Contribution to the Development of Autonomous Satellite Communications Networks

–

The Internet of Satellites

by

Joan Adrià Ruiz-de-Azua Ortega

Submitted to the Department of Network Engineering
in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

at the Technical University of Catalonia – UPC BarcelonaTech

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Thesis Advisors:

Dr. Anna Calveras Augé
Dr. Adriano Camps

September, 2020

Contribution to the Development of Autonomous Satellite Communications Networks: The Internet of Satellites

by

Joan Adrià Ruiz-de-Azua Ortega

Submitted to the Department of Network Engineering on September 2020, in partial fulfilment of the requirements for the degree of Doctor of Philosophy.

Abstract

The space is experiencing a revolution motivated by the emergence of satellite services that can satisfy current environmental, socio-economic, and geo-political demands. The transformation becomes more explicit in the predominant activity sectors, such as Earth Observation and broadband telecommunications. Earth Observation satellite systems have become dependable resources for environmental and climate monitoring, weather forecast, governmental activities, modern agriculture, and countless other commercial and industrial applications. The incursion of 5G in the aerospace domain has motivated the conception of satellite systems as a promising platform to achieve desired applications that require global coverage, and cope the limitations of ground facilities. The numerous demands can be summarized in two main system requirements: (1) increase of data transfer capacity, and (2) decrease the end-to-end communications latency. Some satellite systems are starting to adopt novel structural and functional paradigms grounded upon distribution and increased on-board capabilities in order to address these ever growing user and application requirements.

Distributed Satellite Systems have emerged as an effective solution to meet these new demands by encompassing multiple satellites orbiting and operating simultaneously to satisfy a common goal. The generic definition leads to conceive unique architectures that can be considered instances of this distributed approach. Federated Satellite Systems has become a serious candidate to exploit all the potential of distributed architectures by establishing opportunistic collaborations among satellites to share unallocated satellite resources. These collaborations—formally called federations—allows to conceive the space as a cloud system in which satellites can leverage from other resources to improve their current mission performance. Numerous researches have investigated the potential benefits of applying satellite federations in future satellite missions, emphasizing the synergies with commercial initiatives, and the added-value of combining heterogeneous capacities (e.g. data fusion). The successive investigations have been centered on making feasible these satellite federations by developing novel technologies, such as mechanisms to architect satellite missions with federations, proposals to quantify the value and price of satellite resources, and solutions to provide the data security level required in these interactions. Among the different developments, multiple aspects related to their design and operation are still open fields of study, and some have been expressly tackled in this research. The development of communications capabilities that allows the establishment of these satellite federations are still technology challenges that need to be addressed. This dissertation contributes to fill this gap by defining distinct mechanisms to deploy a network infrastructure for satellite federations.

A networked environment in which satellites are able to establish sporadically, opportunistically, and autonomously federations has been discussed throughout the present research. This context has been called the Internet of Satellites paradigm, and promotes the temporal deployment of inter-satellite networks for this purpose. These networks are composed of heterogeneous satellites with distinct resources, capacities, orbit trajectories, owners, and operators. These features—in conjunction with satellite motion—pose a considerable challenge on the definition of end-to-end communications routes composed of intermediate satellites. In conjunction with the definition of the paradigm, a review of current routing protocols—responsible to define routes in a network—from other satellite network proposals is conducted to identify the ideal protocol that deploy such dynamic networks. The outcome remarks the need to combine multiple capabilities from different domains to achieve the desired performance. Among them, the capability to predict future inter-satellite links becomes crucial to mitigate the fragmentation of the network. This phenomenon entails the isolation of satellites, which limits their capability to communicate with other spacecraft. With this premise in mind, this dissertation presents a predictive protocol that enables to perform the estimation of these inter-satellite contacts in a distributed manner; i.e. in each satellite independently. This new satellite capability may support the routing protocol by allowing the estimation of future routes composed of a sequence of satellite contacts over time.

The research presented in this dissertation also tackles other questions that remained unanswered: How can satellites be aware of the available resources offered by other satellites? What are the necessary mechanisms to deploy a federation in this opportunistic context? Without the proper technology development to address these challenges the routing protocol could become useless. Therefore, a protocol stack to deal with this technology gap has been developed. The federation protocol suite proposes the design of two protocols that allow to deploy satellite federations. The Opportunistic Service Availability Dissemination Protocol allows notifying the services that are available in a satellite, while the Federation Deployment Control Protocol formalizes the different rules to establish and manage a satellite federation. The application of these protocols considerably enhanced the capability of the satellite system to download data, becoming thus enablers of future satellite missions. The achieved performance has motivated the development of a dedicated system that includes the protocol suite, and the hardware components to establish federations. It has been named Federated Satellite Systems Experiment payload, and includes a communications device to create inter-satellite links. This system has been verified in a stratospheric balloon campaign, which emulated a mobile scenario. This dissertation discusses the results of the campaign, which emphasize the benefits and viability of this implementation.

We expect that the contributions of this dissertation may encourage to keep investigating on this inter-satellite communications domain, which are necessary to establish satellite federations in future missions.

Thesis advisors: Dr. Anna Calveras Augé
Associate Professor, Department of Network Engineering

Dr. Adriano Camps
Full Professor, Signal Theory and Communications Department

To my brother Santi,
my parents, Araceli and Francisco,
and my *samwise*, Alba.

Acknowledgements

Many are the people that I want to express my gratitude and I would like to mention. Surely, the following paragraphs do not do them justice, and probably I am missing extraordinary persons that I should also recognize. It has not been easy to write these lines, but all of you deserve at least this effort.

I would like to undoubtedly address my first words to my two advisors: Adriano and Anna. Adriano, I can still remember the first time that we met in your office and talked about the possibility to conduct the doctorate. Certainly, you are one of the most persevering and hard-working persons that I have ever worked with, pushing any boundaries and overcoming them. You have transmitted me your passion for this precious research field, and motivated me to achieve milestones that I would ever imagine that I could achieve. Before meeting you, the space was a theory concept that I could taste only in the books and the university. Now, I have had the fortune of experiencing the development of small-satellites in person. This journey that we have traveled side-by-side would not be possible without the certain vision of Anna. You have encouraged me to keep closing and finishing the different open research lines, while I was releasing my imagination. As a bird cannot fly without two winds, you have become my winds. From them I have learnt so many things that it would be impossible to mention them all. You have encouraged me to openly discuss numerous topics, promoted my critical and innovative thinking, fostered my independence, stimulated my curiosity, and demonstrated that the perseverance is key to succeed. My doctorate—this unpredictable roller coaster—would not be the same without your participation and guidance.

I would like also to mention the amazing companions whom I had the privilege to work with in the NanoSat Lab. JuanFran, Lara, Adri, Carles, Marc, Ricard, Albert, Marco, Oriol, Arnau, Pol, David, Andrea, Guillem, Carlos, Victoria, Hyuk, Jorge, and a long etcetera. I experienced from everyone magical moments that could never be forgotten: the days installing the ground station at Montsec; the days of barbecue; the Ricard's candies; the mysterious friend in the Washington flight; The eternal debates of future radio inventions; the sysadmin blogs; the innumerable passwords created around me; the overflow of sushi; the quotation and cake of the week; the online trivial; the Star Wars sessions; or the efforts to avoid jeopardizing the endless meetings; *but* I would like to remark the figure of Carles Araguz who quickly has become a reference and an endless support in my research. I could not finish this paragraph without also recognizing my four musketeers. We all together survived and—in some way—enjoyed the craziest ever satellite mission. All of you turned the lab into something special, a place where you can feel at home.

The next person to who I am extremely grateful is, undoubtedly, prof. Alessandro Golkar. He opened me the possibility to work in the satellite federations concept and welcomed me in his team in Skoltech. This unique team is composed of admirable engineers and researchers, and from the very beginning considered me as one more. Simone and Nicola, even from the bad moments, I also keep them as fond experiences that matured me. We have overcome different difficulties, and we could experience a magic balloon campaign in the middle of Russia. The memories of recovering the balloon during a cold night with Alessandro will never disappear. Thanks for this incredible experience, which discovered a new incredible domain.

Like them, it is impossible for me not to mention my family and friends. They have always been there when I needed support, making me laughing and enjoying the small—but essential—things. I find it impossible to enumerate every single one of you: my parents, my brother, my entire family, the 'vermut' friends, the 'independitzats' folks, mes Supaéro camarades, the 'hola faig telecos' troop, and all the ones for who I cannot classify. I would not be here without the accumulated experiences that we lived together. You are all for me. I would, however, specially mention Jordi Rancé and Elisenda Bou-Balust who motivated me to start this journey with CubeSats.

Last, but not least, I would not end this without mention the person that has walked with me in this adventure. Alba, you have been the pillar which supported my madness, the numerous nonsense explanations, and the unacceptable schedule. You encouraged me to continue forward with the different ups and downs of this journey, which allowed also to enjoy together the great moments. This dissertation could not be possible without you. It is also yours.

Contents

List of Figures

List of Tables

1

Introduction

1.1 A new space for new demands

Since 1957 with the launch of the first artificial satellite *Sputnik 1*, the space has been populated by a wide variety of satellite systems. The development of new technologies proliferated the emergence of different satellite platforms for multiple purposes. Numerous classifications of satellites have been proposed in the last years, despite the most common categorization relays on their total mass. Previous research (Kramer and Cracknell, 2008; Sweeting, 2018) identified the categories of large-, small-, mini-, macro-, nano- and femto-satellites (Figure 1.1). In this context, the term of large-satellites or big-satellite refers to traditional satellite platforms with large structures, and considerable mass. The global miniaturization of the technology influenced the satellite design by enabling small satellites with reduced mass. This trend not only drove the satellite shape, but also noteworthy impacted on the perception of a satellite, and its development.

The current technological landscape would not be possible without the emergence of the CubeSat platforms (Heidt et al., 2000; Puig-Suari et al., 2001). This well-founded architecture was conceived in 1999 by professors Jordi Puig-Suari from the California Polytechnic State University, and Bob Twiggs from the Stanford University. The goal to develop a spacecraft architecture that would facilitate academic developments surprisingly triggered the creation of a new philosophy to develop satellites: the use of Commercial Off-The-Shelf (COTS) components, and the reduction of satellite dimensions. These small satellites are equipped with all the subsystems of a traditional satellite, which are composed of COTS components. These components are cost-effective commercial products, which are conceived as packaged solutions rather than custom-made. This strategy speeds up the spacecraft development, and drastically reduces the cost of its production. Furthermore, new actors developed standard CubeSat buses, as well as components, which expanded the possibilities to configure the satellite. These commercial components from distinct disciplines promoted the blooming of new capabilities in the satellites. Therefore, the CubeSats are ideal platforms to investigate and develop new technologies.

