UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

PROPOSAL AND EVALUATION OF NETWORK SIZE ESTIMATION MECHANISMS FOR MEDIA ACCESS PROTOCOLS IN SATELLITE IOT

MEMORIA PARA OPTAR AL TITULO DE INGENIERO CIVIL ELÉCTRICO

PABLO ANDRÉS ILABACA PARRA

PROFESORA GUÍA:
SANDRA LORENA CÉSPEDES UMAÑA

PROFESOR CO-GUÍA
SAMUEL MONTEJO SÁNCHEZ

COMISIÓN:
MARCOS DÍAZ QUEZADA

Este trabajo de memoria ha sido apoyado por los proyectos ANID FONDECYT Regular 1201893
y FONDECYT Iniciación No. 11200659

SANTIAGO DE CHILE
2021

MEMORIA PARA OPTAR
AL TITULO DE INGENIERO CIVIL ELÉCTRICO
POR: **PABLO ANDRÉS ILABACA PARRA**
FECHA: 2021
PROF. GUÍA: SANDRA LORENA CÉSPEDES UMAÑA
PROF. CO-GUÍA: SAMUEL MONTEJO SÁNCHEZ

PROPOSAL AND EVALUATION OF NETWORK SIZE ESTIMATION MECHANISMS FOR MEDIA ACCESS PROTOCOLS IN SATELLITE IOT

This work presents the proposal and evaluation of network size estimation mechanisms for media access protocols in a satellite IoT scenario. The study scenario is a wireless network of sensors deployed on the Earth's surface that communicates directly with a low-cost nanosatellite CubeSat as it moves through its orbit around the Earth.

The CubeSat standard is presented as a low-cost tool to overcome the economic barriers that prevent several countries from accessing satellite technology. The latter is of great importance given emerging technologies such as the Internet of Things (IoT), which is anticipated as a significant impact that will require the operation of a large number of interconnected devices. Most of these devices are sensor nodes deployed on the face of the Earth, often in remote areas devoid of signal for which satellite coverage plays an important role. However, the limitations of a low-cost satellite are considerable, and the communication channel's management becomes difficult as the number of nodes that share the channel increases, mainly because the satellite does not know this number. The ignorance of the size of the sensor network deployed in a given area leads to several problems such as unattended nodes, loss of valuable information, and increased costs in implementing wireless networks. Since there is no specific solution in the literature for this scenario, this work proposes the design of a sensor network size estimator and the adaptation of an existing mechanism for this particular case. Secondly, these estimators are validated by simulations of a medium access control (MAC) protocol implementing them and improving network performance.

The work culminates in the successful design and adaptation of two network size estimation mechanisms that allow feedback to the Frame Slotted Aloha (FSA) communication protocol to maintain approximately constant throughput up to a number of 2000 nodes within the satellite footprint.

Finally, the design of a mechanism that ensures fairness between the nodes is proposed as future work, supplying geographic disadvantages.

MEMORIA PARA OPTAR
AL TITULO DE INGENIERO CIVIL ELÉCTRICO
POR: **PABLO ANDRÉS ILABACA PARRA**
FECHA: 2021
PROF. GUÍA: SANDRA LORENA CÉSPEDES UMAÑA
PROF. CO-GUÍA: SAMUEL MONTEJO SÁNCHEZ

PROPOSAL AND EVALUATION OF NETWORK SIZE ESTIMATION MECHANISMS FOR MEDIA ACCESS PROTOCOLS IN SATELLITE IOT

Este trabajo presenta la propuesta y evaluación de mecanismos de estimación de tamaño de red para protocolos de control de acceso al medio en un escenario de IoT satelital. El escenario de estudio es una red inalámbrica de sensores desplegados en la superficie de la Tierra que se comunica directamente con un nanosatélite CubeSat de bajo costo mientras se mueve a través de su órbita alrededor de la Tierra.

El estándar CubeSat se presenta como una herramienta de bajo costo para superar las barreras económicas que impiden que varios países accedan a la tecnología satelital. Esto último es de gran importancia dadas las tecnologías emergentes como el Internet de las cosas (IoT), que se anticipa como un impacto significativo que requerirá el funcionamiento de una gran cantidad de dispositivos interconectados. La mayoría de estos dispositivos son nodos sensores desplegados en la faz de la Tierra, a menudo en áreas remotas sin señal para las cuales la cobertura satelital juega un papel importante. Sin embargo, las limitaciones de un satélite de bajo costo son considerables y la gestión del canal de comunicación se vuelve difícil a medida que aumenta el número de nodos que comparten el canal, principalmente porque el satélite desconoce este número. El desconocimiento del tamaño de la red de sensores desplegados en un área determinada conduce a varios problemas, como nodos desatendidos, pérdida de información valiosa y mayores costos en la implementación de redes inalámbricas. Dado que no existe una solución específica en la literatura para este escenario, este trabajo propone el diseño de un nuevo estimador de tamaño de red de sensores y la adaptación de un mecanismo existente para este caso particular. En segundo lugar, estos estimadores se validan mediante simulaciones de un protocolo de control de acceso al medio (MAC) que los implementa y mejora el rendimiento de la red.

El trabajo culmina con el diseño y la adaptación exitosos de dos mecanismos de estimación del tamaño de la red que permiten la retroalimentación al protocolo de comunicación Frame Slotted Aloha (FSA) para mantener un rendimiento aproximadamente constante hasta una cantidad de 2000 nodos dentro de la huella del satélite.

Finalmente, se propone como trabajo futuro el diseño de un mecanismo que asegure la equidad entre los nodos, supliendo desventajas geográficas.

to my parents, whose effort and dedication have allowed me every step, especially this one,
to my friends and brother, whose unconditional support encourages me to continue,
to all those of whom I have had the good fortune to learn,
Weñilén rayü, mile peuma penihuen.

Acknowledgements

I thank all those who put up with me in these challenging times of the pandemic.

I am very grateful to my two teacher guides, whose support made this process a happy experience that culminated in this dissertation, to which I am very fond.

For all the problems that they taught me to overcome and the opportunities they allowed me to have, I thank Mauricio and María.

Table of Contents

Table index

Illustration index

Chapter 1

Introduction

1.1. Background

Since the first satellite launch in 1957, more than 8,300 satellites have been launched into space to date [1]. The applications for which these have been developed are diverse, such as scientific, military, and telecommunications, the latter concentrating the most significant number of satellites. [2].

As for telecommunications, satellite technology is desirable in scenarios where you want to communicate with devices located in areas devoid of a terrestrial antenna's coverage, either because they are remote places, adverse to human presence, or the construction of antennas. However, several million dollars are required to build a satellite and launch it into space. Just considering the launch, the cost of this is between 10,000 and 20,000 dollars per kilogram [3]. This high cost presents a considerable barrier to entry, which explains why most satellites have been launched by world powers, with satellite technology being inaccessible to other countries.

The accessibility to satellite technology is a factor that gains importance as communication paradigms such as the *Internet of Things* (IoT) are developed. The *Internet of Things* is presented as a technology that proposes to revolutionize the world by interconnecting devices that can exchange information and take actions without a human intermediary. It is estimated that by 2025, the market for IoT technology will have an impact of 4 to 11 trillion dollars per year [4] and that for the same year, 150 billion devices would be connected under this paradigm [5] .

Among these connected devices, it is worth highlighting wireless sensor networks as a key element for collecting the enormous amount of data that the IoT needs for its development [6]. The applications of these sensor networks are practically unlimited. They range from monitoring the environment, meteorological or environmental phenomena, species of animals and plants, tracking, positioning, location, and logistics applications [7]. The sensors of these networks are usually deployed in remote areas or adverse to the human presence [8] where it isn't easy to provide connectivity with terrestrial techniques, which is why satellite technology is an essential factor.

How satellites can enable constant global connectivity to terrestrial devices is by placing constellations of satellites into orbit. An example of this application is the constellation *Starlink*, a US aerospace transport company *SpaceX* project to provide low-cost internet services to remote areas worldwide starting in late 2021 [9]. This would be consolidated by launching more than 30,000 satellites, weighing approximately 227 kilograms and costing the first of them around 250,000 dollars. [10]. An implementation like this is still costly and achievable only by few entities in the world.

In the search for lower-cost access to space, the CubeSat nanosatellite standard is presented as a solution capable of solving satellite technology's significant entry barriers. A Cubesat satellite is characterized by a weight close to 1 kilogram and a cubic shape of 10 centimeters on edge. It shows greater simplicity in relation to other types of satellites and implies limiting conditions lower processing capacity, energy, and memory.

1.2. Motivation

As mentioned in the previous section, the implementation of wireless sensor networks is key to collecting large amounts of data that will serve as resources for the operations of IoT networks. For this reason, thousands of sensors will be deployed on the face of the earth, forming networks of varying topologies and sizes in remote areas without the coverage of terrestrial antennas. As a low-cost solution to these sensors' connectivity, there are the CubeSats, with which constellations of satellites that orbit around the earth can be formed, offering coverage in precise areas as they move, communicating directly with each node under a *master/slave* schema.

However, ensuring good performance in sensor network communication is tricky as it scales the number of nodes within a CubeSat coverage. It is essential to provide equitable and regular service to each of the nodes of the network, since losing the information of a portion of these may imply not detecting an event of interest or not having enough confirmations of an event to validate it, for a minimum number of nodes reporting an event may be necessary for it to be considered in certain applications. However, in a communication scenario between terrestrial nodes and satellites, the wireless communication channel is shared, and the increase in its congestion due to the increase in the number of nodes has as a consequence a greater probability that their messages will collide and, therefore, its information is lost, and these nodes are left unattended.

In a scenario like this, the *Media Access Control* protocols (MAC protocols) play a fundamental role in coordinating each transmitter's transmission to avoid collisions in the shared communication channel or to retransmit and retrieve data if they do occur. Consequently, it is necessary to ensure a good performance of these protocols to have a good performance of the network, which cannot be considered independent of its size [7].

On the one hand, the size of the sensor network that a CubeSat is serving is unknown and may constantly change. On the other hand, several studies show how important it is to estimate the network's size to improve the performance of communication protocols, allowing better coordination and synchronization. Furthermore, estimating the size of the network is essential to provide good network performance [11].

In the literature, there are several size estimation mechanisms proposed for wireless sensor network scenarios, but they do not apply to this case because it involves communication with stationary terrestrial antennas or more expensive satellites and without the limitations that a CubeSat has. Another consideration of these mechanisms is, for example, that the nodes of the network communicate with each other under an *ad hoc* scheme and not a *master slave* scheme. This above suggests the need to delve into the CubeSats and wireless networks scenario to create documentation to exploit their potential.

All of this above motivates the design of a network size estimation mechanism that allows a CubeSat nanosatellite to establish proper communication with all the nodes it has in its coverage area, subject to the conditions that this implementation entails.

1.3. Definition of the studied scenario

The study scenario in this memory work corresponds to a constellation of CubeSat satellites that orbit around the Earth to receive the information generated by sensors deployed on the Earth's surface. These sensors form a wireless network of nodes, the topology of which may constantly be changing, with new sensors appearing and old sensors disappearing. Additionally, the sensors do not have interconnectivity, communicating only with the nanosatellite whose coverage area includes their location when they are active (that is, they have information to transmit). In this way, the study scenario is based on a *master/slave* scheme, where the satellite adopts the role of *master* and the sensors that of *slaves*.

The sensors are randomly distributed along the earth's surface and do not know the network's topology. Each of these is independent of the rest in terms of the amount of information they generate and the moment they transmit. There is no prior temporal synchronization between sensors and the satellite or between the sensors themselves. Given the different geographic locations of the nodes, delays are highly variable.

As for the satellites, they orbit around the planet without any information, a priori, about the number of nodes present or their geographical location. Furthermore, these two aspects can vary continuously. Also, it is assumed that the coverage area of different satellites does not overlap under any circumstances.

The orbit planes in which the nanosatellites are deployed correspond to the *Low Earth Orbit*, with heights that vary between 500 and 600 kilometers in altitude. Communication is carried out using the 400 MHz band, is *half-duplex*, and has a maximum speed of 100 kbps. The rate with which the satellite orbits is approximately 7.5 kilometers per second.

1.4. Definition of the problem

The ignorance of the size of the sensor network deployed in a given area leads to several problems from the moment they want to communicate with a common node.

From a general point of view, nodes can be left unattended, implying that the satellite does not have equitable communication with all the sensors and therefore they are not receiving a fair share of the system's resources (this concept will be used from now on as *fairness*). Likewise, unequal distribution of the communication channel resources can cause the system's effective transfer rate to be high even if there are nodes where it remains low.

From a more particular point of view, some media access control protocols are based on dividing the shared channel according to the number of nodes that want to access it. These by themselves are unusable without prior knowledge of the number of nodes. Other protocols depend on a *buffer* that keeps in a queue the requests from sensors that reserve the channel to transmit on it later. The size of this *buffer* is important not to miss requests, but it must be set according to the potential number of requests that there may be and, therefore, according to the size of the network.

The above restricts the possible protocols to use to those that can dispense with a knowledge of the network's size. However, even these protocols may show a lower performance than desired for this scenario [12], reducing the network's efficiency by scaling the number of sensors to attend.

Not being able to maintain good network performance, regardless of the number of nodes present, leads to several problems such as the loss of valuable information due to having unattended nodes and the increase in costs in the implementation of wireless networks due to installing sensors whose data is not being used.

There is no specific solution for this nanosatellite scenario in the literature since there is no proposed size estimation mechanism for this particular case, although estimation mechanisms could be adaptable to this scenario.

1.5. Objectives and scope

1.5.1. General objective

The general objective is twofold and consists, on the one hand, to design a mechanism that allows estimating the size of the sensor network that is under the coverage of a CubeSat-type satellite and, on the other hand, to adapt an existing mechanism to this particular scenario, in such a way that the satellite can use the estimates by feeding back a media access control protocol to improve network performance.

1.5.2. Specific objectives

- To design a mechanism to estimate the sensor network's size and to adapt an existing one to this scenario to provide the nanosatellite with information on the approximate number of nodes that will share the communication channel at any given time.
- To evaluate the performance of the proposed mechanisms using performance metrics obtained through simulations. These metrics are the convergence time (measured as the number of passes required to get the estimate), the root mean square error and the size estimation mechanism's scalability.
- To evaluate the network's performance with the applicability of the proposed mechanisms, in terms of access to the medium, through throughput analysis of a media access protocol, and thus compare both mechanisms in terms of performance.

1.6. Methodology and tools

In this section, the phases on the basis of which the objectives set for this memory work are developed are presented.

1.6.1. Bibliographic review

A document review is carried out, preparing the state of art related to this report's work. Through this research, media access control protocols and network size estimation algorithms are reviewed, which have been proposed for scenarios similar to the case study to rescue characteristics that can be adapted for the estimation mechanism of size to be designed. Since few documents deal with nanosatellites' specific case with sensor networks, different scenarios are considered for the state of art, including terrestrial sensor networks. Thus, the documents reviewed are works focused on a wireless connectivity scenario characterized by a network of sensors or devices that seek to communicate with a common node. These works may be about the proposal of a media access control protocol, comparing it with other existing ones, or the adaptation of old network size estimation algorithms, looking for better performance.

The approach of the reviewed works responds to the limiting conditions implied by the implementation of CubeSats. For this reason, not only those whose focus is estimating the size of the network are considered, but also those that focus on access to the shared medium and energy efficiency.

1.6.2. Estimation mechanism design and adaptation

Once the previous research has been carried out, an estimation mechanism is adapted and a new one is designed.

1.6.3. Determination of the time required for the estimation

The minimum number of satellite passes through the same area is determined to conceive an estimate of the number of nodes present in it.

