow and wideband speech codecs," in *Proceedings of 8th International Symposium on Signal Processing and its Applications*, Sydney, Australia, August 2005.

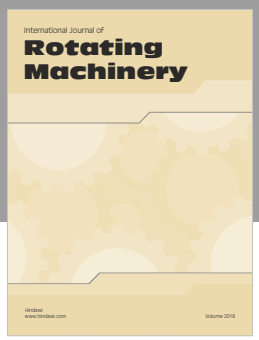
[29] ITU-T P.862.3 Recommendation, *Application Guide for Objective Quality Measurement Based on Recommendations P.862, P.862.1 and P.862.2*, 2007.

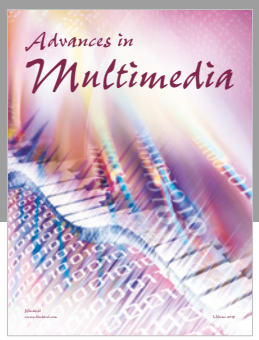
[30] A. Meliones, E. Touloupis, J. Perello, and A. Serrat, "Z-Phone: Design and implementation of embedded voice over Zigbee applications," in *Proceedings of 2014 IEEE Symposium on Computers and Communications (ISCC)*, Funchal, Portugal, June 2014.

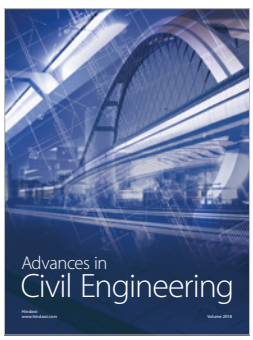
[31] J. Linsky, "Bluetooth and power consumption: issues and answers," *RF Design*, vol. 24, no. 11, pp. 74–79, 2001.

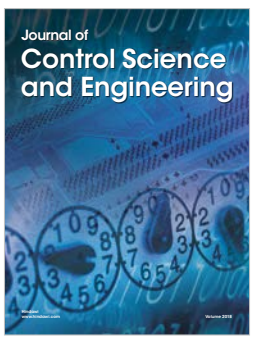
[32] R. Kapoor, L. J. Chen, Y.-Z. Lee, and M. Gerla, "Bluetooth: carrying voice over ACL links," in *Proceedings of 4th International Workshop on Mobile and Wireless Communications Network*, Bombay, India, September 2002.

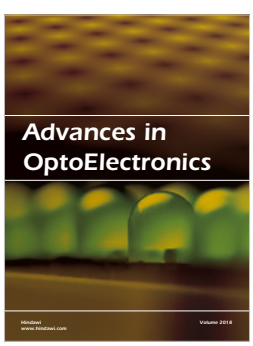
[33] ON Semiconductor, *CFX DSP Architecture Manual*, ON Semiconductor, Phoenix, AZ, USA, 2009.

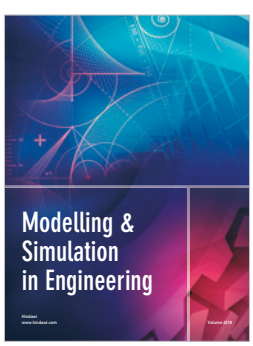


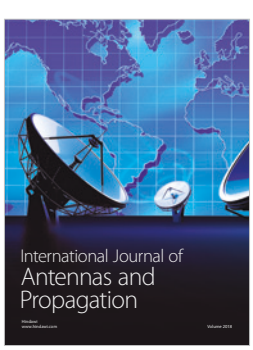


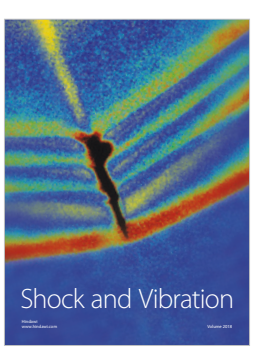


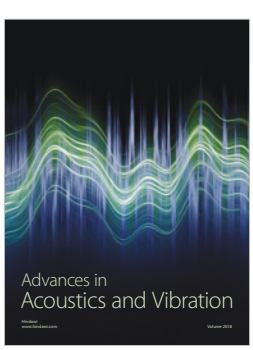












Hindawi

Submit your manuscripts at
www.hindawi.com