



**HAL**  
open science

# RESS-IoT: A Scalable Energy-Efficient MAC Protocol for Direct-to-Satellite IoT

Raydel Ortigueira, Juan A Fraire, Alex Becerra, Tomas Ferrer, Sandra  
Cespedes

► **To cite this version:**

Raydel Ortigueira, Juan A Fraire, Alex Becerra, Tomas Ferrer, Sandra Cespedes. RESS-IoT: A Scalable Energy-Efficient MAC Protocol for Direct-to-Satellite IoT. IEEE Access, 2021, 9, pp.164440-164453. 10.1109/ACCESS.2021.3134246 . hal-04058142

**HAL Id: hal-04058142**

**<https://hal.science/hal-04058142v1>**

Submitted on 4 Apr 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Received November 20, 2021, accepted November 27, 2021, date of publication December 10, 2021, date of current version December 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3134246

# RESS-IoT: A Scalable Energy-Efficient MAC Protocol for Direct-to-Satellite IoT

RAYDEL ORTIGUEIRA<sup>1</sup>, JUAN A. FRAIRE<sup>2,3</sup>, (Senior Member, IEEE), ALEX BECERRA<sup>4</sup>, TOMÁS FERRER<sup>5</sup>, AND SANDRA CÉSPEDES<sup>1,6</sup>, (Senior Member, IEEE)

<sup>1</sup>Department of Electrical Engineering, Universidad de Chile, Santiago 8370451, Chile

<sup>2</sup>CITI, Inria, INSA Lyon, University of Lyon, 69621 Villeurbanne, France

<sup>3</sup>CONICET, Universidad Nacional de Córdoba, Córdoba 5000, Argentina

<sup>4</sup>Aurora Space, Santiago 7750053, Chile

<sup>5</sup>Codelco Chuquicamata, Santiago 8340424, Chile

<sup>6</sup>NIC Chile Research Labs, Universidad de Chile, Santiago 8320198, Chile

Corresponding author: Raydel Ortigueira (rortigueira@ug.uchile.cl)

The work of Raydel Ortigueira was supported by Comisión Nacional de Investigación Científica y Tecnológica - Programa Formación de Capital Humano Avanzado (CONICYT-PFCHA) Doctorado Nacional under Grant 2018-21181930. This work was supported in part by the ANID Project FONDECYT Regular under Grant 1201893, in part by the Regional Program STIC-AMSUD under Grant STIC2020003 Folio 21-STIC-12, in part by the ANID Basal Project under Grant FB0008, in part by the European Research Commission (ERC) Advanced Grant 695614 [Power Verified (POWVER)], in part by the Deutsche Forschungsgemeinschaft (DFG) (as part of TRR 248) under Grant 389792660, in part by the Research and Innovation Staff Exchange Evaluations / Marie Skłodowska-Curie (RISE/MSCA) MISSION Project, and in part by the Agencia I + D + i under Grant PICT-2017-1335.

**ABSTRACT** The Internet of Things (IoT) is enabling unprecedented applications based on the transport of small data volumes to and from constrained devices. When end devices or sensor nodes are located in very remote zones with inaccessible topography, Direct-to-Satellite IoT (DtS-IoT) has been proposed as an appealing solution. In DtS-IoT, isolated sensor nodes can directly relay data to and from inexpensive nanosatellites (i.e., CubeSats). However, because both sensor nodes and CubeSats operate on very limited energy supply and storage, the efficient management of power-hungry communication sub-systems is essential. Thus, specific Medium Access Control (MAC) protocols are needed to ensure the minimal overhead while considering DtS-IoT scalability and channel dynamics. In this work, we contribute with REserve-and-Send Sift-IoT (RESS-IoT), a scalable and energy-efficient DtS-IoT MAC protocol, combining the LoRa physical layer with a novel link scheduling approach. Extensive simulations demonstrate that our solution provides up to four times the energy savings in the satellite and up to seven times the energy savings at the sensor node on the ground with respect to state-of-the-art IoT protocols.

**INDEX TERMS** Direct-to-Satellite IoT (DtS-IoT), energy efficiency, LoRa, medium access control (MAC), nanosatellite.

## I. INTRODUCTION

The Internet of Things (IoT) is defined as the system based on sensors and actuators connected via specific wireless technologies to a network that enables unparalleled data processing and analytics [1]. As a result, a broad range of applications dedicated to healthcare, smart agriculture, monitoring/surveillance of remote areas, and asset tracking are unlocked, all them benefiting many sectors of society [2]. IoT usage is expected to be massive, reaching 45% of the total traffic of the Internet [3] and revenues up to three trillion USD in 2025 [4]. The popularity of IoT networks is further stimulated by the constant decrease in the production costs of

end-devices or sensor nodes, favoring the deployment of such networks at very large scales.

However, many sensor nodes must be deployed in remote places where there is no coverage of terrestrial access networks. In these challenging cases, satellite-based connectivity is an appealing alternative. A cost-effective solution is to leverage nanosatellites designed according to the CubeSat standard [5], [6], which would allow for low production and launching costs as well as reduced manufacturing times, although at the expense of limited power and storage possibilities. For example, on-board power supply elements such as batteries and solar panels need to comply with dimensions and weight on the order of 10 cm × 10 cm × 10 cm and 1.33 kg per unit (i.e., “1U”) respectively [7]. These power and storage constraints notably limit the tasks the satellite can perform, especially in eclipse conditions. Indeed,

The associate editor coordinating the review of this manuscript and approving it for publication was P. Venkata Krishna.

Low Earth Orbit (LEO) satellites, orbiting below 1,000 km, present orbital periods of approximately 90 minutes, during which sunlight exposure and occlusion occurs. Furthermore, on the ground segment, sensor nodes are typically powered by batteries [8], which cannot be easily replaced when deployed in remote zones [9].

In this context, our study scenario consists of a Direct-to-Satellite IoT (DtS-IoT) system [10] where IoT applications depend on low-cost satellite solutions, including CubeSats, to receive massive communication services. This scenario is characterized by an increasing number of nodes, a low data rate, and the use of low-cost technologies, resulting in energy limitations, restrictions on the available hardware in terms of quantity and resources, and a limited processing capacity. Therefore, energy-efficient and scalable communication protocols that require low processing capacity are fundamental for good performance of the DtS-IoT systems. This paper addresses the requirements from the Medium Access Control (MAC) perspective. Since radio communication systems are typically the most power-consuming subsystem [11]–[14], it is crucial to design scalable MAC protocols that operate with limited processing capacity and minimal hardware requirements, and minimize the energy waste of these subsystems.

Although energy efficiency is a relevant aspect of the MAC protocols in both space [15]–[19] and terrestrial [13], [20]–[22] segments, these protocols all fail to perform properly when applied to networks with thousands of nodes as expected for the DtS-IoT case [23]. Then, integrating both scalability and energy efficiency into the design of a MAC protocol is a challenge for DtS-IoT systems, especially if the design needs to consider the sensor nodes and the CubeSat exclusively without increasing the protocol's computational demand. To tackle this problem, we propose *REserve-and-Send Sift-IoT* (RESS-IoT): a scalable energy-efficient MAC protocol inspired by the Sift protocol [21]. The proposed protocol uses variable-length Clear-to-Send (VCTS) frames and reduces the energy waste in the terrestrial and space segments. The contributions of this paper can thus be summarized as follows:

- We propose a scalable energy-efficient MAC protocol that depends exclusively on the sensor nodes and the CubeSat; the protocol does not demand high processing capacity and does not require the resources of the ground station or channel information in advance.
- We define an analytical model for the proposed MAC protocol, which is validated through extensive simulations. Our comparative evaluations demonstrate that the proposed protocol reduces energy waste for the satellite and for the thousands of nodes participating in a DtS-IoT network.

The remainder of this paper is organized as follows. In Section II, we provide a review of the related work. In Section III, we describe the system model and then introduce the proposed RESS-IoT MAC protocol in Section IV. In Section V, we evaluate the performance of the RESS-IoT

protocol in comparison to state-of-the-art satellite IoT MAC protocols. In Section VI, we provide the concluding remarks.

## II. RELATED WORK

In recent years, several MAC protocols with low or no processing load and minimal hardware requirements have been proposed for both satellite IoT environments with LEO satellites and terrestrial IoT environments. In this section, we discuss the most representative of these low-overhead MAC protocols.

### A. TERRESTRIAL IoT MAC PROTOCOLS

#### 1) ADC-MAC [13]

The MAC protocol based on the Adaptive Duty Cycle (ADC-MAC) is inspired by the Class A LoRaWAN protocol. In ADC-MAC, nodes adjust the transmission duty cycle taking into account the following performance indicators: residual power, traffic load, and network congestion rate. However, in networks where the number of nodes is high, where the nodes have similar energy levels in their batteries, and where the traffic loads are similar, energy waste increases due to collisions among data packets.

#### 2) ACO-MAC [20]

In the MAC protocol based on the Ant Colony Optimization (ACO-MAC), each data packet transmission starts when the sink node transmits a beacon message to the nodes in the area of interest. Then, each node with available data calculates an energy efficiency index based on information about the residual energy and channel condition. The node with the highest index will wait the shortest period to transmit its data packet and will obtain access to the channel. This protocol uses opportunistic carrier sensing, which is a useful strategy when the nodes are located in a small area. However, if a high number of nodes is distributed in an extensive and/or complex region, the energy waste increases due to collisions among data packets generated by the hidden terminals.