1.6.4. Scenario simulation

Using MATLAB software, the described scenario is simulated by deploying nodes in space and emulating their communication with the nanosatellite. During this communication, the size estimation is carried out through a precise MAC protocol described later. Three different experiments are considered:

1. The nanosatellite is static, and the wireless network has sensors located under his coverage area. Different scenarios are considered, increasing the total number of nodes at each iteration.

2. The nanosatellite moves and the wireless network has sensors distributed throughout the path of the satellite coverage area. Different scenarios are considered, increasing the total number of nodes at each iteration.

First, the operation of the size estimation mechanisms is simulated. Secondly, a validation is performed by simulations of a MAC protocol implementing these estimations.

1.6.5. Performance evaluation metrics extraction

Thanks to the simulation of the scenario, metrics are extracted to evaluate the performance of the size estimation mechanisms, considering the following metrics:

- **Convergence time**: the time it takes for the mechanisms to converge to an estimated number of the network size, measured in the number of passes of the satellite required.

- **root-mean-square error** (RMSE): this error compares the estimated value and the actual value of the number of nodes, measuring the amount of error there is using the following expression:

  $$RMSE = \sqrt{E((\hat{\theta} - \theta)^2)} \tag{1.1}$$

  Where $\hat{\theta}$ represents the estimate and θ the value observed.

- **Scalability**: this metric measures qualitatively the maximum size of the network for which the size estimation mechanism is efficient.

- **Memory cost**: an analysis is made of the amount of memory needed to run the estimations.

For the validation step, the following metrics are calculated to analyze the impact of the estimation mechanisms on a MAC protocol:

- **Throughput**: measured as the quotient between the number of packets successfully transmitted and the time elapsed for it.

- **Normalized energy efficiency**: expressed as the quotient between the energy used in successful transmissions and the total energy expended during the transmission period.

- **Collision probability**: the probability that a packet will collide with that of another sensor.

1.6.6. Comparative evaluation

The protocol's performance using the feedback from the size estimation mechanisms is compared to the protocol itself without that feedback to perceive if it offers a performance improvement.

1.6.7. Software tools

MATLAB [13]: is a numerical computing software that has an integrated development environment and a programming language. In this, calculations with highly complex mathematical expressions can be developed with ease.

The rest of the document is structured as follows: Chapter 2 sets out the theoretical framework and the bibliographic review that makes up the state of the art, Chapter 3 details the development to design a network size estimation mechanism and adapt one already existing to the study setting. Then chapter 4 details the simulations carried out in order to meet the objectives set out above, chapter 5 shows the result and discussion of the first iteration of the experiments carried out, and then in chapter 6 are shown the results of the second iteration, where corrections detailed in the previous chapter are applied. Finally, chapter 7 presents the conclusions regarding the results obtained, the objectives set, and future work.

Chapter 2

Theoretical framework and state of the art

2.1. Technical concepts

In this section, concepts necessary to fully understand the work carried out and its objectives are described. The choice of these is closely related to the development of the state of the art review, which can be found in the next section.

In this way, the notions described here correspond, first of all, to the definition of the *CubeSat standard*, the type of satellite mentioned during the definition of the problem, describing the most important characteristics to take into account for solving it. Then, the function of the *Media Access Control protocols* is explained, as it is the tool to be used to develop the solution to the problem, and the three types of protocols addressed in this work are described: *fixed assignment protocols*, *random access protocols* and *reservation protocols*. Finally, the concept of *wireless sensor network* is defined as it is the main way to characterize the structure of the network addressed both in the different documents reviewed in the next section, and in the development of the solution for this work.

2.1.1. CubeSat standard

CubeSat is a nanosatellite characterized by its low cost. The CubeSat standard is based on a cube-shaped satellite of 10 cm per side, weighing less than 1.33 kg [14]. Given its small size and focus on low cost, it has quite limiting restrictions compared to other larger satellites. Among the main limitations considered in this document are those related to energy storage capacity, processing capacity and memory capacity. [12], considering the following specifications for these capacities: 2 GB of permanent memory, 256 kb flash memory, 16 Kb RAM memory, 16 MHz processing speed and accumulated energy in battery less than 100 Watt / hour, characteristics of the Suchai CubeSat [14]. It is worth mentioning the Low Earth Orbit (hereinafter, LEO) where these nanosatellites are deployed in the scenario proposed for the work of this document. This orbit covers heights between 300 and 2000 kilometers and is characterized by low communication delays compared to other more distant orbits. The speed in this orbit is approximately 7.5 km per second, for a satellite located 500 km high.

2.1.2. Media Access Control Protocols

The Media Access Control protocols (hereinafter "MAC protocols") are those that govern the way in which different nodes communicate by transmitting information through a shared medium. There are several types of MAC protocols, depending on the way in which they manage this shared medium to avoid collisions and then achieve successful information transmissions.

The types of MAC protocols discussed in this document are:

- *Fixed assignment protocols*: They are protocols that divide the channel evenly (and fixedly) between the nodes that want to communicate. This division can be done in different dimensions depending on the protocol, such as frequency (Frequency-division multiple access, hereinafter FDMA), time (Time-division multiple access, hereinafter TDMA), code (Code-division multiple access, hereinafter CDMA). Given the particularity of the channel being divided, collisions cannot occur, sacrificing channel capacity (since it is divided among users).

- *Random Access Protocols*: They are protocols that govern communication when the number of users is not fixed and the channel is occupied by them without prior coordination. In this case there may be collisions, in the event that one or more users want to transmit and their signals overlap. In this family of protocols are contention-based (or contention-based) protocols. The Aloha protocols and their variants described in [2], Reservation Aloha, Enhanced Aloha, Enhanced Spread Spectrum Aloha, Multi-Slots Coded Aloha (hereinafter R-Aloha, E-ALoha, E-SSA and MuSC respectively) can be mentioned as well as CSMA protocols.

- *Reservation protocols*: protocols by which users reserve the channel prior to the transmission of their data. One way to make the reservation is through a request message that is sent to a central node, which processes the requests of the different nodes that seek to transmit in order to assign each one, one or more *slots* during which the node that requested it has the channel reserved for itself. Another way to reserve the channel is through the use of a token message, which rotates between the users who want to transmit, being able to do so only while in possession of the token. As reservation protocols we can mention Packet Reservation Multiple Access (PRMA) [15], which proposes the use of *control time slots* for reservation and allocation management, and data time slots for data to transmit.

2.1.3. Wireless sensor network

Hereinafter “WSN”, it denominates networks whose nodes are made up of sensors and may have other elements depending on the type of network. There are *ad hoc* networks, decentralized networks where the presence of routers or gateways is not necessary, but rather sensors are responsible for the routing process, forwarding data from node to node. Another way to design a network is through the *master/slave* scheme, where the sensors (*slaves*) depend centrally on a gateway (*master*) to which they send their information so that it is processed and/or transmitted outside the network. An existing phenomenon in WSN is that of *Hidden nodes*, by which collisions can occur in a shared medium without the transmitting nodes detecting the superposition of their signals, since for physical reasons (distance or other phenomena such as *shadowing*) the nodes are not in each other's coverage.

The way in which sensor nodes communicate with the satellite in this particular case of study is called *direct to satellite* (DtS) and thus there is no intermediate ground gateway nor interconnection in between nodes. As the satellite moves in his orbit, nodes will enter and leave the coverage area as it is shown in Fig. 2.1, where red nodes are outside and green nodes are inside of it and can communicate with the satellite.

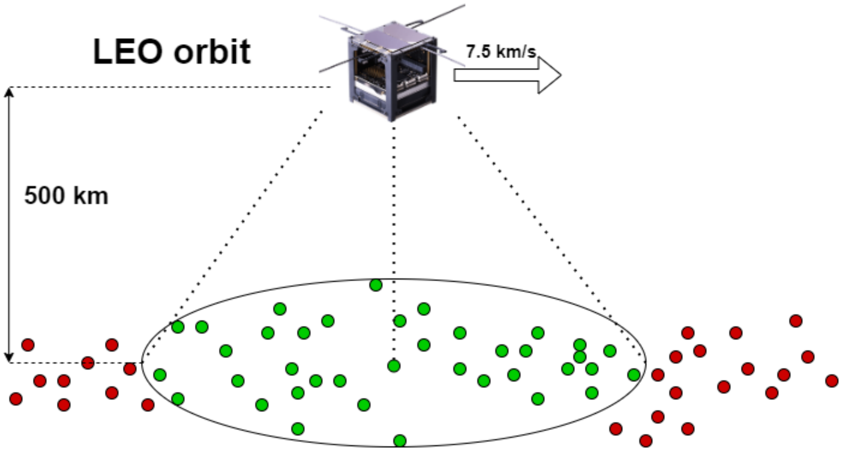


Figure 2.1: Communication scheme between satellite and sensor network

Altitude and speed parameters shown in the figure are those mentioned before in the definition of the studied scenario.

2.1.4. Performance metrics

In this subsection four metrics are described:

- **RMSE**: the root mean-square error is calculated as shown in equation (1.1), by taking the square root of the average squared errors. Since these errors are squared before the average is calculated, there's a relatively high weight to large errors. This is the reason to choose RMSE as a metric to measure estimation performance, as large errors are undesirable in a scenario where the number of nodes that are perceived has a direct implication in the management of the channel resources and therefore the performance of the network.

- **Throughput**: it is a measure of the total amount of data that flows through the channel and differs from *goodput* because it is restricted to only useful data. Throughput is chosen since the work in this document focuses on the communication of the nodes regardless of the nature of their packets (retransmissions, redundant messages, etc.).

- **Fairness**: metric used to determine whether nodes are receiving a fair amount of the channel's resources and it will be measured according to the possibilities of the nodes to transmit within the satellite footprint according to the throughput.

- **Normalized Energy Efficiency**: under the consideration that when a sensor node transmits it consumes 1.0 unit of energy as that in [16], the energy of successful transmissions is divided by the total energy consumed during the process.

2.2. Review and evaluation of the state of the art

This section presents the review carried out on the state of the art related to the report's work. For this research, emphasis was placed on documents that detailed MAC protocols used in wireless sensor networks. In particular, the study focused on those who developed the most important problems to solve according to the scenario outlined above, such as access to the medium, the estimation of the size of the sensor network and energy efficiency.

In the first place, among the documents reviewed with access to the medium as the main thread, the work carried out by T. Ferrer et al. in [12], in which different MAC protocols used for satellite systems are reviewed, with the limitations of the characteristics of CubeSat technology. After these protocols are described, quantitative and qualitative comparisons are provided resulting from the evaluation using different criteria. These criteria include three communication performance metrics, *throughput*, *channel load* and *packet loss*, in addition to other measures such as dependency on network topology, complexity of implementation and power consumption. After this comparison, the results are discussed and the challenges for this wireless network paradigm are raised.

Several conclusions emerge from this research. Although several options offer high performance in terms of communication (R-Aloha, FC-TDMA, E-SSA, and MuSCA), these are accompanied by high complexity and, therefore, high cost of implementation and also energy consumption. On the other hand, protocols based on Aloha show, in general, low complexity and are suggested as promising candidates for the MAC layer of the case studied. However, the latter have important deficiencies in scalability, showing poor performance when increasing the number of nodes and traffic. Then, examining protocols based on interference cancellation (E-SSA, CRDSA, MuSCA, CSA), it is estimated that its application for the case study is unfeasible given the adverse conditions that prevent correct estimation of the channel in LEO orbit, together with the processing capacity limitations of the CubeSats. On the other hand, *carrier sensing* protocols prove to be relatively inefficient given topology conditions such as *hidden nodes* and highly variable delays between nodes. This last factor is also critical for TDMA protocols since, to synchronize the channel, a security time of the order of the variability of the delay between nodes must be incorporated, translating into considerable losses of channel resources.

For their part, K. Jamieson et al. present in [17] the Sift medium access protocol for wireless sensor networks, specially designed for cases of spatially correlated contention. Sift is a CSMA protocol with a fixed-size contention window and a non-uniform probability distribution to transmit in each window slot. This probability distribution corresponds to an increasing geometric distribution, which is an increasing probability of choosing slots as the channel remains silent in the first slots because no node transmits. This design results in a quick way to pick a "winning" node (that is, one that manages to transmit) within a wide range of population sizes (sensor array size), reducing the chance of collision. The authors' results show a latency improvement of seven times compared to 802.11, with a population that scales up to 500 nodes. Given its independence from the topology, Sift is an attractive protocol for the case study. However, [12] on CSMA-type protocols and synchronization impose specific barriers so that Sift could perform well in a medium as described above.

A study of the performance of PRMA protocols for WSN is presented in [15]. This document describes the main characteristics of the PRMA protocols: the queuing process for *requests* and the parameters that influence the performance of the protocol (described in the previous section). The experiments carried out are based on varying the latter, obtaining *Throughput* and *Message delay* measurements for each case. The main conclusions of this study show that the *master* must have a *buffer* for reservations, with a threshold to reject the excess in the queue, as well as, the longer the data time slot length, the greater it has to be the threshold, but choosing a higher threshold does not significantly improve performance. Lastly, the threshold should be determined according to the length of the data time slot and the average number of packets sent per *message request.* Although no explicit value ranges are offered for an optimal behavior of PRMA protocols, it follows that by simulation, a combination of parameters that exhibit good behavior can be found, adapting these to the characteristics of the WSN where the protocol is implemented. Unfortunately, synchronization complications are also present if this protocol is to be adapted to the CubeSat scenario.

Approaching the paradigm of an IoT system, G. Tsoumanis et al. in [18], they propose the implementation of a MAC protocol topology independent, TiMAC, which is also based on TDMA. This protocol is designed to operate within a low-cost IoT system, where the devices must consume the least amount of energy possible. Its main objective is to allow at least one successful transmission per node, per frame, regardless of the network topology. The authors propose solving two challenges: implementing TiMAC (achieving the objective described above) and synchronizing the devices using a decentralized *multi-hop* technique. The assignment of *time-slots* is done through the arbitrary assignment of Galois polynomials, described in [18], and it is possible to grant each node a transmission without collision, even in cases where changes occur in the topology. Then decentralized synchronization *multi-hop* is also a success, although it is a feature that does not adjust to the CubeSat scenario's characteristics.

In [19] and [20], C. Wang et al. propose LST-MAC, a low latency hybrid MAC protocol designed for IoT systems with LEO satellites. The topology proposed in these documents comprises a satellite whose coverage area contains several IoT Gateways, each of these connected to sensors. These satellites have an antenna of the *satellite phased array antenna* type (described in [20]), which can make their rays converge, increasing the power of their signal but sacrificing coverage area. LST-MAC is a TDMA-based protocol, which uses a strategy to allocate *time-slots* based on the geographic location of the *gateways*, to be able to manage the slots at the moment the satellite beam changes position and contact other *gateways* within coverage. In a simple way, the satellite allocates *slots* of time to the *gateways* which in turn allocate *slots* of time to their sensor nodes. The experimentation results are favorable, obtaining improvements in performance comparing the protocol to other existing ones [20].

N. Shanin et al. in [21] propose the protocol *hybrid slotted-CSMA/CA – time-division multiple access* (hereinafter, HSCT), designed to fill the flaws of the IEEE 802.11ah standard at the time of the registration process where a massive amount of devices simultaneously try to access a single and centralized *access point* (hereinafter, AP), in an IoT scenario. The topology of the case study is a star. The protocol is based on the use of three different periods, first a *Beacon period* (hereinafter, BP), then a *Slotted CSMA/CA period* (hereinafter, SCP) divided into multiple *C-slots* (which are multiple *CSMA/CA access windows*) and finally a *Slotted TDMA period* (hereinafter, STP) similarly divided into multiple *T-slots*. During BP, the AP announces with a *broadcast* the start of the next SCP period, as well as the number of *C-slots* and their duration (analogously with the number and duration of *T -slots*). During SCP, the authentication processes occur, and during STP, the association processes. The simulation and evaluation results of the protocol are such that a speed improvement of between 64 % and 87 % is achieved in the registration process, compared to existing schemes, allowing the registration of up to 8000 devices. The periods mentioned above can be visualized in the diagram of Fig. 3.1.