The original CubeSat proposal, which quickly became a standard, only defined the structure of 1 Unit (1U) CubeSat. This unit corresponds to a cube structure with dimensions of 10 cm x 10 cm x 10 cm, which encapsulates a maximum mass of 1.33 kg. The standard gained notoriety

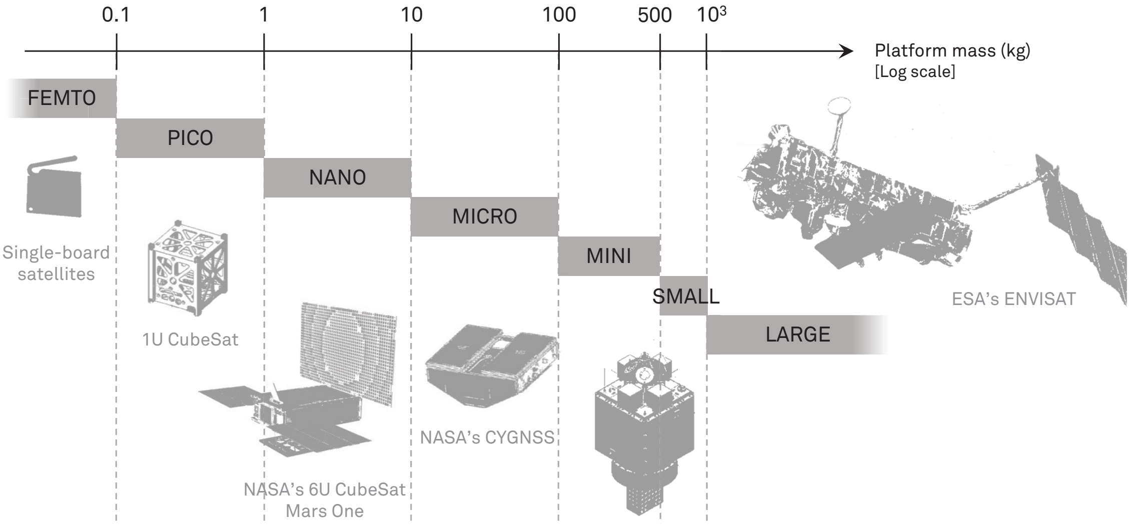


Figure 1.1: Satellite platform classification according to their mass. Figure from (Araguz, 2019).

with the development of dispensers that enabled to place in orbit multiple CubeSats in the same launch. CubeSats were also transported as secondary payloads in big satellite launches, which in conjunction to the dispensers reduced considerable the launch cost. Nevertheless, these small dimensions also constrain the capabilities or resources of the satellite, such as memory storage or available power. Therefore, combination of multiple units rapidly emerged new CubeSat formats with additional capacities. Platforms of 3U have predominated in satellite missions thanks to their well cost-performance balance, enabling to deploy commercial missions. Currently, 6U platforms are becoming the successor of the 3U formats, despite other architectures are being evaluated, such as 12U, 16U, and 27U. Figure 1.2 shows the number of small satellite launches performed in the last years, and a forecast according to each CubeSat format. These statistics highlights the growth on the CubeSat activity during the following years, thanks to the affordable cost of accessing space with this satellite platform. This peculiarity promoted the emergence of new agents who accepted the cost-benefit risk to develop novel and innovative applications with these platforms.

The inventiveness experienced with CubeSats influenced also on the traditional big-satellite activities. New private and adventurous proposals have proliferated in the last years encouraged by their commercial prospects. The most distinguished innovative application for this big platform is the deployment of massive satellite constellations with thousands of satellites to provide Internet connection to the entire globe (Lunden, 2014; Pultarova, 2015; Selding, 2015). Private companies, like OneWeb or Space Exploration Technologies Corp. (SpaceX), are competing to deploy this kind of constellation as a reference infrastructure in the space. Although the production of these satellites is not comparable to the CubeSats, the philosophy to minimize the cost and reproduce generic components has transformed the conventional process. The advent of manufacturing chains allows to make these potential systems a reality (Caleb, 2019c), and placing in orbit a set of prototypes of the final satellites (Thompson, 2020). Nevertheless, other technological and regulation challenges need to be addressed before deploying these constellations.

These new initiatives remark that satellite activity has evolved over the years from the traditional monolithic satellites developed only by space agencies. This trend is, in fact, motivated by the apparition of new end-user demands or needs to be satisfied with this type of systems. Normally, satellite systems have been developed to address a variety of purposes, such

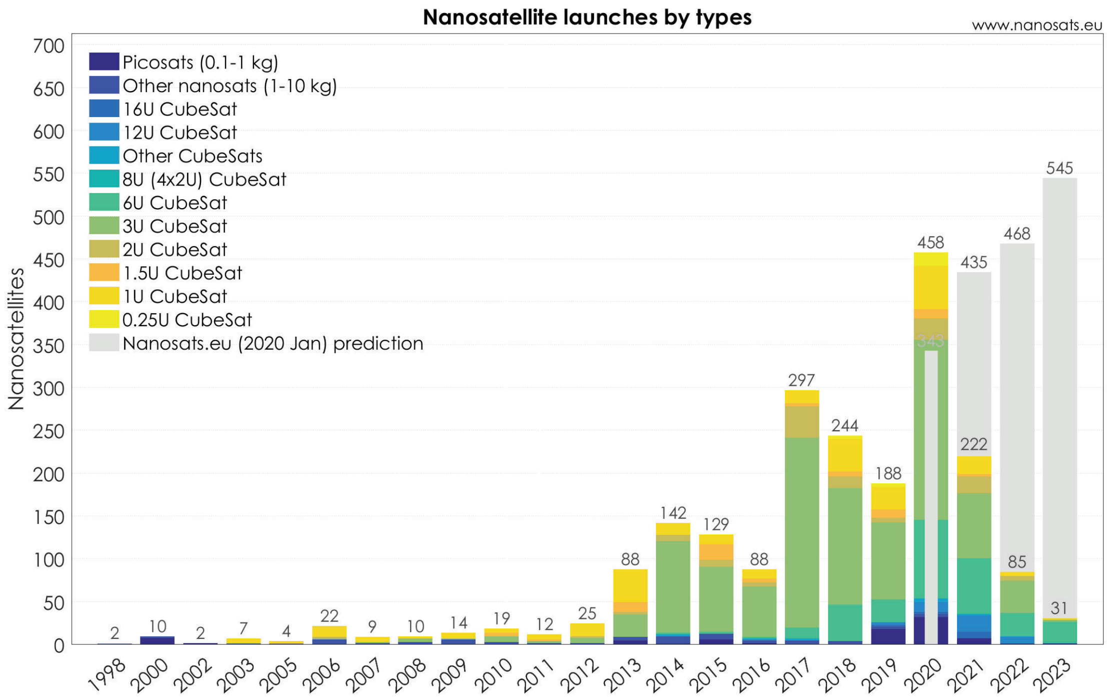


Figure 1.2: Nano-satellite launches, and launch prediction for the next three years with respect to the CubeSat type. Data and figure from the on-line database Nanosats.eu [accessed 30th June 2020].

as global navigation, space exploration, observation of the Earth, or broadband telecommunications. These last two segments are indeed the ones that experienced more satellite activity in the last years, motivated by the apparition of new environmental, socio-economic, and geopolitical demands. Figure 1.3 breaks down all the satellite launches according to the mission type during 2018. More than half of the satellites (61%) were designated for Earth Observation (EO), and broadband telecommunications services. Therefore, the end-user needs of these two predominant communities may drive the development of future satellite missions, and the corresponding technology development. Therefore, the understanding of these demands becomes crucial for future research.

1.1.1 The Observation of the Earth from the space

Most of satellite developments are conceived to sense critical environment parameters of our planet. Although some of them could certainly be observed from fixed ground stations, only satellite systems allow measuring in a global manner. This leads satellite-based EO systems to have a profound impact in a large number of strategic pursuits, such as ocean and polar monitoring, climate and atmospheric monitoring, among others. Many commercial, industrial, or governmental endeavors are also taking advantage of this environmental information to succeed with new applications. These applications and added-value services have enhanced the productivity of numerous human activities for years through the analysis of sea surface, soil, and vegetation characteristics. In fact, geospatial data acquired from spaceborne instruments is a fundamental contributor of multiple ecosystemic indicators: water stress, nutrient, and pest control (Pinter Jr et al., 2003); yield forecast (Bolton and Friedl, 2013); monitoring of plant growth and vigor (Kurosu et al., 1995); etc. Moreover, the alarming increase on weather disasters in the last years have encouraged the development of new systems to observe, monitor, and forecast these phenomena (Awange and Kiema, 2019; Voigt et al., 2016). The feasibility of these services

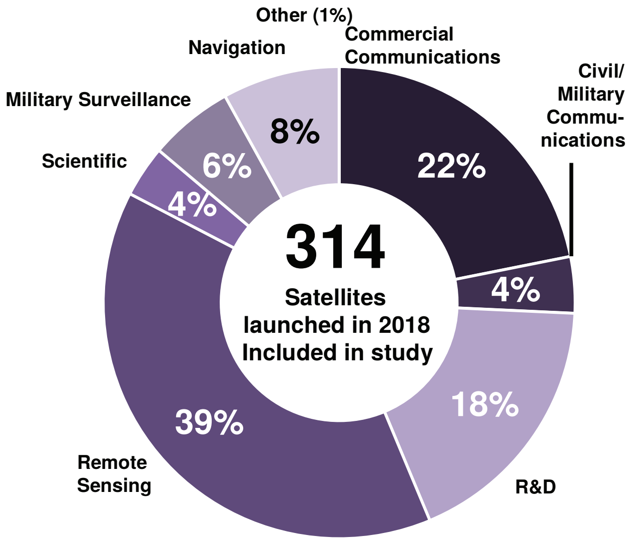


Figure 1.3: Satellites launched per mission in 2018. Source: 2019 State of the Satellite Industry report. Figure from (Space and Technology, 2019).

or the relevance of the corresponding observations are nowadays constrained by the performance limitations of monolithic satellites.

Distributed Satellite Systems (DSS) have emerged as a potential solution to provide these services (Selva, 2012; Shaw et al., 1999). This proposed architecture encompasses multiple satellites orbiting and operating simultaneously to satisfy a common goal. The generic definition leads to conceive unique architectures that can be considered instances of DSS, such fractionated satellites, satellite swarms or satellite trains. Selva et al. (2017) perform an in-depth study of the DSS taxonomy, and identify the six different architectures currently under research. Despite the DSS term became relatively well established at the end of 1990s, previous investigations conceived scenarios in which multiple satellites could observe the same specific targeted area for navigation or broadband communications purposes. This organized architecture became the so-called satellite constellation. In this way, a satellite constellation is a category of a DSS. A detailed research on the benefits of applying DSS architectures in EO missions is presented in (Jilla, 2002). The potential of the DSS resides not only in the possibility to retrieve simultaneous observations of the same target area, but also on distributing the system itself. An excellent exploration of the consequences and benefits of performing this distribution is discussed in (Araguz, 2019). As a summary, the DSS proposal leverages on the incremental satellite deployment, the combination of multiple measurements from different views, and the cost savings to achieve considerable performance. As a result, DSS may achieve wider coverage, better resolutions (temporal, spatial, spectral and angular), to fulfill the needs of a number of data products that could not be satisfied with monolithic satellites. Many programs have evaluated the feasibility of these DSS architectures with current technologies which could greatly improve EO services.

