IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 16, 2020, accepted June 20, 2020, date of publication June 23, 2020, date of current version July 2, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004480

An SDR-Based Satellite Gateway for Internet of Remote Things (IoRT) Applications

CRISTINEL GAVRILĂ[1], VLAD POPESCU[1], (Member, IEEE), MARIAN ALEXANDRU[1], MAURIZIO MURRONI[2,4], (Senior Member, IEEE), AND CLAUDIO SACCHI[3,4], (Senior Member, IEEE)

[1]Department of Electronics and Computers (DEC), Transilvania University of Brasov, 500019 Braşov, Romania
[2]Department of Electrical and Electronic Engineering (DIEE), University of Cagliari, 09123 Cagliari, Italy
[3]Department of Information Engineering and Computer Science (DISI), University of Trento, 38123 Trento, Italy
[4]Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), 43124 Parma, Italy

Corresponding author: Vlad Popescu (vlad.popescu@unitbv.ro)

This work was supported in part by the Romanian Ministry of National Education and Scientific Research, RDI Programme for Space Technology and Advanced Research (STAR), under Project STAR CTR 147, and in part by the Autonomous Region of Sardinia-Italy (POR Marinanow 3.0 Research Project and Piano Sulcis Research Grant SULCIS-821101).

ABSTRACT Internet-of-Things (IoT) represents a breakthrough for the current ICT market. In many IoT applications, sensors and actuators are distributed over very wide areas, sometimes not reached by terrestrial networks. In such scenarios, the satellite plays a significant role. In this paper, a Software-Defined Radio (SDR) - based satellite gateway for Internet-of-Remote-Things (IoRT) is proposed. The use of SDR allows to decrease equipment cost and provides higher flexibility. The proposed architecture has been implemented by using a standalone SDR platform and Commercial Off-The-Shelf (COTS) modules for covering the main terrestrial IoT standards. Extensive proof-of-concept results are presented and discussed. Uplink and downlink tests showed the correct functionality implementation and transmitted signal generation, while the integration tests allowed to assess the reliability of the end-to-end information processing. Reconfigurability tests confirmed the capability of the gateway of dynamically updating in real-time its protocol settings. The overall test results showed the validity of the proposed SDR-based gateway for IoRT applications.

INDEX TERMS Internet of Remote Things (IoRT), satellite networking, software-defined radio (SDR).

I. INTRODUCTION

In these last ten years, a second revolution in Internet technology has grown around the concept of Internet of Things (IoT) [1]. In the past decades, Internet had become the paradigm of the ''global network of the people'', where the information content has always been generated by the intellectual efforts of Internet users and driven by their communication needs. IoT is based on a totally different paradigm: in the IoT, instead of manned content producers, smart objects can autonomously process and communicate data and interact among themselves and with the surrounding environment. The application fields of such a revolutionary technology are very wide and range from public safety, intelligent transportation systems, energy management, healthcare, industrial automation, to domotics, retail, etc. As stated in [2], satellite communications may potentially play an important role in IoT. Indeed, smart objects are often remote or dispersed over wide geographical areas or they are inaccessible. Such a situation is referred in [2] as Internet of *Remote* Things (IoRT). In this framework, satellite communication could offer a broadband cost-effective solution for the interconnection of networks of smart objects with the rest of the world. A valuable example is shown in [3], where IoRT supported by satellite backhaul is studied in the framework of video transmission from remote geographical areas involved in emergency situations. In another work [4], the use of two most common Machine-to-Machine (M2M) and IoT protocol stacks, namely: Message Queueing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP) is tested on a satellite random channel, based on the DVB-RCS2 standard.

In this paper, we tackle the issues of satellite connectivity for IoRT from a different viewpoint. The variety of communication standards, which are currently used for IoT and M2M applications, makes it very difficult to have a single device able to act as gateway. On the satellite side, the hardware cost of a gateway could be significantly lowered considering

The associate editor coordinating the review of this manuscript and approving it for publication was Faissal El Bouanani.

a SDR implementation. From the IoRT system side, the SDR implementation of the gateway can provide a high degree of flexibility in dynamically operating multiple wireless interfaces, especially combined with the plethora of terrestrial communication standards.

The main goal of the paper is to present and assess the hardware implementation of the SDR-based gateway for IoRT, along with its operational characteristics. The analysis of the proposed hardware/software architecture is performed at ''proof-of-concept'' level, without open-field testing that would require the presence of an operational satellite link.

The proposed and tested SDR gateway can find application in heterogeneous distributed IoRT scenarios, where a plethora of sparse devices, transmitting information using different protocols and standards over wide areas, is considered. At the best of our knowledge, the state-of-the-art SDR-based solutions for IoT do not tackle such kind of long-range and sparse multi-standard scenarios, being mainly focused on providing connectivity among devices operating in dense areas of limited extension.

The paper can be regarded as an evolution of three prior papers, in particular [5], [6], and [7]. These papers presented in sequence: the initial proposed architecture [5], the implementation of the terrestrial interface [6] and part of the implementation of the satellite interface [7]. This work presents, together with the complete implementation of the IoRT gateway, the performed tests for the terrestrial and the satellite interfaces, the overall tests and the final results of the entire project.

The paper is structured as follows: section II will discuss the state-of-the-art framework and highlight the novel contribution provided by our work. Section III will describe the architecture of the proposed SDR-based IoRT gateway, while Section IV will be devoted at illustrating the hardware/software implementation of the gateway itself. In-lab testing and proof-of-concept results will be shown in Section V. Finally, Section VI will draw the conclusions.

II. BACKGROUND AND INNOVATION

As shown in Fig. 1, the present work can be regarded as the intersection of a threefold state-of-the-art framework that can be summarized as: M2M/IoT communications over satellite links, SDR for satellite communications, and SDR for IoT.

A. M2M/IoT COMMUNICATIONS OVER SATELLITE LINKS

Some recent publications have dealt with the design of satellite systems for IoT. In [8], Qu *et. al* propose an architecture of a Low Earth Orbit (LEO) satellite constellation-based IoT network, discussing, among the others, aspects of efficient spectrum allocation, heterogeneous networks compatibility and routing protocols. In such a perspective, the authors of [9] showed that the Iridium Short Burst Data (SBD) network is one of the technologies best-suited for long-range IoT communications, with the implementation and testing of a hybrid Iridium-LoraWAN prototype. Another work dealing with transmission aspects for IoT-satellite connections is [10], where ultra-narrowband waveforms, based on Chirp Spread Spectrum (CSS) are designed for direct access of IoT to GEO satellites. In some recent contributions, performance evaluation and optimization of the IoT application protocols in the perspective of future integration of M2M with satellite networks are analyzed in detail [11], [12]. TCP performance of satellite M2M applications is investigated in [13] for random access links, when Contention Resolution Slotted (CRS) ALOHA protocol is used for managing the multiple access. In order to increase the spectral efficiency of satellite-based IoT systems, a novel forward link multiplexed scheme is presented in [14], where the signals of different users can be simultaneously transmitted over the satellite link using the same frequency by means of constellation coding. Finally, an interesting application of satellite-based IoRT is shown in [15], where the IoT sensors are located inside a flying aircraft and a satellite provides the backhaul both with aircraft manufacturers and airline operators.

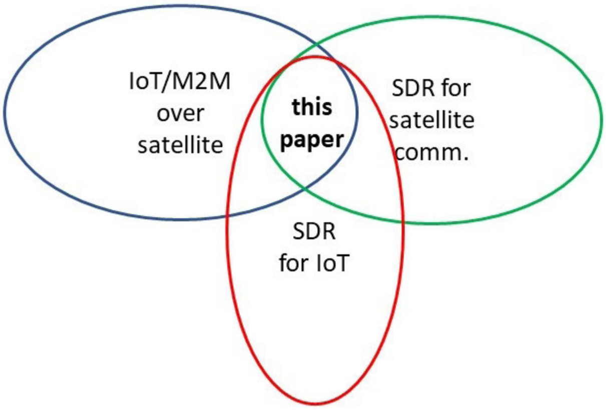


FIGURE 1. State-of-the-art framework of the present paper.

B. SDR FOR SATELLITE COMMUNICATIONS

Due to their flexibility, SDRs can provide a major contribution to the improvement of satellite communications. Through the use of Field Programmable Gate Arrays (FPGAs) and other dedicated processors, SDRs offer great versatility, that enables them to be used with multiple bands and modulation techniques, without requiring any change to hardware components. In such a framework, Digdarsini *et. al.* present in [16] a FPGA-based automatic modulation recognizer that uses the wavelet transform for identifying PSK, FSK and QAM signals. The authors of [17] have implemented a satellite negotiator devoted at ensuring the interoperability between different satellite missions, characterized by different transmission protocols. Another work worth of citation is [18], where an SDR-based reprogrammable Very Small Aperture Terminal (VSAT) terminal is proposed in order to restore multi-modal connectivity in emergency situations. Many recent works consider small satellites and CubeSats as valuable space platforms for the satellite SDR technology deployment. For example, the authors of [19] explore the SDR architecture in order to support the detection of multiple signals coming from multiple CubeSats in parallel. In other related works, the main focus is on the use of very low-cost COTS hardware components and open source software tools in order to enable satellite experiments also for students and researchers working in small labs [20], [21].

The SDR technology can be used for designing and implementing ground stations or user terminals. In [22] a multimodal VSAT unit capable of communicating both through LTE or Wireless LAN (WLAN) terrestrial links, as well as through satellite links is presented. A similar hybrid radio access system (i.e. a system that provides both satellite and terrestrial connectivity) is showcased in [23]. It consists of combining terrestrial Long Term Evolution (LTE) networks used in urban areas with co-channel satellite LTE cells for low populated areas.