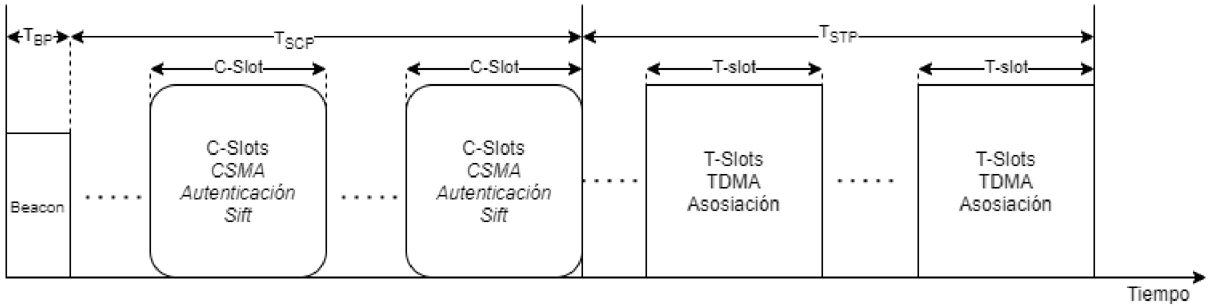


Figure 2.2: Time diagram of HSCT periods, adapted from [21]

Several mechanisms are those that allow HSCT to control the massive amount of contention: the division of *Slotted CSMA/CA period* into multiple slots, an adjustment of the SCP and STP periods to minimize losses of channel capacity, a period without TDMA contention to process the registration of those devices that managed to request authentication and the use of the geometric probability distribution of Sift during SCP, which reduces the probability of collisions.

On the other hand, R. Ali et al. present in [22] two algorithms for estimating the size of *Master/Slave* and Ad hoc networks. These algorithms are adaptations of the already existing *Random Tour* and *Gossip-based Aggregation* methods to fit *Master/Slave* architectures and to be compatible with random changes in the topology. First, the *Random Tour* method is based on the transfer of a *token*, registering each new node visited. The adaptation of this work consists of adding steps to follow in case the transfer of the *token* is not successful (for example, if a node crashes before being able to transmit the *token*) and in a protocol of routing so that the *token* returns to the beginning in case a node does not find neighboring nodes, not visited by the *token*, to which to deliver it. Second, the *Gossip-based Aggregation* method is based on communication between adjacent nodes. The initiating node starts with a value called *avg*, equal to 1 while the other nodes start with a value of 0, then the nodes are successively averaging the value *avg* with their neighbors until, in a network of N nodes, each node will have a *avg* value equal to $\frac{1}{N}$. The adaptation carried out consists of dividing the network into *clusters* with a *master* for each one, these being responsible for averaging the value *avg* with their *slaves*, and then sending this updated value to all its *slaves*. By repeating this process a certain number of times, the *avg* value converges to the value mentioned above, estimating the size of the network. However, both methods require communication between nodes.

In [23], Zanella proposes a collision set size estimator for Framed Slotted Aloha (FSA) wireless networks and Radio Frequency Identification (RFID) systems. FSA is an Aloha-based contention MAC protocol that divides time into slots. These slots are grouped in frames so that active nodes transmit their packet in each frame choosing a random slot inside with uniform probability. An RFID system is generally composed of a reader and several tags, each having a unique ID. Collisions due to simultaneous tag responses can occur in the same way as in wireless sensor networks. A collision set is the group of nodes contending for a time slot in a given contention window of FSA. If two or more nodes in the set transmit concurrently, a packet collision will occur, that is to say, the loss of the transmitted data due to the receiver's interference. The receiver can perceive these collisions and therefore can keep track of the number c of collisions that have occurred, as well as the number of successful transmissions s and idle slots i. These three numbers together compose an observation of the result at the end of a frame. Then, having these values, Zanella applies a maximum likelihood estimator, which returns the value of n that maximizes the conditional probability of having that observation, given the number n of transmitting nodes in that frame. It is essential to mention that the number of transmissions in each slot is considered independent Poisson random variables, a necessary approximation for the calculation to be more straightforward and possible to be handled by common devices. The results obtained show that the method has the lowest estimation error among the methods reviewed in the document.

Next, documents that focus on energy efficiency are presented. A problem presented by various sources regarding the scheme of a WSN ad hoc network is that in the vicinity of the *sink* node (the node to which all the information from the sensor network is sent), the adjacent nodes They must be active for a more extended period than the rest, as they serve as a bridge for communication between distant nodes and the *sink*. Being this *sink* static, there is an imbalance in energy consumption that can eventually deplete the neighboring nodes' energy to the *sink* and therefore cut communication with it. Furthermore, the *sink* itself is constantly faced with the energy load of receiving the WSN nodes' data. Faced with this problem, the two works [24], by W. Heinzelman et al. and [25], by M. Abo-Zahhad et al., present protocols for the role of *sink* to be mobile, seeking to distribute the energy load of the wireless sensor network equitably.

Thus, in [24] LEACH (*Low-Energy Adaptive Clustering Hierarchy*) is proposed, a distributed protocol that dispenses with a global knowledge of the network in order to operate. The idea of LEACH is to rotate the role of *sink* between the nodes. For its part, [25] proposes *Mobile Sink based adaptive Immune Energy-Efficient clustering Protocol* (MSIEEP), which uses an algorithm to guide the mobile *sink* and find the number optimal *cluster heads* (nodes that communicate directly with the *sink*). The result is a more energy-efficient protocol than LEACH.

Looking at similar cases, the authors C. Cheng et al. elaborated in [26] *Sink Location Service Protocol for Mobile Sinks* (from now on SLSPMS), to reduce the energy consumption of the sensor system with a mobile *sink*. Finally, the authors B. Malhotra et al. en [27] propose an energy-efficient way for a system of several mobile sinks that require coordination with each other.

Finally, a proposal is presented focused on the feedback of a MAC protocol that adjusts its parameters according to the number of active devices that want to transmit. In [16] a slotted Aloha Game is proposed to control access to the medium in satellite networks. The proposal focuses on determining the transmission probabilities of a group of independent terminals (nodes) that share the communication channel and transmit via the framed slotted Aloha protocol. In the document, a payoff function based on the probabilities of successful transmission is considered. A Nash equilibrium is sought in which the transmission probability of each node is adjusted to reduce costs (failed transmission attempts) and maximize the probability of successful transmissions. The probability of transmitting of each node depends on the number of active nodes, in such a way that if this amount is less than or equal to the number of slots in the frame, then they are allowed to transmit with a probability equal to 1, however, if the number of active nodes exceeds the number of slots, the probability of transmitting decreases so that on average only a number of nodes equal to the size of the frame transmit. The simulations' results show that the slotted Aloha game proposed allows the communication of large numbers of nodes (up to 500), maintaining an approximately constant throughput and presenting low access delay and low power consumption of the system in general.

Several documents described here solve the problems they pose, but mostly under conditions that would be difficult to replicate in this memory work's study setting. Such conditions are the synchronization of the sensor network for the CubeSat scheme, communication between sensor nodes, and low *delay* in communication. For this reason, it is necessary to choose elements that could be adaptable to the case in the dimensions of access to the medium, estimation of network size, and energy saving.

Next, a comparison between the different previously reviewed works is presented in Table 2.1. In this, each one's positive and negative points are indicated in relation to their adaptability to the case study of this memory work. In this way, interesting proposals can be identified according to their positive aspects, being able to adapt these to the case study or finding a way to deal with the negative aspects.

Table 2.1: Comparative and evaluation table in the study setting

Works	focus	Positive points	Negative points
Sift [17]	Media Access	Scalability with good performance up to 500 nodes	Requires synchronization Susceptible to variant delays and Hidden nodes
PRMA [15]	Media Access	Parameters adaptable to WSN characteristics	Requires synchronization
TiMAC [18]	Media Access	Topology independence Energy efficiency	Requires interconnected nodes
LST-MAC [20]	Media Access	Made for WSN with LEO satellites	Requires geographic location of nodes Requires synchronization
HSCT [21]	Media Access	Scalability up to 8000 devices	Requires synchronization
Random Tour [22]	Size estimation	-	Requires interconnected nodes
Gossip-based Aggregation [22]	Size estimation	The CubeSat can manage the process	Requires interconnected nodes
Zanella [23]	Size estimation	Estimation adaptable to case study	Requires synchronization
LEACH [24]	Energy efficiency	Topology independence	Requires interconnected nodes Rotation of the Master role
MSIEEP [25]	Energy efficiency	-	Requires interconnected nodes Requires geographic location of nodes
SLSPMS [26]	Energy efficiency	-	Requires interconnected nodes Requires geographic location of nodes
B. Malhotra et al. [27]	Energy efficiency	-	Requires interconnected nodes
Slotted Aloha Game [16]	Network size feedback to MAC protocol	Satellite network	Tested only up to 500 nodes Requires synchronization

As can be seen in the table, several proposals require time synchronization of the network as they depend on the use of *time slots*. Another negative aspect found repeatedly is the requirement of interconnectivity between nodes, that is, that the nodes can transmit messages to each other. This characteristic is not found in the case study of this memory work since the nodes communicate directly with the CubeSat and not with each other. Zanella's estimator can be used among these works as long as time synchronization is assured, letting FSA manage the communication channel. Given the latter, Slotted Aloha Game can process the estimator feedback to adjust the transmission probability and improving performance when a number of nodes bigger than the frame's slot number are being attended.

Chapter 3

WSN estimation mechanisms for DtS-IoT

In this chapter, two estimators are explained and proposed for later simulations. The MAC protocol for communication between nanosatellite and nodes during estimation rounds is FSA. Space is divided discretely so that each position corresponds to a single frame. Time synchronization is assumed, and the frame is composed of a downlink beacon announcing the start of the contention window, followed by an uplink period where nodes inside the nanosatellite footprint transmit, choosing a random frame slot with uniform probability.

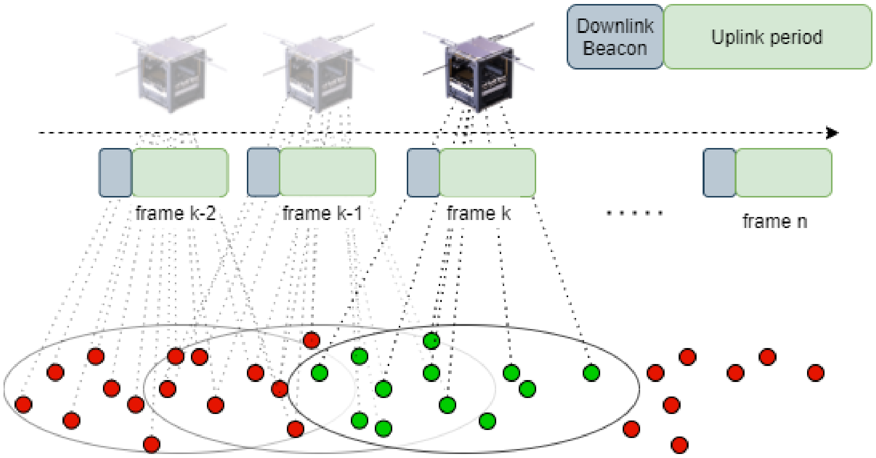


Figure 3.1: Time diagram of HSCT periods, adapted from [21]

Let n be the number of nodes to be estimated in a given area, with the assumption they are distributed in a way that the n can be covered by the satellite coverage area at the same time. Let w be the number of time slots of a FSA frame (also called frame length) and for every contention window let c be the number of collided slots, s the number of successful transmissions and i the number of idle slots. For a transmission to be successful in one slot it is necessary that only one among the n nodes transmits in that slot. This means that there is the following relationship between the variables described: $c + s + i = w$. However, the only one that provides direct information about the number of nodes that are active and trying to transmit in the frame is s, since multiple nodes can be involved in a collision and no nodes are transmitting in an idle slot. It is important to mention that it is assumed that the satellite can perceive collisions, considering cases where a message cannot be decoded because of interference of two or more packets as a collision. Moreover, channel noise is not considered as a potential factor that can lead to problems decoding a message. Because of this assumptions, the nanosatellite is aware and can keep track of c, s and i of every position where FSA protocol took place.

The following list summarizes the assumptions and details considered for the WSN size estimators presented in the sections of this chapter:

- The path of the satellite coverage footprint is divided in a discrete manner, which means that the amount of positions is a natural number.
- For every position there is an unique number n of nodes inside the satellite footprint to be estimated.
- In every position, the n nodes contend for the channel picking a slot randomly within a frame. This means there is a single frame for each position.
- The only way a transmission can fail is a collision.
- The satellite can perceive collisions.
- A node can try to transmit only once per frame.
- The particular position of each node within the footprint does not influence communication. Therefore, there is no difference between the delay of two nodes within the satellite coverage even if they are at different distances from the center of this.
- There's time synchronization between nodes and the satellite, so that they know the beginning and end of frames and slots.

In the following sections, two estimators are proposed.

3.1. Adaptation of Zanella's collision set size estimator

As presented in the state of the art, in [23] Zanella proposes a collision set size estimator for RFID systems using FSA as the medium access control protocol. Although the mechanism is made for an environment composed of RFID tags, it can be adapted to a satellite scenario under the assumptions mentioned in the last section. The reason for this is that at the end of each frame the satellite will still have an observation of c, s and i that serves as input for the collision set size estimator, as explained below.

Following the mathematical development of [23], let $v =< c, s, i >$ be one observation of the vector $V =< C, S, I >$ which stores the number of collided, successful and idle slots in the frame. Then, the conditional probability of observing v given that n nodes transmit is equal to:

$$P_n(v) = P[V = v|n] = \binom{w}{s}\binom{w-s}{c}\sum_{j=0}^{c}\sum_{l=0}^{c-j}\binom{c}{j}\binom{c-j}{l}\frac{(-1)^{c-j}n!j^{n-l-s}}{w^n(n-l-s)!} \tag{3.1}$$

The classical maximum likelihood estimator is the one that returns the value of n that maximizes the probability shown in equation (3.1). However, there are several problems with the calculation of this such as the numerical instability of the computation of binomial terms, along with high computational cost, which make the maximum likelihood estimator inappropriate to implement on low-end devices and much less on a CubeSat. The solution to this problem found by the author of the document is to simplify the conditional probability by considering the number of transmissions in each slot as independent Poisson random variables of mean $\mu = \frac{n}{w}$. The conditional probability of observing v given μ can be expressed as:

$$P_\mu(v) = \mu^s e^{-\mu w}(e^\mu - 1 - \mu)^c \tag{3.2}$$

Setting to zero the derivative of (3.2) in μ to find the value that maximizes that expression the following is obtained:

$$\frac{\mu w - s}{c} = \frac{\mu(e^\mu - 1)}{e^\mu - 1 - \mu} \tag{3.3}$$

Equation (3.3) admits only one non-negative solution μ' and can be determined by bisection search method. Then, the estimate is the product $w\mu'$.