At the beginning of 2016, the Operational Network of Individual Observation Nodes (ONION) project (Alarcón et al., 2018) investigated the technical feasibility of distributed satellite architectures to the benefit of current trends in the EO community. Its main goals were to evolve and foster DSS technologies, and to evaluate and identify the unaddressed user needs. The research was centered on proposing complementary assets to the European programme Copernicus (European Union, 2014) to be applied beyond 2020. This programme encompasses six thematic areas: land monitoring, marine monitoring, atmosphere monitoring, emergency man-

Table 1.1: Ranking of the top 10 use-cases not satisfied by the existing EU Copernicus infrastructure according to the ONION project. Reproduced from (ONION, 2016).

Use-case name	Num. of users	Related need score	Related service score					Final score norm.
			FPBI* Coverage	FPBI* Accuracy†	FPBI* Frequency‡	FPBI* Access¶	Service score	
Marine weather forecast	14	0.8823	<10%	60–70%	50–60%	50–60%	1.00	1.0000
Sea ice monitoring	15	0.8749	<10%	60–70%	50–60%	50–60%	1.00	0.9916
Fishing pressure, stock assessment	12	0.6829	<10%	60–70%	50–60%	50–60%	1.00	0.7740
Land for infrastructure status assessment	17	1.0000	30–40%	10–20%	10–20%	50–60%	0.67	0.7556
Agriculture: hydric stress	24	0.9972	30–40%	10–20%	10–20%	50–60%	1.00	0.7535
Land for basic maps	18	0.9055	30–40%	10–20%	10–20%	50–60%	0.67	0.6842
Sea ice melting emissions	15	0.7135	<10%	60–70%	50–60%	50–60%	1.00	0.6739
Atmosphere for weather forecast	14	0.8823	<10%	30–40%	50–60%	30–40%	0.67	0.6667
Climate for ozone layer & UV	14	0.7058	<10%	40–50%	30–40%	50–60%	0.83	0.6666
Natural habitat monitoring, protected species monitoring	18	0.6903	<10%	40–50%	40–50%	50–60%	0.83	0.6520

* Fraction of Products that would Benefit from Improvement.
† Relates to a product's information content (i.e. resolution, radioelectric accuracy, map scale).
‡ Update frequency. Usually depends on revisit time.
¶ Delivery time, or product's latency.

agement, security, and climate change. Copernicus delivers services, and products to a myriad of end users by combining satellite and in situ local observations. ONION explored how to fulfill unsatisfied user needs of current Copernicus services by conducting an in-depth analysis of stakeholder needs (ONION, 2016). The methodology was founded on an extensive database that comprised EO users, explicit needs, and products, among other categories (Matevosyan et al., 2017). The study conducted in ONION concluded with a ranked list of ten needed services according to their priority, relevance, technical maturity, and suitability in the project context. Table 1.1 presents this resulting service list, in which marine services stand out to be the most promising ones. Three specific services lead the ranking:

- Marine Weather Forecast — The constant observation of the sea conditions would enhance current offshore activities. In particular, Oil/Gas/Mining industry, as well as fishing and aquaculture industry could be improved if a warning system can deliver in near-real-time information about marine weather.

- Sea Ice Monitoring — The ice located in the oceans and seas is constantly evolving due to the fluctuations of the weather. The periodic observation of the ice thickness and coverage would allow to monitor the sea ice status to predict weather phenomena or trace new commercial routes (Gabbatiss, 2018). This information would enable the creation of status maps, service alerts, and route optimization, among other applications.

- Marine Fishery Pressure — The monitoring of oceanographic conditions, and fishing pressure would enhance the fishing activity with the knowledge on fish shoal movements, as well as the identification of natural vulnerabilities, and anthropogenic factors. The surveillance of marine resources would additionally detect illegal and unregulated fishing activity.

These three top use-cases converge to common characteristics that have to be satisfied before their foundation. Reductions of both revisit time and product delivery delay to roughly 1 hour, and the observation of key Earth parameters related to marine nature were highlighted as critical goals that could enable polar navigation, enhance weather forecast, and bring about near-real-time monitoring. Current monolithic and single satellites cannot satisfy these temporal requirements. Instead, DSS composed of multiple observation nodes are naturally well suited to ameliorate revisit time and deliver global observations. Nevertheless, low-latency cannot be afforded by this architecture in its original approach, being a must the deployment of satellite-to-satellite interactions; i.e. Inter-Satellite Links (ISL) to relay data to the ground fast enough. Furthermore, the instruments to observe and retrieve the necessary parameters must be evaluated in detail to understand the nature of the generated data.

The OSCAR[1] is one of the most comprehensive databases on EO of instruments and satellite platforms which has catalogued up to 122 variables that are currently sensed from space. Based on this database and using a quantitative optimization method, Lancheros et al. devised in (Lancheros et al., 2018) a set of instruments that could be used to sense the Earth parameters for the previous use-cases. The spatial resolution of an instrument was one of the performance characteristics which was evaluated, and concluding that resolution from tenth of meters up to kilometers were required. Microwave instruments could be used to provide these observables, such as Global Navigation Satellite System Reflectometers (GNSS-R), radar altimeters, microwave radiometers, and others. Nevertheless, Synthetic Aperture Radar (SAR) and imagery instruments stand out to be the most suggested systems in (Lancheros et al., 2018) for this purpose. These instruments are characterized by generating large amounts of data. In this way, satellites would generate more data than they can download. Current missions that use these instruments are limited to a small set of observing areas in order to properly manage the amount of generated data to avoid overflowing the satellite memory. Therefore, current solutions may not be able to retrieve constantly information with these instruments over wide marine areas, such as the oceans. Mechanisms to optimize the amount of generated data or to improve its transfer must be conceived in order to deploy these future and demanded use-cases.

Researches centered on compressing raw data on-board of the satellite reduced the generated data with minimum information loss, such as in SAR instruments (Benz et al., 1995; Romano et al., 2020), imagery instruments (Barrios et al., 2020), among others (Huang, 2011). The Flexible Microwave Payload - version 2 (FMPL-2) system—entirely developed in the Technical University of Catalonia (UPC BarcelonaTech)—combines a microwave radiometer and a GNSS-R instruments to monitor ice cover, ice thickness, and soil moisture (Munoz-Martin et al., 2020). This instrument includes compression algorithms designed specifically for CubeSat processing capacity. Despite this data compression, measurements may be affected by unexpected conditions which degraded them to an untrustworthy quality. This useless data is also downloaded to the ground, wasting downlink capacity for better measurements. The ϕ-sat-1 experiment (Esposito et al., 2019), developed in the FSSCat mission frame (Camps et al., 2018), integrates an Artificial Intelligence (AI) processor into an hyperspectral camera for cloud detection. Images with high presence of clouds—hiding the relevant information—would be discarded with this solution. Even though this optimization, this solution does not address the problem of downloading the large volumes of useful data generated. The possibility to establish ISL for data relay to the ground would perfectly complement the limitations presented in the previous solutions. The

[1] The Observing Systems Capability Analysis and Review Tool (OSCAR) database is an expert system and database realised by the World Meteorological Organization (WMO) (Organization, 2011). OSCAR is implemented as a web-based application and is publicly available at `https://www.wmo-sat.info/oscar/`.

three development axes for managing the generated data—compression, filtering, and relay—would enable the execution of the necessary instruments to achieve the future demands.

Additionally to the required satellite-to-satellite interaction to enhance data download and latency, the benefits of this communication interface for autonomous decision-making DSS is discussed in (Araguz, 2019). This cognitive capability would provide some levels of autonomous control in order to improve their learning capabilities, to allow for self-organization, to optimize simultaneous observations, or to coordinate collective measurements to maximize performance, resiliency, and survivability. The use of ISL to enable the necessary interaction to coordinate multiple satellites for observing a common target area would also support the development of the previous use-cases.

1.1.2 The renaissance of telecommunications services over satellites

Satellites have always been bounded to broadband telecommunications thanks to their large visibility of the Earth. Since the launch of the Telstar 1 at 1962, geostationary large-satellites became the standard configuration to provide these services. Geostationary spacecraft orbit above the equator with an altitude of 35,786 km following a trajectory with the same period as the Earth rotation, which remains them as fixed points in the sky. These characteristics are ideal to provide broadband services for specific geo-political regions, such as television delivery, military applications, or generic telecommunications. The emergence of Internet encouraged multiple proposals that leverage on these fixed satellites to provide Internet access for ground users. The Spaceway proposal was conceived to deploy ten of these geostationary satellites for military purposes (Shankar, 1996). Nevertheless, new satellite configurations with low-altitude orbits started to be investigated.

Gaffney et al. analyzed the feasibility of the new non-geosynchronous or non-geostationary satellite systems proposed on the US Federal Communications Commission (FCC) by different private companies (Gaffney et al., 1996). These innovative proposals enhanced the communications end-to-end latency by means of Medium Earth Orbit (MEO), and Low Earth Orbit (LEO) satellites. Satellites orbiting at LEO region are located in altitudes bellow 2,000 km, while the MEO region—also known as Intermediate Circular Orbit (ICO)—is delimited by the LEO region and geosynchronous orbits. Those orbit regions were not originally proposed because of the continuous relative movement—unlike the geosynchronous case—that satellite experience. Satellites in this configuration suffer from a small field of view, which leads to service disruptions. Therefore, satellite constellations have risen as a promising architecture to deal with this challenge, by performing user handovers between adjacent satellites. An in-depth study of the benefits of applying these constellations was presented in the dissertation (Kashitani, 2002). Kashitani remarks that satellite constellations at LEO region are able to serve larger number of customers with a relative reduced deployment cost as compared to their MEO homonyms. These performance expectations encouraged the development of numerous LEO satellite constellations, as Iridium and Globalstar (Pratt et al., 1999; Wiedeman and Viterbi, 1993).