C. SDR FOR IoT

The literature about SDR for IoT applications has become richer and richer in these last years. Some examples of related works are reported in the following. In [24], Schadhauser, Robert and Hueberger propose an SDR-based autonomous base station for low-power wide area sensor networks. Ma, Zeng and Sun proposed in [25] a multi-function radar for human body detection in human-centric IoT applications. Software radio is used in [26] in order to implement and test an IEEE 802.15.4k transceiver for low-energy critical infrastructure monitoring. In the same framework, it is worth citing [27], where the standard LTE-based R13 downlink transmission for IoT has been implemented and tested with SDR and USRP boards. Smartphones with SDR capabilities are considered in [28] to overcome the IoT language barriers, while a low-power basedband processor is designed in [29] to effectively support SDR usage in IoT applications. The security aspects of device-to-device communications are considered in [30], where a SDR platform is used to implement a cryptographic mechanism. Another work related to IoT security is [31], where SDR is used to assess the vulnerability in IoT devices to RF jamming attacks and to implement appropriate mitigation strategies. Other works deal with the combination of SDR techniques with Network Function Virtualization (NFV) in order to support multiple programmable air interfaces on top of one radio frequency (RF) front-end [32] and to provide the abstraction of the entire IoT stack to heterogeneous, generic, IoT devices [33].

D. NOVEL CONTRIBUTION OF THE PAPER

Considering the aforesaid state-of-the-art framework, the present work contains some significant innovation points. First of all, the published contributions about M2M/IoT communications on satellite links mostly concern with testing and optimization of terrestrial M2M/IoT protocol stacks in satellite-based IoRT configurations and/or on the study of efficient PHY-layer transmission solutions for direct connection of IoT devices with satellites. At the best of our knowledge, the study and the development of a multi-standard satellite gateway has not yet been addressed. Furthermore, satellite-based SDR applications are mostly focused on the deployment of re-programmable payloads, multi-modal ground stations and terminals and re-configurable links, but never considered the use of SDR for implementing a re-programmable multi-modal satellite gateway for IoRT. In the field of SDR for IoT, the examples found in the literature are focused on software implementation and testing of IoT transmission standards, analysis and mitigation of security threats, implementation of specific signal processing tasks, and, in few cases, implementation of autonomous terrestrial gateways, like in [24], but never on the design, implementation and test of re-configurable satellite infrastructures for IoRT. For this reason, we consider our work a clear step-ahead with respect to the state-of-the-art. In the last years, more IoT and M2M devices already have means to deliver directly IP or MQTT data, but especially for the IoRT field these options are not feasible, and even if they were, there is still the need of the satellite interface.

Also, the state-of-the-art analysis revealed that even though SDR-based solutions are starting to make their way into the market, there are not mature satellite gateway products capable of providing support for the main protocols currently used in M2M and IoRT applications. Hence, the originality of the present paper resides in the functionality of the proposed gateway, as a standalone embedded platform which can handle data from both the terrestrial and satellite sides.

Another novelty of the presented approach is the compactness of the platform, implemented on a low cost, battery-powered device. The use of an SDR-based approach ensures also the necessary flexibility for covering the future developments of IoT standards and applications, that could be implemented easily on the computational platform through the available reconfigurability mechanisms.

III. REQUIREMENTS FOR THE SDR-BASED IoRT GATEWAY

A. TERRESTRIAL INTERFACE

IoRT applications are based on low-power radio communications and need a gateway to connect with the external environment. The backbone of this type of systems is represented by data protocols and communication technologies capable to seamlessly integrate thousands of nodes, while ensuring a reliable data transfer. They operate at different levels of the protocol stack, the most important in this context being data link, network and session level (see table 1).

A key aspect for achieving the goal of covering most of the presented protocols and standards, is the use of a SDR platform as it provides both multi-protocol access for receiving data from different smart devices and the part of directing / transferring satellite traffic.

To ensure multi-protocol access for terrestrial interface of the gateway, two approaches are available:

- implementation of the IoRT protocols on the FPGA of the SDR platform;
- use of COTS modules.

TABLE 1. Different layer protocols used in IoT applications.

ISO Layer		Protocol standard
Session		MQTT, SMQTT, CoRE, DDS, XMPP, CoAP
Network	Encapsulation	6LowPAN, 6TiSCH, 6Lo, Thread
	Routing	RPL, CORPL, CARP
Data link		Wi-Fi, BLE, Z-Wave, ZigBee, 3G/LTE, NFC, LoRaWan

The first approach (implementation in FPGA) has the main advantage that it eliminates the need to use external devices, but significantly increases the complexity of implementation. Depending on the protocol that is intended to be implemented, the requirements regarding the FPGA's capabilities and performance can significantly increase, leading to a higher hardware cost for the SDR gateway. Also, another potential risk may be the inability of the SDR platform to host all the required protocols, due to the imposed constraints that directly affect the platform's performance. The second approach reduces significantly the complexity of implementation by using COTS modules for the desired protocol. The aspects to be addressed here are the interface with the SDR platform and also the additional hardware costs of the COTS modules themselves.

In the specific case of our gateway, based on the aforementioned considerations, combined with the subsequent analysis of the existing SDR platforms [6], the COTS implementation was preferred due to its extended flexibility, low cost and modularity.

B. SATELLITE INTERFACE

The IoRT networks are generally characterized by the existence of a large number of low-power, small sized terminals that share the resources available in very dynamic traffic conditions. In particular, on the return link (ground station to the satellite), users are expected to generate short, bursty traffic with extended periods of inactivity, therefore asynchronous random access schemes are being preferred, to the detriment of the high-cost synchronous ones. The absence of the synchronization mechanism simplifies the deployment and activation of the terminals. The key requirements for choosing an efficient access scheme are:

- high aggregate throughput with very low transmission probability of packets;
- effective support of small packages with very low transmission cycle;
- minimum signaling overhead;
- low EIRP (Effective Isotropic Radiated Power) needs;
- robustness against the energy imbalance of the received packets;
- accessible complexity.

Although the requirements listed above are particularly relevant for the intended satellite messaging system, they may also have general applicability for satellite narrowband mobile communications systems. An important step of our work was dedicated to the investigation and identification of the radio interfaces that best fit the IoRT message transmission needs.

1) UPLINK

The generic requirements of the uplink link for IoRT applications imply low data rates and an elevated number of nodes accessing the transmission resources in short bursts, leading to the necessity of the use of simple, uncoordinated and random access based medium access (MAC) protocols.

Starting from an initial comparative study, extended with our own considerations we concluded in [7] that the most suitable uplink radio interface for the use of M2M and IoT applications would be S-MIM (S - Band Interactive Mobile Multimedia). The E-SSA access mechanism offers the highest MAC throughput and robustness to power imbalance, while allowing for full-time asynchronous packet transmission over time and the lowest EIRP requirements.

2) DOWNLINK

For the downlink, based on our initial requirements analysis [5] the most suitable radio interface would be DVB-S2 [34]. Following the study of the specific standard, we came to the conclusion that DVB-S2X would be even more suitable to M2M/IoT message forwarding requirements [7]. DVB-S2X is an extension of DVB-S2 specifications that promotes additional technologies and features. DVB-S2X uses the LDPC (Low Density Parity Check) Forward Error Correction (FEC) scheme proven and strong in combination with Broadcast Control Channel (BCH) FEC as the external code but introduces additional features, which fulfill the M2M/IoT Forward link requirements [35].

C. GATEWAY ARCHITECTURE

The requirements identified in the previous sections, led us to the specific design of the gateway architecture. The main specific requirements to be satisfied by the architecture are: i) to support or to be able to embed M2M and IoT standards and communication technologies, and ii) to have an embedded architecture with overall low complexity, which can assure portability and flexibility together with reprogrammability and reconfigurability in the framework of a non-proprietary programming environment (Linux, GNU Radio). Also, the system has to ensure adequate FPGA capacity for running complex protocol stacks, large RF bandwidth, MIMO capabilities, interfacing capabilities (AXI, I2C, USB, IP) with external components, embedded GPS module or GPS interface capabilities, reduced power consumption and a low price tag. The envisioned architecture is depicted in Fig. 2 and is based on the computational platform that links the satellite communication with the IoRT one. The FPGA-based platform is also responsible for the implementation of the protocol stack of the most common communication standards

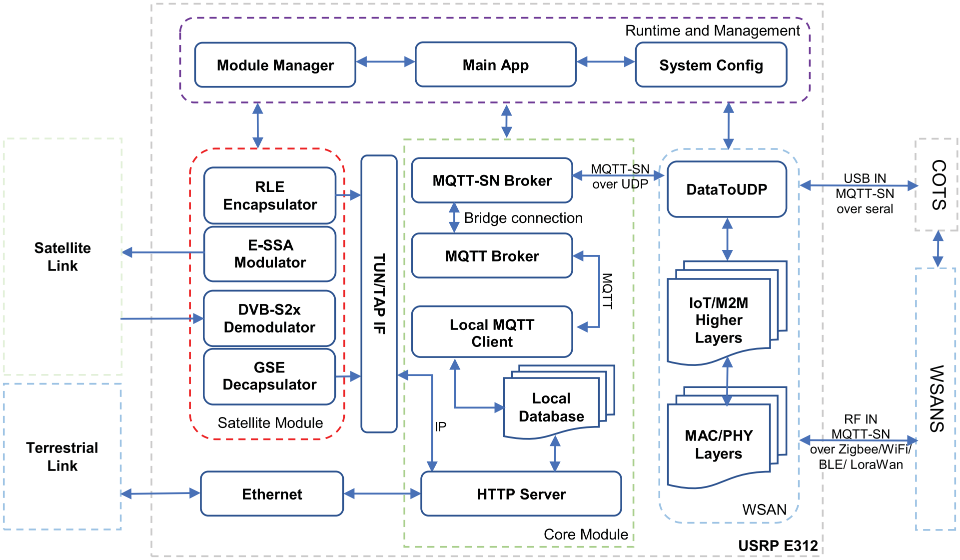


FIGURE 2. Architecture of the proposed IoRT gateway.

for IoT and M2M applications and for the general baseband – RF bidirectional conversion. On the software side, a modular design was adopted to ensure a high degree of reconfigurability and flexibility. Each main feature/functionality has a designated module which can be replaced or updated as needed either before deployment or remotely while the system is in the field.