Since the values necessary to produce this estimate come from the result of communication through the FSA protocol between devices and a receiver, it can also be applied to the scenario between a satellite and nodes of a WSN, as long as the communication is effective and the only cause of failed transmissions are collisions perceivable by the satellite. The code for MATLAB implementation of this estimator is provided by the author and can be found in [28]. Some testing shows that the estimation has the appearance of a noisy curve that oscillates around the actual value of nodes. One way to refine the estimate is to smooth the curve using a polynomial fit. The latter is to obtain a smaller error, which is important in locations where a large number of nodes (of the order of a thousand) can be present and then the estimation error can be of the order of hundreds, which would have a strong negative impact on the feedback to the MAC protocol. Hereinafter, the name to refer to this estimation mechanism in the document is Zan. It is important to mention that the code in [28] uses binary search to solve equation (3.3).

3.2. Design of a new estimator

If every single one of the n nodes transmit in the frame choosing any slot with probability $\frac{1}{w}$, and considering the only way a transmission can be a failure is a collision, a simple way of estimating n is calculating $s + 2c = n'$, n' being the estimation of n at a given position and assuming no more than 2 nodes are involved in each collision. This assumption is valid as long as $n \leq w$, otherwise the probability of triple, quadruple and higher order collisions gets bigger when n grows larger than w, and thus n' becomes an underestimation.

Setting the frame length $w = 128$, as done in [23], the probabilities of having at least one double, triple, quadruple or higher order collision can be empirically calculated, as shown in

Fig. 3.2, where each curve is the result of the average of 1000 simulations of the choice of slots within a frame, increasing the value of n.

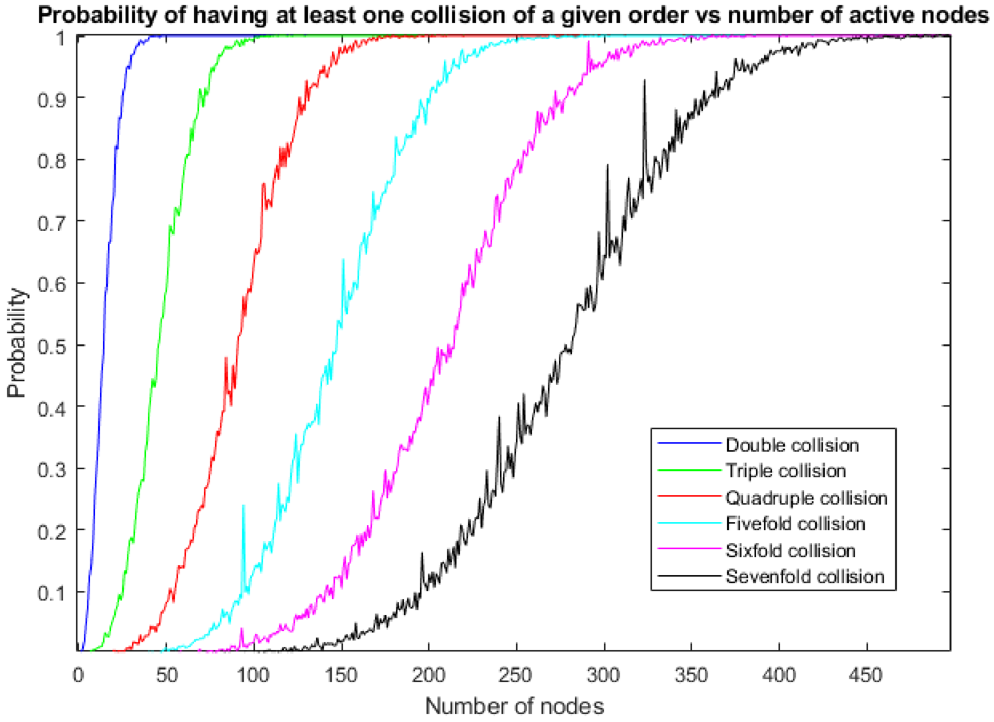


Figure 3.2: Probability of having at least one collision of a given order as a function of number of active nodes transmitting in a w=128 frame.

As can be seen in Fig. 3.2, when the amount of nodes n approaches the length of the frame, the probability of a triple collision to occur tends to 1. This behavior is similar for larger sizes of w. Considering that n will be in most cases bigger than w, triple collisions are to be considered.

Let j denote the number of the turn around the world and k the order of a collision so that c_j represents the total number of collisions at a given position, $c_{\{j,k\}} = c_{\{1,3\}}$ represents the number of triple collisions at that position in the first turn of the satellite around the globe and s_j represents the number of successfully decoded messages in that same position. Since the nanosatellite is deployed in the LEO orbit, with an orbit speed of 7.5 km/s, it takes approximately 90 minutes to make one turn, which means that it can make 14 turns around the earth in a day at minimum. Given the latter, $1 \leq j \leq 14$.

For each position (i.e. frame), the number of collided slots c_j must be equal to the sum of different orders collisions and the number of nodes n is equal to the sum of the number of successful transmissions s_j plus all the nodes involved in collisions. Because of this, the next system of equations must be fulfilled for every position:

$$\begin{aligned} n &= s_j + 2c_{\{j,2\}} + 3c_{\{j,3\}} + 4c_{\{j,4\}} + \ldots \\ c_j &= c_{\{j,2\}} + c_{\{j,3\}} + c_{\{j,4\}} + \ldots \end{aligned} \tag{3.4}$$

Where s_j and c_j are perceived by the satellite and thus are known. If there are no collisions, i.e. $c_j = 0$, it is straightforward that $n = s_j$, but if this isn't the case and $c_j > 0$, then $n > s_j$ and, if higher order collisions than triple ones are considered, then the system of equations has no solution.

Taking into account only up to triple collisions is a relatively simple way to solve this problem. For each position there are two equations and three unknowns, which means that one of this unknowns has to be calculated in order to solve the system. If, for a given position, we start from the pass of the satellite where c_j is the lowest and assume that all the collisions that occurred there were double, such that $n_0 = s_{j'} + 2c_{j'}$, since $c_{\{j',3\}} = 0$ and $c_{j'} = c_{\{j',2\}}$. Then, for another pass of the nanosatellite over the same area $j \neq j'$, if n_0 is effectively the number of nodes the next system of equations has an integer solution:

$$\begin{aligned} n_0 &= s_j + 2c_{\{j,2\}} + 3c_{\{j,3\}} \\ c_j &= c_{\{j,2\}} + c_{\{j,3\}} \end{aligned} \tag{3.5}$$

It is worth remembering that for a given area or nanosatellite position, the number of nodes n has to be the same for any j. If the system has no solution, then n_0 can be increased to $n_1 = n_0 + 1$ and the system can be re-evaluated, increasing the estimation by one until there's a solution for the system of equations.

If m is the number of estimation turns the satellite performs, then these iterations are repeated for *m-1* turns, since one of them is considered as the start for the process. The variable m is important because not only it affects the outcome of the mechanism by refining its estimation the larger it is, but also because it restricts the number of passes devoted to effective communication between the nodes and the satellite. This last topic will be discussed later in the document.

To provide a clear explanation of the process of estimation, the following steps are described:

1. The nanosatellite passes j times over an area where the number of nodes has to be estimated, $1 \leq j \leq m$. Each time, the beginning of a FSA frame of length w is announced to these nodes so they contend picking a frame slot at random with probability $\frac{1}{w}$.
2. For each j, the nanosatellite stores the number of successful transmissions s_j and collided slots c_j.
3. Once the last pass is over, the nanosatellite begins the process starting from j' where the number of collisions is the lowest, i.e. $c_{j'} = min(c_j)$, and calculates $n_0 = s_{j'} + 2c_{j'}$, assuming that all the collisions that occurred on $j = j'$ were double, i.e. $c_{\{j',3\}} = 0$ and $c_{j'} = c_{\{j',2\}}$.
4. Then, the process is continued by taking the lowest j, such as $j \neq j'$ and solving (3.5) with $n = n_0$. If there is no integer solution, then $n = n_1 = n_0 + 1$ is considered and (3.5) is solved again with the new value of n. These iterations are held until there is a $n_x = n_0 + x$ which assures an integer solution of (3.5).

5. This process is repeated for every $j \neq j'$, considering the latest value of n at each repetition, until n increases to a value $\hat{n}$ that solves the system of equations for every j.
6. The nanosatellite stores in its long-term memory $\hat{n}$ as the size estimation of the network at that given area.

These steps are carried out for each discrete position of the satellite along its path around the Earth.

From now on, the name to refer to this estimation mechanism in the document is Frame Slot Collision Based Estimator (FSCBE).

Chapter 4

Software simulations

This chapter presents how the simulations are implemented following the methodology proposed above, under the fulfillment of the objectives of evaluating the estimation mechanisms' performance and of a MAC protocol that uses those estimations. In this way, the development is directed to the fulfillment of the specific objectives and obtaining a simulation package for subsequent analysis.

For every simulation, the two estimation mechanisms considered are the ones proposed in chapter 3. The software used for the simulations is MATLAB [13]. Multiple passes are simulated by repeating the estimation as many times as passes are considered. For FSCBE, each pass is integrated into the iterative process, while for Zan, the results of each pass are averaged with the previous results.

4.1. Medium access control protocol for estimations

As stated previously, the MAC protocol used for communication between satellite and nodes is Framed Slotted Aloha (FSA) [23]. Both mechanisms require that every node inside the area of coverage transmit on a randomly chosen slot from the frame with probability $\frac{1}{w}$. A node can only attempt to transmit once per frame regardless of the result of its transmission.

4.2. Static simulation

In the first experiment, the simulated scenario consists of n nodes distributed within the satellite footprint. This footprint remains static in its position, and the value of n increases, varying from 10 nodes to 2000, with increments of 10.

For each iteration, an FSA communication window is simulated as many times as satellite passes are necessary for the estimation mechanisms. For both estimation mechanisms, up to four passes of the satellite are considered. The estimations are made from the communication simulation results, stored, and compared to the real number of nodes present inside the footprint by calculating the RMSE. These results will determine how many estimation passes will be considered for the rest of the simulations.

The way in which more than one pass are integrated into the Zan estimate is by calculating the average of the estimates from each pass. For this, two ways are proposed, the first consists of calculating the average of all the estimates at each pass, and the second consists of progressively calculating the average between the estimate of the current pass with the previous average. The first way requires memorizing each pass's estimates, while in the second, only the previous average is remembered. In the case of FSCBE, the way to integrate several passes is to include them in the iterative process to calculate the estimate.

4.3. Dynamic simulation

In the second experiment, the simulated scenarios consist of N nodes distributed in a predetermined discretized area. For this section, N represents the total number of nodes deployed and n the specific number of nodes included within the satellite footprint at a precise position, so that $n \leq N$. The footprint of the satellite moves horizontally, containing a different number n of nodes at each position.

For each position, there's a single window w for which the nodes inside the satellite footprint contend. Concerning the simulations, estimations and comparisons, the same previous process is carried out.

Two ways to deploy nodes are contemplated:

- **Nodes deployed randomly**: N nodes are deployed randomly in the simulated area. As in the previous simulation, the value of N increases its value but now varying from 10 nodes to 2010 with increments of 100.

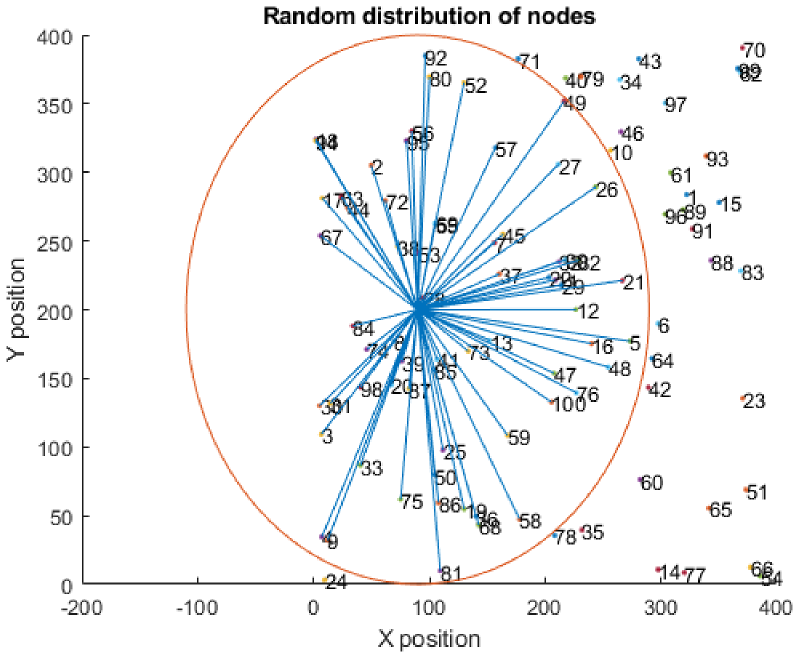


Figure 4.1: Visual example of the dynamic simulation with nodes deployed randomly and movement of the satellite footprint along the X axis.

Fig. 4.1 shows a visual example of the dynamic simulation with 110 nodes deployed randomly throughout the simulated area. As it is shown in the figure, the area where nodes are deployed has dimensions of 400x400 and the satellite has a radius of 200, both measurements are of arbitrary unit. The blue lines that join the nodes with the center of the nanosatellite footprint serve to indicate the nodes that are within the coverage area and therefore can establish communication with the nanosatellite.

- **Nodes deployed in clusters**: there's three clusters containing $\frac{N}{3}$ nodes each. N increases its value varying from 10 nodes to 6010, with increments of 300, so that every cluster ends with around 2000 nodes each in the last simulation.

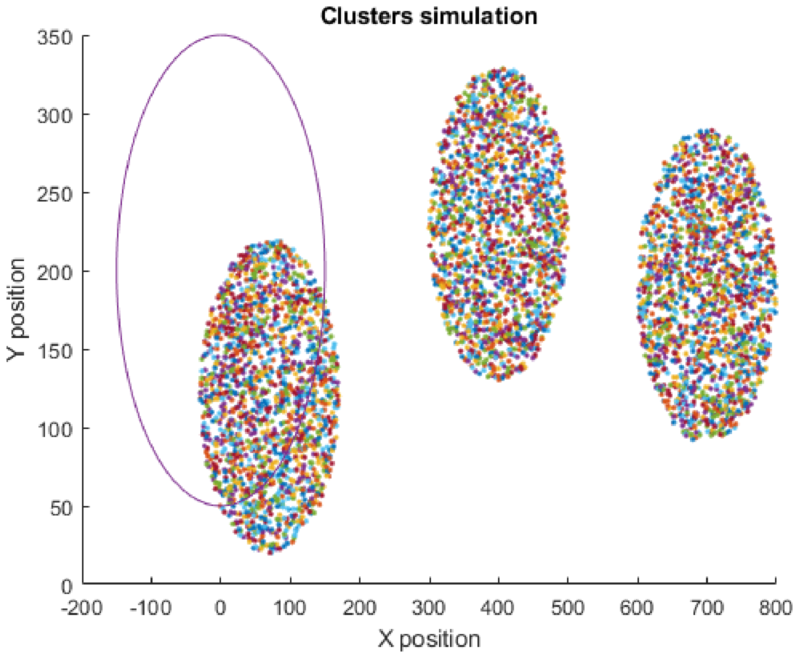


Figure 4.2: Visual example of the dynamic simulation with nodes deployed in clusters and movement of the satellite footprint along the X axis.

Fig. 4.2 shows a visual example of the dynamic simulation of clusters composed of 2000 nodes each deployed randomly throughout the simulated area. The area where nodes are deployed has dimensions of 800x400 and the satellite has a radius of 150, both measurements are of arbitrary unit.

The objective of these two simulations is to subject the estimation mechanism to a changing scenario of nodes in order to analyze its ability to estimate the number of devices being in motion and to face it in a scenario as close to the reality where nodes are agglomerated in high concentration in fixed spaces.

4.4. Feedback simulation

This experiment consists of the feedback of the size estimate for a medium access control protocol. For this, the first scenario described above is simulated to compare the performance of a MAC protocol with and without the feedback.