#### 3) SIFT [21]

When  $N$  sensor nodes close to each other detect the same event, they send one data packet to the sink node to report the event. To send this packet, each node exchanges Request-to-Send (RTS) and Clear-to-Send (CTS) messages with the sink node, which sends an acknowledgment (ACK) message every time it receives a data packet. Once the event is detected, each node creates a fixed-length slotted time window. Then, it randomly selects one time slot according to a reverse exponential probability distribution to begin transmitting an RTS message. When a node receives the ACK message and its transmission is not confirmed, it resends the RTS message. Nodes stop resending RTS messages when they receive  $R$  acknowledgment messages from the sink node, where  $R < N$ . In this protocol, nodes begin the data transmission process immediately after detecting an event. However, this behavior

is not useful if the sink node is a LEO satellite because the nodes would transmit in the absence of a passing-by satellite.

#### 4) UCAL [22]

Unconfirmed Class A LoRaWAN (UCAL) is an Aloha-based protocol where the end-device transmits a data packet to the gateway and then listens to the channel for two periods. This protocol increases the energy waste due to data packet collisions when the number of nodes increases.

### B. SATELLITE IoT MAC PROTOCOLS

#### 1) ENCODED-SA [18]

In Encoded Slotted Aloha (Encoded-SA), each message sent by a node to the satellite is divided into sub-packets. These sub-packets are encoded using an  $(n, k)$  erasure code and transmitted using Slotted Aloha. At the receiver, each message can be decoded using at least  $k$  correctly received sub-packets of the  $n$  encoded sub-packets.

#### 2) SLOTTED-AlohaCA [17]

Slotted Aloha with Collision Avoidance (Slotted-AlohaCA) is a protocol based on Slotted Aloha, where relay nodes (RNs) send data packets to the satellite. When an RN successfully transmits a data packet in a time slot and has more data packets to send, it informs the spacecraft using the transmitted data packet. Consequently, the satellite sends an ACK message at the end of the time slot to inform all RNs that the next time slot is reserved. As a result, the successful RN reserves the radio channel until all its data packets are transmitted. However, in this scheme, sensors on the ground require the RNs to transmit their data to the spacecraft; thus, the network topology is more complex than in DtS-IoT.

#### 3) SA-LoRaWAN [15]

Sensor Activation based on LoRaWAN (SA-LoRaWAN) is a protocol that uses two mechanisms: a percentage activation control mechanism that disperses the activation instants of the nodes according to the fill level in their buffers, and a mechanism that adapts the transmission power of the nodes based on the location information of the spacecraft.

#### 4) EA [19]

Enhanced Aloha (EA) is the protocol used in the Argos satellite telemetry system. In this protocol, the data packets are transmitted to the spacecraft introducing a random deviation at the moment of transmission, which avoids collisions of data packets caused by periodic transmissions.

In general, the surveyed schemes have shown to be energy-efficient for small networks. However, in these protocols, the energy waste due to collisions among data packets increases dramatically as the number of nodes increases. Therefore, integrating energy efficiency and scalability into the design of a MAC protocol will be a challenge for DtS-IoT systems, in particular if the design requires reduced computational complexity and no involvement from entities other than the

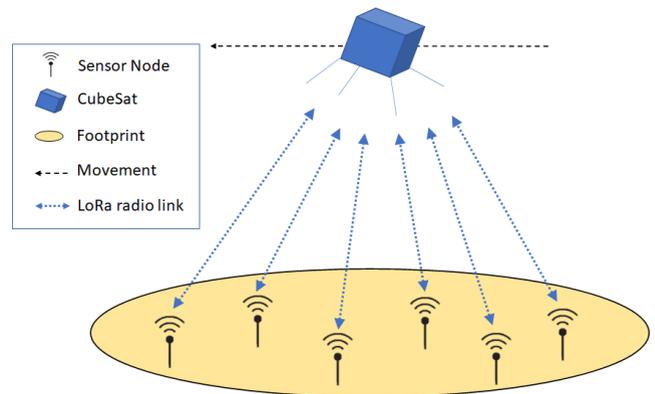


FIGURE 1. System model for Direct-to-Satellite IoT.

IoT nodes and the satellite. In the following sections, we propose a scalable energy-efficient MAC protocol inspired by the Sift protocol. The proposed protocol does not depend on the ground station, does not demand high processing capacity, and does not require channel information in advance. Furthermore, this protocol reduces energy waste in both the spacecraft and the nodes when a large number of nodes are present.

### III. SYSTEM MODEL

The model comprises  $N$  stationary sensor nodes randomly distributed in a region over which a CubeSat nanosatellite in LEO passes, as shown in Fig. 1. The spacecraft has a velocity  $V_s$  and an average orbit height  $h_s$ . Sensor nodes are the only source of useful data traffic, and the traffic model follows a deterministic distribution whose generation pattern is periodic with frequency  $f_{gen}$ . The proposed protocol is designed for telemetry systems, whose applications are tolerant to delays; this allows the nodes to store the generated traffic until it is possible to transmit it to the satellite. Sensor nodes are placed in a square region with side  $L$ , and they keep their radios off while no useful data is available to transmit. The nanosatellite receives these packets, stores them, and downloads them at the closest ground station on its path. Communications between the spacecraft and the nodes are performed using LoRa technology, and the time-varying propagation delays between each node and the nanosatellite are considered.

The proposed protocol model is based on the following assumptions:

- The data packets have fixed and equal lengths.
- The channel is ideal with no capture effect at the receiver. Therefore, a reception is successful only when no interference from other transmissions is present. In other words, a collision occurs if two or more packets overlap in time, in which case, no data is received.
- The nodes have information about the trajectory and the position of the satellite through the frequent update of the satellite ephemeris data [15], [24] or the Keplerian orbital parameters [17].

- The nodes turn on their radios to receive the announcement message from the nanosatellite only when the spacecraft is visible and they have data available to transmit or require synchronization. The sighting instant of the nanosatellite will be an estimated value because the node's clock and the spacecraft's clock may lose synchronization. The de-synchronization effect can be bounded using mechanisms such as the one proposed in [25].

#### A. PHYSICAL LAYER: LoRa

In this work, LoRa technology is assumed at the physical layer for direct communications from sensor nodes to the nanosatellite. LoRa is a technology intended for low-power sensor nodes that contributes to extending the lifetime provided by their batteries. This technology includes a modulation developed by SemTech and a network architecture typically based on a star topology [26]. LoRa modulation is based on Chirp Spread Spectrum, which allows multiple signals to coexist simultaneously on the same physical channel without degrading the quality of the received signal [15], [27].

Signals in LoRa are modulated with orthogonal Spreading Factors (SF) defined with different bit rates. The modulation bit rate and the data signal bit rate are defined by the following equations [28], [29]:

$$R_m = \frac{SF}{(2^{SF}/BW)}, \quad (1)$$

$$R_{data} = \frac{SF}{(2^{SF}/BW)} \times \frac{4}{(4 + CR)}, \quad (2)$$

where  $SF$  is the number of bits per symbol,  $BW$  is the channel bandwidth, and  $CR \in \{1, 2, 3, 4\}$  stands for the different coding rates 4/5, 4/6, 4/7, and 4/8, respectively [26], [30]. LoRa defines a packet format whose duration is defined by [22]:

$$T_{packet} = T_{sym} \times N_{sym}, \quad (3)$$

where

$$T_{sym} = 2^{SF}/BW, \quad (4)$$

where  $T_{sym}$  is the duration of the symbol, and  $N_{sym}$  is the number of symbols that the frame contains.  $N_{sym}$  includes a preamble, a header, and a payload. It is expressed as follows:

$$N_{sym} = (N_{sym}^{preamble} + 4.25) + N_{sym}^{h\&p}, \quad (5)$$

where  $N_{sym}^{preamble}$  is the number of symbols in the preamble.  $N_{sym}^{h\&p}$ , expressed in (6), as shown at the bottom of the page, is the sum of symbols in the header and the payload [22], where  $PL$  is the payload length in bytes;  $CR \in \{0, 1\}$  indicates the presence of an optional 16-bit CRC for the

payload when  $CR = 1$  and the absence of these bits when  $CR = 0$ ;  $IH \in \{0, 1\}$  indicates the presence of the implicit header when  $IH = 1$  and the absence when  $IH = 0$ ;  $DE \in \{0, 1\}$  indicates the activation of Low Data Rate Optimization (LDRO) when  $DE = 1$  and deactivation when  $DE = 0$  [22]. LDRO must be enabled for  $BW = 125$  kHz and SF equal to 11 or 12 [31].

In recent years, LoRa technology has been increasingly considered for the interconnection between CubeSats and stations on the ground. In [32], data transmitted using LoRa from a CubeSat (GPS, altitude, and temperature) to an Internet gateway on the ground allowed real-time visualization and tracking via both TV and Facebook. Also, a study carried out in [33] showed that LoRa modulation has high immunity to the Doppler effect demonstrating the feasibility of using this technology in CubeSat radio communication systems. Furthermore, in [34], the authors describe a mission with a Store-and-Forward communication system that uses LoRa modulation in the TRICOM-1R CubeSat.