The central part of the generic architecture is the MQTT broker, part of the Core Module, that acts as a universal transport mean between the terrestrial and the satellite side. The broker is connected to a local database and to a HTTP server. The data to be sent in the database is organized by communication protocol, node id and node type (e.g.temperature, humidity, etc.). All the data present in the database will be sent via the satellite interface to la cache server.

Towards the satellite interfaces (Satellite Module) the communication is done using IP packets, while towards the terrestrial side, inside the Wireless Sensor and Actuator Network (WSAN) module the packets are UDP. The proposed architecture leaves space for both the satellite and the terrestrial interfaces to be implemented as FPGA code or by using external COTS modules connected through the interfacing capabilities of the gateway.

The Runtime and Management module is responsible for the management and the update for all building blocks.

IV. SDR IMPLEMENTATION OF THE IoRT GATEWAY

A. SDR PLATFORM

In order to implement the envisioned architecture of the satellite gateway, several SDR platforms have been taken into consideration. Price, computational power, power consumption, use of non-proprietary development environments and the ability to run autonomously without the aid of an external computational platform were the main factors taken into consideration for the choice of the SDR platform. In [5] a thorough analysis was performed on the SDR platforms available on the market that are compliant with the previously drawn requirements. Based on this analysis, a portable, stand-alone SDR platform was chosen: the USRP E312 comes with a 2 × 2 MIMO transceiver able to cover bands from 70 MHz to 6 GHz, a Xilinx Zynq 7020 SoC, a dual-core ARM CPU and an Open Embedded Linux OS which offers the support and flexibility needed for custom software development. A 3200 mAh on-board rechargeable battery allows a high degree of autonomy, with more than 2 hours of battery life.

B. IMPLEMENTATION OF THE TERRESTRIAL INTERFACE

As interconnecting satellites with WSANs is best achieved via indirect access - each sensor and actuator communicates with the satellite through a gateway node – and given the fact that an interface which allows third party applications to collect sensor data via terrestrial WAN links is also an important requirement, the terrestrial side of the satellite gateway becomes a key architectural element which enables data pass through between short range wireless communication protocols and traditional terrestrial/satellite networks. For this, the gateway has to offer support for a rigorous set of requirements, the most important being: communication management, security and authentication, data aggregation, protocol translation and power management. As one of the main objectives to be achieved is the capability of communicating with M2M/IoT applications by using the most common wireless protocols and transmission standards, the SDR-based terrestrial gateway offers support for the following communication standards: LoRaWan, BLE, IEEE 802.11 and ZigBee IEEE 802.15.4.

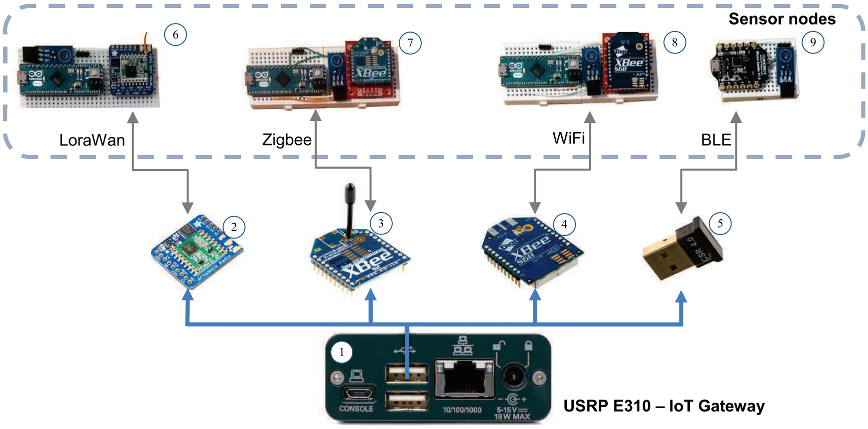


FIGURE 3. **Modular hardware scheme of the terrestrial side of the IoRT gateway.**

The hardware components adopted for implementing the terrestrial side of the satellite gateway are depicted in Fig. 3 and listed as follows:

- Xbee S6B Wi-Fi (figure 3 – item 4)
- Adafruit RFM95W LoRaWan (figure 3 – item 2)
- CSR 4.0 BLE (figure 3 – item 5)
- Xbee S2 Zigbee (figure 3 – item 3)

The sensor nodes layer consists of four nodes, each able to gather information from various types of sensors (temperature, humidity, pressure, gas etc.), send data to the gateway and perform actions received from the gateway (i.e. relay/actuator control). The LoRaWan, ZigBee and Wi-Fi sensor nodes consist of an Arduino Micro board, a LM50 temperature sensor and the corresponding COTS module. For the BLE sensor node a Beetle BLE board with an integrated BLE module was used. All the modules are connected via the USB ports of the SDR platform.

To ensure the end-to-end device-to-device or device-to-user communication between the satellite and terrestrial interfaces, the MQTT/MQTT-SN (Sensor Networks) protocol [29] has been used. This data protocol has been specifically designed for constrained environments requiring low bandwidth consumption and good performance. The gateway acts as a broker which classifies the data received from the publishers into topics and relays it onward to the interested subscribers, independent of the communication protocols involved. The sensor nodes act as publishers for passing sensor data to the gateway as well as subscribers for receiving commands from the gateway. The UDP protocol is used internally as a payload bridge between WSAN layer and the internal MQTT-SN broker. The packet recovery in case of lost/erroneous packets is handled at the WSAN protocol level and at the MQTT-SN level by properly selecting the quality of service (QoS). In case of the presented architecture, the QoS level is set to 1 which ensures that the sender stores the message until it gets a PUBACK packet from the receiver that acknowledges receipt of the message.

Furthermore, still using GNU Radio we have implemented a multi-band spectrum sensing procedure for detecting wireless activity in the RF bands specific for each considered short-range terrestrial communication standard. The results of the spectrum sensing procedure are returned to the user interface, in order to allow the user to decide for the use of one or more of the aforementioned standards. Both the software implementation of the protocol stacks and the sensing procedure are making use of the RF interface of the USRP E312 platform, capable to cover the terrestrial RF bands used for wireless sensor networks (i.e. 433 MHz, 860 MHz, 2.4 GHz). The user interface of the gateway [6] allows the coordination of an initial spectrum sensing and packet sniffing and the activation of one or more communication standards for the connection with the IoT end devices via the COTS modules.

C. IMPLEMENTATION OF THE SATELLITE INTERFACE

1) UPLINK

The implementation of the E-SSA modulator has been completed using the GNU Radio software package in the FPGA of the USRP E312 platform, using as starting point the block diagram of in Fig 4. In order to ensure the data transport service, the following functions have been implemented:

- error detection mechanisms on the transport channel and reporting to the higher levels;
- FEC (forward error correction) coding;
- matching the rate of the transport channel to the physical channel;
- transport channel mapping to the physical data channel;
- power allocation and combination of physical channels;
- modulation and spreading on the physical channels;
- time and frequency synchronization (bit, chip, burst);
- RF processing.

The Random Access Channel (RACH) is the only data transport channel, characterized by a limited length of the data burst. Two possible dimensions of the length of the RACH burst (total number of bits), namely 1920 and 3600 bits, are supported, matching the rate achieved after the FEC coding. There are two physical channels, the PDCH (Physical Data Channel) used for the RACH data burst and the PCCH (Physical Control Channel), used for physical level

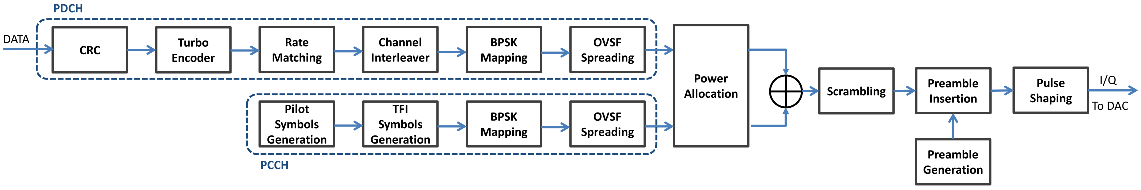
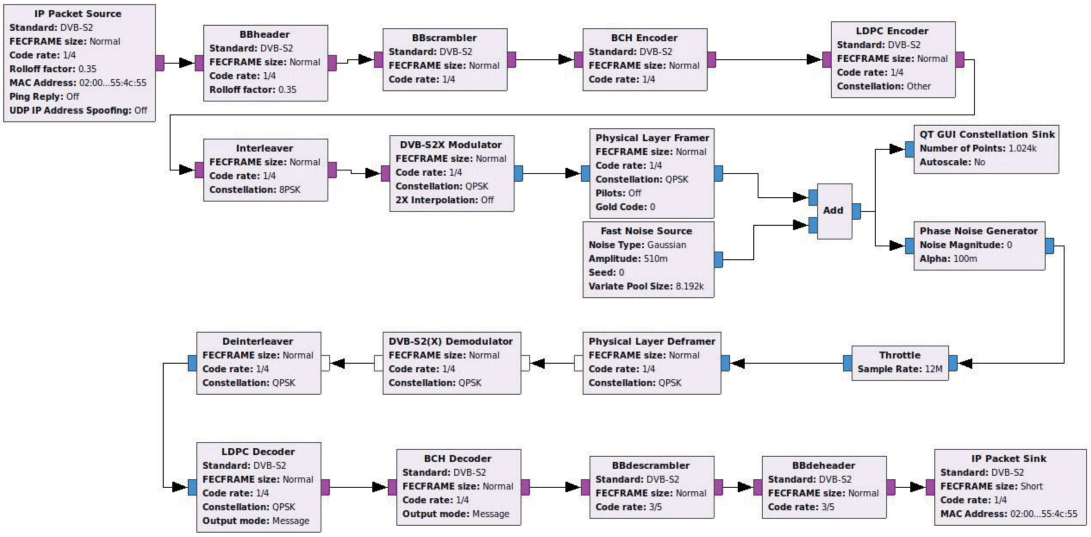


FIGURE 4. Block diagram of the E-SSA modulator.