The protocol to be tested is Framed Slotted Aloha, and the way the feedback is carried out is the one presented in [16] with the Slotted Aloha Game proposal, varying the transmission probabilities of each node according to the amount that wants to access the channel. The metrics used to compare the effect of the mechanisms are the throughput calculated for each frame according to the quotient between the number of packets transmitted successfully and the amount of time the frame lasts in time slots, the normalized energy efficiency calculated as the quotient between the energy consumed for successful transmissions and the total energy consumed in the process, and the slot collision probability.

Chapter 5

First iteration of results and discussion

In this chapter some corrections to the new estimation mechanism are proposed, as well as changes to the window value w for both estimators. These changes and corrections emerge from preliminary results obtained through early simulation of the mechanisms.

5.1. Initial version of the estimation mechanisms

On the one hand, the newly designed estimator tends to a particular estimation value depending on the frame's length. Its process begins by assuming that all collisions are double in the satellite's pass with the least amount of collisions. However, when the amount of nodes present n grows beyond the size of w not only do the errors associated with considering only collisions up to the third order appear, but it also reaches a point where there are collisions in all the slots, and thus the estimation will always start its process from the same number, which explains that the estimated value of nodes tends to a number equal to the length of the window $n' = 2w$, as can be seen in Fig. 5.1

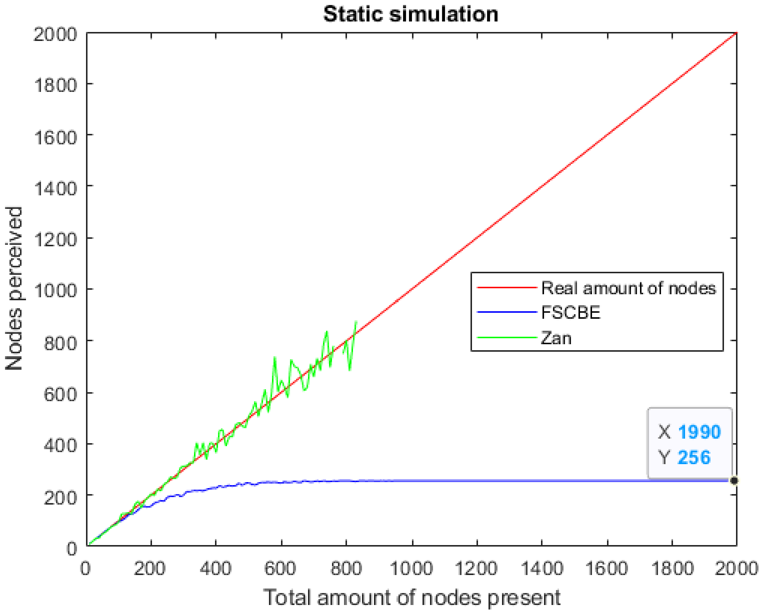


Figure 5.1: Preliminary results of the estimated number of nodes by both mechanisms as a function of the real number of nodes present with a frame of length $w = 128$.

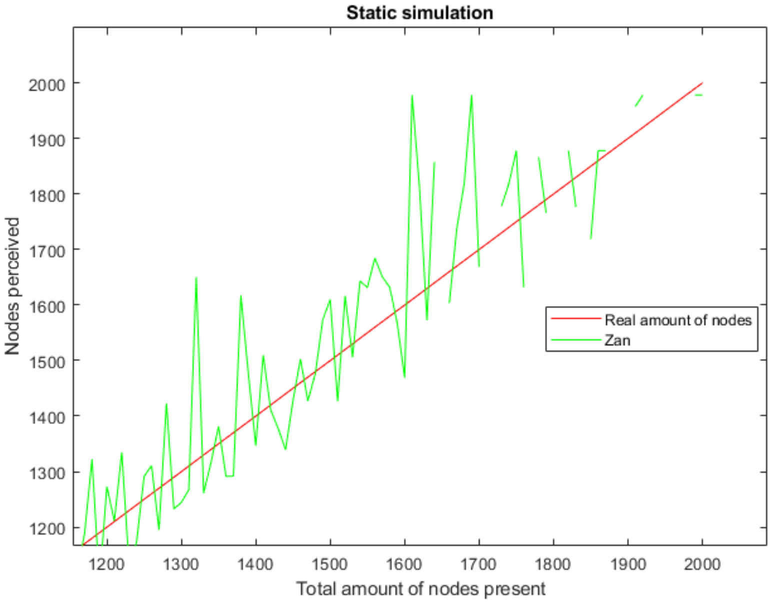
Fig. 5.1 was obtained by simulating twice each estimation mechanism on the static simulation so that two passes are considered, but both estimators' behavior is the same regardless of the number of passes.

A way to solve this issue is to create a function that relates each point of the estimate to the real value of nodes, but for this to work correctly, bijectivity must be ensured between the two data. Using a polynomial fit can solve this matter as long as the window frame is long enough so that no value is repeated, and thus injectivity is assured.

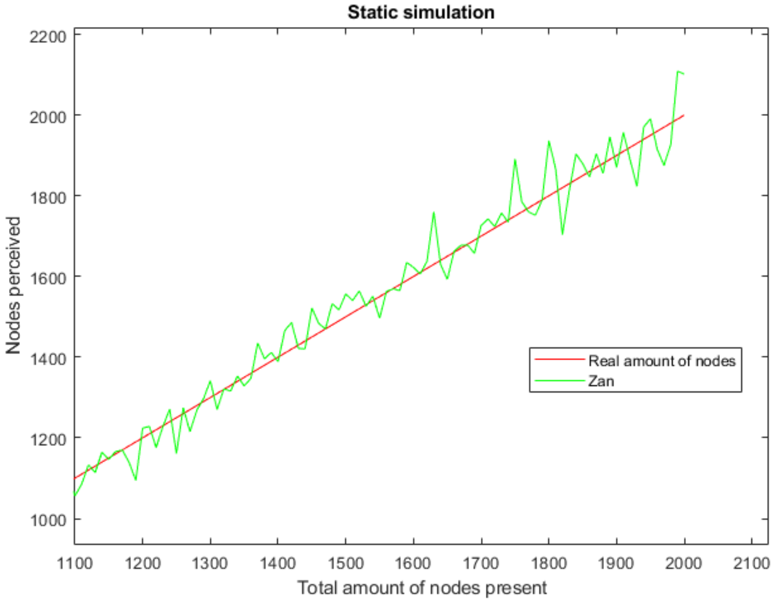
On the other hand, Zan estimator also has problems with the frame length. Once the number of nodes increases past the point where all slots collide, i.e. $c = w$, the estimate is infinite, and no further estimation can be done. Setting a frame $w = 256$ is not enough to handle up to 2000 nodes, as shown in Fig. 5.2a by the discontinuities presented by the curve.

5.2. Modifications required by the estimation mechanisms

Changing the frame length to $w = 512$ slots shows good results, see Fig. 5.2b. If the number of slots increases, the probability that collisions will occur in all of them decreases, and the probability of indeterminacy of the estimation mechanism decreases. The reason for choosing window lengths equal to powers of 2 is to facilitate binary analysis.



(a) $w = 256$.



(b) $w = 512$.

Figure 5.2: Preliminary results of Zan as a function of the real number of nodes present with frames of different sizes.

Having set the window value to $w = 512$ and using a polynomial fit of order 7, the FSCBE estimation can be corrected by first calculating the polynomial fit of the FSCBE estimate, then calculating the coefficients required to relate each point of the fit to the real number of nodes by using a second polynomial fit that maps the elements of the estimate to the value of the real number of nodes present. The order of the polynomial fit is chosen to be high enough so that the correction oscillates less around the red curve, however the impact of the order is discussed later in the document. The resulting curve is shown in Fig. 5.3

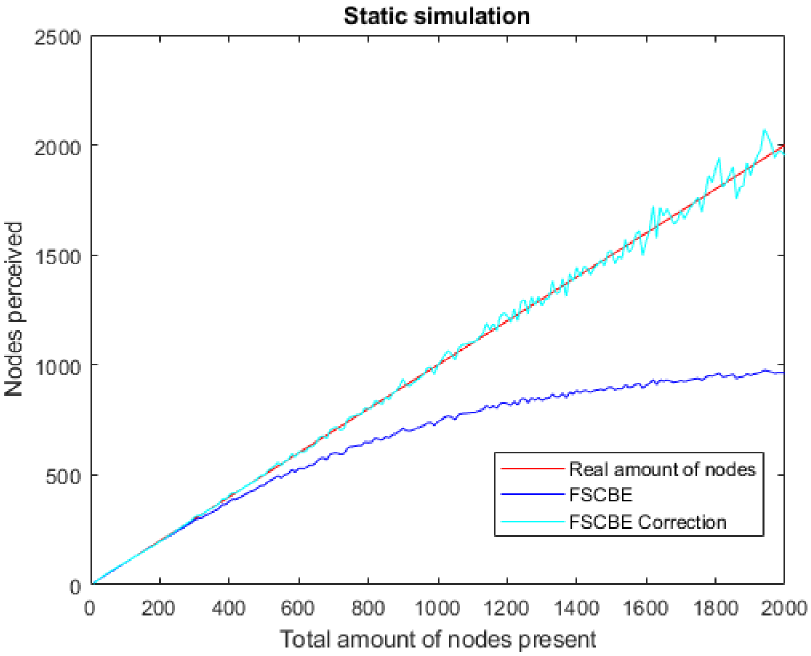


Figure 5.3: Preliminary results of FSCBE and its correction as a function of the real number of nodes present with a frame of length $w = 512$.

Fig. 5.3 shows that the correction is comparable to the Zan estimate and the coefficients calculated can be used on future simulations, which means the CubeSat would not have to calculate them at every estimate. Using this correction method, it is possible to implement FSCBE in a single pass of the satellite, which is tested in the next iteration of experiments. As the FSCBE correction curve has a noisy behavior, a polynomial fit is also used in the next iteration of simulations to smooth the curve.

From the point of view of the objectives set for this work, the results are already satisfactory. Up to this point, an estimation mechanism designed and one adapted to the scenario of interest have been tested, showing good behavior when refined with the details mentioned in this chapter. In the next chapter, the second iteration of simulations and results is presented to go further with the experimentation and analyze the impact that both estimators can have on the sensor network's performance.

Chapter 6

Second iteration of results and analysis

This chapter is dedicated to showing the results of the implementation of the simulations described in chapter 4, taking into account the corrections of chapter 5 to both estimation mechanisms.

6.1. Static simulations

The results of the static simulations are shown in table 6.1.

Table 6.1: RMSE results of the static simulation for both estimators

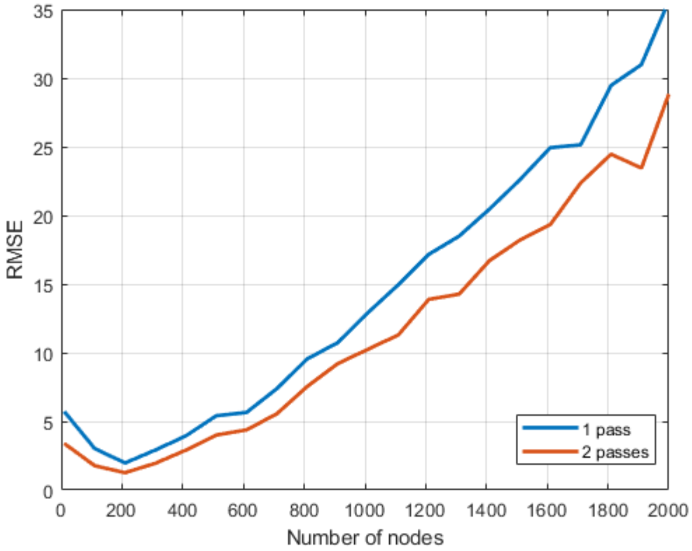
	RMSE			
	1 pass	2 passes	3 passes	4 passes
Corrected FSCBE	32.39	26.28	25.10	25.01
FSCBE polynomial fit	7.85	6.48	4.87	4.85
Zan	34.24	21.14	17.97	16.62
Zan with cumulative averaging	-	-	20.68	20.64
Zan polynomial fit	3.63	5.97	3.26	3.25
Zan polynomial fit with cumulative averaging	-	-	5.48	5.37

In this table, the RMSE between the array of estimated nodes and the array of actual number of nodes is calculated. It can be seen that considering a second pass greatly reduces the error for both estimators, but considering 3 or more passes does not significantly improve performance. Given this, it is preferable to keep only two passes of the satellite, since the improvement in RMSE is not as great as the damage of losing one pass for effective communication between the satellite and the WSN. Furthermore, in the case of Zan, averaging the estimates recalling the previous passes shows better performance than successively averaging with the previous estimate. However, since no more than two passes will be considered in subsequent simulations, this detail is not of great importance. Finally, applying a polynomial fit to the estimators improves performance, but the results of the subsequent sections are necessary to verify if the polynomial fit could be improving performance due to the shape of the curve of nodes to be estimated and it does not necessarily work in all the cases.

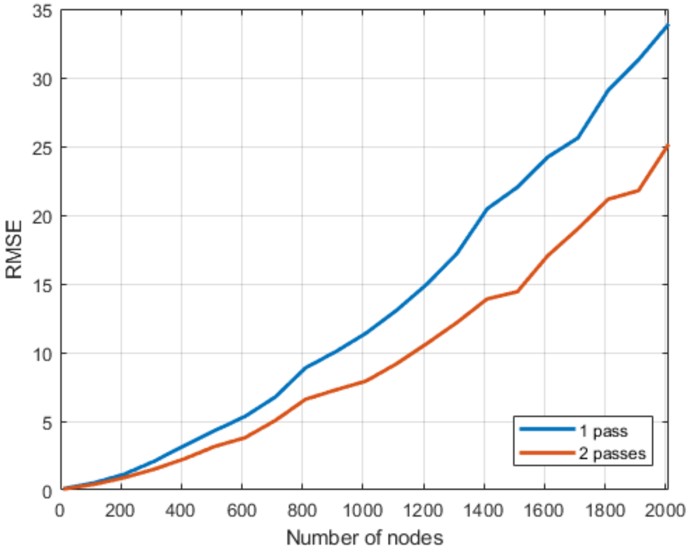
6.2. Dynamic simulations

6.2.1. Randomly distributed nodes

First, Fig. 6.1 shows the change in RMSE when using two passes instead of just one, for the two estimation mechanisms.



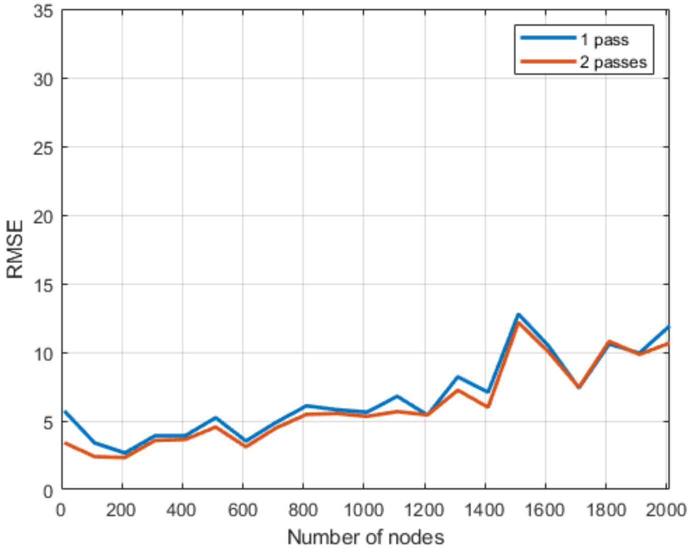
(a) RMSE of corrected FSCBE.