#### B. LINK LAYER: SIFT PROTOCOL INSPIRATION

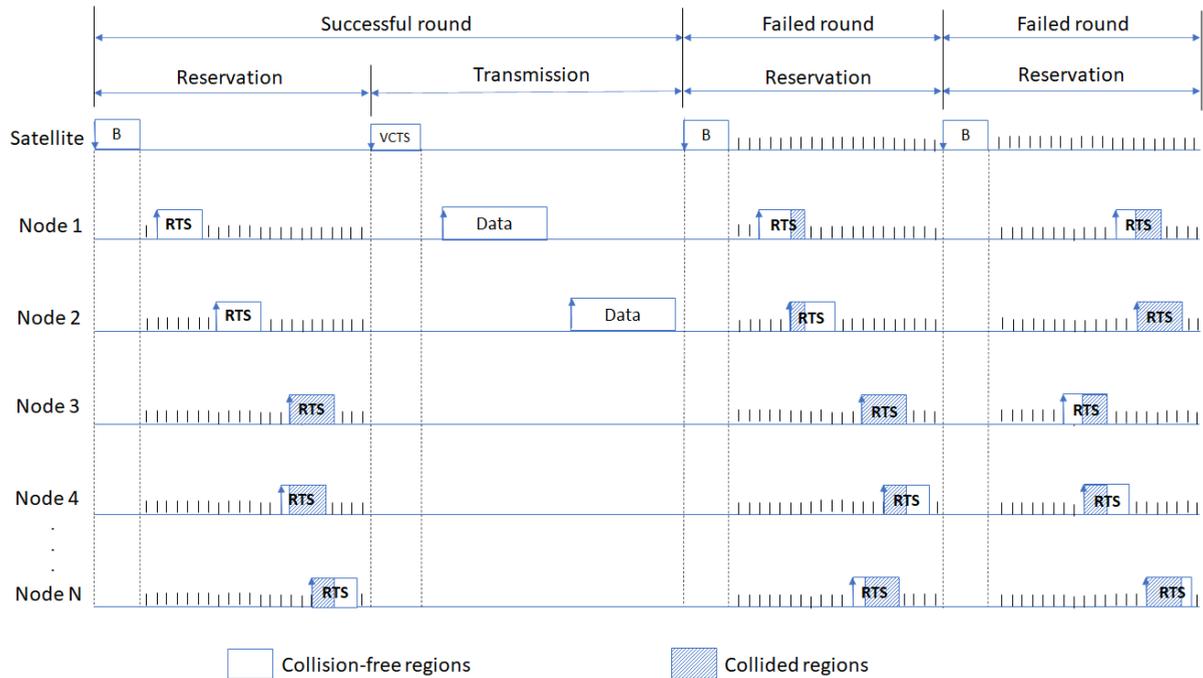
The proposed RESS-IoT MAC protocol is inspired by some of the features of the Sift protocol [21], such as the RTS/CTS frames exchange and the use of a fixed-length slotted time window with contention periods that follow an inverse exponential probability distribution.

However, different from fixed wireless sensor networks, in RESS-IoT, the sink node is replaced with a CubeSat, whose mobility results in limited service periods. Considering this limitation, in RESS-IoT the request frames are not transmitted instantly to the CubeSat as soon as an event is detected because they may occur in the absence of line-of-sight to the satellite. Instead, each sensor node independently extracts data from the environment and keeps it available in a buffer until the satellite's service period starts. As a result, data generation and request-frames transmission become independent processes, which require a different protocol design, discussed in the following section.

#### IV. RESS-IoT PROTOCOL

Our RESS-IoT protocol works through communication rounds, as shown in Fig. 2. Each round is divided into two consecutive phases: a reservation phase and a transmission phase. In the reservation phase, the CubeSat announces its presence to the sensor nodes by means of a beacon frame. Next, each sensor node with data transmits a Request-to-Send (RTS) frame to the spacecraft. In the transmission phase, the satellite assigns channel access to each node whose request frame was received correctly. Finally, these nodes send their data packet to the spacecraft. The details of the phases are presented as follows.

$$N_{sym}^{h\&p} = 8 + \max \left( \text{ceil} \left[ \frac{8 \times PL - 4 \times SF + 28 + 16 \times CR - 20 \times IH}{4 \times (SF - 2 \times DE)} \right], (CR + 4), 0 \right) \quad (6)$$



**FIGURE 2.** Link scheduling for RESS-IoT protocol. The figure illustrates three consecutive rounds where only the first one is successful because the satellite receives two RTS frames without collisions during the reservation phase. In contrast, the last two rounds fail due to collisions among the transmitted RTS frames.

**A. RESERVATION PHASE (RP)**

A round begins with the reservation phase, which comprises the following steps:

1. The satellite broadcasts a beacon frame to the sensor nodes to announce its presence. The beacon frame consists of 2 fields: a field to identify the frame type (i.e., beacon) and a field to identify the spacecraft.
2. After receiving the beacon frame, each node with data in the buffer attempts to reserve a time slot. The reservation period begins when the node creates a slotted time window, whose duration  $T_W$  is fixed. In this window, each node selects a time slot using a reverse exponential probability distribution, whose mean  $\mu$  is calculated by  $\mu = \alpha \times T_W$ . The duration and number of time slots and the  $\alpha$  parameter are selected to adjust guard times among the transmitted RTS frames.
3. After selecting the slot, each node begins transmitting its RTS frame to the satellite when the slot begins. The RTS frame consists of 2 fields: one to identify the frame type (i.e., RTS) and another to identify the sensor node. Sensor nodes keep the radio turned off while they are not transmitting an RTS frame.
4. After sending the beacon frame, the CubeSat remains in listening mode for a period with duration  $W_{rx}$ , which comprises the duration of the slotted time window and a guard time, which was included to consider the variable propagation delays caused by the different distances between the satellite and the nodes. During  $W_{rx}$ , the spacecraft receives RTS frames from nodes.

**B. TRANSMISSION PHASE (TP)**

After completing the reservation phase, the satellite counts the successfully received RTS frames. If all RTS frames collide, the transmission phase does not occur, and the spacecraft immediately resumes another reservation phase by sending a beacon frame. Instead, if at least one RTS frame is received successfully, the satellite starts the data transmission phase by sending a variable-length Clear-to-Send (VCTS) frame. This phase comprises the following steps:

1. The satellite broadcasts a VCTS frame at the beginning of the transmission phase. This frame indicates which nodes successfully transmitted RTS frames and when they should transmit data packets. The VCTS frame consists of 3 fields: one to identify the frame type (i.e., VCTS), another to identify the spacecraft, and a variable-length field containing the identifier code (ID) of each successful sensor node.
2. After the slotted time window ends in the reservation phase, each node begins a listening period. While listening, the node may receive a VCTS frame from the satellite. Upon VCTS reception, each node turns off the radio and examines the frame to determine if it can transmit a data packet. If no RTS frame was successful during the reservation phase, the spacecraft sends a beacon frame instead of a VCTS frame; thus, a new reservation phase begins.
3. When a node finds its ID in the VCTS frame, it proceeds to send a data packet. To determine when to transmit, the node uses the duration of the LoRa packet

whose payload is the data frame (all data frames have fixed and equal lengths) and the position of its ID in the identifier's field that the VCTS frame contains. If a node is placed at the  $p$ -th position, then it must transmit after  $(p - 1)$  data packets are transmitted. To consider the variable propagation delays caused by the different distances between the spacecraft and the nodes, nodes with reservations increase their wait times by adding a guard time to the duration of each of the  $(p - 1)$  data packets. This process allows nodes to transmit the data packets sequentially to the satellite. A node only turns on its radio to transmit the data packet.

4. If a node does not find its ID in the VCTS frame, then its radio remains turned off during the transmission phase. The idle period is determined by multiplying the number of IDs in the VCTS by the packet transmission time plus a guard time. If a node erroneously calculates the idle period during the transmission phase (i.e., incorrectly counts the IDs received or loses some of them), the node may turn on the receiver before the transmission phase ends. This situation may cause energy waste by idle listening but with no harm to the network.
5. Once the transmission phase ends, each node begins a listening period. During this period, the node receives a new beacon frame, which indicates the start of the next reservation phase.
6. After sending the VCTS frame, the CubeSat remains in listening mode for a period with duration  $D_{rx}$ , which is the duration of all data packets plus a guard time per packet. During the listening period, the spacecraft receives data frames from the successful nodes. Once  $D_{rx}$  ends, the satellite transmits a beacon frame to start a new round, as shown in Fig. 2.

### C. ENERGY CONSUMPTION DURING THE RESERVATION PHASE

During this phase, the energy consumed is defined as

$$E^{RP} = E_{sat}^{RP} + E_N^{RP}, \quad (7)$$

where  $E_{sat}^{RP}$  is the energy consumed by the satellite, and  $E_N^{RP}$  is the energy consumed by  $N$  nodes (we assumed  $N$  nodes with available data).  $E_{sat}^{RP}$  is calculated as

$$E_{sat}^{RP} = P_{tx} \times T_{beacon} + P_{rx} \times W_{rx}, \quad (8)$$

where  $P_{tx}$  is the transmit mode power consumption,  $T_{beacon}$  is the duration of the LoRa packet whose payload is the beacon frame, and  $P_{rx}$  is the receive mode power consumption.

On the other hand, the energy consumed by the  $n$ -th node with available data ( $E_n^{RP}$ ) and the energy consumed by  $N$  nodes ( $E_N^{RP}$ ) are calculated as

$$E_n^{RP} = P_{rx} \times T_{beacon} + P_{tx} \times T_{RTS} \quad (9)$$

and

$$E_N^{RP} = N \times E_n^{RP}, \quad (10)$$

where  $T_{RTS}$  is the duration of the LoRa packet whose payload is the RTS frame.

### D. ENERGY CONSUMPTION DURING THE TRANSMISSION PHASE

During this phase, the energy consumed depends on the number of RTS frames that the satellite successfully receives during the reservation phase. As a result, the energy consumed during the transmission phase is different for each round. If no RTS frame is received successfully during the  $r$ -th round, then the round is unsuccessful, the transmission phase does not occur, and the energy consumed during this phase is zero. Conversely, if at least one RTS frame is received successfully during the  $r$ -th round, the round is successful, and the transmission phase occurs.