IP Packet Source
Standard: DVB-S2
FECFRAME size: Normal
Code rate: 1/4
Rolloff factor: 0.35
MAC Address: 02:00...55:4c:55
Ping Reply: Off
UDP IP Address Spoofing: Off

BBheader
Standard: DVB-S2
FECFRAME size: Normal
Code rate: 1/4
Rolloff factor: 0.35

BBscrambler
Standard: DVB-S2
FECFRAME size: Normal
Code rate: 1/4

BCH Encoder
Standard: DVB-S2
FECFRAME size: Normal
Code rate: 1/4

LDPC Encoder
Standard: DVB-S2
FECFRAME size: Normal
Code rate: 1/4
Constellation: Other

Interleaver
FECFRAME size: Normal
Code rate: 1/4
Constellation: 8PSK

DVB-S2X Modulator
FECFRAME size: Normal
Code rate: 1/4
Constellation: QPSK
2X Interpolation: Off

Physical Layer Framer
FECFRAME size: Normal
Code rate: 1/4
Constellation: QPSK
Pilots: Off
Gold Code: 0

Fast Noise Source
Noise Type: Gaussian
Amplitude: 510m
Seed: 0
Variate Pool Size: 8.192k

Add

QT GUI Constellation Sink
Number of Points: 1.024k
Autoscale: No

Phase Noise Generator
Noise Magnitude: 0
Alpha: 100m

Throttle
Sample Rate: 12M

Physical Layer Deframer
FECFRAME size: Normal
Code rate: 1/4
Constellation: QPSK

DVB-S2(X) Demodulator
FECFRAME size: Normal
Code rate: 1/4
Constellation: QPSK

Deinterleaver
FECFRAME size: Normal
Code rate: 1/4
Constellation: QPSK

LDPC Decoder
Standard: DVB-S2
FECFRAME size: Normal
Code rate: 1/4
Constellation: QPSK
Output mode: Message

BCH Decoder
Standard: DVB-S2
FECFRAME size: Normal
Code rate: 1/4
Output mode: Message

BBdescrambler
Standard: DVB-S2
FECFRAME size: Normal
Code rate: 3/5

BBdeheader
Standard: DVB-S2
FECFRAME size: Normal
Code rate: 3/5

IP Packet Sink
Standard: DVB-S2
FECFRAME size: Short
Code rate: 1/4
MAC Address: 02:00...55:4c:55

FIGURE 5. GRC flowgraph of the DVB-S2X modulation-demodulation chain.

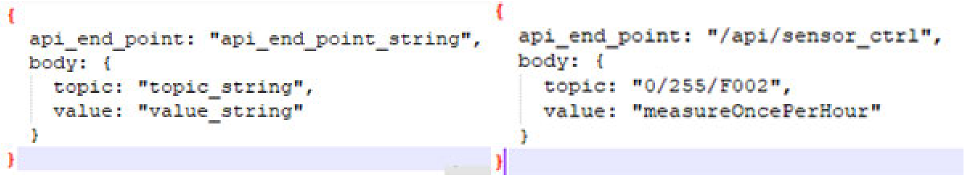
signaling information. PDCH and PCCH are multiplexed to form an uplink burst (ULB - Up-Link Burst), before which preamble is inserted. The physical level signaling includes the transmission of a TFI (Transport Format Indication) to the receiving party. The UPB is multiplexed I-Q using two Walsh OVSF (Orthogonal Variable Spreading Factor) sequences [36], as in the case of W-CDMA 3GPP. The PDCH and the associated control channel have a burst length multiple of the W-CDMA 3GPP frame period, i.e., 10ms. The purpose of the PCCH is also similar to that of W-CDMA 3GPP. In particular, it allows for a robust estimation of the channel even in difficult channel transmission conditions, ensuring the possibility of transmitting physical level signals that inform the receiver about the burst format without forcing its blind detection. In this respect, the four possible lengths of PDCH / PCCH are associated with two possible signal data rates for after FEC coding, 15 kBaud (corresponding to the information bit rate 5 kbit/s) and 30 kBaud (information bit rate 10 kbit/s) and three possible message lengths (300, 600 and 1200 bits). The power allocation is performed by weighting and combining the physical channels, considering indeed the frame length. The power is controlled by 4 bits in the SSA configuration table that indicate the voltage gain of the control channel PCCH relatively to the data channel PDCH.

The CPU usage of the code implementing the uplink of the gateway was evaluated varying the previously mentioned parameters, i.e. signal data rate and message length. The CPU load remained steady at 42% regardless of the variation of both parameters.

2) DOWNLINK

As it can be seen in Fig. 5, a complete DVB-S2X modulation - demodulation chain has been implemented in the FPGA of the SDR platform using GNU Radio software. Their combined implementation is motivated by the need to validate the developed demodulator scheme. The Physical Layer (PL) Deframer block provides the de-scrambling part at the physical level, the frame synchronization and the PL header decoding in order to automatically determine the constellation and coding parameters according to ETSI EN 302 307-1 section 5.5. The output of this block provides an XFECFRAME type frame. The adopted implementation offered us the possibility of manually selecting the constellation and the coding parameters mainly to carry out tests, the actual implementation being in full accordance with the specifications of the standard.

The DVB-S2X demodulator receives at the input a XFECFRAME and ensures the constellation-bit mapping. The I/Q sequences are mapped into parallel sequences, depending on the chosen constellation, and then converted to FECFRAME frames (a sequence of 64.800 bits for normal FECFRAME, or 16.200 bits for a short FECFRAME) and supplied at the output. In the case of 8PSK, 16APSK and



```
{
  api_end_point: "api_end_point_string",
  body: {
    topic: "topic_string",
    value: "value_string"
  }
}
```

```
{
  api_end_point: "/api/sensor_ctrl",
  body: {
    topic: "0/255/F002",
    value: "measureOncePerHour"
  }
}
```

FIGURE 6. Basic structure of JSON control message (left side) and data message (right side).

32APSK, the frames supplied by the DVB-S2X Demodulator are also passed through the Deinterleaver block. The frames are subsequently supplied to the LDPC and BCH Decoder blocks for correction and detection of residual errors, and a BBFRAME (Base Band Frame) frame is generated at the output. The implementation of these blocks was carried out in accordance with ETSI EN 302 307-1 section 5.3.1 to 5.3.3. The de-randomization section of the BBFRAME framework processes the header and create the data flow (payload) through the BBdescrambler and BBdeheader blocks. The resulting bitstream, encapsulated in accordance with the GSE (Generic Stream Encapsulation) protocol, is provided to the IP Sink block for processing and conversion into IP packets, which are then transmitted to a TUN virtual interface.

3) ModCod SELECTION

In order to determine the optimal MODCOD for the proposed architecture, the following aspects must be considered based on the prevoiusly analysed requirements:

- the requirements regarding the bit rate required to send the control messages to WSAN;
- the minimum band to be allocated so as to ensure the transmission at the determined bit rate;
- ensuring the requirements regarding power consumption (as low as possible);
- reduced use of computational resources (limited).

The bit rate required for control messages is determined by the structure and size of the messages. Fig. 6 shows the basic structure, in JSON format, of a control message (left side), and an example of an actual message to be transmitted to the gateway to adapt the measurement frequency to an hourly WSAN node identified by the "F002" ID (right side). The message size from the previous example is 86 bytes. Based on the current implementation, we determined the maximum size of the control message to be 158 bytes. Considering the analysis of the requirements for any further development, but also future optimizations of the message structure, we concluded that a maximum size of 500 bytes provides the necessary requirements.

Due to the small size of the control packages, a net rate of 500 kbps is sufficient to meet the system requirements. Therefore, the optimal options would be:

- MODCOD 1 - QPSK, 1/4 FEC, 400 kSym/s;
- MODCOD 2 - 8PSK, 3/5 FEC, 400 kSym/s.

Both MODCODs provide a sufficient net data rate, and the occupied band is relatively similar, whereas MODCOD 1 allows the use of a higher FEC and requires less computational resources. Following the analysis, MODCOD 1 was chosen as the optimal variant for our system.

Regarding the symbol rate, an analysis has been performed to determine the optimum rate based on the time required for the frame lock, i.e., the time between the control message is transmitted and the moment it is received and the computational resources used. Fig. 7 provides more details about the average frame lock time as well as the average time difference between the time a control message was transmitted to the transmitter's TAP interface and the time it was received from the receiver's TAP interface. An inversely proportional dependence of these values can be observed relative to the symbol rate, the best results being obviously obtained at a symbol rate of 800 kbps.