The Iridium constellation extended the original concept by interconnecting satellites with radio interfaces to relay data down to the ground; i.e. the development of ISL. This architecture revolutionized the perception of these constellations which started to be conceived as LEO satellite networks. Werner discusses the challenges on deploying these networks, and concludes that satellite mobility is a key factor on the stability of the links that compose the network (Werner, 1997). Satellite interconnection through ISL has the drawback of service disruptions due to the relative motion between satellites. Ekici et al. (2000) started to work with a satellite constellation configuration that mitigates this mobility impact on the communications performance. Therefore, a well-organized architecture with specific planes, and satellites locations was defined to transform a LEO satellite network into a mesh network (Ekici et al., 2000). Services that required low-latency communications were provided over this infrastructure, such as phone calls. Furthermore, the stability of the network was enhanced on the Earth regions with larger population. Akyildiz et al. also extended this concept by conceiving a satellite network composed of satellites at different altitudes. This multi-layered satellite network (Akyildiz et al., 2002) was substantiated on an organized architecture in which a LEO satellite network could benefit of MEO and GEO satellites to relay data. Despite this theoretical architecture never was deployed in space, it influenced future proposals as discussed in the following paragraphs.

During the last years, research has been focused on the optimization and adjustment of the Internet mechanisms, techniques, and protocols to enhance the throughput over satellites. Henderson studied the challenges to use the well-standardized Transport Control Protocol (TCP) for satellite communications in his dissertation (Henderson, 1999), concluding that minor changes on the original standard could drastically improve the communications performance. With the advent of the different versions of the Digital Video Broadcast - Satellite (DVB-S) protocols (Mignone et al., 2011), the television broadcast over satellites was also investigated as part of the digitalization of this service. The outcome of all these efforts was the development of the high throughput satellites, so-called next-generation satellites (Vidal et al., 2012). Research remained on adjusting the Internet protocols to the satellite constraints, meanwhile the Internet itself was evolving with new services, such as cellphone integration or cloud computing. The advent of CubeSat platforms became the necessary catalyst to transport the new Internet technologies in satellites. The growth of cellular activity influenced considerably the role of the satellites in the current epoch.

The fifth generation technology standard for cellular networks (5G) has been already established on the ground infrastructure as a fast, reliable, and high-connectivity communications interface for cellphones and other devices. The 3rd Generation Partnership Project (3GPP) developed the multiple specifications that conform this standard over the years (3GPP, 2019a) to satisfy the requirements of future use-cases. These requirements were presented by the International Telecommunication Union Radiocommunication (ITU-R) sector in the International Mobile Telecommunications (IMT) for 2020 and beyond (Sector, 2017a). The standard presents three use-cases according to the current and future telecommunications activity: the enhanced Mobile Broadband (eMBB), the massive Machine Type Communications (mMTC), and the Ultra Reliable Low Latency Communications (URLLC). The eMBB scenario is an evolution of the mobile broadband applications developed in the previous generation standard (4G) by improving the data transfer performance, and increasing the seamless experience. Both URLLC and mMTC are new use-cases which were defined due to the emergence of the Internet of Things (IoT) and the apparition of Critical Communications (CC). The IoT paradigm (Atzori et al., 2010) promotes the interconnection of multiple devices which can exchange data without requiring human interaction. In this way, the mMTC represents this trend which is characterized by a myriad of connected devices that typically transmit a relatively low volume of delay-tolerant data. The URLLC scenario is centered on safety and critical applications that require real-time and reliable communications, such as control of industrial manufacturing, remote medical surgery, autonomous driving, and other emergency applications.

The 5G specifications were developed to satisfy the requirements published by the ITU-R in (Sector, 2017b), which are summarized in Table 1.2. This development has been conducted to achieve three main goals:

Table 1.2: Requirements for each use-case presented in the IMT-2020. Adapted from (Sector, 2017b)

Capability	Requirement	Applicable use-case
Maximum data rate for the data downlink (i.e. downlink peak data rate)	20 Gbps	eMBB
Maximum data rate for the data uplink (i.e. uplink peak data rate)	10 Gbps	eMBB
Average data rate for the data downlink (i.e. downlink data rate)	100 Mbps	eMBB
Average data rate for the data uplink (i.e. uplink data rate)	50 Mbps	eMBB
Network contribution to packet travel time (i.e. end-to-end latency)	4 ms	eMBB
	1 ms	URLLC
Maximum node speed for handoff (i.e. mobility)	500 km/h	eMBB | URLLC
Total number of devices per unit area (i.e. node density)	10^6 devices/km^2	mMTC
Total traffic across coverage area (i.e. area traffic capacity)	10 Mbps/m^2	eMBB
Throughput per unit wireless bandwidth (i.e. spectrum efficiency)	30 bits/Hz	eMBB

- Provide high data speeds — efforts were focused to conceive new transmission techniques that satisfy the necessity of high data rates of the eMBB use-case. Therefore, enhanced downlink and uplink communications would provide data rates up to 20 Gbps and 10 Gbps respectively, ensuring hundreds of Mbps in average.

- Reduce the end-to-end latency — the URLLC applications require the data deliver process to be more instantaneous, because critical services cannot suffer from delay. Therefore, the enhancements would allow to reach real-time access with end-to-end latency less than 1 ms.

- Ensure seamless and global connectivity — the emergence of autonomous and mobile devices encourage the development of an infrastructure that enhance the continuous connection with the network. Services deployed over this network would not suffer any disruption which could compromise the performance in mMTC, eMBB, and URLLC scenarios.

Is in this last goal in which satellites have stood out as a promising platform to be integrated in the 5G infrastructure. In March 2017, 3GPP started new activities to study the role of the satellites in the 5G (3GPP, 2019b). The outcome of these studies was the definition of the Non-Terrestrial Networks (NTN) which encompasses the multiple systems not located on the ground, such as satellites, Unmanned Aerial Vehicles (UAV) or High Altitude Platforms (HAP). This network leverages from this high altitude architecture which awards the satellites with unique qualities for the 5G (3GPP, 2019c). In this way, the NTN are conceived following the multi-layered satellite network premise, but including in the architecture other systems than only satellites. The large coverage area of spaceborne telecommunications systems enhances the service continuity in case that not being ensured by ground infrastructure. Furthermore, satellite coverage enhances the network capacity by serving a myriad of end-users with a single spot. Finally, the orbit trajectory of a satellite allows to reach the service ubiquity on the entire globe, being able to provide services in remote and typically inaccessible areas.

These qualities have led to the definition of multiple satellite applications in the eMBB, mMTC, and URLLC scenarios (Kodheli et al., 2020). The eMBB scenario would leverage on satellite systems working as a complementary traffic backhauling nodes of the network, or by reducing the handovers of those mobile nodes that perform large trajectories, such as trains or airplanes (Liolis et al., 2019). Satellites could enhance the services in the mMTC scenario depending on the area in which the devices are deployed. For wide area services, the satellites play the important role of large visibility to become as a central node that feeds on device traffic. Otherwise in local area services, the satellites become a complementary infrastructure to backhaul the traffic of massive number of devices, like the eMBB case. Unlike the other cases, satellite altitude prevents from achieving the required end-to-end latency for URLLC cases. Nevertheless, the satellites enhance these services by providing a supporting role that broadcasts information over a wide area.

Satellites remain as a support node to access Internet in all these use-cases. As in the LEO satellite constellation case in the 1990s, these satellites must have a satellite-to-satellite interface over which relay data from ground devices. The relative low-latency could not be achieved without this interface, because the satellite only would download data over ground stations. This situation becomes relevant when the satellite serves devices located in remote areas distant from the ground infrastructure, and the satellite cannot relay directly to the ground. As in the EO community, the current demands of the new telecommunications applications are subject to the development of ISL technologies.

1.1.3 Distributed Satellite Systems: concept and taxonomy

DSS are potential architectures to address current and future user demands, which are unsatisfied due to the limitations of monolithic satellites. The distribution of the system in multiple satellites naturally allows the observation of distinct target areas simultaneously, and provides other substantial system-level benefits, such as scalability, flexibility, and reusability. This powerful concept encouraged the emergence of distinct DSS instances or proposals over the years. Several characteristics can classify DSS into different categories. The most frequent aspects that drive this categorization are the distribution of satellites, the dynamic nature of system structure (i.e. static, dynamic, opportunistic), and structural functions (e.g. availability of inter-satellite links, exchange of resources, etc.) Table 1.3 presents the founded taxonomy of DSS with the architecturally identifying characteristics of each category. Six distinct categories encompass this classification:

(i) Constellations — satellite constellations could be generically defined as groups of satellites orbiting independently to achieve continuous over numerous geographical areas. This well-founded DSS has largely deployed over the years, being the ones for broadband telecommunications the most predominant. Nevertheless, constellations started to be a serious solution for EO applications, such as Planet Labs (Marshall and Boshuizen, 2013). These constellations are characterized by not leveraging on satellite-to-satellite communications, unlike the telecommunications ones.

(ii) Trains — an ensemble of coordinated and heterogeneous satellites which closely follow the same orbit track is defined as a satellite train. This configuration enhances the observation of the same ground track by distinct satellites. Therefore, the satellites that conform the train are normally equipped with different EO instruments, which provides more information of the same region. The NASA Afternoon-train (A-train) mission is the most representative mission with seven satellites that conform the train (L'Ecuyer and Jiang, 2010).

(iii) Clusters — satellites orbiting close to each other in a strict flight formation are considered a satellite cluster. Unlike the trains, satellites in a cluster maintain the orbit formation by means of exchanging control messages between them. This formation enables the critical measurement for different applications, such as astronomy or data reconstruction.

Table 1.3: Summary of characteristics for each DSS mission topology. Reproduced from (Araguz, 2019)

Architecture type	**Mission goals**	**Geometry configuration**	**Satellite distances**	**Maturity of the concept**	**Year**
Constellation	Single, shared among elements.	Structured: walker, rosetta, flower...	$\sim 10^3$ km.	Many deployed constellations (e.g. GPS, Iridium)	60-70s
Train*	Coordinated, shared among elements.	Structured: same ground track.	$\sim 10^2$ km.	Operative mission: A-train.	2002
Cluster	Some mission goals are shared.	Tightly-controlled close flight.	~1 m. to ~1 km.	Demonstrator missions: TerraSAR-X / TanDEM-X, FASTRAC.	Late 90s
Swarm	Some mission goals are shared.	Loose, close flight.	~1 m. to ~1 km.	Demonstrator missions: ESA's SWARM	00s
Fractionated Spacecraft	Multiple mission goals.	Close flight.	~1 m. to ~100 m.	Pending demonstrator mission.	1984 & 2006
Federated Satellite System*	Unrelated mission goals	Opportunistic, unstructured.	$\sim 10^3$ km.	Proof of concept: FSSCat (Camps et al., 2018), launch expected in 2020.	2013

* Three fundamental differences between a satellite train and a FSS are: (1) the structured nature of the constellation (in trains), (2) the wider notion of opportunistic cooperation, mission synergies (in FSS), and (3) the likely need of inter-satellite communication (in FSS).

(iv) Swarms — similar to the previous class, a satellite swarm is composed of numerous of small satellites orbiting close to a bigger satellite. These necessarily in flight formation, these complementary small satellites are equipped with different sensors or instruments, and use the central satellite to download the generated data. The ESA swarm mission is currently working with three satellites following this configuration (Macmillan and Olsen, 2013).