When the round is successful, the energy consumed during the  $r$ -th round is defined as

$$E^{TP_r} = E_{sat}^{TP_r} + E_N^{TP_r}, \quad (11)$$

where  $E_{sat}^{TP_r}$  is the energy consumed by the satellite, and  $E_N^{TP_r}$  is the energy consumed by  $N$  nodes.  $E_{sat}^{TP_r}$  is calculated as

$$E_{sat}^{TP_r} = P_{tx} \times T_{VCTS}(s_r) + P_{rx} \times D_{rx}, \quad (12)$$

where  $s_r$  is the number of nodes that successfully sent an RTS frame to the spacecraft during the reservation phase in the  $r$ -th round, and  $T_{VCTS}$  is the duration of the LoRa packet whose payload is the VCTS frame. The length of this frame is defined as a function of  $s_r$ .

Furthermore, the energy consumed by the  $n$ -th node ( $E_n^{TP_r}$ ) and the energy consumed by  $N$  nodes ( $E_N^{TP_r}$ ) are calculated as

$$E_n^{TP_r} = P_{rx} \times T_{VCTS}(s_r) + B_{n,r} \times P_{tx} \times T_{data} \quad (13)$$

and

$$E_N^{TP_r} = \sum_{n=1}^N E_n^{TP_r}, \quad (14)$$

where  $T_{data}$  is the duration of the LoRa packet whose payload is the data frame, and  $B_{n,r} \in \{0, 1\}$  indicates whether the  $n$ -th node transmits a data frame in the  $r$ -th round. If  $B_{n,r} = 1$ , then the node sent one of the  $s_r$  successful RTS frames; therefore, the node sends a data frame during the transmission phase. Instead, if  $B_{n,r} = 0$ , the node turns off its radio until the successful nodes send the  $s_r$  data frames.

### E. TOTAL ENERGY CONSUMPTION

The total energy consumed is defined as

$$E^{total} = E_{sat}^{total} + E_N^{total}, \quad (15)$$

where  $E_{sat}^{total}$  is the total energy consumed by the satellite and  $E_N^{total}$  is the total energy consumed by  $N$  nodes.

Considering that each round consists of a reservation phase and a transmission phase,  $E_{sat}^{total}$  is defined as

$$E_{sat}^{total} = R \times E_{sat}^{RP} + \sum_{r=1}^R E_{sat}^{TP_r}, \quad (16)$$

where  $R$  is the number of rounds.

Similarly, the total energy consumed by the  $n$ -th node ( $E_n^{total}$ ) and the total energy consumed by  $N$  nodes ( $E_N^{total}$ ) are defined as

$$E_n^{total} = R \times E_n^{RP} + \sum_{r=1}^R E_n^{TP_r} \quad (17)$$

and

$$E_N^{total} = R \times E_N^{RP} + \sum_{r=1}^R E_N^{TP_r}. \quad (18)$$

## V. RESULTS AND DISCUSSION

In this section, we provide a comparative performance evaluation of the RESS-IoT protocol in terms of energy efficiency, throughput, and fairness, in a DtS-IoT system using LoRa technology for communication between sensor nodes and the nanosatellite. The energy efficiency of the proposed RESS-IoT is first evaluated using the analytical model presented in Section IV. For comparative purposes, we simulated the Enhanced Aloha (EA) protocol [19] and the Unconfirmed Class A LoRaWAN (UCAL) protocol [22].

Although the UCAL protocol, as part of LoRaWAN, is widely used for terrestrial IoT networks, it is also used in satellite IoT systems. For example, Lacuna Space operates its satellite IoT system using LoRaWAN to provide satellite connectivity to low-cost devices [35], [36]. On the other hand, the EA protocol is the MAC protocol used by the Argos satellite system. Both these protocols were selected for comparison because they are actively used in current satellite IoT systems and are energy-efficient when the number of nodes in the network is small. MAC protocols that include interference cancellation or similar mechanisms were not considered due to their high demand for channel/processing resources [23]. When simulating UCAL, we assumed all nodes had already joined the network, and there was no presence of downlink transmissions [37].

Table 1 lists the LoRa Modem parameters used in both the analytical evaluation and the simulations (i.e., modulation bit rate, data signal bit rate, LoRa packet length, LoRa packet duration) using (2) to (6). As for the modem power consumption, values correspond to the recently introduced Semtech LR1110 transceiver [38]. The values were obtained considering typical current values and a voltage of 3.3 V for 25°C. We assumed that both ends of the LoRa link had identical transceivers. Furthermore, we assumed that during the simulation time, the nanosatellite and the nodes have line-of-sight through which the nodes have coverage of the satellite signal and vice versa.

TABLE 1. LoRa modem and LR1110 transceiver parameters.

LoRa modem		Values	
Bandwidth		125 kHz	
Number of symbols in the preamble		8	
CR		1	
CRC		1	
IH		0	
Spreading Factor		10	12
DE		0	1
Modulation Bit Rate		1221 bps	367 bps
Data Signal Bit Rate		977 bps	293 bps
LoRa packet duration	PL = 3 B	206.84 ms	827.39 ms
	PL = 5.5 B	247.80 ms	827.39 ms
	PL = 8 B	247.80 ms	991.23 ms
	PL = 10.5 B	288.76 ms	991.23 ms
	PL = 63 B	698.36 ms	2793.47 ms
LoRa packet length	PL = 63 B	107 B	128 B
<b>LR1110 transceiver</b>			
Power consumption in reception		25.74 mW	
Power consumption in transmission		389.4 mW	

Results were plotted using a base 10 logarithmic scale on the x-axis and the y-axis. Accordingly, values equal to 0 are not plotted on the graph.

### A. PERFORMANCE METRICS

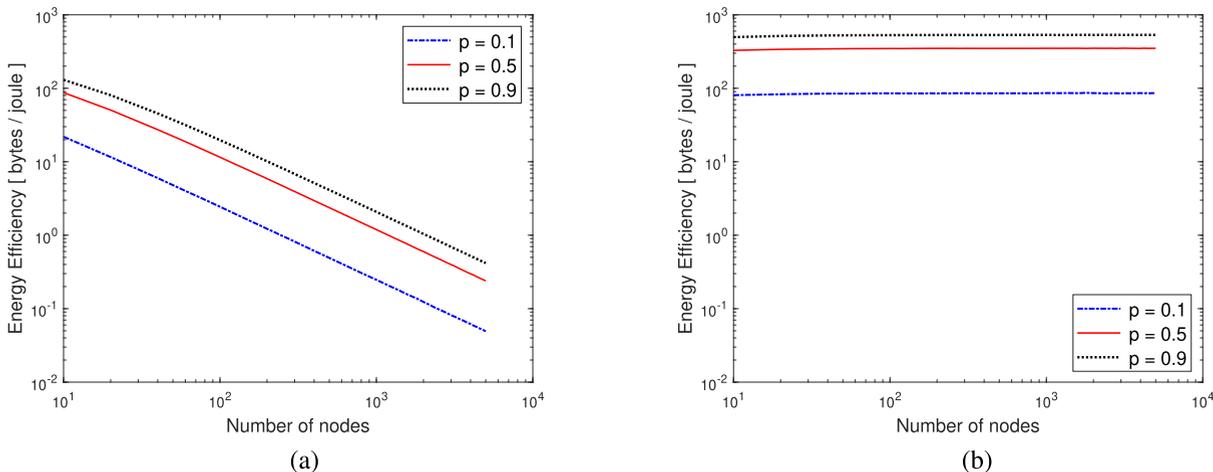
To evaluate the protocol performance, we employ three metrics: energy efficiency, throughput, and fairness. Energy efficiency is calculated as

$$\xi = \frac{K}{E}, \quad (19)$$

where  $K$  is the number of bits contained in useful data packets that are successfully received by the satellite. In this work, “useful data packets” do not consider any control/overhead packet used by the protocols under evaluation, such as RTS, VCTS, or beacon packets.  $E$  is the total energy consumed by the transceiver during the transfer process (i.e.,  $E$  is the energy consumed by the reception/transmission periods of both useful data packets and control/overhead packets, as described in the energy consumption model shown in sections IV-C and IV-D).

Based on this,  $K$  and  $E$  are calculated in different ways. At the network level,  $K$  corresponds to useful data packets successfully received by the satellite, and  $E$  corresponds to the total energy consumed by nodes and the satellite, as expressed in (15). For the satellite,  $K$  corresponds to useful data packets successfully received by the satellite, and  $E$  corresponds to the total energy consumed by the satellite, as expressed in (16). For each node,  $K$  corresponds to its useful data packets successfully transmitted to the spacecraft, and  $E$  corresponds to the total energy consumed by the node, as expressed in (17).

The average energy efficiency per node is obtained by calculating the energy efficiency of each node and averaging the calculated efficiencies. The average energy efficiency per



**FIGURE 3.** Energy efficiency of the RESS-IoT protocol obtained through analytical expressions: (a) average energy efficiency per node and (b) energy efficiency for satellite.

node is calculated as

$$\xi_{node} = \frac{1}{N} \times \sum_{n=1}^N \xi_n, \tag{20}$$

where  $\xi_n$  is the energy efficiency of the  $n$ -th node.

Throughput is calculated as

$$S = \frac{D}{T}, \tag{21}$$

where  $D$  is the number of bits contained in the useful data packets successfully received by the satellite during a period  $T$ .

Fairness is defined to quantify how fair a system is in allocating resources. In this work, fairness quantifies how fair the proposed protocol is when it allocates access to the channel for nodes to transmit useful data packets. As a fairness metric, we have used the Jain’s Fairness Index, which is calculated as

$$JFI = \frac{(\sum_{n=1}^N x_n)^2}{N \times \sum_{n=1}^N x_n^2}, \tag{22}$$

where  $N$  is the number of nodes, and  $x_n$  is the number of useful data packets that the  $n$ -th node successfully transmits to the satellite.