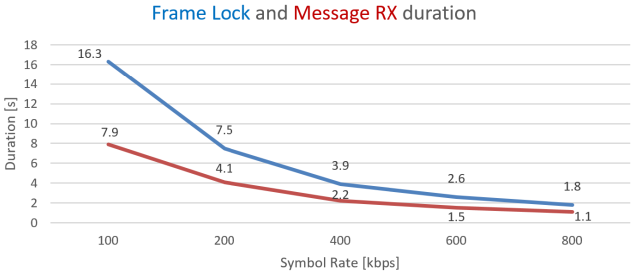


FIGURE 7. Average frame lock time vs. symbol rate for the SDR-implemented DVB-S2X transceiver.

Fig. 8 shows the average level of CPU utilization relative to the symbol rate. A directly proportional dependency to the symbol rate can be seen, the best results being obtained at a symbol rate of 100 kbps. Therefore, the 400 kbps symbol rate is the optimal value ensuring both acceptable duration and relatively low CPU use.

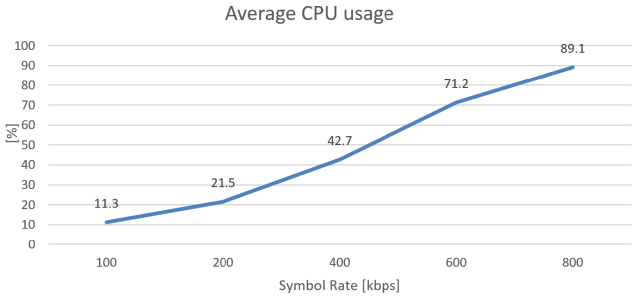


FIGURE 8. Average CPU usage vs. symbol rate for the SDR-implemented DVB-S2X transceiver.

Re-assuming, form the implementation of the up- and downlink satellite interfaces, we can conclude positive on the feasibility of the software implementation with a maximum of 82% of the CPU usage (39.3% for the uplink and 42.7% for the downlink), leaving enough capacity for the implementation of the remaining software blocks from Fig. 2. The overall CPU occupation confirms also the necessity to implement the terrestrial side using COTS modules and not the SDR's FPGA.

D. RECONFIGURABILITY

Reconfigurability is one of the key aspects of the proposed architecture, for which we defined a set of mechanisms which enables satellite and terrestrial link reconfiguration, both remotely and on site, via its Runtime and Management module. The reconfigurability mechanism is illustrated in Fig. 9. The data link can be, as previously stated, a satellite or a terrestrial one.

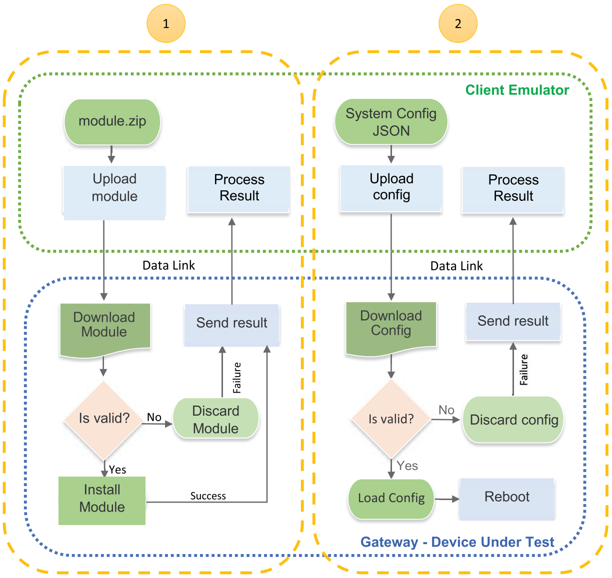


FIGURE 9. **Reconfigurability Mechanism.**

Upon startup, the System Config component is queried and the current configuration is retrieved. Depending on the values provided, the following are selected: uplink/downlink protocols, uplink/downlink frequencies, (active) IoT protocols. Based on the selection, the corresponding modules are loaded and initialized. For example, if in the current configuration DBV-S2X is set as the downlink protocol, the DVB-S2X Demodulator will be loaded and initialized. The configuration can be adapted on site, by manually overwriting the configuration file available at the System Config component, or remotely by pushing the configuration file via the satellite link. After each configuration change a system reboot is required, as depicted on the right side (2) of Fig. 9. Uploading new modules or updating the currently existing ones is also supported (both remotely and on site). A new module may be required for ensuring support for a different uplink/downlink protocol or for adding an FPGA base implementation for an IoT protocol which was previously supported via a COTS module. In both cases, the module implementation (e.g. a zip file) can be pushed via the satellite link and installed on the Gateway device. The Gateway will load and use the newly added module, only after explicitly adapting the system configuration as defined above and graphically described on the left side (1) of Fig. 9. The proposed system offers fallback mechanisms as well: if the system cannot be started after a configuration change, it will automatically fallback to the last known configuration. If a new module was uploaded and set as the one which shall be used or in case of an uplink/downlink protocol change, the system will fall back to the last used module/protocol if no data is received via the satellite link after a fallback timeout defined in the System Config component.

E. SECURITY

The security issues which can arise are the unauthorized access to the gateway and the deployment of malicious software during the reconfiguration phase. The satellite access to the gateway is granted via a web interface (Fig. 10) which implements a credential-based access and allows connections only for a restricted set of IPs. All the payloads are Triple DES (Data Encryption Standard) encrypted. The software upgrades via satellite were initially performed through HTTPS over GSE, but the SSL/TLS handshake procedure caused additional delay. Therefore, we opted for plain HTTP over GSE with the payloads encrypted using the Triple DES symmetric-key mechanism. The same access mechanism is applied in case of terrestrial link access. Furthermore, in this case, access is also allowed via a direct MQTT connection as long as valid credentials (username and password) are provided.

F. COSTS AND SIZE

One of the main goals of the present work is to build a low-cost and compact gateway. These requirements have been taken into consideration when selecting the USRP platform and when opting for a COTS implementation of the terrestrial side communication. The overall hardware costs of the gateway, including the COTS modules, are around 3500 €. These costs do not include the costs of up- and downconverters, amplifiers, satellite dishes and other equipment necessary to connect to the satellite, as the present paper is intended to be a proof of concept and does not present a commercial prototype. As for the size of the proposed gateway, the ETTUS E312 platform has a small form factor, that can be fit, together with the COTS modules and a 3200 mAh battery in a case 140 X 80 X 50 mm. External power supplies and the GPS antenna are not included.

V. IN-LAB TESTING RESULTS

For the up- and downlink in-lab tests, we set up the system presented in Fig. 11. The test transmitter and receiver are implemented in the FPGAs of two identical SDR USRP E312 platforms. The first SDR platform runs the GNU Radio implementation of a DVB-S2X transmitter and of an E-SSA receiver. Its RF part is connected through a coaxial cable with 50 ohm impedance and a 30 dB attenuator to the device

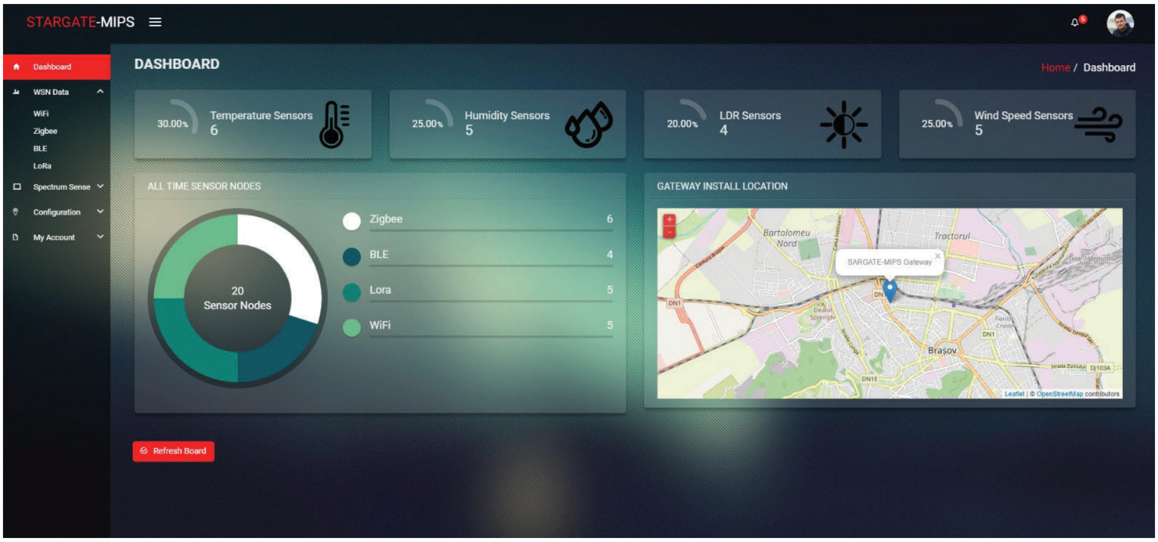


FIGURE 10. User Interface.

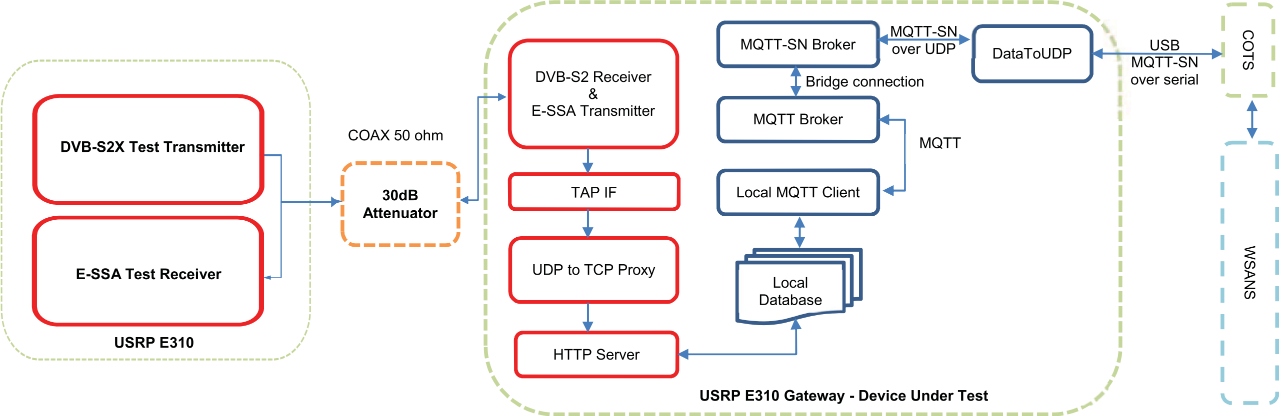


FIGURE 11. In-lab system testing setup.

under test, i.e. the gateway prototype embedded on the second USRP platform, as depicted in Fig. 11.