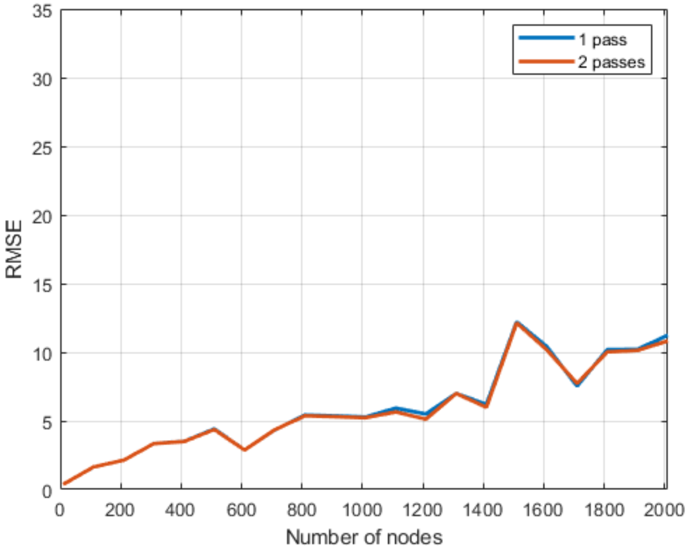
(b) RMSE of Zan.

Figure 6.1: Comparison between the RMSE of both estimators when using one and two passes as a function of the number of distributed nodes in the simulated area.

Second, a similar comparison to the previous one is presented in Fig. 6.2 but using polynomial adjustment in both methods.



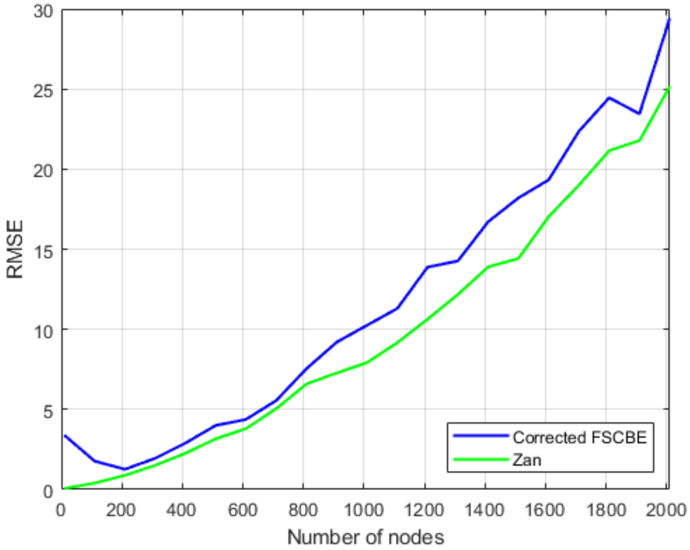
(a) RMSE of corrected FSCBE with polynomial fit.



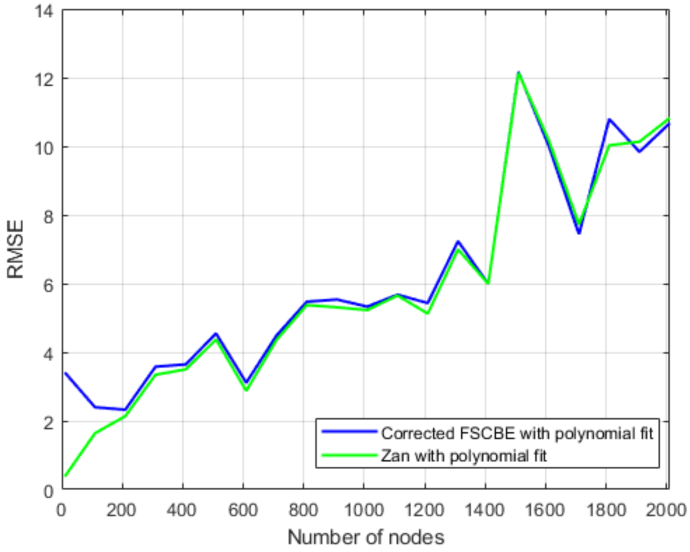
(b) RMSE of Zan with polynomial fit.

Figure 6.2: Comparison between the RMSE of both estimators with a polinomial fit when using one and two passes as a function of the number of nodes distributed in the simulated area.

Third, in Fig. 6.3 a comparison is made between the best results of both methods.



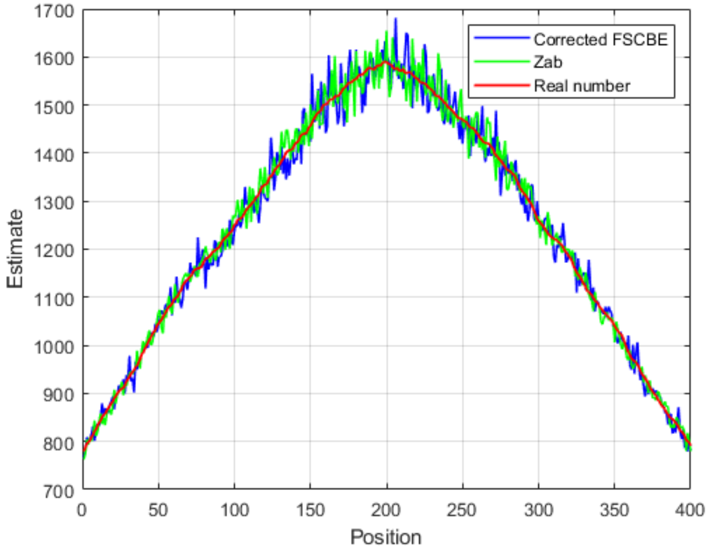
(a) Without polynomial fit.



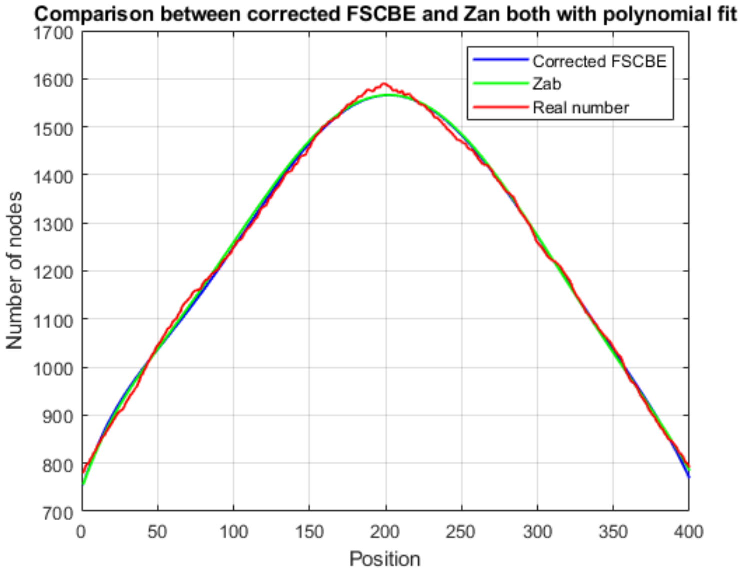
(b) With polynomial fit.

Figure 6.3: Comparison between the RMSE of both estimators using two passes with and without the polynomial fit.

Finally, for this simulation, Fig. 6.4 shows two examples of the estimation performed by both methods, in the case of 2010 nodes distributed randomly, with and without polynomial adjustment and contrasted with the real value of nodes inside the footprint according to position. In Fig 6.4 (a) Zan's curve has a RMSE of 26 while the corrected FSCBE has a RMSE of 29. In 6.4 (b) Zan's curve has a RMSE of 10.6 while the corrected FSCBE curve has a RMSE of 10.8.



(a) Estimation without polynomial fit.

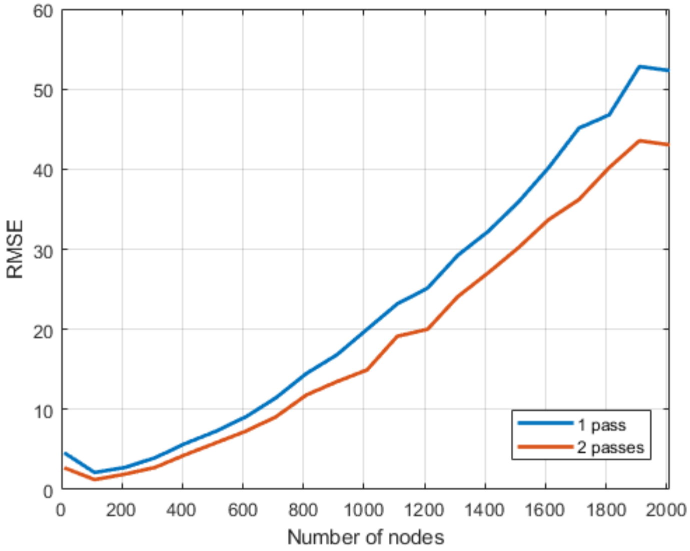


(b) Comparison of both estimators' estimation with polynomial fit.

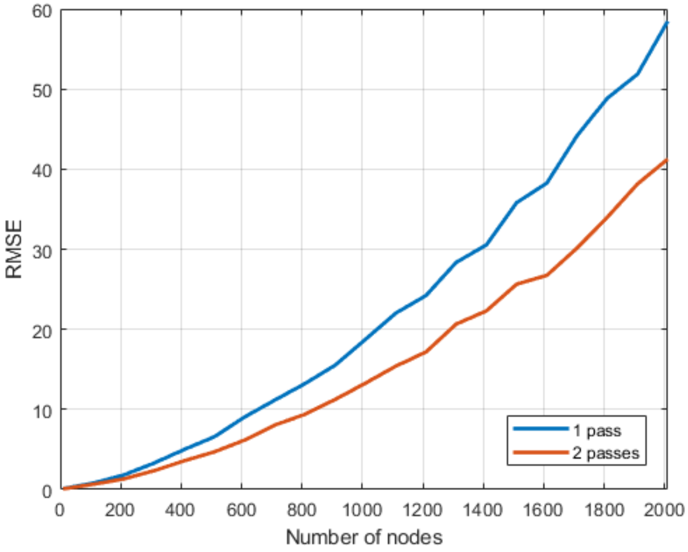
Figure 6.4: Comparison of the estimation of both mechanisms, using two passes, with the real curve of nodes present inside the satellite footprint.

6.2.2. Nodes distributed in clusters

As in the previous subsection, first the RMSE of each estimator is shown using one and two satellite passes in Fig. 6.5. Then, in Fig. 6.6 both estimation mechanisms use polynomial fit and the same comparison is established.

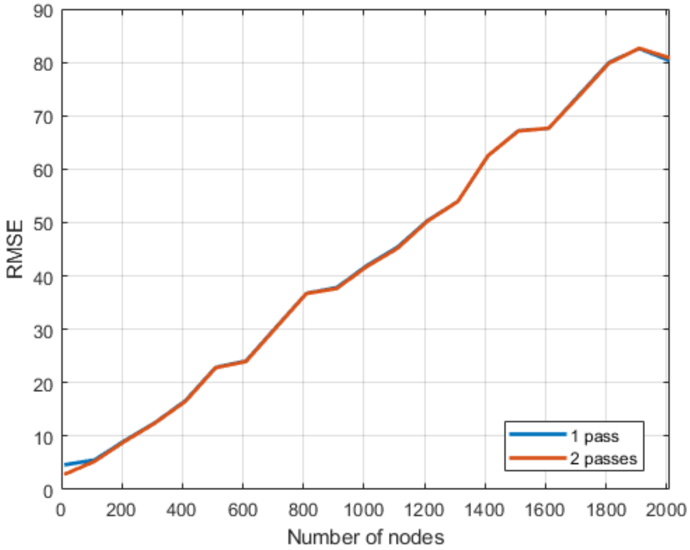


(a) RMSE of corrected FSCBE.

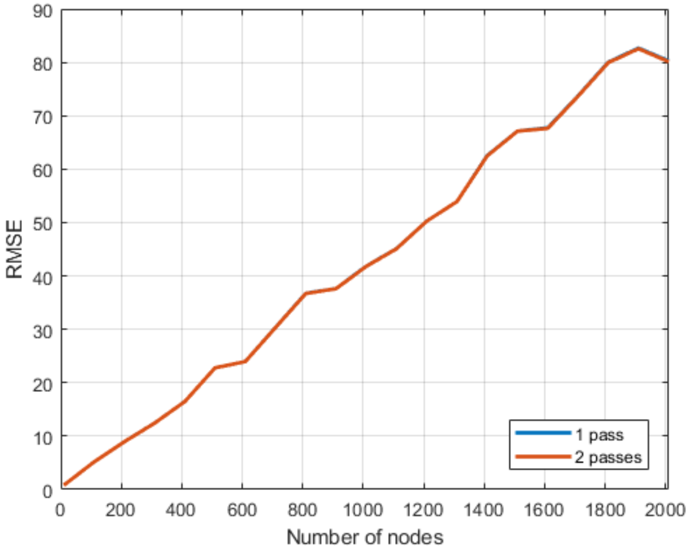


(b) RMSE of Zan.

Figure 6.5: Comparison between using one or two passes by calculating the RMSE of both estimators as a function of the number of nodes in each cluster.



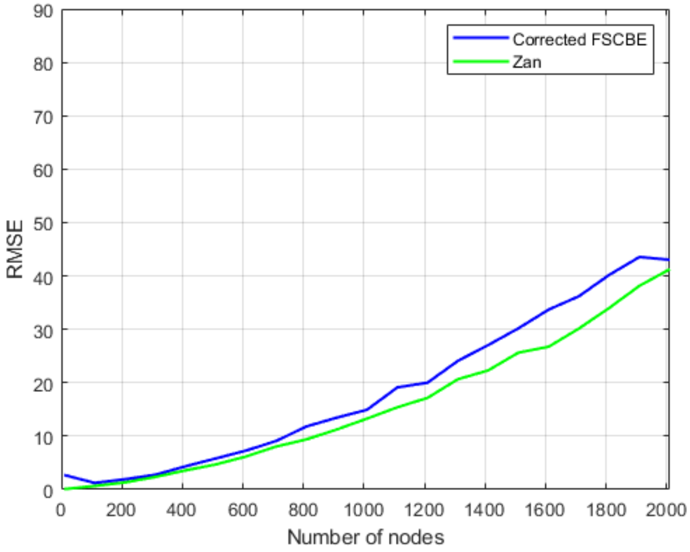
(a) RMSE of corrected FSCBE with polynomial fit.



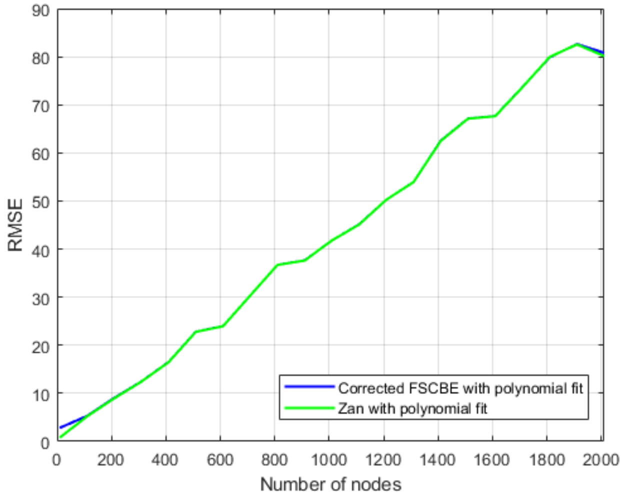
(b) RMSE of Zan with polynomial fit.

Figure 6.6: Comparison between using one or two passes by calculating the RMSE of both estimators with a polinomial fit as a function of the number of nodes in each cluster.