### B. NUMERICAL RESULTS FOR ENERGY EFFICIENCY

Energy efficiency was calculated as a function of the number of sensor nodes in the network. Additionally, different values were assigned to the probability of a successful round ( $p$ ) to analyze the behavior of the calculated energy efficiency. When  $p = 0.3$ , the spacecraft correctly receives RTS frames in 3 out of 10 rounds. During each successful round, the maximum number of IDs that a VCTS frame can contain was limited to 3 because the probability of receiving more than 3 RTS frames without collisions is very low. As a result, during each successful round, the number of useful data packets received by the satellite was assigned by a random variable

**TABLE 2.** Parameters for numerical results.

Parameters	Values
Spreading Factor	10
Number of rounds	1000
Number of nodes	10 to 5000
Number of repetitions	100
Success round probabilities	0.1, 0.5, 0.9

uniformly distributed among 1, 2, and 3. Table 2 shows the parameters used during calculations.

Fig. 3a shows the expected behavior of energy efficiency, whose value decreases when the number of collisions among RTS frames increases. The collisions occur because the number of nodes that send RTS frames during the slotted time window increases. As a result, nodes that do not successfully transmit useful data packets obtain an energy efficiency equal to zero. Consequently, the average energy efficiency per node decreases because the number of nodes whose energy efficiency is zero increases. Instead, Fig. 3b shows that the energy efficiency remains almost constant for the spacecraft. This behavior results from receiving similar amounts of useful data packets even though the number of nodes in the terrestrial segment varies. In both terrestrial and space segments, the increase of  $p$  implies an increase in energy efficiency because the number of rounds with no data transmitted decreases.

### C. SIMULATION RESULTS

To recreate a DtS-IoT scenario, we implemented a simulation of a nanosatellite that flies over a region of South America, where a high number of nodes were placed, as shown in Fig. 4. The selected region includes parts of Chile and Argentina, over which the satellite moves following a polar LEO from south to north for ten minutes. The STK software was used to design the simulation scenario with the parameters shown in Table 3. A video that clarifies the dynamics of the scenario is available in [39].

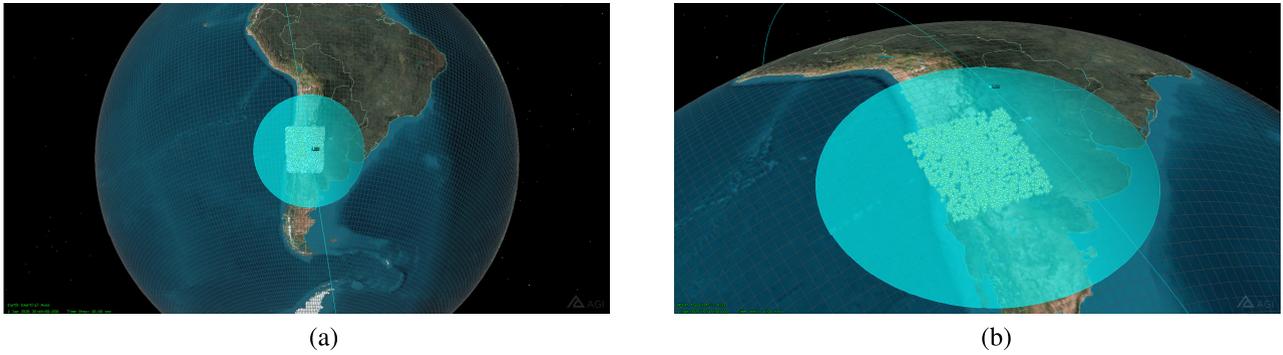


FIGURE 4. Views of the simulation scenario, which includes parts of Chile and Argentina in the South American region.

TABLE 3. Parameters for simulation results.

Simulation		Values
Number of nodes		5000
Number of repetitions		30
Simulation time		600 s
Data generation frequency		0.01 Hz
Scenario		
Dimensions of the region		960 x 960 km
Satellite	Speed	8 km/s
	Altitude	600 km
	Epoch	1 Jan 2020
	Mean Anomaly	0°
	Argument of Perigee	0°
	Orbital Inclination	98°
	Right Ascension of Ascending Node	340°

TABLE 4. Protocols parameters.

RESS-IoT	Values	
Spreading Factor	10	12
Reverse exponential distribution mean	0.691 s	2.566 s
Slot duration	57.55 ms	213.81 ms
Number of slots	100	
$\alpha$ parameter	0.12	
Maximum number of reservations allowed per transmission phase	3	
VCTS frame length (for 1 ID)	5.5 B	
VCTS frame length (for 2 IDs)	8 B	
VCTS frame length (for 3 IDs)	10.5 B	
Beacon frame length	3 B	
RTS frame length	3 B	
Data frame length	63 B [37]	
UCAL		
Duration of each reception window	401.4 ms [22], [40]	
Duty cycle relative to the sub-band	1% [22], [40]	
Data frame length	63 B [37]	
EA		
Random level of the transmission interval	22.3 % [19]	89.4 % [19]
Average transmission interval	100 s [19]	
Data frame length	63 B [37]	

For the simulation of MAC protocols, we implemented the logic in the OMNeT++ network simulator using the parameters shown in Tables 3 and 4. For simplicity, the protocols were simulated using only 2 logical channels, corresponding to SF10 and SF12. The selected spreading factors

provide the most robust conditions for transmission but are also the most demanding in terms of energy consumption.

As for fairness, Fig. 5 shows that for the proposed protocol, fairness decreases when the number of nodes increases because more RTS packets collide, whose source nodes are not successful in being allocated a frame to transmit a useful data packet. Thus, fewer nodes send useful data packets when the number of nodes increases. Nevertheless, RESS-IoT provides more transmission opportunities, and therefore more successful transmissions, to all nodes when compared to the state-of-the-art protocols.

As for the throughput, Fig. 6 shows that the proposed protocol outperforms the state-of-the-art protocols. When using SF12 (see Fig. 6a), RESS-IoT has better performance for networks with more than 30 nodes when compared to EA; UCAL shows the lowest throughput. In the case of SF10 (see Fig. 6b), RESS-IoT has a comparable performance for less than 100 nodes, but it outperforms the other protocols for larger network sizes. For a 100-node network (SF12), RESS-IoT achieved 24.7 kB/h, EA achieved 2.68 kB/h, and UCAL fell to 0 kB/h. Accordingly, the throughput obtained by our proposed protocol represents an increase of 822% with respect to EA. In the case of a 1000-node network (SF10), the proposed protocol achieved 78.5 kB/h, EA achieved 7.65 kB/h, and UCAL achieved 0 kB/h. Based on this, the throughput obtained by RESS-IoT represents an increase of 926% with respect to EA. These percentages continue increasing for networks of up to 5000 nodes, where the comparison protocols achieved even smaller throughput values.

Additionally, the results that are shown in Fig.6 include a theoretical maximum throughput. Since data transfers only happen during the successful rounds in RESS-IoT, we consider a maximum amount of data transferred during each transmission phase to calculate the maximum throughput. As a result, we assume the VCTS frame always announces the maximum number of reservations allowed per transmission phase, thus enabling the transmission of the maximum amount of data allowed per round. In the calculation, we assume the duration of a successful round is minimum. Therefore, the reception periods of the satellite do not include guard times to simulate a continuous reception of useful

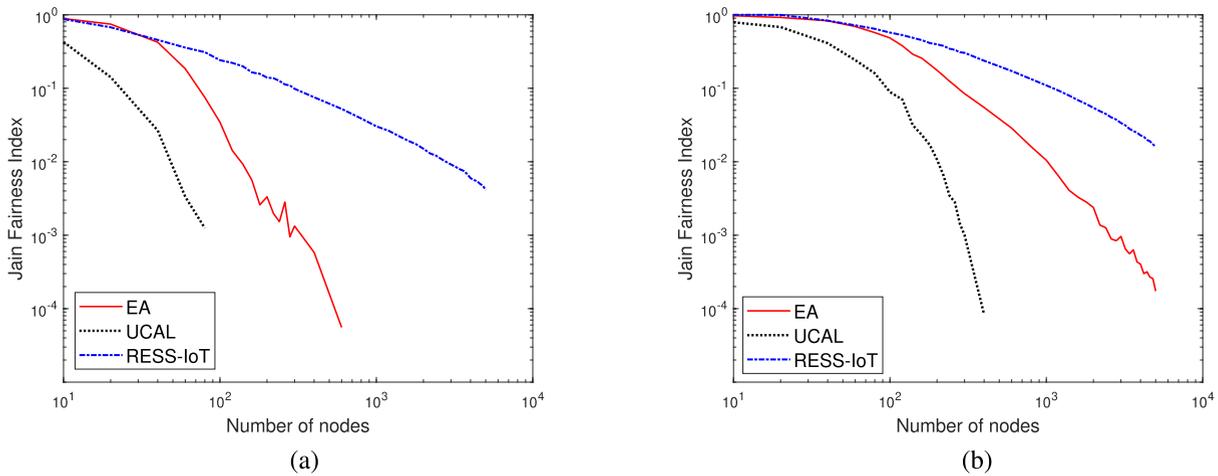


FIGURE 5. Jain fairness index: (a) SF12, and (b) SF10.