(v) Fractionated satellites — a different approach is proposed in fractionated satellites in which individual satellites are built from physically-detached modules which orbit in close formation. Each module is centered on a satellite functionality, such as power supply, downlink capacity, etc. A technological demonstrator was conducted in 2006 with the Future, Fast, Flexible, Fractionated, Free-Flying Spacecraft (F6) mission (Brown and Eremenko, 2006).

(vi) Federated satellites — satellites that opportunistically share in-orbit unallocated spacecraft resources are federated satellites. These satellites are normally conceived for different missions with distinct capabilities, and at certain moment decides to create this collaboration. The FSSCat mission would become the first in-orbit proof-of-concept of federated satellites that will be launch in 2020 (Camps et al., 2018).

Among the different DSS categories, satellite federations provide the necessary characteristics to fully exploit the benefits of distributed satellite architectures. The opportunistic and adaptive nature of this approach is suitable to satisfy the new end-user demands by optimizing future satellite missions. Therefore, this federation concept is the main object of study in this dissertation. Despite this section introduces in details the research on this type of satellite system, we encourage to review (Lluch and Golkar, 2015b) for further details of the other DSS types.

1.1.4 Federated Satellite Systems

The Federated Satellite Systems (FSS) term was introduced in 2013 by Prof. Golkar during the celebration of the 9th *Symposium on Small Satellites for Earth Observation* (Berlin, Germany). However, it was not until two years later that its definition was formally detailed in (Golkar and Lluch, 2015). FSS essentially consist of spacecraft networks in which satellites trade unused or inefficiently allocated resources commodities, such as data storage, data processing, downlink capacity, power supply, or instrument time. This concept is analogous to terrestrial applications, such as peer-to-peer file sharing, cloud computing, and electrical power grids. In this way, FSS tried to avoid the underutilization of expensive space assets in already existing missions. The same optimization goal is the focus of the Heterogeneous Spacecraft Networks (Faber et al., 2014). Both approaches argued on the necessity to establish cooperation frames beyond the common mission interactions based on ground post-processing and merging of instrument data. Distinct-stakeholder satellite missions would leverage on the establishment of in-orbit collaborations by improving current system performance or by achieving new goals.

The terminology presented in (Golkar and Lluch, 2015) allows to understand the nature of these collaborations. A *satellite federation* is composed of a group of satellites which decide to engage in a collaboration with each other during their mission. These federations allow the satellites to share or trade available *resources* which are the tangible and intangible assets that a spacecraft has (e.g. propellant, power, data processing, downlink capacity, etc.) Although the original definition encompasses generic assets, the later work investigated federations with data-centered resources, such as processing, downlink, and storage capacity. The establishment of a satellite federation has a decision-making component—not necessarily autonomous—with which a satellite *opportunistically* deems if the collaboration is beneficial, where benefit is defined as either economic profit, or generic value. Despite the opportunity refers to the federation profit, a temporal aspect also is integrated in these characteristics. Due to the mission lifecycle of a satellite, resources are not constantly available, enabling temporal windows of opportunity in which they can be traded. During this transaction, the satellite that supplies the corresponding resources are defined as satellite *suppliers* or *providers*. When instead a satellite is seeking to request the resources, it plays the role of a *customer* in the federation. These roles may be switched over satellite lifetime depending on their interest, being also possible that a satellite acts as both customer and supplier at the same time. At the end, the joint set of customers and suppliers makes a satellite federation.

Federations enable two high-level functions for interoperability among the satellites: the resource transaction, and the resource negotiation. The transaction refers to the interaction between the satellites to exchange the corresponding resources. The negotiation refers to the ability of satellites to receive customer requests, and allocate them efficiently according to the demand and resource availability over time. While transaction is effectively performed in-space, negotiation could even be implemented on the ground via diverse auction mechanisms and scheduling. Considering these interactions, three families of federated spacecraft configurations are defined (Figure 1.4): (1) a centralized federation is composed of spacecraft acting as central negotiators, which regulates and negotiates all the future transactions; (2) an alternative is proposed in the negotiated federations, in which the negotiators only fulfill the negotiating role, while the transactions are directly conducted satellite to satellite; (3) distributed federations are composed of satellites which negotiates and transacts independently, as a peer-to-peer network. The distributed scheme corresponds to the full integration of the FSS concept in the spacecraft design, which represents the final goal for satellite federations.

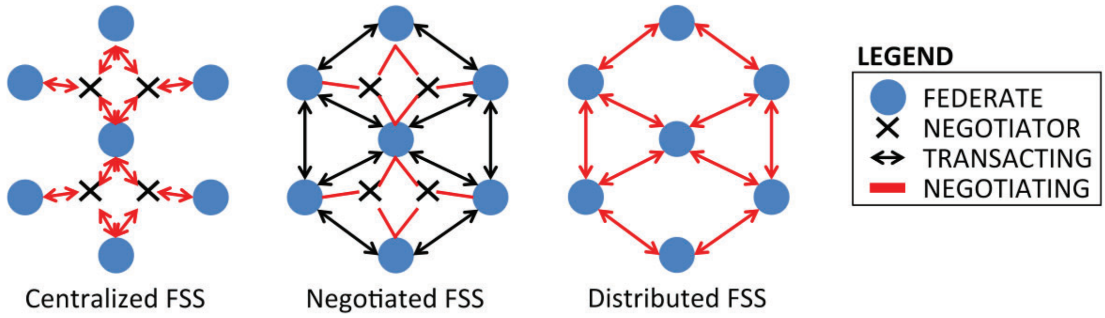


Figure 1.4: Representation of the federation configurations that satellite interactions determine. Figure from (Golkar and Lluch, 2015).

The FSS represented a breakthrough with respect to the multiple DSS proposals that posed numerous questions in the research community and experts about its technological feasibility, and mainly on the benefits of these collaborations. Golkar explored the commercial possibilities and identified the areas of opportunity associated with federated systems in (Golkar, 2013a). The study was conducted by means of the Stakeholders Value Network (SVN), an analytic tool previously detailed in (Cameron et al., 2008) for the US Vision for Space Exploration programme. The benefits and costs expectations are evaluated with a custom SVN centered on the FSS context. The presented benefits for the stakeholders were identified as the possibility to provide on-demand and scalable services, create secondary streams of revenue for their mission, repurpose and leverage resources of partially failed spacecraft, to use small satellites in new contexts, and improve the overall reliability of the mission. These benefits were compared with technical costs (e.g. complexity on the satellite architecture with ISL, impact on the original mission success, etc.); and stakeholders and policy issues (e.g. acceptance of federations by the stakeholders, malicious users that drive to liability issues, etc.) The analysis evaluated different macro-stakeholders that could be interested on deploying federations in their future missions, and categorized them as focal organization, governments, and private companies. The study concludes with three main statements on the potential and realistic benefits of using federations: (1) commercial space missions seem to feature stronger synergies with FSS developments than what exhibited by government missions; (2) missions playing the dual role of suppliers and customers seem to leverage more of the FSS infrastructure; and (3) added-value data products from data fusion of different space sensors are valuable for several markets including the science community and government space agencies. The study is extended in (Golkar, 2013c) by evaluating the possible threads that the FSS concept could have. No strongly dominant threat was identified in the analysis; this latter result represents a double-edge sword. While the absence of a dominant threat is reassuring for a successful development of the FSS infrastructure, it creates uncertainty in decision-makers as to which threats mitigate first in a scenario of limited development resources. It is concluded that FSS are powerful and potential architectures for future missions.

A study of the potential market, and possible satellite missions that would leverage on federations were presented in (Golkar and Lluch, 2015). The methodology was founded on an extensive database that comprises the Two-Line Elements (TLE) of operational satellites. The study outcome remarked that LEO satellites are predominant in multiple orbit inclinations, being near-polar orbits—roughly 90 º of inclination—predominant. These characteristics are driven by the numerous satellite missions oriented on observing the Earth, which normally use Sun Synchronous Orbits (SSO). Satellites that follow this kind of orbit pass over any given area of the planet surface at the same local mean solar time. Golkar and Lluch considered that EO satellite market is potential for the FSS approach. A case study of FSS in support of the EO missions is also presented in the same work. The study highlights that the FSS approach suffers from different business challenges based on stakeholder acceptance, the achievement of a critical mass of participants in a progressive deployment, and legal implications of sharing systems among several nations and organizations. The study results show the versatility of FSS which could potentially improve satellite missions for different purposes. Nevertheless, EO satellite missions seems to be the ones that would leverage more on deploying satellite federations.

Seeking the most advantageous conditions to engage in this space-based market requires to understand the complex interplay balance between the capabilities enhancement, and the system-level impacts in spacecraft design. Lluch addressed this quandary in his dissertation by studying the feasibility to deploy satellite federations with current satellite technology (Lluch, 2017). An analytical framework was defined, that was later extended in (Lluch and Golkar, 2019), to quantify at early stages of the mission the performance enhancement with respect to the challenges of introducing complexity on default satellite design (e.g. use of satellite-to-satellite interfaces, on-board decision-making capability, etc.) Three distinct steps are required to follow the proposed framework: (1) a preparatory step that gathers the information about the systems to be part of the federation; (2) an exploration step where different performance metrics are explored (e.g. downlink data rate, communications latency, etc.); (3) an analysis step were the results of the exploration are consolidated and compared. This framework also recognizes each local system architect and local stakeholders as the decision-making actors, and that the case for federating shall arise from local cost advantages for all systems involved rather than global utility figures. The framework is founded on a satellite model which includes multiple subsystems, and allows to derive their impacts up to spacecraft mass and power. These two system-level metrics are relevant to determine the federation influence on mission cost, available resources, and original concept of operations.

The numerous EO use cases presented as part of the framework evaluation (Lluch and Golkar, 2014a,b, 2015a) generated interesting outcomes on the implications of applying federations according to each satellite role. The results suggest that satellites working as suppliers may experience the following situations: (1) a growth on the power consumption per federation established, which provoke the increase of spacecraft mass to satisfy power requirements (e.g. solar panel array, larger battery pack, etc.); (2) enhancement of satellite communications systems with larger data rates is mass-beneficial than transmission times; (3) LEO satellites experience a nearly linear increase on loaded mass per established federations rather than MEO and GEO satellites; (4) LEO satellites require an important mass growth and ISL capabilities to communicate with GEO satellites (e.g. large antennas, large power transmission, etc.). This kind of communications should only occur between designated spacecraft or at a low data rate; (6) small-satellite mass (100 kg) experience larger increases due to federations rather than medium-, and large-satellites (900 kg and 2000 kg respectively); (7) satellites orbiting at near-polar orbits are suitable to work as suppliers to achieve maximum satellite system performance. This situation is ideal for EO satellite missions, as stated previously. On the other hand, satellite customers seem to experience certain mass-benefits when applying federations: (1) the establishment of numerous federations enables to download a larger amount of data, reaching an ideal case in which customers may not require their own downlink equipment. In this case, they would experience a considerable reduction of mass; (2) as in the supplier case, the mass reduction is more significant—at percentage level—in small-satellites rather than medium-, and larger-satellites; (3) despite this reduction, satellite mass keeps increasing per federation established with a smoother gradient than in the supplier case. In summary, a satellite supplier experiences a considerable increase of resources while satellite customers can reduce their complexity thanks to federations. **In this sense, CubeSat platforms are suitable to work as customers, and large-satellites as suppliers**. A well-balanced situation seems to be a hybrid scenario in which satellites work as customers and suppliers with common capabilities. Furthermore, results suggest that LEO satellite interactions are currently more affordable than multi-layered ones.