A. UPLINK TESTS

For the validation of the functionality of the E-SSA modulator, considering the capabilities of the used USRP platform, we opted for the use of the L Band, with a center frequency of 1.2 GHz. By doing this, we avoided for the beginning to use up- and downconverters to the Ka / Ku band, for better assessing the behavior of the generated waveform. A constant payload for the PDCH channel was used, with following characteristics: bit rate 5 kbps, symbol rate 15 ksps and a spreading factor of 16. The chip rate is 240 ksps and the roll of factor for the pulse shaping block is 0.22. The uplink transmitted signal spectrum was evaluated using a Tektronix Vector Signal Analyzer and can be viewed in Fig. 12.

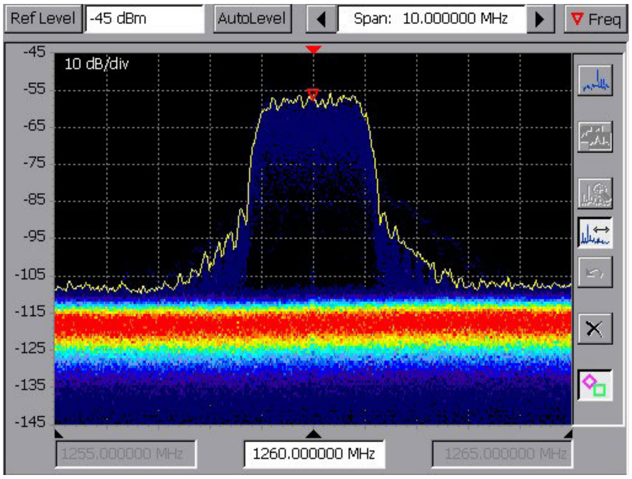


FIGURE 12. Transmitted signal spectrum visualized with a signal analyzer.

B. DOWNLINK TESTS

For validation of compliance with DVB-S2X standard, but also for performance evaluation, a number of automatic tests have been implemented. The components of the automatic test system, shown in Fig. 13, can be listed as follows:

- data generator controllable by the test application;

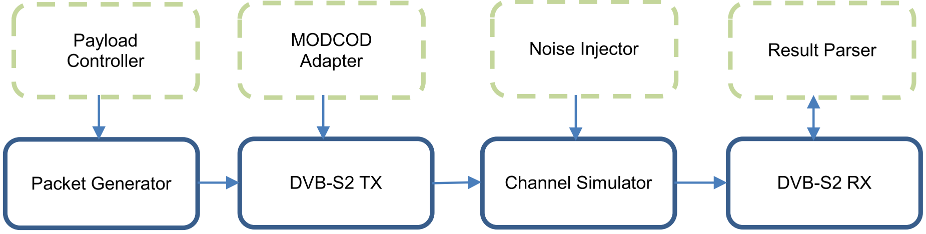
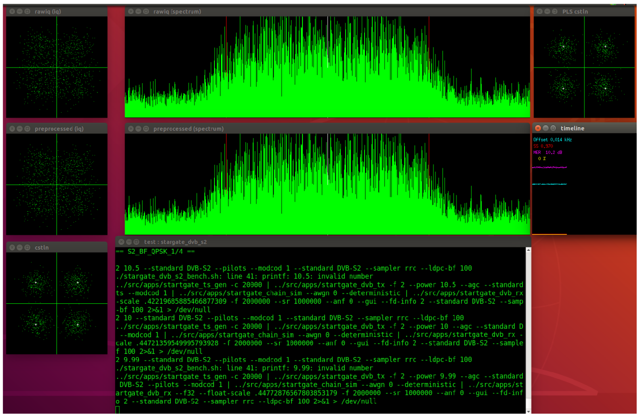


FIGURE 13. **DVB-S2X downlink system components.**

- DVB-S2X transmitter with a dedicated module that allows the automatic modification of MODCODs;
- channel simulator with noise injector (for performance tests);
- the DVB-S2X receiver implemented in section C, with integrated status check modules at different key points (necessary to compare the status of the component blocks at key times with a number of expected predefined states).

The downlink tests permitted us to assess the correct functionality of the entire downlink chain, starting form the constellation of the received signal up to the decoding of the transmitted data (Fig. 14) according to the DVB-S2X standard. Besides adding AWGN noise to the channel we performed also phase noise and adjacent channel interference (ACI) measurements. For the phase noise measurements we used components of the TOPCOM library written in C++. The threshold for the reception was the QEF (Quasi Error Free) requirement as defined in the DVB-S2X standard as the ratio between the useful FECFRAMEs correctly received and those affected by errors, after forward error correction FER $= 10^{-5}$ [37]. The same reception threshold was used when testing the ACI by generating more DVB-S2X carriers in 4 adjacent channels. The ACI was generated using an Agilent N5182A MXG RF Vector Signal Generator instead of the second USRP platform.



```
== S2_BF_QPSK_1/4 ==

2 10.5 --standard DVB-S2 --pilots --modcod 1 --standard DVB-S2 --sampler rrc --ldpc-bf 100
./stargate_dvb_s2_bench.sh: line 41: printf: 10.5: invalid number
../src/apps/startgate_ts_gen -c 20000 | ../src/apps/startgate_dvb_tx -f 2 --power 10.5 --agc --standard
ts --modcod 1 | ../src/apps/startgate_chain_sim --awgn 0 --deterministic | ../src/apps/startgate_dvb_rx
-scale .42219685885466877309 -f 2000000 --sr 1000000 --anf 0 --gui --fd-info 2 --standard DVB-S2 --samp
-bf 100 2>&1 > /dev/null
2 10 --standard DVB-S2 --pilots --modcod 1 --standard DVB-S2 --sampler rrc --ldpc-bf 100
../src/apps/startgate_ts_gen -c 20000 | ../src/apps/startgate_dvb_tx -f 2 --power 10 --agc --standard D
 --modcod 1 | ../src/apps/startgate_chain_sim --awgn 0 --deterministic | ../src/apps/startgate_dvb_rx -
cale .44721359549995793928 -f 2000000 --sr 1000000 --anf 0 --gui --fd-info 2 --standard DVB-S2 --sample
f 100 2>&1 > /dev/null
2 9.99 --standard DVB-S2 --pilots --modcod 1 --standard DVB-S2 --sampler rrc --ldpc-bf 100
./stargate_dvb_s2_bench.sh: line 41: printf: 9.99: invalid number
../src/apps/startgate_ts_gen -c 20000 | ../src/apps/startgate_dvb_tx -f 2 --power 9.99 --agc --standard
 DVB-S2 --pilots --modcod 1 | ../src/apps/startgate_chain_sim --awgn 0 --deterministic | ../src/apps/st
artgate_dvb_rx --f32 --float-scale .44772876567803853179 -f 2000000 --sr 1000000 --anf 0 --gui --fd-inf
o 2 --standard DVB-S2 --sampler rrc --ldpc-bf 100 2>&1 > /dev/null
```

FIGURE 14. **DVB-S2X downlink tests.**

All reception tests were validated also with a DVB-S2X compliant DekTek DTU-315 board.

C. INTEGRATION TESTS

The integration tests verify the functioning of the interconnected internal processing modules, providing performance information based on a test architecture depicted in Fig. 15.

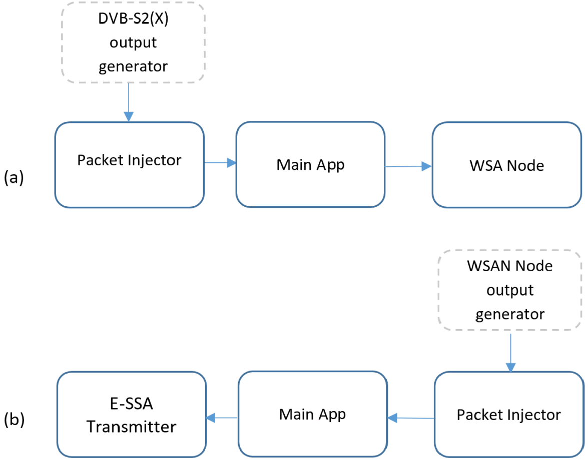


FIGURE 15. **Integration test architecture.**

The test results of the integration tests allowed the evaluation of the end-to-end processing latency of the gateway, the average value being 121 ms for 1000 data packets transmitted. This value increases to 688 ms if there are other packet requests in the processing queue. The higher value of processing time is motivated by the serial processing of requests. Therefore, if other requests are received (e.g. a request to return the data recorded by a sensor) when a control package is received (request to change the operating parameters of a particular WSAN), they will be processed first the requests already present after which the control package will be processed. The serial processing is sped up through a multi-threading implementation that was adopted to ensure parallel processing of client requests and data provided from WSAN (if at the time the application processes a request from a client, a WSAN node transmits data, they will be processed at the same time).