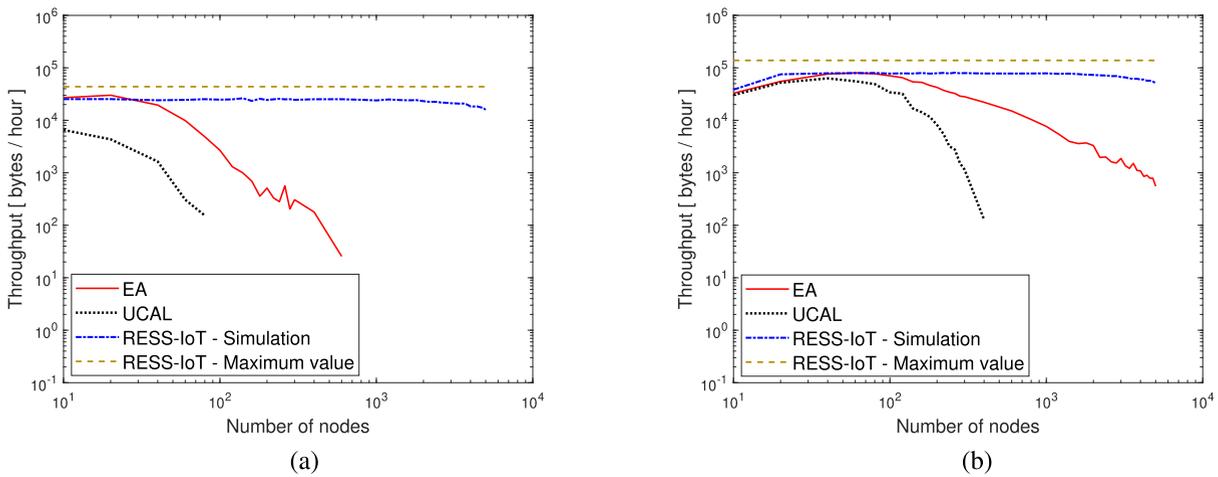


FIGURE 6. Throughput: (a) SF12, and (b) SF10.

TABLE 5. Calculation of maximum throughput.

Parameters	Values	
Spreading Factor	10	12
$D_{max}$ [B]	321	384
$T_{min}$ [ms]	8345.68	31580.03
<b>Maximum throughput</b>		
$S_{max}$ [B/s]	38.46	12.15
$S_{max}$ [kB/h]	138.46	43.77

data packets. Table 5 shows the maximum throughput calculated when using the simulation parameters of Tables 1 and 4.

Figures 7 and 8 show the comparison among the energy efficiencies obtained with UCAL, EA, and RESS-IoT operating in the DtS-IoT scenario. The most energy-efficient protocol is the one that allows the satellite to receive more bytes of data per joule of energy.

In the space segment, Fig. 7 shows that the efficiency obtained by the satellite when using the proposed

protocol outperforms the state-of-the-art protocols starting from network sizes of 50 nodes (SF12) and 200 nodes (SF10). For example, in a network of 100 nodes (SF12), the proposed protocol achieved 147.2 bytes/joule, EA achieved 28.9 bytes/joule, and UCAL fell to 0 bytes/joule. Accordingly, the efficiency obtained by RESS-IoT represents an increase of 409% with respect to EA. In a 1000-node network (SF10), the proposed protocol achieved 473.9 bytes/joule, EA achieved 82.5 bytes/joule, and UCAL fell to 0 bytes/joule. In the latter case, the efficiency obtained by RESS-IoT represents an increase of 474% with respect to EA.

In the terrestrial segment, Fig. 8 shows that the efficiency obtained by the nodes when using the proposed protocol outperforms the state-of-the-art protocols starting from network sizes of 40 nodes (SF12) and 450 nodes (SF10). For a 100-node network (SF12), RESS-IoT achieved 4.9 bytes/joule per node, EA achieved 0.6 bytes/joule per node, and UCAL fell to 0 bytes/joule per node. Accordingly, the efficiency per node obtained by the proposed protocol represents an

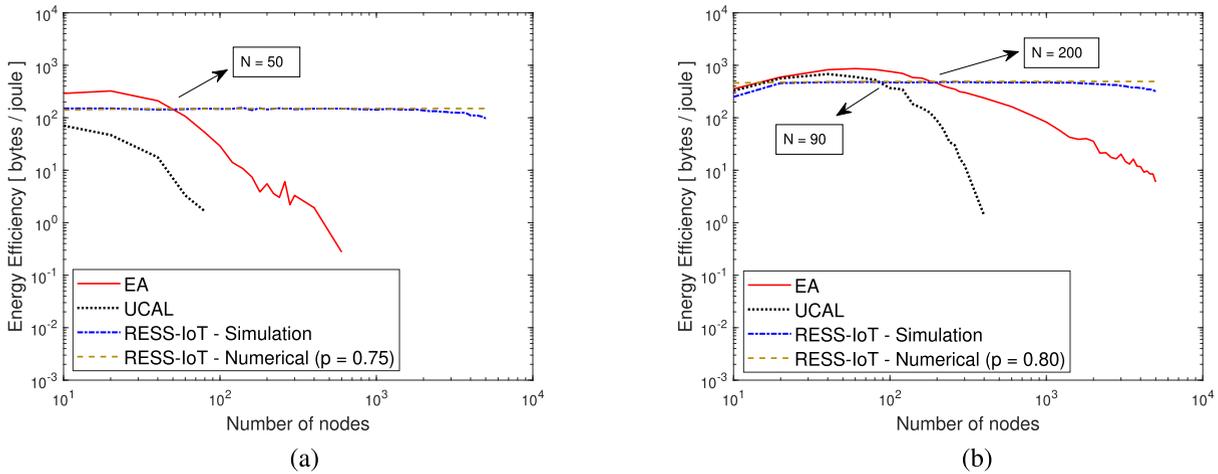


FIGURE 7. Energy efficiency obtained by the satellite: (a) SF12, and (b) SF10.

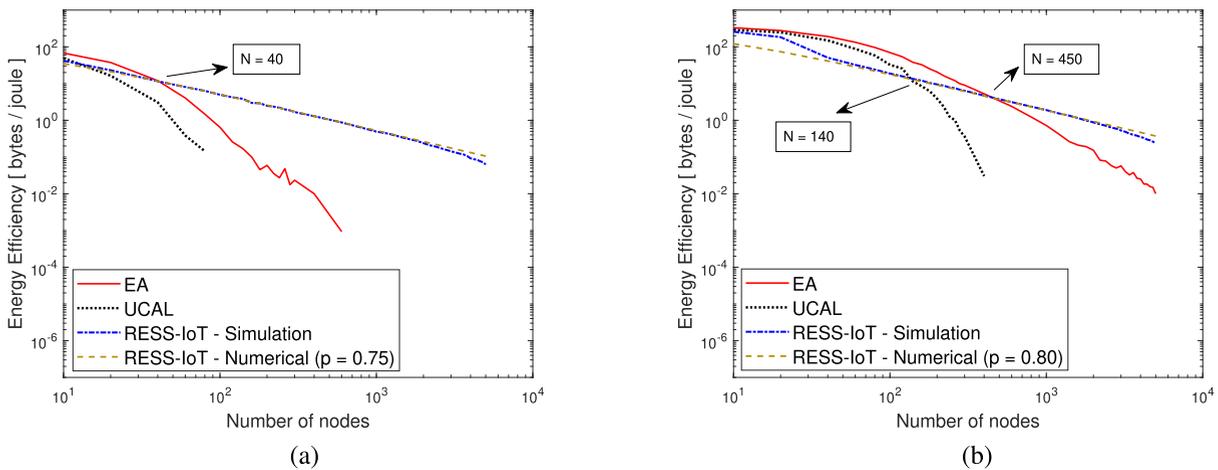


FIGURE 8. Average energy efficiency per node: (a) SF12, and (b) SF10.

increase of 716% with respect to EA. In the case of a 1000-node network (SF10), the proposed protocol achieved 1.8 bytes/joule per node, EA achieved 0.7 bytes/joule per node, and UCAL achieved 0 bytes/joule per node. Based on this, the efficiency per node obtained by RESS-IoT represents an increase of 157% with respect to EA. These percentages continue increasing for networks of up to 5000 nodes, where the comparison protocols achieved even smaller energy efficiency values.

The small bump in Fig. 8b is due to the small number of nodes that make up the network (i.e., from 10 to 20 nodes). When this small number of nodes send RTS frames to the satellite, most of these nodes transmit their useful data packets and obtain energy efficiency values different from zero. Instead, from 20 nodes and higher, the situation changes abruptly because the number of nodes sending RTS frames that cannot successfully transmit useful data packets increases sharply. For these nodes, the energy efficiency is zero; therefore, the average energy efficiency per node is

calculated with a high number of nodes whose energy efficiency is zero. As a result, the average energy efficiency per node decreases sharply.

As for the impact of the physical layer configuration, the energy efficiency achieved by all protocols increases when the bit rate increases from SF12 to SF10. Also, Figures 7 and 8 show the values that result from the proposed energy efficiency model, where the most adjusted successful round probabilities are 0.75 for SF12 and 0.80 for SF10.