Then, Fig. 6.7 shows the comparison between both estimators using two passes each, with and without polynomial fit.



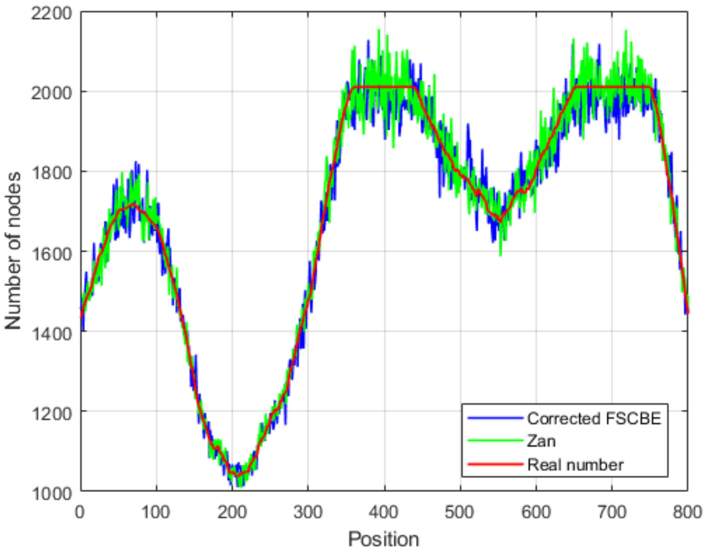
(a) Without polynomial fit.



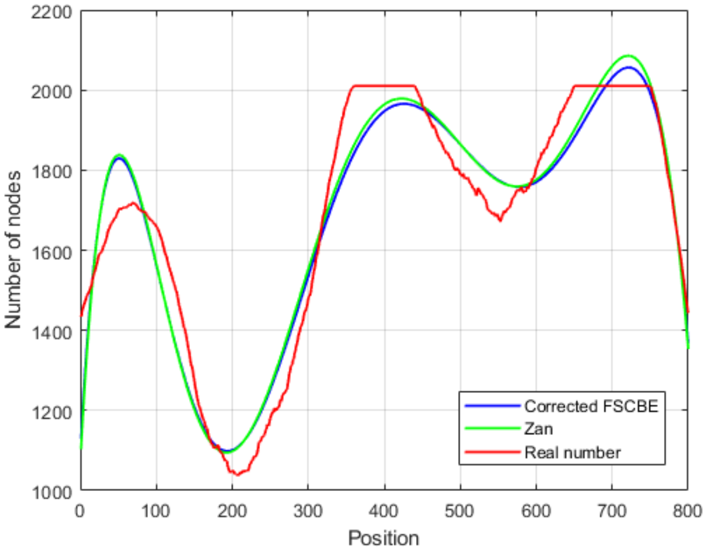
(b) With polynomial fit.

Figure 6.7: Comparison between the RMSE of both estimators, with and without polynomial fit, as a function of the number of nodes on each cluster .

Finally, Fig. 6.8 compares the estimates made for the case with the higher number of nodes in each cluster, with and without polynomial fit. In Fig 6.8 (a) Zan's curve has a RMSE of 41 while the corrected FSCBE has a RMSE of 43. In 6.8 (b) Zan's curve has a RMSE of 80.1 while the corrected FSCBE curve has a RMSE of 80.8.



(a) Without polynomial fit.



(b) With polynomial fit.

Figure 6.8: Comparison of the estimation of both mechanisms in the case of the greater number of nodes distributed in clusters, with and without polynomial fit.

6.3. Feedback to communication protocol

This section shows the results of implementing feedback from both estimators to the FSA communication protocol.

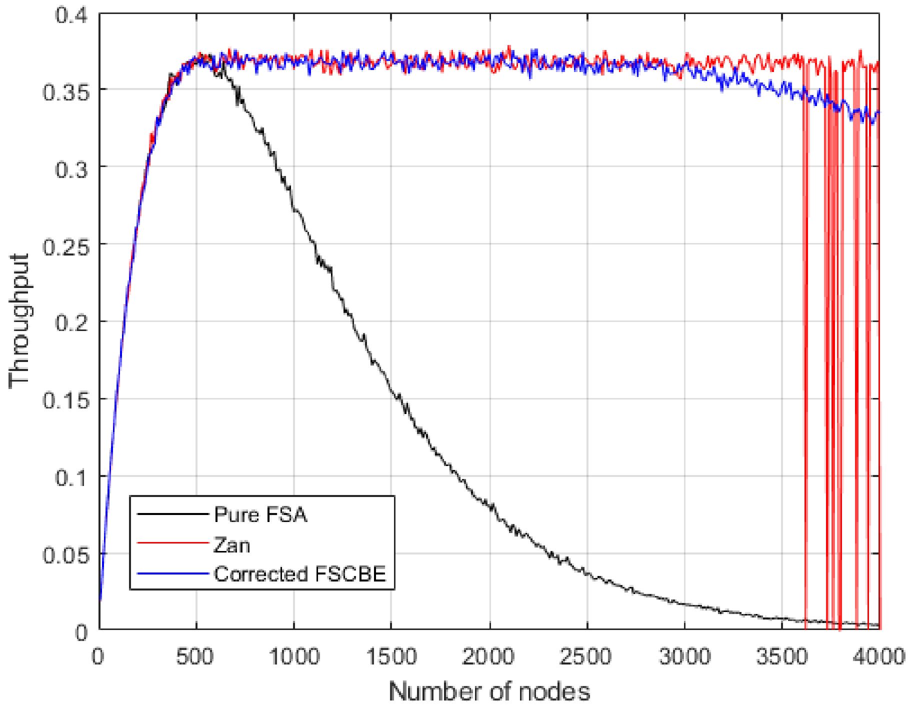


Figure 6.9: Throughput comparison of pure FSA and FSA with estimation mechanisms' feedback.

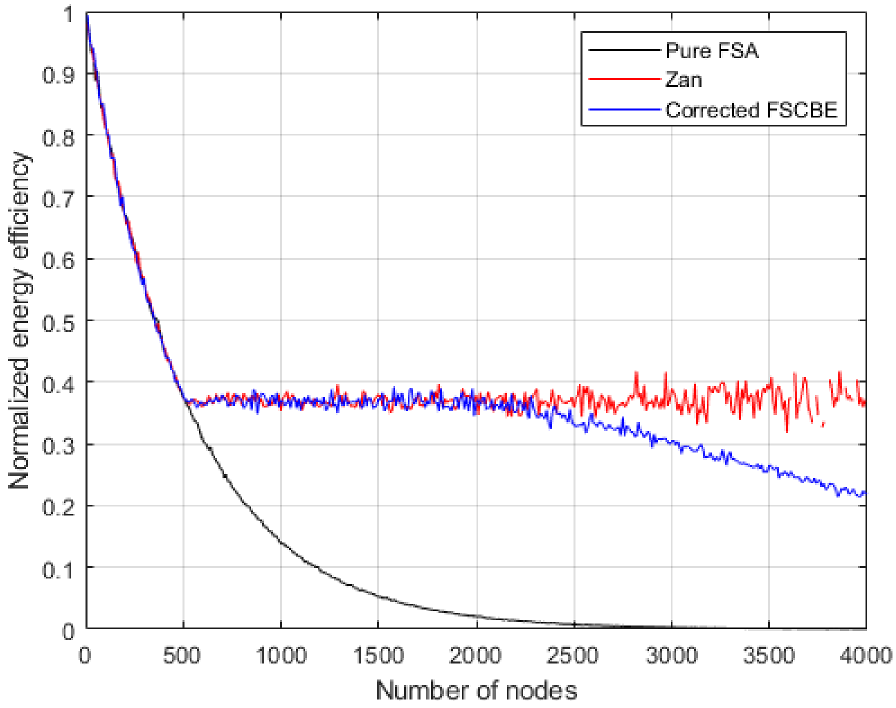


Figure 6.10: Energy efficiency comparison of pure FSA and FSA with estimation mechanisms' feedback.

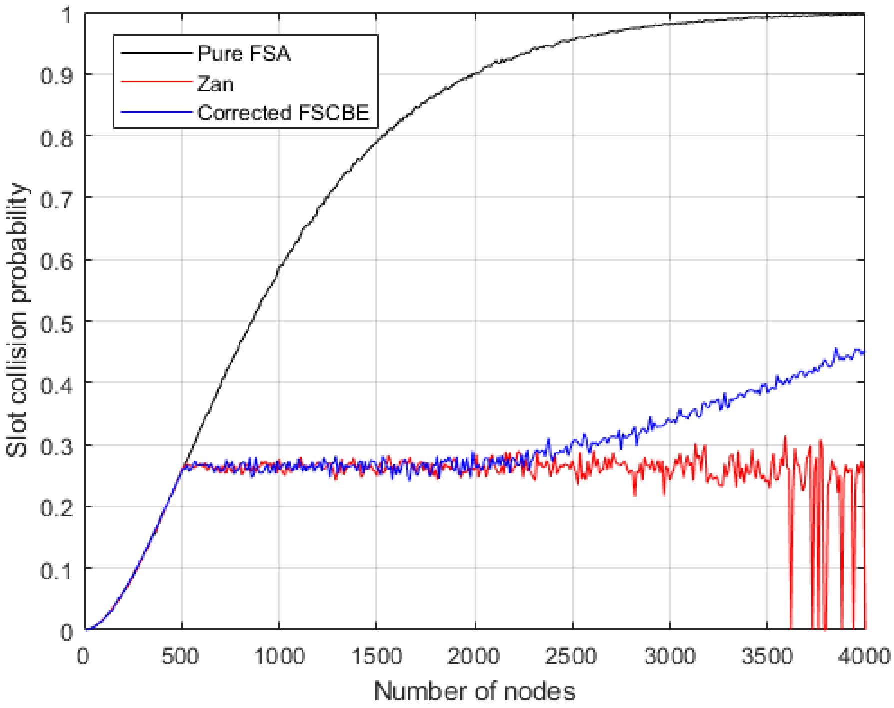


Figure 6.11: Slot collision probability comparison of pure FSA and FSA with estimation mechanisms' feedback.

In figures 6.9, 6.10, and 6.11 is shown the performance of Framed Slotted Aloha according to its throughput, energy efficiency and slot collision probability when it is fed back with the size estimation carried out by corrected FSCBE and Zan estimators, following the Slotted Aloha Game method. The results represent the average values of 30 iterations of each estimator, considering two passes of the nanosatellite.

6.4. Analysis

This section presents the analysis performed on the results obtained in the previous sections. In the first place, the results of the static simulation on Table 6.1 directly show that although an improvement in the performance of the estimation mechanisms is achieved by opting for two satellite passes, this improvement does not remain of the same magnitude when choosing more passes still. For this reason, it is decided to opt for two passes and discard a greater number, since choosing a greater number of passes does not provide a significant improvement that compensates for the loss of a lap around the earth that could be dedicated to communication with the WSN. If more than two estimation passes were implemented, the results of Fig 6.1 show that performing an average between the results of each pass's estimates has less RMSE than progressively averaging. However, the advantage is very small compared to the memory cost of having the CubeSat remember each position's estimates to perform the average calculation.

Furthermore, the graphs of the dynamic simulations Fig. 6.1 and Fig. 6.5 show the advantages of choosing two satellite passes for the estimation, which are more pronounced for the case of Zan than for FSCBE. This implies that it is more attractive to choose two passes for Zan than for FSCBE. On the contrary, Fig. 6.2 and Fig. 6.6 show that the advantage of choosing two passes is not present in the case of polynomial fits as it is in the pure versions of the estimators. Fig. 6.2 shows an advantage of around 1 in terms of RMSE in the largest case, while in Fig. 6.6 the advantage is non-existent. This is because the changes from one pass to the other for both estimators have to be very large for the polynomial fit to look different. Since the change is not that steep, the curves for the one-pass and two-passes polynomial fits are very similar. In the case of Fig. 6.6 this phenomenon is more accentuated because as the number of passes increases, the general appearance does not change enough to establish a difference in the polynomial fit. Moreover, Fig. 6.7 (b) shows that the fits are almost identical for both estimators.

Afterwards, Table 6.1 also shows that implementing a polynomial adjustment to smooth the results allows to have a lower RMSE value, a result that is also supported by comparing the results shown in Fig. 6.1 and Fig. 6.2 but which are nevertheless contradicted by the high RMSE value presented in Fig. 6.6 in contrast to Fig. 6.5. The above's explanation is that the polynomial fit works very well to fit a set of data to a parabolic curve, so it increases the error when applying it to a scenario of more abrupt variations such as the case of clusters. For this reason, whether or not to choose a polynomial fit depends on the application for which the satellite is intended. If this corresponds to orbit around the earth, offering coverage to the different WSN of the earth's surface, it is not the optimal option, and it is better to choose pure mechanisms without polynomial adjustment. Otherwise, if it is an application in which nodes are accumulated, as shown in Fig. 6.4, then the polynomial fit can be an option.

On the other hand, figures 6.3 (a), 6.7 (a) and the table 6.1 show that the corrected version of FSCBE has higher RMSE in general in comparison to Zan, so the designed mechanism presents a viable method to solve the network size estimation problem but has not the best performance. The corrected FSCBE estimator shows good performance in the different simulations carried out, with a number of nodes that can reach up to 2000 nodes. Overall, the results on RMSE are good for both estimators since even when confronted with a large number of nodes, the error between the estimate and the real number is little. RMSE is a very punishing metric when an estimate deviates far from the actual observed value, and for this reason, it is used in cases where significant errors are undesirable. In the case of satellite communication with a large number of nodes, it is desirable to have an error as low as possible to meet the demands of the channel as faithfully and accurately as possible. This previous allows characterizing both methods as adequate for the proposed scenario.

In terms of computational cost, an analysis can be made of how the time required to make the estimates increases when parameters increase when the satellite is in orbit. If the number of nodes increases, the Zanella estimator still uses the same amount of data to estimate (successes, failures, and the constant window size), and the necessary calculations do not have a more significant influence from the number of nodes that are contending. The same can be said of FSCBE since it uses the results of an FSA communication window in terms of collisions and successful transmissions but not of what happens in greater detail within such communication.

Since the satellite travels approximately 7.5 km/s in its orbit, it goes around the world in 90 minutes, that is, 5400 seconds. It is common in the literature that the length of a slot in a communication protocol such as FSA measures an amount of time equal to the time of a packet transmission. In the existing literature, applications with packet lengths of around 1000 bits can be found [29], also the packet transmission time varies depending on the application, such as 5ms [30]. If the transmission of a packet takes 5 ms, then a frame takes $512 \cdot 5 = 2560$ ms, that is 2.56 seconds. Then, with that frame length, there could be 2109 contention windows, i.e. positions, in a world tour. FSCBE needs two integers of 3 bits maximum for each position for successes and failures of the FSA frame to calculate the estimate. The most simple way of storing a number implies the use of 10 bits for a 3 digit number, thus $2 \cdot 10 \cdot 2109 = 42.180$ bits are needed in the first pass. In the second pass it is not necessary to store integers since the result of the communication through FSA is used to perform the estimation. Therefore, only the estimated number of nodes is stored in memory. According to the experiments carried out, each position's size estimate is an integer of up to 4 digits, which means $14 \cdot 2109 = 29.526$ bits are needed after the estimations are done, plus the bits needed for the correction coefficients. On the other hand, Zan only needs to store the latest estimate of each position if two passes are considered, and thus 29.526 bits are needed.

Regarding the number of bits required to store FSCBE's correction coefficients, Fig. 6.12 shows the effect that the order of the polynomial correction has on the RMSE of corrected FSCBE for the cluster's experiment. It can be seen that a coefficient of order 4 is sufficient to have a low RMSE compared to lower orders.