D. RECONFIGURABILITY TESTS

To ensure that the reconfigurability mechanisms are fully functional, a set of manual tests were performed. The focus was on uploading a new satellite module which would offer support for a different set of uplink/downlink protocols. As the modulator and demodulator implementations were already available for both DVB-S2X and E-SSA protocols, a protocol swap test scenario was defined and executed. First, the DVB-S2X receiver and the E-SSA transmitter were loaded on the first USRP. Then, the zip files containing the DVB-S2X Test transmitter and the E-SSA Test receiver were transmitted via the coax link to the device under test (Fig. 11). When the confirmation was received, that the modules were successfully installed, a new System Config file was sent to the device under test in which DVB-S2X was set as uplink protocol and E-SSA as downlink protocol. While the device under test was rebooting, the first USRP was reconfigured to use the E-SSA transmitter and the DVB-S2X receiver. After the reboot was completed, a control message was sent via the DVB-S2X uplink. It was observed that the sensor node reacted accordingly and that a confirmation message was received via the E-SSA downlink.

As for the timing, an overall reconfiguration performed with the new module available locally, took up to 10 seconds (including the time required for the configuration file to be sent over the satellite link). In case of a satellite link protocol reconfiguration, the overall duration was longer, especially for the case when the module for the desired satellite communication protocol was not available on the gateway. In this case the time required to upload the new module has to be taken into consideration. For example uploading a new module with a size of 1.5 Mb over a DVB-S2X link with MODCOD 1 (QPSK, 1/4 FEC, 400 kSym/s) selected will add an average of 6 seconds to the overall duration. Adding the installation time, configuration change and system reboot resulted in an overall duration of up to 25 seconds.

E. UNIT TESTS

The last functional tests performed were the unit tests which are needed to verify the proper implementation and functioning of the individual modules. We integrated the implemented software code both for the up- and downlink parts with the GitLab Continuous Integration / Continuous Development (CI / CD) environment and defined a suite of 70 automatic tests covering the key points of the component modules. The integration with the CI / CD environment ensured the automatic execution of the tests whenever a change in the source code takes place. A picture of the tested prototype, together with the WSAN nodes is presented in Fig. 16. We also tested the system with 20 real, physical nodes, and simulated a maximum workload with 5.000 nodes.

F. POWER CONSUMPTION

The power consumption of the gateway varies much as a function of the data that has to be translated from the terrestrial side to the satellite side and vice-versa. The gateway's consumption varies between 3 and 6 W depending on the performed tasks, reaching its maximum when simultaneously decoding the incoming signal and encoding and transmitting the outgoing signal. On top of this value, we need to add the COTS module with the highest power consumption, i.e. the WiFi module which has a maximum power rating of 1 W. adding all the variables, the mean power consumption is 4W and the maximum power rating is 7W. On the chosen Ettus USRP E312 platform with on-board 3200 mAh battery, that lead to an minimum autonomy of 2 hours.

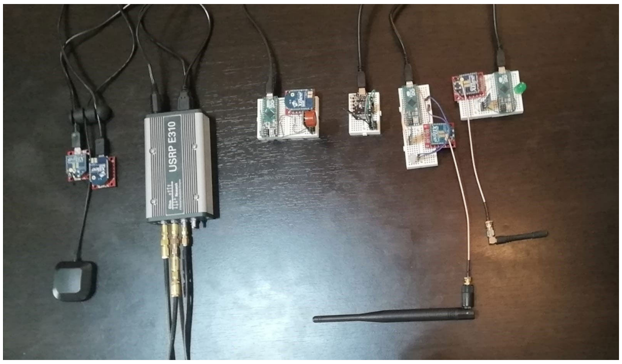


FIGURE 16. Gateway Prototype together with WSAN nodes.

VI. CONCLUSIONS

This paper presents a solution for satellite connectivity for IoRT applications, implemented on a single device that acts as a gateway between the terrestrial and the satellite sides. The main goal of the adopted implementation was to significantly lower the hardware and implementation costs, while assuring a high degree of re-configurability by using SDR technologies. Based on an initial requirements analysis, a generic architecture was conceived and subsequently implemented using a standalone SDR platform and COTS modules for covering the main terrestrial IoRT standards. To lower the implementation cost, the satellite up- and downlink was written in the FPGA of the SDR platform, leading to an overall CPU occupation of 82%. This left enough space for the programming of the control interface, local databases, HTTP server, MQTT brokers and all the other necessary software modules. The memory and CPU constraints of the chosen SDR platform led us to the use of COTS modules connected to the gateway by means of the serial USB interface for the terrestrial side.

The in-lab testing of the entire gateway validated the chosen architecture and the solutions used for its implementation on a low-cost, standalone SDR platform. Although preliminary and still in the "proof-of-concept" stage, the achieved results confirmed the correct and effective SDR-based implementation of the gateway functionalities and fully demonstrated the reconfiguration capabilities of the proposed architecture. The integration tests confirmed the reliability of the end-to-end information processing performed by the gateway in real-time. Finally, the unit tests validated the implementation and the functioning of the single SDR modules. To sum up, the proposed SDR-based IoRT gateway architecture can represent an interesting, innovative, prototypical realization that can be further on upgraded for future open-field testing, where a real satellite will actually support the remote connection. Future work will include the optimisation of the developed solution for reducing the occupation of the FPGA area and also for minimizing the overall delay. Other future work will include tests on the implementation using the new ETTUS E320 board and, alternatively, last generation SoC (System on Chip) devices combined with dedicated RF hardware, in order to lower the implementation costs even more.

ACKNOWLEDGMENT

The research activities presented in this paper fall within the field of interest of the IEEE AESS Technical Panel on Glue Technologies for Space Systems.

REFERENCES

[1] J. A. Stankovic, "Research directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.

[2] M. De Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, "Satellite communications supporting Internet of remote things," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 113–123, Feb. 2016.

[3] I. Bisio, F. Lavagetto, and G. Luzzati, "Cooperative application layer joint video coding in the Internet of remote things," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1418–1426, Dec. 2016.

[4] M. Bacco, M. Colucci, and A. Gotta, "Application protocols enabling Internet of remote things via random access satellite channels," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[5] C. Gavrila, C.-Z. Kertesz, M. Alexandru, and V. Popescu, "SDR-based gateway for IoT and M2M applications," in *Proc. Baltic URSI Symp. (URSI)*, May 2018, pp. 71–74.

[6] C. Gavrila, C.-Z. Kertesz, M. Alexandru, and V. Popescu, "Reconfigurable IoT gateway based on a SDR platform," in *Proc. Int. Conf. Commun. (COMM)*, Jun. 2018, pp. 345–348.

[7] C. Gavrila, M. Alexandru, V. Popescu, C. Sacchi, and D. Giusto, "Satellite SDR gateway for M2M and IoT applications," in *Proc. IEEE Aerosp. Conf.*, Mar. 2019, pp. 1–9.

[8] Z. Qu, G. Zhang, H. Cao, and J. Xie, "Leo satellite constellation for Internet of Things," *IEEE Access*, vol. 5, pp. 18391–18401, 2017.

[9] I. I. Lysogor, L. S. Voskov, and S. G. Efremov, "Survey of data exchange formats for heterogeneous LPWAN-satellite IoT networks," in *Proc. Moscow Workshop Electron. Netw. Technol. (MWENT)*, Mar. 2018, pp. 1–5.

[10] C. A. Hofmann and A. Knopp, "Ultranarrowband waveform for IoT direct random multiple access to GEO Satellites," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10134–10149, Dec. 2019.

[11] M. Bacco, L. Boero, P. Cassara, M. Colucci, A. Gotta, M. Marchese, and F. Patrone, "IoT applications and services in space information networks," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 31–37, Apr. 2019.

[12] R. Soua, M. R. Palattella, and T. Engel, "IoT application protocols optimisation for future integrated M2M-satellite networks," in *Proc. Global Inf. Infrastruct. Netw. Symp. (GIIS)*, Oct. 2018, pp. 1–5.

[13] A. Abdelsalam, C. Roseti, and F. Zampognaro, "TCP performance for satellite M2M applications over random access links," in *Proc. Int. Symp. Netw., Comput. Commun. (ISNCC)*, Jun. 2018, pp. 1–5.

[14] D. Hu, L. He, and J. Wu, "A novel forward-link multiplexed scheme in satellite-based Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1265–1274, Apr. 2018.

[15] R. L. Sturdivant and J. Lee, "Systems engineering of IoT connected commercial airliners using satellite backhaul links," in *Proc. IEEE Topical Workshop Internet Space (TWIOS)*, Jan. 2018, pp. 1–4.

[16] D. Digdarsini, M. Kumar, G. Khot, T. V. S. Ram, and V. K. Tank, "FPGA implementation of automatic modulation recognition system for advanced SATCOM system," in *Proc. Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Feb. 2014, pp. 464–469.

[17] R. Akhtyamov, I. L. Cruz, and A. Golkar, "Software defined radio implementation of a negotiator node testbed for federated satellite systems," in *Proc. Int. Astron. Congr.*, 2015, pp. 1–8.

[18] N. Suematsu, H. Oguma, S. Eguchi, S. Kameda, M. Sasanuma, and K. Kuroda, "Multi-mode SDR VSAT against big disasters," in *Proc. Eur. Microw. Conf.*, Oct. 2013, pp. 842–845.

[19] M. R. Maheshwarappa, M. D. J. Bowyer, and C. P. Bridges, "Improvements in CPU & FPGA performance for small satellite SDR applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 310–322, Feb. 2017.

[20] O. Ceylan, A. Caglar, H. B. Tugrel, H. O. Cakar, A. O. Kislal, K. Kula, and H. B. Yagci, "Small satellites rock a software-defined radio modem and ground station design for cube satellite communication," *IEEE Microw. Mag.*, vol. 17, no. 3, pp. 26–33, Mar. 2016.