The results indicate that the CubeSat, whose function is to collect useful data packets, is more energy efficient for networks with a high number of nodes if the RESS-IoT protocol is implemented in the DtS-IoT scenario. Similar behavior is observed from results obtained for the sensor nodes. Nevertheless, when the bit rate increases (i.e., when using SF10 instead of SF12), the duration of the transmitted packets decreases; therefore, the number of collisions among useful data packets decreases for both the EA protocol and the UCAL protocol. The decrease improves the energy efficiency

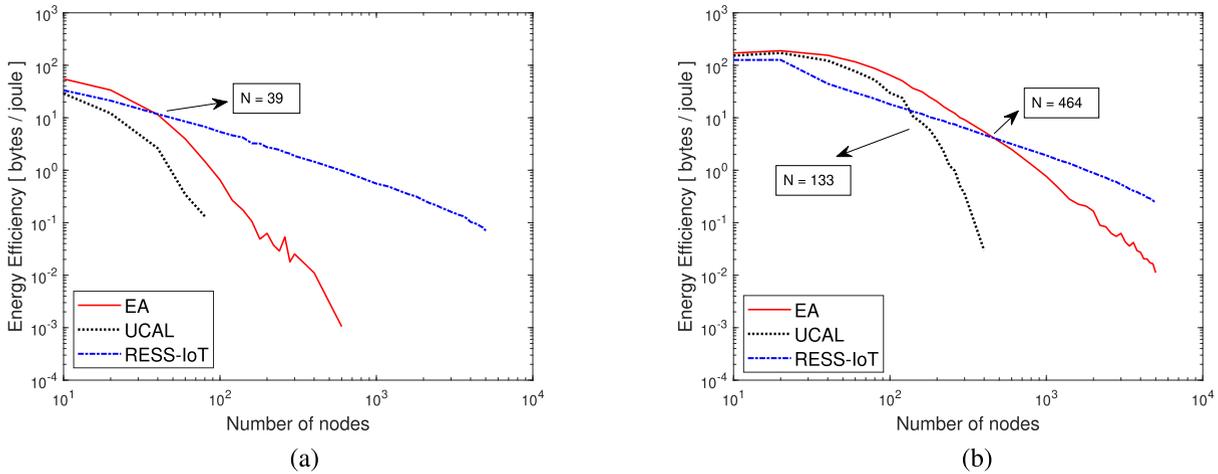


FIGURE 9. Energy efficiency achieved by the network: (a) SF12, and (b) SF10.

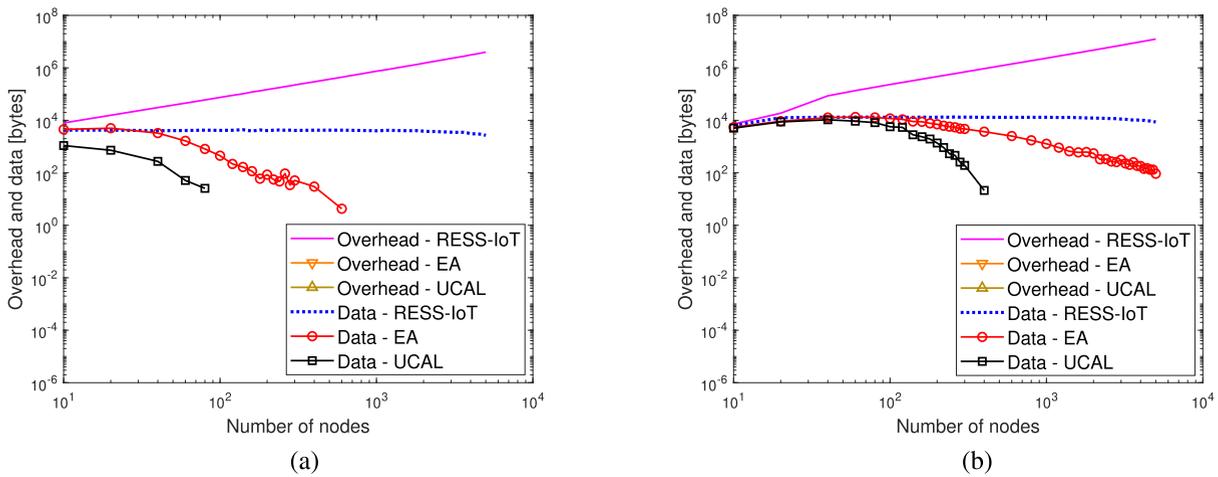


FIGURE 10. Overhead (control packets transmitted in the network) and data (useful data packets successfully received by the satellite): (a) SF12, and (b) SF10.

of both protocols because the satellite receives more bits of data without collisions.

Regarding the network, Fig. 9 shows that the energy efficiency decreases because the overhead increases without achieving an increase in the number of useful data packets received successfully, as shown by the trends of the pink and blue curves in Fig. 10. In addition, the decrease in energy efficiency of the network is similar to the decrease in the average energy efficiency per node. (see Fig. 8). Since the nodes constitute the segment of the network that transmits the most overhead (i.e., RTS packets), the energy consumed by this segment is the predominant component in the energy consumed by the network.

The above-mentioned results show that the RESS-IoT is more efficient while providing scalability to the DtS-IoT network. However, there is a tradeoff of energy efficiency and scalability in the overhead. RESS-IoT requires control packets to be transmitted, which is not the case for the other two protocols, as shown in Fig. 10 (with logarithmic axes, values

equal to zero cannot be graphed). The lack of control packets in the contention-based protocols reduces energy waste due to overhead. As a result, these protocols achieve higher energy efficiency values than in RESS-IoT as long as the number of nodes involved is small. However, when the number of nodes increases to large values, EA and UCAL experience a high number of collisions since there is no coordination for data transmission attempts, causing RESS-IoT to outperform the protocols under comparison in terms of energy efficiency.

Although the simulations consider the time-varying propagation delays that the satellite’s movement causes, we do not consider the variations on the ground track of the satellite (i.e., the projection on the ground of the satellite’s path) or the different link durations observed by the set of nodes in the footprint. These simplifications do not significantly impact the results obtained since the scenario reflects a straightforward but representative approach in which there are always nodes entering/leaving the satellite coverage. Furthermore, the different link durations observed by each node can be

compensated for in the successive passes of the satellite; in a practical network, a typical LEO provides different line-of-sight angles between the satellite and the nodes in each pass [41]. To improve future simulations, we will address these cases through heterogeneously distributed nodes on the ground and will consider line-of-sight angles that generate different link durations. Additionally, to improve the accuracy of the proposed model, we will include a probabilistic model to describe the successful reception of RTS frames as a function of the number of nodes.

## VI. CONCLUSION

In this work, we proposed a scalable MAC protocol called RESS-IoT, designed to increase energy efficiency in a Direct-to-Satellite IoT scenario, where sensor nodes connect directly to a CubeSat using LoRa technology. The proposed protocol uses a variable-length Clear-to-Send frame and reduces the collisions among useful data packets when the number of nodes is high. The results show that the proposed protocol is more energy-efficient than the Enhanced Aloha protocol and the Unconfirmed Class A LoRaWAN protocol for networks with a high number of nodes. In this context, the proposed protocol is more energy-efficient despite having more overhead than the other protocols evaluated, whose number of control packets is zero. As part of our future work, we will consider different line-of-sight angles and improve the simulations by including heterogeneously distributed nodes on the ground. In addition, we will include a probabilistic model to improve the accuracy of the proposed model.