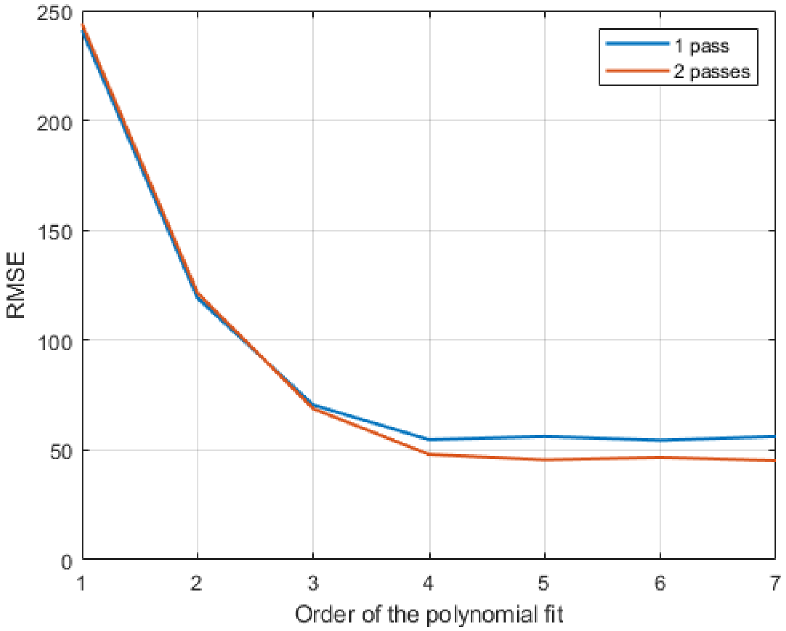


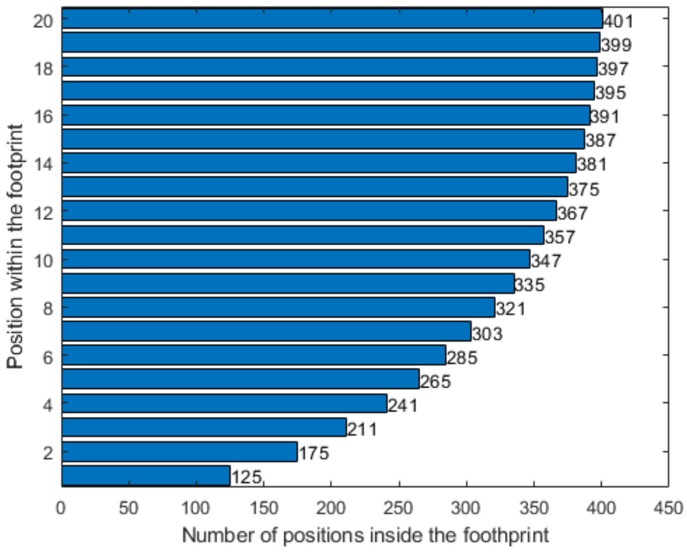
Figure 6.12: RMSE of corrected FSCBE as a function of the order of the polynomial fit used for the correction under clusters simulation.

These coefficients are numbers with up to 15 digits that can require up to 64 bits for each one, which then implies that if we chose a polynomial order of 4, 256 bits would be needed.

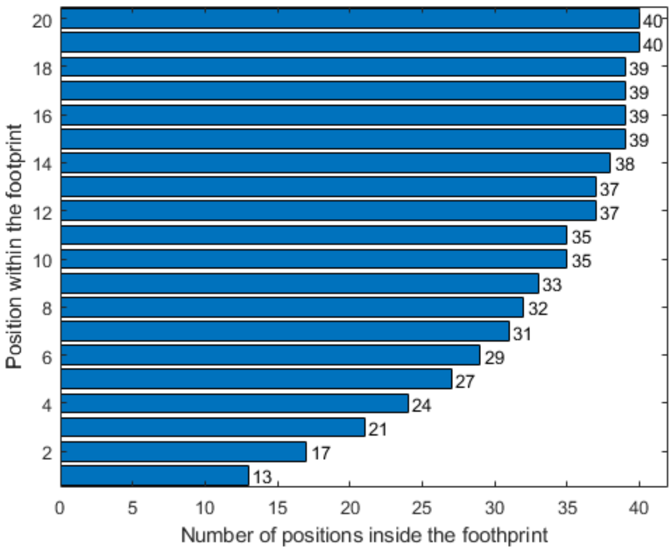
In terms of the impact of both estimators in FSA's performance, figures 6.9, 6.10 and 6.11

show for both methods that there is an improvement in throughput when using FSA as the communication protocol, articulated with Slotted Aloha Game that uses the estimation to set the threshold from which to restrict the transmissions. Pure FSA's throughput decreases once the number of nodes contending exceeds the number of slots in the frame, its energy efficiency decreases as the number of nodes increases, and its collision probability increases with the number of nodes. When either of the two estimators' feedback is applied, these three metrics have the same behavior until the network's size exceeds the frame length. From that point, throughput oscillates close to 0.37 for throughput, 0.37 for energy efficiency, and 0.26 for slot collision probability. However, with a number of nodes higher than 2000, corrected FSCBE starts to show worse performance than Zan, and this latter shows discontinuities when the network's size is closer to 4000 nodes. These results allow us to affirm that the feedback from both estimators contributes a benefit to the MAC protocol's performance even if the number of nodes for which its parameters were adjusted is exceeded. Zan shows the best result between the two estimators. The discontinuities that Zan presents and the worse performance of corrected FSCBE are explained in Chapter 5. To obtain better performance for a more extensive network size, the frame length has to be increased, and the coefficients for FSCBE correction recalculated.

The metrics' results vary little if 1 or 2 passes of the nanosatellite are considered as the difference in error is not large enough. The latter motivates the choice of a single estimation pass.



(a) 3600 positions.



(b) 360 positions.

Figure 6.13: number of positions within the satellite footprint as a function of the distance to the center of it.

In Fig 6.13 both graphs show an example of how the number of positions in which a node is within the coverage area of a satellite changes depending on how close to the circumference's extremes it is. Position 0 represents the farthest distance from the center of the footprint, while position 20 represents a node placed right in the center's trajectory. Given the symmetry, only the results of one half of the nodes are shown, and the difference between graphs (a) and (b) is the granularity of the discretized space. These results are of great importance in terms of the notion of *fairness*, since nodes that are far from the center of the circumference have fewer opportunities to transmit. Slotted Aloha Game bases its operation on restricting the number of transmitting nodes. This can be detrimental to a node that has few opportunities to transmit if, at those times, there are a large number of nodes that also seek to contend for the channel. One way to solve this can be to propose, based on game theory, a mechanism that makes each node choose to transmit in the window where there is the greatest probability of transmitting according to the Slotted Aloha Game raised above. That probability would be communicated to them by the nanosatellite through a broadcast. Since all nodes would choose positions without knowing the others' decision, the process would be iterative, in such a way that a Nash Equilibrium would be reached if all nodes choose a position from a certain number of iterations from which they would not choose any other position because of not perceiving an advantage in choosing another one.

Chapter 7

Conclusions

The CubeSat standard is presented as a way to overcome the economic barriers that prevent a large part of the world's countries from accessing satellite technology. However, this type of satellite has several limitations, which makes it necessary to research new ways to deal with the challenges of a satellite but in a more restricted setting. In the case of this work, the subject related to the management of the communication channel is addressed to control access to the medium of a sensor network with DtS communication. To properly manage the channel in this particular scenario, it is necessary to know the number of nodes that can potentially contain each other at any given time, and since this information is unknown to the CubeSat, it is necessary to develop mechanisms that allow estimating the size of the sensor network.

Regarding the latter, first, an estimation mechanism is designed that allows providing the nanosatellite with approximate information on the number of nodes that share the communication channel. In addition, an existing mechanism for RFID scenarios is adapted to the particular scenario of the case study.

Second, software simulations are implemented to extract performance metrics from the estimate convergence time, the RMSE error, and the scalability of both estimation methods.

Third, the estimates of these two size estimation mechanisms are applied to the operation of a media access control protocol, being able to compare how they improve network performance.

Having carried out these three points, the specific objectives of the work of this document are met and, in this way, the general objective of providing two network size estimation mechanisms that allow a satellite to feed back a MAC protocol in order to obtain an improvement in network performance.

The results obtained are satisfactory, not only in terms of the good performance of the two estimators, but also because they offer versatility and adaptability to the specific scenario in which they can be implemented, not being limited to a specific number of satellite passes to perform the estimation and being able to vary its parameters to adjust to the characteristics of the implementation. Zan estimator shows overall the best results in terms of RMSE and FSA performance. Although for both estimators, a single nanosatellite pass is sufficient for the estimation process, the choice of how many passes to consider depends on the application and the variability of the network topology.

Finally, it is proposed as future work the design of a mechanism based on game theory that allows distributing the positions in which the nodes are within the satellite footprint, in such a way that the channel resources are distributed equitably and that the management of the communication channel is not detrimental for nodes that are at a disadvantage due to their position in relation to the satellite and the few positions in which they are present within the coverage area.

BIBLIOGRAPHY

[1] Andy, "How many satellites orbiting the Earth in 2019? | Pixalytics Ltd." https://www.pixalytics.com/satellites-orbiting-earth-2019/, 2019. Accessed: 2020-01-10.

[2] X. P. F. Verger, R Ghirardi, I Sourbès-Verger, *L'espace nouveau territoire : atlas des satellites et des politiques spatiales.* Paris: Belin, 2002.

[3] A. Duret, *Alain Duret, Conquête spatiale : du rêve au marché.* Paris: Gallimard, 2002.

[4] James Manyika, Michael Chui, Peter Bisson, Jonathan Woetzel, Richard Dobbs, Jacques Bughin and D. Aharon, "Unlocking the potential of the Internet of Things | McKinsey." https://www.mckinsey.com/business-functions/mckinsey-digital/our-insights/the-internet-of-things-the-value-of-digitizing-the-physical-world, 2015. Accessed: 2020-07-10.

[5] D. Helbing and E. Pournaras, "Society: Build digital democracy." http://www.nature.com/news/society-build-digital-democracy-1.18690, nov 2015. Accessed: 2020-01-10.

[6] M. Lazarescu, *Wireless Sensor Networks for the Internet of Things: Barriers and Synergies*, pp. 155–186. 09 2017.

[7] C. Buratti, A. Conti, D. Dardari, and R. Verdone, "An overview on wireless sensor networks technology and evolution," sep 2009.

[8] S. L. Peng, S. S. Li, X. K. Liao, Y. X. Peng, and N. Xiao, "Estimation of a Population Size in Large-Scale Wireless Sensor Networks," *Journal of Computer Science and Technology*, vol. 24, no. 5, pp. 987–997, 2009.

[9] A. Mann, "Starlink: SpaceX's satellite internet project | Space." https://www.space.com/spacex-starlink-satellites.html, 2020. Accessed: 2020-07-20.

[10] B. Wang, "SpaceX Starlink Satellites Could Cost $250,000 Each and Falcon 9 Costs Less than $30 Million – NextBigFuture.com." https://www.nextbigfuture.com/2019/12/spacex-starlink-satellites-cost-well-below-500000-each-and-falcon-9-launches-less-than-30-million.html, 2019. Accessed: 2020-07-20.

[11] M. Radi, B. Dezfouli, K. Abu Bakar, and S. Abd Razak, "Integration and analysis of neighbor discovery and link quality estimation in wireless sensor networks," *The Scientific World Journal*, vol. 2014, 2014.

[12] T. Ferrer, S. Céspedes, and A. Becerra, "Review and evaluation of mac protocols for satellite IOT systems using nanosatellites," *Sensors (Switzerland)*, vol. 19, no. 8, pp. 1–29, 2019.

[13] Mathworks, "MathWorks - Makers of MATLAB and Simulink - MATLAB Simulink."

https://www.mathworks.com/?s_tid=gn_logo. Accessed: 2020-03-29.

[14] T. Opazo, “Requerimientos, implementación y verificación del nano-satélite Suchai,” tech. rep., Universidad de Chile, 2013.

[15] C. S. Wu and G. K. Ma, “Performance of packet reservation MAC protocols for wireless networks,” *IEEE Vehicular Technology Conference*, vol. 3, pp. 2537–2541, 1998.

[16] B. Zhao, G. Ren, and H. Zhang, “Slotted ALOHA game for medium access control in satellite networks,” *2019 IEEE/CIC International Conference on Communications in China, ICCC 2019*, no. Iccc, pp. 518–522, 2019.

[17] K. Jamieson and H. Balakrishnan, “Sift : A MAC Protocol for Event-Driven Wireless Sensor Networks,” *European Workshop on Wireless Sensor Networks, Springer*, pp. 1–23, 2003.

[18] G. Tsoumanis, A. Papamichail, and V. Dragonas, “Implementation of a Topology Independent MAC (TiMAC) Policy on a Low-Cost IoT System,” pp. 1–13, 2020.

[19] C. Wang, L. Liu, H. Ma, and D. Xia, “SL-MAC: A Joint TDMA MAC Protocol for LEO Satellites Supported Internet of Things,” *Proceedings - 14th International Conference on Mobile Ad-Hoc and Sensor Networks, MSN 2018*, pp. 31–36, 2018.

[20] C. Wang, L. Liu, H. Ma, and D. Xia, “Lst-mac: A low-latency hybrid mac protocol for leo satellite supported iot,” *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS*, vol. 2019-Decem, pp. 480–487, 2019.

[21] N. Shahin, R. Ali, and Y. T. Kim, “Hybrid Slotted-CSMA/CA-TDMA for Efficient Massive Registration of IoT Devices,” *IEEE Access*, vol. 6, pp. 18366–18382, 2018.

[22] R. Ali, S. S. Lor, and M. Rio, “Two algorithms for network size estimation for master/slave ad hoc networks,” *2009 IEEE 3rd International Symposium on Advanced Networks and Telecommunication Systems, ANTS 2009*, pp. 23–25, 2009.

[23] A. Zanella, “Estimating collision set size in Framed Slotted Aloha wireless networks and RFID systems,” *IEEE Communications Letters*, vol. 16, no. 3, pp. 300–303, 2012.

[24] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” *Proceedings of the Hawaii International Conference on System Sciences*, vol. 00, no. c, p. 223, 2000.

[25] M. Abo-Zahhad, S. M. Ahmed, N. Sabor, and S. Sasaki, “Mobile Sink-Based Adaptive Immune Energy-Efficient Clustering Protocol for Improving the Lifetime and Stability Period of Wireless Sensor Networks,” *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4576–4586, 2015.

[26] C. H. Cheng, S. W. Liu, J. Y. Lin, and Y. F. Huang, “Energy-efficient sink location service protocol to support mobile sinks in wireless sensor networks,” *Proceedings - International Symposium on Parallel and Distributed Processing with Applications, ISPA 2010*, pp. 103–108, 2010.

[27] B. S. Malhotra and A. A. Aravind, “A master-sink based model for on-site tracking of multiple mobile targets in wireless sensor networks,” *Proceedings - 2005 International Conference on Intelligent Sensing and Information Processing, ICISIP’05*, vol. 2005, pp. 104–109, 2005.

[28] dgt Dei Gruppo Telecomunicazioni, “MATLAB source code.” `http://dgt.dei.unipd.it/pages/read/54/`. Accessed: 2020-10-01.

[29] J. Su, G. Ren, Q. Wang, and H. Zhang, “Randomly Pre-coded Packets based Random Access Scheme for IoT-Oriented Satellite Networks,” *IEEE Access*, vol. 8, 2020.

[30] D. Li, S. Wu, Y. Wang, J. Jiao, and Q. Zhang, “Age-optimal harq design for freshness-critical satellite-iot systems,” *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2066–2076, 2020.