[21] R. P. Hudhajanto, H. Wijanarko, M. Arifin, B. Sugandi, S. Aisyah, N. S. Asaad, and A. A. Syardiansyah, "Low cost nano satellite communication system using GNURadio, HackRF, and raspberry Pi," in *Proc. Int. Conf. Appl. Eng. (ICAE)*, Oct. 2018, pp. 1–4.

[22] N. Suematsu, S. Kameda, H. Oguma, M. Sasanuma, S. Eguchi, and K. Kuroda, "Multi-mode portable VSAT for disaster-resilient wireless networks," in *Proc. Asia–Pacific Microw. Conf.*, Nov. 2014, pp. 549–551.

[23] J. Ylitalo, A. Hulkkonen, M. Höyhtyä, A. Byman, M. Leinonen, J. Janhunen, and A. Roivainen, "Hybrid satellite systems: Extending terrestrial networks using satellites," in *Cooperative and Cognitive Satellite Systems*, S. Chatzinotas, B. Ottersten, and R. D. Gaudenzi, Eds. New York, NY, USA: Academic, 2015, ch. 11, pp. 337–371. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780127999487000116

[24] M. Schadhauser, J. Robert, and A. Heuberger, "Design of autonomous base stations for low power wide area (LPWA) communication," in *Proc. Smart SysTech; Eur. Conf. Smart Objects, Syst. Technol.*, Jun. 2017, pp. 1–8.

[25] Y. Ma, Y. Zeng, and S. Sun, "A software defined radio based multi-function radar for IoT applications," in *Proc. 24th Asia–Pacific Conf. Commun. (APCC)*, Nov. 2018, pp. 239–244.

[26] X. Xiong, T. Wu, H. Long, and K. Zheng, "Implementation and performance evaluation of LECIM for 5G M2M applications with SDR," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2014, pp. 612–617.

[27] C.-F. Li, J.-K. Hwang, C. Ma, and C.-J. Lin, "Software defined radio implementation of LTE R13 NB-IoT downlink vector signal generator," in *Proc. IEEE Int. Conf. Consum. Electron. Taiwan (ICCE-TW)*, Jun. 2017, pp. 69–70.

[28] Y. Park, S. Kuk, I. Kang, and H. Kim, "Overcoming IoT language barriers using smartphone SDRs," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 816–828, Mar. 2017.

[29] Y. Chen, S. Lu, H.-S. Kim, D. Blaauw, R. G. Dreslinski, and T. Mudge, "A low power software-defined-radio baseband processor for the Internet of Things," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Mar. 2016, pp. 40–51.

[30] T. Balan, A. Balan, and F. Sandu, "SDR implementation of a D2D security cryptographic mechanism," *IEEE Access*, vol. 7, pp. 38847–38855, 2019.

[31] S. Majumder, "Energy detection spectrum sensing on RTL-SDR based IoT platform," in *Proc. Conf. Inf. Commun. Technol. (CICT)*, Oct. 2018, pp. 1–6.

[32] M. Kist, J. Rochol, L. A. DaSilva, and C. B. Both, "SDR virtualization in future mobile networks: Enabling multi-programmable air-interfaces," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[33] T. Ahmed, A. Alleg, and N. Marie-Magdelaine, "An architecture framework for virtualization of IoT network," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jun. 2019, pp. 183–187.

[34] V. Mignone, M. A. Vazquez-Castro, and T. Stockhammer, "The future of satellite TV: The wide range of applications of the DVB-S2 standard and perspectives," *Proc. IEEE*, vol. 99, no. 11, pp. 1905–1921, Nov. 2011.

[35] A. Morello and V. Mignone, "DVB-s2X: The new extensions to the second generation DVB satellite standard DVB-s2: Editorial," *Int. J. Satell. Commun. Netw.*, vol. 34, no. 3, pp. 323–325, May 2016, doi: 10.1002/sat.1167.

[36] E. H. Dinan and B. Jabbari, "Spreading codes for direct sequence CDMA and wideband CDMA cellular networks," *IEEE Commun. Mag.*, vol. 36, no. 9, pp. 48–54, Sep. 1998.

[37] C. Marchand and E. Boutillon, "LDPC decoder architecture for DVB-S2 and DVB-S2X standards," in *Proc. IEEE Workshop Signal Process. Syst. (SiPS)*, Hangzhou, China, May 2016, pp. 1–5, doi: 10.1109/SiPS.2015.7345034.

CRISTINEL GAVRILĂ received the B.Sc. degree in technologies and telecommunication systems and the M.Sc. degree in electronic and communication integrated systems from the Transilvania University of Brasov, Romania, in 2014 and 2016, respectively, where he is currently pursuing the Ph.D. degree in electronics, telecommunications and informational technologies. He is a Research Assistant with the Faculty of Electrical Engineering and Computer Science, Transilvania University of Brasov. His current research interests include wireless networks, the Internet of Things (IoT) and M2M applications, cognitive radio, hybrid television, OTT services, e-Learning, and artificial intelligence. His scientific activity includes nine scientific articles (author/coauthor) in fields, such as hybrid television, streaming, OTT and web services, hybrid location tracking or e-Learning.

VLAD POPESCU (Member, IEEE) received the M.Sc. degree in electronics and computer engineering, and the Ph.D. degree in telecommunications from the Transilvania University of Brasov, Romania, in 1999 and 2006, respectively. In 2000, he was with Malmö University, Sweden, specializing in multimedia applications for four months. From 2001 to 2002, he was a Research Fellow with Technical University, Aachen, Germany, where he finished the experimental part of his Ph.D. studies on wireless communication in underground environments, from 2004 to 2005. Since 2009, he has been a Visiting Professor with the Department of Electrical and Electronic Engineering, University of Cagliari, Italy. Since 2000, he has been an Associate Professor with the Department of Electronics and Computers, Transilvania University of Brasov. He also collaborates close with the Department of Electrical and Electronic Engineering, University of Cagliari, both on research and didactical level. His main research interests include telecommunications, cognitive radio systems, multimedia applications, and data acquisition. He is a member of the IEEE BTS, the IEEE VTS, and the IEEE AES. He is the Chair of the IEEE BTS Chapter in Romania.

MARIAN ALEXANDRU received the Ph.D. degree from the Transilvania University of Brasov, Romania, in 2005. He is currently an Associate Professor with the Faculty of Electrical Engineering and Computer Science, Transilvania University of Brasov, where he has been teaching radars and telecommunication courses, since 1996. His current research interests include applications of virtual and remote laboratories, digital communications, wireless networks, cognitive radio, DVB, and e-Learning. His scientific activity include author/coauthor of 12 books or chapters (RO), 1 book (EN), and more than 40 scientific articles (journals and international conferences) in the field of indoor location and sensing, wireless systems, microwaves, antennas, and educational technologies. He collaborated to nine national and international engineering research and educational projects. He is a coauthor of three patents. He was in the Management Committee of the IC1004 COST Action, Cooperative Radio for Green Smart Environments. He is a member of the Management Committee of COST CA 15104 IRACON (inclusive radio communications networks for 5G and beyond).

MAURIZIO MURRONI (Senior Member, IEEE) received the M.Sc. degree in electronic engineering and the Ph.D. degree in electronic engineering and computers from the University of Cagliari, in 1998 and 2001, respectively. He is currently an Associate Professor with the Department of Electrical and Electronic Engineering (DIEE), University of Cagliari, and a member of the National, Inter-University Consortium for Telecommunications (CNIT). He is a coauthor of an extensive list of journal articles and peer-reviewed conference papers, and received several best paper awards. His research interests include quality of experience, multimedia data transmission and processing, broadcasting, cognitive radio systems, and signal processing for radio communications. He is a Senior Member of the IEEE ComSoc, the IEEE BTS, the IEEE VTS, and the IEEE SPS. He was a coauthor of the 1900.6-2011 IEEE Standard for Spectrum Sensing Interfaces and Data Structures for Dynamic Spectrum Access and other Advanced Radio Communication Systems. He has served as the Chair of various international conferences and workshops, and a Guest Editor for several journals. He is a Distinguished Lecturer of the IEEE Broadcast Technology Society and an Associate Editor of the IEEE TRANSACTIONS ON BROADCASTING INFORMATION.

CLAUDIO SACCHI (Senior Member, IEEE) received the Laurea degree in electronic engineering, and the Ph.D. degree in space science and engineering from the University of Genoa, Italy, in 1992 and 2003, respectively. From 1996 to 2002, he was a Research Cooperator with the Department of Biophysical and Electronic Engineering (DIBE), University of Genoa, and the National Italian Consortium in Telecommunications (CNIT), managing project activities in the field of multimedia surveillance systems and satellite communications. Since August 2002, he has been an Assistant Professor with the Department of Information Engineering and Computer Science (DISI), University of Trento, Italy. He is the author or coauthor of more than 100 papers published in international journals and conferences. His research interests include wideband mobile and satellite transmission systems based on space, time, and frequency diversity, MIMO systems, array processing, multirate and multiaccess wireless communications, EHF broadband aerospace communications, software radio and cognitive radio, and radio communications for emergency recovery applications. He is a member of the IEEE ComSoc, the IEEE Broadcast, the IEEE VT, and the IEEE AESS Society. In 2011, he was a Guest Editor of the special issue of PROCEEDINGS OF THE IEEE: Aerospace Communications: History, Trends and Future. In 2015, he was also a Guest Editor of the featured-topic special issue of *IEEE Communications Magazine*: Toward the Space 2.0 Era. Since May 2019, has been a Coordinator of the IEEE AESS Technical Panel: Glue Technologies for Space Systems.

• • •