## REFERENCES

- [1] S. Li, L. Xu, and S. Zhao, "The Internet of Things: A survey," *Inf. Syst. Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.
- [2] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generat. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, 2013.
- [3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.
- [4] Z. Qu, G. Zhang, H. Cao, and J. Xie, "LEO satellite constellation for Internet of Things," *IEEE Access*, vol. 5, pp. 18391–18401, 2017.
- [5] California Polytechnic State University and Stanford University's Space Systems Development Lab. *Origin of the New Space Revolution*. Accessed: Jan. 2, 2021. [Online]. Available: <https://www.cubesat.org>
- [6] Alén Space. *Guía Básica de Nanosatélites*. Accessed: Jul. 14, 2020. [Online]. Available: <https://alen.space/es/guia-basica-nanosatelites>
- [7] O. Popescu, "Power budgets for cubesat radios to support ground communications and inter-satellite links," *IEEE Access*, vol. 5, pp. 12618–12625, 2017.
- [8] A. Khalifeh, K. A. Aldahdouh, K. A. Darabkh, and W. Al-Sit, "A survey of 5G emerging wireless technologies featuring LoRaWAN, sigfox, NB-IoT and LTE-M," in *Proc. Int. Conf. Wireless Commun. Signal Process. Netw. (WiSPNET)*, Mar. 2019, pp. 561–566.
- [9] M. Doudou, D. Djenouri, N. Badache, and A. Bouabdallah, "Synchronous contention-based MAC protocols for delay-sensitive wireless sensor networks: A review and taxonomy," *J. Netw. Comput. Appl.*, vol. 38, pp. 172–184, Feb. 2014.
- [10] J. A. Fraire, S. Céspedes, and N. Accettura, "Direct-to-satellite IoT—A survey of the state of the art and future research perspectives," in *Proc. Int. Conf. Ad-Hoc Netw. Wireless Cham, Switzerland: Springer*, 2019, pp. 241–258.
- [11] Q. Li, Y. Lan, D. Lu, and J. Sun, "Research on improved S-MAC energy conservation based on adaptive mechanism," in *Proc. IEEE 19th Int. Conf. Commun. Technol. (ICCT)*, Oct. 2019, pp. 962–966.
- [12] A. Rajasekaran and V. Nagarajan, "Adaptive intelligent hybrid MAC protocol for wireless sensor network," in *Proc. Int. Conf. Commun. Signal Process. (ICCSPP)*, Apr. 2016, pp. 2284–2289.
- [13] T. Deng, J. Zhu, and Z. Nie, "An improved LoRaWAN protocol based on adaptive duty cycle," in *Proc. IEEE 3rd Inf. Technol. Mechatronics Eng. Conf. (ITOEC)*, Oct. 2017, pp. 1122–1125.
- [14] B. Denby and B. Lucia, "Orbital edge computing: Nanosatellite constellations as a new class of computer system," in *Proc. 25th Int. Conf. Architectural Program. Lang. Operating Syst.*, Mar. 2020, pp. 939–954.
- [15] T. Deng, J. Zhu, and Z. Nie, "An adaptive MAC protocol for SDCS system based on LoRa technology," in *Proc. 2nd Int. Conf. Autom., Mech. Control Comput. Eng. (AMCCE)*, 2017, pp. 825–830.
- [16] T. P. F. Briceño, "Diseño y evaluación de protocolo de acceso múltiple para sistema de IoT nanosatelital," Bachelor's thesis, Facultad de Ciencias Físicas y Matemáticas, Departamento de Ingeniería Eléctrica, Universidad de Chile, Santiago, Chile, 2019.
- [17] A. Addaim, A. Kherras, and Z. Guennoun, "Enhanced MAC protocol for designing a wireless sensor network based on a single LEO Picosatellite," *Int. J. Sens. Netw.*, vol. 23, no. 3, pp. 143–154, 2017.
- [18] V. Almonacid and L. Franck, "Extending the coverage of the Internet of Things with low-cost nanosatellite networks," *Acta Astronautica*, vol. 138, pp. 95–101, Sep. 2017.
- [19] H. Ma and L. Cai, "Performance analysis of randomized MAC for satellite telemetry systems," in *Proc. 5th Int. Conf. Commun. Netw. China (ICST)*, 2010, pp. 1–5.
- [20] P. S. Lakshmi and M. G. Jibukumar, "Energy efficient MAC protocol for lifetime maximization in wireless sensor networks," in *Proc. IEEE 13th Int. Conf. Ind. Inf. Syst. (ICIIS)*, Dec. 2018, pp. 191–195.
- [21] K. Jamieson, H. Balakrishnan, and Y. Tay, "Sift: A MAC protocol for event-driven wireless sensor networks," in *Proc. Eur. Work. Wireless Sens. Netw.* Berlin, Germany: Springer, 2006, pp. 260–275.
- [22] M. A. Ertürk, M. A. Aydın, M. T. Büyükkaktaşlar, and H. Evirgen, "A survey on LoRaWAN architecture, protocol and technologies," *Future Internet*, vol. 11, no. 10, p. 216, Oct. 2019.
- [23] T. Ferrer, S. Céspedes, and A. Becerra, "Review and evaluation of MAC protocols for satellite IoT systems using nanosatellites," *Sensors*, vol. 19, no. 8, p. 1947, Apr. 2019.
- [24] Semtech Corporation. *Update the GNSS Almanac on the LR1110*. Accessed: May 3, 2021. [Online]. Available: <https://loro-developers.semtech.com/learning-center/hands-on-labs/build-end-to-end-solution-using-lorawan-and-loraedge/update-the-gnss-almanac-on-the-lr1110>
- [25] J. A. Fraire, S. Henn, F. Dovis, R. Garello, and G. Taricco, "Sparse satellite constellation design for LoRa-based direct-to-satellite Internet of Things," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.
- [26] A. Augustin, J. Yi, T. Clausen, and W. Townsley, "A study of LoRa: Long range & low power networks for the Internet of Things," *Sensors*, vol. 16, no. 9, p. 1466, Sep. 2016.
- [27] Z. Xuan, Z. Xiaojie, W. Shuo, and Z. Ye, "An attempt to utilize LoRa for intersatellite communications," in *Proc. 10th Int. Conf. Commun., Circuits Syst. (ICCCAS)*, Dec. 2018, pp. 260–264.
- [28] R. Sharan Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *ICT Exp.*, vol. 3, no. 1, pp. 14–21, Mar. 2017.
- [29] Semtech Corporation. *LoRaT Modulation Basics, Application Note AN1200.22*. Accessed: May 28, 2020. [Online]. Available: <https://www.semtech.com/products/wireless-RF/loro-transceivers/sx1272>
- [30] K. Mikhaylov, J. Petaejaervi, and T. Haenninen, "Analysis of capacity and scalability of the LoRa low power wide area network technology," in *Proc. 22th Eur. Wireless Conf.*, 2016, pp. 1–6.
- [31] J. M. Marais, R. Malekian, and A. M. Abu-Mahfouz, "LoRa and LoRaWAN testbeds: A review," in *Proc. IEEE AFRICON*, Sep. 2017, pp. 1496–1501.
- [32] P. Techavijit, P. Sukchalern, N. Wongphuangfuthaworn, A. Plodpai, S. Manuthasna, and S. Chivapreecha, "Internet network in space for small satellites: Concept and experiments," *Sensors Mater.*, vol. 30, no. 10, pp. 2295–2311, 2018.
- [33] A. A. Doroshkin, A. M. Zadorozhny, O. N. Kus, V. Y. Prokopyev, and Y. M. Prokopyev, "Experimental study of LoRa modulation immunity to Doppler effect in CubeSat radio communications," *IEEE Access*, vol. 7, pp. 75721–75731, 2019.

- [34] Q. Verspieren, T. Matsumoto, Y. Aoyanagi, T. Fukuyo, T. Obata, S. Nakasuka, G. Kwizera, and J. Abakunda, "Store-and-forward 3U Cube-Sat project TRICOM and its utilization for development and education: The cases of TRICOM-1R and JPRWASAT," *Trans. Jpn. Soc. Aeronaut. SPACE Sci.*, vol. 63, no. 5, pp. 206–211, 2020.
- [35] (2021). *Lacuna Space Ltd*. Accessed: Sep. 30, 2021. [Online]. Available: <https://lacuna.space/first-successful-lacunasat-launch-in-2021/>
- [36] The Things Network. *LoRaWAN Connectivity Everywhere, From Space—Thomas Telkamp (Lacuna Space)*. Accessed: May 10, 2021. [Online]. Available: <https://www.youtube.com/watch?v=5Pofqck4gd0>
- [37] R. B. Sørensen, D. M. Kim, J. J. Nielsen, and P. Popovski, "Analysis of latency and MAC-layer performance for class a LoRaWAN," *IEEE Wireless Commun. Lett.*, vol. 6, no. 5, pp. 566–569, Oct. 2017.
- [38] Semtech Corporation. *LR1110 Data Sheet*. Accessed: May 10, 2021. [Online]. Available: <https://www.semtech.com/products/wireless-RF/lora-transceivers/lr1110>
- [39] J. Fraire. *Direct-to-Satellite IoT-MAC Protocol Study Scenario*. Accessed: Dec. 5, 2020. [Online]. Available: <https://www.youtube.com/watch?v=RNT9ItnFwY8&feature=youtu.be>
- [40] LoRa Alliance. *LoRaWAN Specification V1.0*. Accessed: Mar. 30, 2020. [Online]. Available: <https://lora-alliance.org/resource-hub/lorawan-specification-v100>
- [41] H. Chelle, M. Crosnier, V. Deslandes, R. Dhaouz, and A.-L. Beylot, "Modelling discontinuous LEO satellite constellations: Impact on the machine-to-machine traffic and performance evaluation," in *Proc. 8th Adv. Satell. Multimedia Syst. Conf., 14th Signal Process. Space Commun. Workshop (ASMS/SPSC)*, Sep. 2016, pp. 1–7.



**ALEX BECERRA** received the B.S. degree in electrical engineering from the University of Chile, Santiago, Chile, in 2010. From 2010 to 2017, he worked as a Project Engineer at the Space and Planetary Exploration Laboratory (SPEL), University of Chile, leading the design, manufacturing, assembly, testing, and operation of the first nanosatellite—SUCHAI—developed in Chile. In 2015, he was part of the team that founded Aurora Space. In that company, he has developed projects in space technology, the mining industry, the IoT connectivity, and agriculture. His research interests include project management, printed circuit board design and testing, software development, and communication link evaluation and testing.



**TOMÁS FERRER** received the B.S. degree (Hons.) in electrical civil engineering from the University of Chile, in 2019. In this university, he was a member of the WiNet Group, where he worked on the design and study of communication protocols for nanosatellite networks. His results in this group were presented in the 32nd Annual Small Satellite Conference and published in the *Sensors* journal. He is currently an Operations Supervisor with the Chuquicamata Division, Codelco. His research interest includes the application of engineering in sustainable technological development in the face of current challenges.



**RAYDEL ORTIGUEIRA** received the B.S. degree in telecommunications and electronic engineering from the Higher Polytechnic Institute "José Antonio Echeverría," currently the Technological University of Havana (CUJAE), in 2013. He is currently pursuing the Ph.D. degree in satellite communication with the Department of Electrical Engineering, University of Chile. From 2013 to 2018, he worked as a Professor with the Department of Telecommunications and Telematics, CUJAE. His research interest includes medium access control protocols for Direct-To-Satellite IoT communications.



**JUAN A. FRAIRE** (Senior Member, IEEE) is currently a Researcher at the National Institute for Research in Digital Science and Technology (INRIA), France, and an Associate Professor at the Universidad Nacional de Córdoba (UNC) and Saarland University, Germany. His research interests include spaceborne networking and distributed applications enabled by state-of-the-art informatics techniques. He is the Founder of the annual Space-Terrestrial Internetworking Workshop (STINT) and has been its Chair, since 2014. He has coauthored more than 55 papers published in international journals and conference proceedings, and the *Delay-Tolerant Satellite Networks* book. He participates in joint projects with NASA's Jet Propulsion Laboratory (JPL) and the Argentinian Space Agency (CONAE) and collaborates with world-renowned space companies and start-ups.



**SANDRA CÉSPEDES** (Senior Member, IEEE) received the B.Eng. and Specialization degrees in telematics engineering and management of information systems from Universidad Icesi, Cali, Colombia, in 2003 and 2007, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Canada, in 2012. She is currently an Associate Professor with the Department of Electrical Engineering and the Head of the Wireless Networking Research Group (WiNet), Universidad de Chile, Santiago, Chile. She holds an honorary adjunct professorship at Universidad Icesi. She is the Head of research at NIC Chile Research Labs and an Associate Researcher with the Advanced Center of Electrical and Electronic Engineering (AC3E). Her research interests include routing and protocol design for restricted devices and the Internet of Things, vehicular communication systems and networking, and cyber-physical systems. She serves as an Associate Editor for the IEEE INTERNET OF THINGS JOURNAL and the *IEEE Vehicular Technology Magazine*.