The studies conducted by Golkar and Lluch stood out the necessity to develop new technologies that address specific technical challenges: the development of extensive tools to properly architect FSS by assess alternative designs, the management or mitigation of fast communication handovers between spacecraft in heterogeneous orbits, the mechanism to protect resources with an in-orbit transaction process, and the application of cybersecurity techniques to ensure data confidentiality. Subsequent research activities followed these paths.

Enhancements on techniques to design federated architectures were achieved in the different works conducted by Grogan. Following the game theory, a new approach in which the FSS paradigm was considered a game table board where different stakeholders compete or "play" to deploy federations and garner benefits was presented in (Grogan et al., 2014a). This approach not only enables to estimate the system benefits of applying federations, but also to understand the social interactions between stakeholders. From the experience achieved, a formal mathematical model to assess FSS design alternatives was defined (Grogan et al., 2016b). Following the System Value Model (SVM) definition, a specific SVM model was determined for federated architectures that is able to mathematically represent the preferences of the stakeholders, and determine the system benefits. The model encompasses the uncertainties due to probabilistic collaboration intention of stakeholders, and generates a risk method applied in the resulting system value. This model enabled the possibility to bound the value of stakeholder collaboration delimited by an independent satellites strategy (i.e. each satellite executing their own mission independently), and a centralized federation strategy (i.e. a central entity manages the collaboration between stakeholders). Results suggest that these boundary values are indeed accurate when a fixed-cost communications service is defined, being applied to different scenarios. The proposed work opened the possibility to keep enhancing the model with multiple stakeholders—currently two are applied—, and situations with market saturation and competition as strong negative interactions between players.

Is in this last situation in which a satellite resource may become a precious commodity that its value or price may fluctuate according to the stakeholder demand. A mechanism based on pricing and auctions was proposed to manage this situation in (Pica and Golkar, 2017). A sealed-bid reverse auction is implemented between the satellites to determine the pricing of the available resources in a satellite federation. Satellites participating in this auction periodically exchange information to each other about the auction status, and the available resources that multiple suppliers are providing. In this context, the suppliers are the bidders, which compete with each other to sell their resources to satellite customers. The final price of the service depends on the provider-issued bids, which is determined at the moment of awarding the commodity to the customer. This one tries to find the best supplier among the ensemble. In the end, this approach enables to manage the existence of multiple instances of the same service, and to optimize the benefits of the provider to offer the service. Further investigation was conducted to determine a pricing mechanism that fairly represents the value of the satellite resource over time. An operational model was developed in (Ehsanfar and Grogan, 2020b) to allocate processing, storage and communication resources to computational demands. An optimal solution to process satellite tasks, allocate links, store and delivery data is determined by means of a mixed-integer linear programming (MILP) formulation. A trusted auctioneer leverage on a mechanism to allocate the resources, and suggest prices for exchanging resources across federations. This centralized approach can compensate the adverse effects of strategic bids on collective value, and increases the number of resource exchanges in a federation. A complete review of the numerous auctions-based algorithms for satellite operations in federated scenarios was presented in (Ehsanfar and Grogan, 2020a).

The lack of trust between stakeholders would prevent the deployment of satellite federations owned and operated by multiple parties. Therefore, satellite transactions which exchange sensitive information would require certain level of data protection, and user authentication. A study of the security threads in FSS was conducted in (Sanchez Net et al., 2016), identifying possible attacks that may occur in this context: (1) the impersonation of a satellite participating in a federation after the theft of its credentials by means of eavesdropping attacks; (2) misappropriation of exchanged proprietary data by another participant satellites; (3) corruption of the data during its propagation to the destination; (4) selective dropping of forwarded data by intermediate malicious satellites; (5) duplication of forwarded data by intermediate satellites which may lately use it for their own benefits. Security mechanisms need to be integrated in satellite federations to prevent these attacks. A similar review (Maurich and Golkar, 2018) concluded with seven security requirements that must be satisfied in federations: (1) the verification of whether received data is authentic; (2) the restriction to the data by means of data encryption, and authorization; (3) stakeholder authentication; (4) the audition that data has been correctly delivered; (5) the no modification of federation conditions by intermediate satellites; (6) reading protection against non-participant satellites; (7) the guarantee that the stakeholder provider of the resources will be properly remunerated. Each investigation addressed the same security challenges with distinct approaches.

Sanchez Net et al. discussed the possibility to integrate Information Security (InfoSec) services by analyzing the security-risk conditions—in form of security states—that may materialize in federations. These services are broadly defined as the practice of protecting information by means of security mechanisms to reduce the probability of unauthorized access to data, the unlawful use, deletion, corruption, modification, inspection, or recording of information. The analysis of point-to-point security risks allowed the identification of four security states determined by the nodes and channel reliability. The security mechanisms of hashing, signing, certifying, encrypting, and encoding data are evaluated for the negotiation and transaction of the federation. The use of each specific mechanisms depends on the security state that the satellite remains, enabling the satellite itself to determine the state for each federation. As an example, a satellite that is using an optical communications interface may decide to use hashing, signing, certifying, and encrypting mechanisms to transact data in a distributed federation, because it trust the channel and not the pairing satellite. Otherwise, no security mechanism would be required in a centralized federation with the same interface, because channel and pairing satellites are trusted. The possibility to jump over the different security states depends on the conducted actions, and type of federation established. This approach allows to independently decide which security functionality to implement, and enforce to provide the specified set of security mechanisms between the satellites. The study concluded that, as expected, centralized and negotiated federations require less security mechanisms, rather than distributed ones. Furthermore, it paved the development of the required mechanisms in future investigations.

The development of a public key infrastructure (PKI) protocol was also proposed for distributed federations (Maurich and Golkar, 2018). The proposed architecture is based on the fact that each satellite owns a pair of public and private keys which allow the satellite to authenticate itself, and encrypt data blocks before the transaction. These keys would be generated in advance by a central certificate authority located on ground, once the satellite is registered in this infrastructure. Satellites interested on participating in federations have to exchange their certificates to authenticate each other. After this authentication process, the satellites trade data blocks previously encrypted with their private key. This authentication and encryption mechanism are normally deployed in current computers, with distinct capacities than constrained satellites. Therefore, these mechanisms were executed in a resource-constrained platform to analyze their viability in future satellite platforms. Despite the overhead generated with the authentication and encryption mechanisms is acceptable, the computation overhead is intrinsically bounded to the amount of transacted data. Satellites with high-capacity ISL would require powerful platforms to accommodate the encryption process, rather than less-capacity platforms. These results suggest that future investigations may address the challenge to provide the necessary security services or functionalities considering the resource constraint of satellites.

Among the different characteristics of FSS, the heterogeneity of the federated satellites poses technical challenges regarding how to interface distinct ISL technologies. Techniques to confer flexibility and re-configurability on communications systems were discussed in (Akhtyamov et al., 2016). The study was centered on the challenges in negotiated federations (Figure 1.4) where specific nodes work as negotiators and must interact with numerous satellites. These nodes require certain level of flexibility on the communications system to interface with other satellites that are not equipped with the same technology. Software Defined Radio (SDR) devices are suitable for this purpose in case that Radio Frequency ISL (RF-ISL) are used. A stratospheric balloon campaign was conducted in the same work to evaluate the performance of a Nuand's BladeRF. The results suggest that this kind of platform may be suitable for large ranges, and its capabilities to modify its RF characteristics can enhance the deployment of federations. Despite this outcome encourages to keep investigating on the application of SDR devices in FSS, the proposed solution based on a preliminary technology demonstrator that may be extended with further development on communications protocols.

Communication among satellites is a fundamental characteristic of the FSS concept. Satellite federations mostly rely upon inter-satellite networking and coordination mechanisms to create virtual, opportunistic infrastructures with emergent functionalities. Despite the research performed on the federation feasibility and its trading process, further investigations may address the development of communications technologies to deploy satellite networks that allow the establishment of satellite federations.

1.2 Motivation: satellite networks in the FSS paradigm

So far the current trends in EO and broadband telecommunications fields have been presented. During the last years, the users in these communities have experienced a change demanding new satellite services that could provide near-real-time information, and high-capacity to download data. Current monolithic satellites cannot satisfy these new user demands, which require new approaches. DSS are a new concept of satellite system which encompass a group of satellites orbiting and operating simultaneously to a common goal. How DSS and their potential characteristics can be a cornerstone to address these ever-increasing user demands have been summarized. A generic understanding of distributed satellite architectures has been introduced revealing some of their operational traits, and its founded taxonomy. Many of the DSS approaches present structural functions that could allow them to implement complex communication networks, maintain coordinated flight formation, or distribute computational load among spacecraft.

The FSS approach has become a potential distributed architecture to fully exploit the potential of DSS (Selva et al., 2017). This concept defines a satellite federation as a sporadic and opportunistic collaboration between satellites to leverage on inefficiently allocated or unused resources. Studies to quantify the potential and the benefits of applying this concept in current satellite missions have also been previously presented. The outcome of these studies (Golkar, 2013a; Lluch, 2017) motivates the deployment of federations to satisfy these new demands, and materialize new commercial stakeholder perspectives. Despite these expectations are applicable in numerous satellite activities, the potential of FSS is mainly remarkable in the EO market with new capabilities based on heterogeneous satellites, such as the fusion of data from different instruments.

This future perspective encouraged multiple lines of research to enhance the integration of federations in satellite missions. We have presented the developments conducted so far, and mainly centered at different aspects of system definition: (1) a framework to quantify—in early stages of mission definition—the benefits of applying federations in return for increasing the complexity of satellites (Lluch and Golkar, 2019); (2) a mechanism to design satellite missions which includes federations taking in consideration the uncertainty of stakeholder intentions (Grogan et al., 2016b); (3) an auction-based mechanism to in-orbit determine the fair value of the available resources according to the satellite concurrence and demands (Pica and Golkar, 2017); and (4) the identification of the data security requirements in federations, and a preliminary approach to satisfy them (Maurich and Golkar, 2018). Despite the interesting technology maturity achieved, all these developments assumed that the necessary communication means to deploy federations were available, functional, and suitable for federation nature (e.g. ISL technologies, protocols to deploy and manage networks, protocols to negotiate and transact resources, and protocols to notify available services). However, this assumption needs still to be discussed in depth.

A summary of the satellite networks proposed in the last years have been briefly presented with the review of the broadband telecommunications satellite services. Although further information is presented in Section 1.3, the main connected architectures were introduced: the LEO satellite network, and the multi-layered satellite network. These satellite systems found on the custom architecture design—number of satellites, altitude orbit, constellation planes, etc.—to mitigate the influence of satellite motion on the communications performance. The technology developments for these architectures were under the premise of this custom satellite constellation design. Despite some studies (Lluch and Golkar, 2014b) have identified potential orbits for each federation role, homogeneous orbits and specific pre-defined satellite architectures are not included as part of the federation definition. Therefore, the application of previous solutions in FSS seems per se not suitable to coexist with its heterogeneous nature.

When this custom satellite system architecture is not included in the design process, the satellite mobility drives the capability to communicate. Satellite-to-satellite interfaces are characterized by being temporal links, which can be concatenated to generate per se a temporal route between two remote satellites. Due to this temporality, the routes and links may be destroyed over time, disrupting a satellite network into multiple connected fragments. As discussed in (Maurich and Golkar, 2018) the imperative necessity to address this situation for the sake of the established federations. This phenomenon—characteristic of heterogeneous satellite networks—were investigated with the concept of Delay and Disruption Tolerant Networks (DTN). The solutions developed in this research frame anticipate or coexist with this network disruption by deploying delay-tolerant applications (e.g. non-time-sensitive data, messaging, etc.) Despite these mechanisms may be suitable for certain types of federations, others that are more time-constraint require different solutions. A hybrid solution which is able to manage both situations seem to be a well-balanced approach between satellite network limitations and federation constraints. This strategy was followed in in (Lluch et al., 2015) by complementing a routing protocol from the Mobile Ad-hoc Networks (MANET) theory with one from the DTN research to deploy federations. This combination could deploy a network that constantly monitors all the possible routes between each satellite pair, while the data delivery is protected against the network disruption. Despite this approach seems well-oriented, the proposed routing protocol was not conceived to respect the sporadic and opportunistic nature of federations. Specifically, the proposed protocol provokes the periodic transmission of control messages to be part of the satellite network without considering the intention of each satellite. In this regard, the decision-making of a satellite to participate in a federation would be missed due to the mandatory interaction to compose the network. This situation provokes the waste of resources for those satellites that do not expect to participate in any federation—neither as intermediate nodes—and compromise their original mission.

The previous research motivates the need to conceive a communications environment over which federations can properly be deployed. Therefore, the different protocols that enable this communications capacity must respect the opportunistic, sporadic, and satellite decision-making nature of federations. A similar observation was also made in (Araguz, 2019): the decision-making capacity of satellites becomes a potential feature for future autonomous DSS which must be respected in all the different protocols that enables the establishment of the necessary communications interfaces. Therefore, the definition of this space paradigm and the exploration of tangible solutions to deploy satellite networks following the federation premises, naturally constitute a fruitful field of study and has become one of the main motivations of this doctoral research. The following section continues by exploring the specific works that relate to this body of knowledge, detailing the state-of-the-art and introducing the theoretical background for this thesis.

1.3 Literature review

The trends in current user demands, and satellite federations presented before has hopefully outlined a clear understanding of the context in which this doctoral research is framed: the research on inter-satellite communications to enable satellite federations. The work presented here is committed to the exploration of the design aspects and challenges that interconnected satellite systems are exposed. In order to do so, this thesis has leveraged existing theory and constructs belonging to the distinct concepts of satellite networks, and the corresponding ISL technologies currently developed. This field of study and its critical research is presented in this section in order to later identify the objectives of this thesis. At the end, Section 1.3.12 summarizes the most critical aspects and points to some of the research gaps that motivated this thesis.

1.3.1 Nature of satellite-to-satellite contacts

Satellites are celestial bodies which constantly move tracing an elliptical trajectory around another larger celestial body. The EO and broadband telecommunications satellites orbit around the Earth. This distinctive motion is defined by means of the Keplerian orbital parameters or

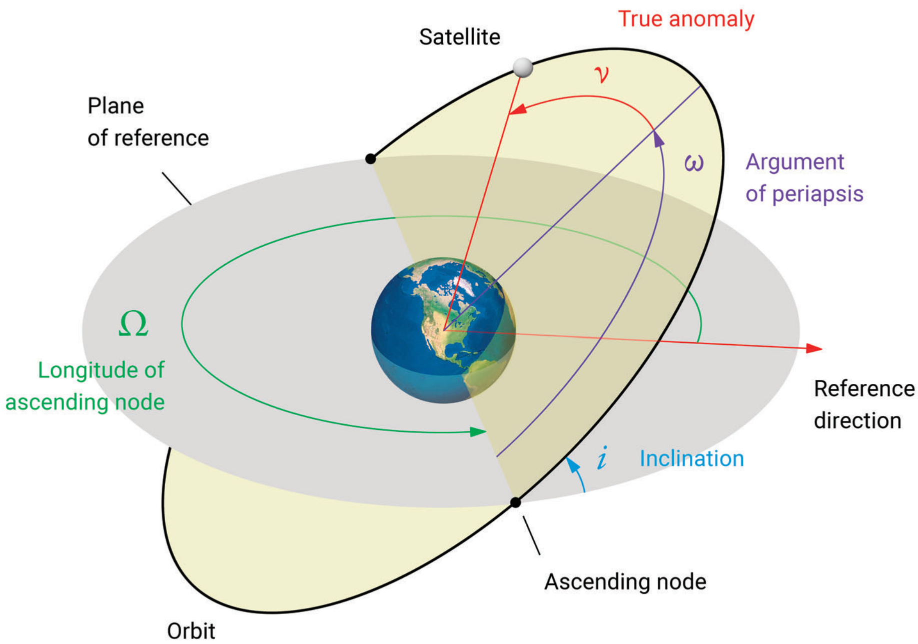


Figure 1.5: Representation of the Keplerian elements that represents the orbit trajectory of a satellite (gray sphere), and its plane (yellow surface) with respect to the equatorial plane (gray surface)

elements. These six parameters determine the shape and size of the ellipse, the orientation of the orbital plane with respect to the Earth, the orientation of the ellipse in this plane, and the location of the satellite along this trajectory. Satellite orbits are gravitationally curved trajectories which can be represented in a two-dimensional trajectory located in a plane. This plane is known as orbital plane, and two Keplerian elements determine its orientation with respect to the Earth: (1) the longitude of the ascending node (Ω) corresponds to the horizontal angle between the plane and the origin of the longitudes (the Greenwich meridian or prime meridian); (2) the inclination angle (i) determines the vertical orientation of the plane with respect to the origin of the latitudes (the equator). The orbit shape traced in this plane is determined by (3) the semi-major axis (a) that corresponds to half the distance between the periapsis and apoapsis of the orbit, and (4) the eccentricity (e) which describes how much the ellipse is elongated compared to a circle. The resulting ellipse can be rotated in the same plane determined by (5) the angle known as argument of periapsis (ω). A satellite travels this elliptical curve over time, moving periodically among numerous locations. The position of a satellite in this ellipse is represented by the (6) the true anomaly (ν) which corresponds to the angle of the satellite location at a specific epoch or time with respect to the direction of the periapsis. The ensemble of Keplerian elements allow the characterization of the satellite position, and its complete trajectory. Nevertheless, natural orbit perturbations may—in long-term—influence the Keplerian elements over time, resulting in a modification of the ellipse that conforms the satellite trajectory. Figure 1.5 represents these parameters to clarify their meaning.

Numerous orbits exist depending on the values of the Keplerian elements, which laid the foundation of different classifications. Besides the categorization based on the semi-major axis parameters—determining LEO, MEO, and geosynchronous orbits—, others are centered on the remaining parameters. Orbits are distinguished according to their inclination, resulting as the polar orbits—with an inclination of roughly 90º—that allow satellites to pass over the Earth poles, and the non-inclined or equatorial orbits—with an inclination of roughly 0º or 180º—that allow satellites to pass over the tropical areas. Another categorization is focused on the orbit eccentricity, determining the circular orbit with zero eccentricity, the elliptic orbit with an eccentricity smaller than 1, the parabolic orbit with an eccentricity of 1, and the hyperbolic orbit with greater eccentricities. These taxonomies demonstrate the abundant distinct types of orbits that current satellites travel, and present the space as a huge heterogeneous scenario.

The elliptical motion of a satellite has still some peculiarities that make it unique. This motion is driven by the coexistence of various energies: the kinetic energy, and the potential energy. Objects that are in motion with a specific velocity possess the kinetic energy. Meanwhile, objects located at a relative distance with respect to the Earth are stressed by gravitational factors, which allow them to possess a potential energy. The specific orbital energy—determined by the gravitational two-body problem—is the constant sum of the potential energy and the kinetic energy. Therefore, satellites traveling in an orbit keep constant the orbital energy, although the nature of this energy changes. Satellites located at high altitudes have predominant potential energy, while in the opposite case the main one is the kinetic energy. Heterogeneous satellites—with distinct orbits—experience different energies, which is reflected in different velocities. Furthermore, satellites in eccentric orbits experience a variation of their velocity along the orbit, while in circular orbits they move with a constant velocity. These velocity profiles play a key role in the apparition of satellite-to-satellite contacts.

A satellite-to-satellite contact corresponds to an approach of two satellites during which the distance between the spacecraft allows the establishment of a link. For a given set of parameters—frequency, antenna gains, transmitted power, receiver sensitivity, etc.—the establishment of these contacts or ISL is determined by the distance between the satellites over time. This distance profile is mainly determined by the relative velocity between the satellites, which results from the combination of each velocity vector. Previous researches studied the nature of these satellite approaches for other purposes rather than inter-satellite communications. The space object proximity started to concern the research community with the apparition of space debris. These human-made objects correspond to defunct systems—like satellites—which no longer serve a useful function. These orbiting debris may collide with operational satellites that could per se generate more debris. The same situation of two satellites colliding becomes a tragic undesired event. Consequently, large efforts intended to statistically appraise these spacecraft approaches. One of the outcomes of this research was the definition of the Probability of Close Approach (PCA) (Richey, 1986) between a *deputy* of two satellites. This mathematical formulation allows to estimate the probability of two satellites become close enough to consider it a nearness. This close condition was determined by a sphere around one of the satellites, while this could naturally restrict certain conditions. A volumetric method extending the PCA in which generic volume shape determined the approach of two satellites was presented in (Alfano and Oltrogge, 2015). More recently, it was extended to determine the probability that a satellite walks into a generic volume that belongs to another satellites (Alfano and Oltrogge, 2018a). This situation is lately named as satellite walk-ins and walk-outs. The research centered on estimating this probability enhanced the development of mathematical models to estimate the evolution of space debris (Braun et al., 2019).

Another investigation centered on determining the relative orbit of a spacecraft with respect to another. This relative position represents per se the distance between the satellites. Rendezvous missions, like the International Space Station (ISS), required the development of mathematical models to estimate this relative orbit between a chief satellite and a deputy satellite.

Those models are based on the Hill coordinate frame (Hill, 1878), which is a Cartesian coordinate system centered in the *chief* position. The relative position vector of a *deputy* satellite with respect to the *chief* one is just expressed with three components. The initial development models were non-linear, like the Clohessy-Wiltshire (CW) relative equations (Clohessy and Wiltshire, 1960). This formulation determines the relative motion of a satellite with respect to the chief satellite that is following a circular orbit. Schaub extended these equations in (Schaub, 2004) to formulate a linear expression of this kind of orbit, based on the difference between the Keplerian parameters previously defined in (Alfriend and Schaub, 2000).

These models enhanced the estimation of the distance between the satellites, and stood out the its variability over time. Therefore, the condition to establish a satellite-to-satellite contact or ISL may not be always satisfied, concluding that these ISL are temporal events that happens throughout the time. Lluch and Golkar illustrate this temporal nature in (Lluch and Golkar, 2015a) by evaluating numerous orbits with different Keplerian elements that generate distinct satellite-to-satellite contacts. Despite the study was centered on the definition of the best orbits to establish federations, the results demonstrate that the orbit difference determines when the contacts could happen, and their duration. This phenomenon presents a scenario in which ISL are created and destroyed over time, influencing the communications between satellites. In particular, communications routes between two remote satellites may be defined by the concatenation of multiple ISL. These routes become per se temporal because of the temporal nature of the ISL. Therefore, these communications interfaces between remote satellites is available during lapses of time, and the route composition may also change over time.

An additional phenomenon happens due to this temporality. The dynamism of the ISL leads to the possibility that two satellites may never communicate due to the lack of a route at a specific instant. This phenomenon is known as network disruption, and it provokes the fragmentation of isolated parts or sub-networks that may be connected by sporadic ISL over time. Therefore, the routes are also determined by a sequence of ISL concatenated over time. This behavior drives the possibility to deploy satellite networks, and the proper performance. Without any doubt, this has been the major challenge that various researchers have tried to mitigate or coexist with different solution in the last years.

1.3.2 Taxonomy of satellite networks

A satellite network is an ensemble of satellites which have communications interfaces that interconnect satellites with satellites to exchange messages. As previously stated, the availability of these connections—established with ISL—changes over time. This temporal nature is caused by the continuous motion of the satellites that conforms the network. The satellite movement also unleashes the fragmentation of the network at certain moments. This disruption phenomenon generates sub-networks or independent satellites which are temporally isolated, and cannot establish the corresponding communications interfaces. These isolated satellites may degenerate to not being able to establish any end-to-end communications route—composed of satellite-to-satellite links—due to the lack of instantaneous connectivity. The definition of a communications route in this tough network becomes an important hindrance to deploy this kind of infrastructure. Efforts to investigate this behavior were dedicated in the last years with the proliferation of multiple and assorted approaches. Among the aspects that can be regarded for classification purposes, we categorized them according to the strategy deployed to confront this network disruption. This section briefly introduces the proposed taxonomy of satellite networks, which its class is separately presented in the following sections.

Proposals that coexist with the network disruption were initially presented to demonstrate the feasibility of satellite networks. Mechanisms to characterize the connection dynamism enhanced the concept of defining communications routes over time by means of the store-and-forward mechanisms. This concept relaxes the necessity to have a route at a specific moment by concatenating satellite contacts that conforms this route over time. Nevertheless, the applications and services deployed over this context may accept this time forward-looking perspective, becoming delay-tolerant applications and services. Two different satellite network categories present this adaptation to the network disruption:

Snapshot networks (Werner, 1997) — The dynamism of satellite links encourage the perception of a satellite network as a sequence of topology instances in which the overall links remain stable during a lapse of time. Each of these states represents a snapshot of the topology configuration with the corresponding established ISL. The connections that conform the snapshot remain untouched until the satellite motion changes one of them. These snapshots are ordered in a sequence which is driven by the satellite movement, determined by the periodic trajectory of the orbits. Therefore, the sequence of snapshots itself becomes also periodic. Communications routes between remote satellites can be determined in each snapshot, and among the snapshots with in-advance route transitions along the sequence.

Delay and disruption tolerant networks (Vinton et al., 2007) — The intermittent connectivity experienced in satellite networks encourages to define an environment in which network partitions are frequent, and satellites must leverage on opportunistic satellite contacts to communicate. These temporal satellite contacts may be scheduled to identify communications routes available at specific moments, and over time. Therefore, a route corresponds to a sequence of satellite contacts that not necessarily may occur at the same moment. Mechanisms to store the received data in a satellite contact to lately forward it through the next contact become essential to define these over-time routes.

Alternatively to these adapted proposals, interventionist approaches were lately presented to mitigate the network disruption by controlling the ISL dynamism. The demand of services that required time-sensitive communications (e.g. satellite phone calls) motivated the conception of new satellite architectures that conform a novel satellite network design. The evolution of satellite links over time was under control by deploying satellites in an orchestrated architecture that conformed a specific and custom constellation. Three distinct satellite networks can be identified depending on the type of satellite architecture that is proposed:

LEO satellite networks (Ekici et al., 2001) — A satellite constellation with the spacecraft orbiting at LEO region conform this type of network. Despite this generic definition, this network has a unique architecture aspect that enables to mitigate satellite mobility influence on the network topology. The number of satellites, and their location in the constellation—number of planes, and satellites per plane—orchestrate a specific polar architecture that represents a mesh network over the most populated areas of the Earth. This mesh network connects each satellite with four neighbors, located in the same and adjacent orbit planes. Nevertheless, the satellites keep moving, and this mesh structure cannot be satisfied over the poles. Although the network cannot be established in these regions, the mesh structure is ensured over the areas where the users are located.

Massive satellite constellations — Following the same premise of LEO satellite networks, massive satellite constellations pursuit the eradication of the network disruption by creating a satellite architecture with seamless connectivity. In this context, end-to-end routes between satellites are always feasible pointing the quandary to how their composition changes over time.

The name of this type of satellite network—also known as mega-constellations—comes from the number of deployed satellites, which can grow up to thousands. These macro structures allow achieving this non-disruption network at the expense of increasing the complexity of the system (e.g. satellite operations, frequency coordination, in-orbit maintenance, etc.)

Multi-layered satellite networks (Akyildiz et al., 2002) — The combination of multiple satellite constellations located at different altitude regions enhances the visibility of the satellite network. Satellites located at higher altitudes may interact with a large group of satellites deployed in lower altitudes. Under this premise, a hierarchical and heterogeneous structure is defined by satellites located at LEO, MEO, and GEO regions which conform the layers of the satellite system. Despite the connections between the satellites still changes over time, upper-layer satellites offer more stable connections that can be used to avoid the disruption of the network. Groups of low-layer satellites represent the members associated to an upper-layer satellite. The fluctuation of this association due to the satellite motion is the main factor that characterizes the dynamism in this network. This hierarchical architecture normally integrates a LEO satellite network—with its mesh—as the low layer to control the member transitions over time. The end-to-end routes in this architecture can be defined in the same layer or through different layers.

The pursuit to transport current Internet technologies in satellites, and the emergence of architectures that mitigate the network disruption encouraged the definition of new satellite networks with non-conventional approaches. Two main technologies stood out among the different proposals in the evaluation of its applicability in satellite systems:

Mobile ad-hoc networks — The nodes that conform these networks are in motion, and capable to learn about the topology of the network. The end-to-end routes are defined according to this information that is updated over time. The network adapts itself to the mobility of the nodes thanks to this approach. This autonomy resides on the capability to learn about the network topology by each node, which makes it suitable for satellite systems. Nevertheless, the typical mobility pattern characterized in these networks does not include the disruption phenomenon that may be degraded by traditional solutions. However, thanks to its flexibility, scalabality, and autonomy this model has been adopted more and more in some predictable scenarios.

Wireless sensor networks — satellites may be modeled as a resource-constrained embedded system which is furnished with the capability to communicate. The definition of satellite networks becomes similar to a wireless sensor network, in which numerous sensors are interconnected to conform a network over which exchange data. These sensors, as in the satellite case, are resource-constrained devices with communication capabilities. Despite these similarities, satellite networks are normally characterized by larger communication distances than in the sensor case. Therefore, the use of this technology was targeted mainly for cluster and swarm satellite systems. The proposed satellite sensor network (Song et al., 2015) defines a central satellite that interact with pico- or femto-satellites using wireless sensor network techniques. The end-to-end routes are determined by unusual criteria in this kind of network, such as minimizing the consumption of the nodes. Therefore, the routes are defined taking in consideration the resources of the nodes or satellites.

These satellite networks present mechanisms to determine an end-to-end route, because they are centered in different scenarios. Their use to facilitate the deployment of satellite federations is still a discussion that needs to be addressed. The following sections extend the explanation of each network type by presenting details of the different protocols used to determine a route in each context. This protocol is formally known as routing protocol.

1.3.3 Snapshot networks

A snapshot of a network is the topology representation with the connections established between the satellites that remain stable during a lapse of time. The creation or destruction of an ISL results on the generation of another snapshot. The overall generated snapshots compose a sequence that represents the evolution of the topology over time (Figure 1.6). This evolution is characterized by the movement of the nodes, which in this case corresponds to the orbit trajectory. Just as a brief reminder, this motion is determined by a set of parameters that allow to estimate the complete trajectory of a satellite. Furthermore, this trajectory follows a periodic pattern, if no orbit disturbances are considered. Consequently, the sequence of snapshots is periodic, and predictable. Nevertheless, the periodicity of this sequence depends on the orbits of all the satellites that conform the system, being in some cases a long period (e.g. weeks or months). Therefore, the sequence does not remain periodic in local views.

The notion of a snapshot was firstly presented by Werner at 1997. He presented the snapshot sequence in (Werner, 1997) as a virtual topology representation of a satellite constellation with the objective to define a routing protocol suitable to establish Asynchronous Transfer Mode (ATM) connections through satellites. The Discrete-Time Dynamic Virtual Topology Routing (DT-DVTR) protocol (Werner et al., 1997) was designed to compute the best route between all the satellite pairs in each snapshot independently—called Instantaneous Virtual Topology (I-VT). The numerous snapshots were computed in-advance with ground facilities to generate the corresponding snapshot sequence. This off-line sequence computation allows to define in-orbit by each satellite the different routes in each snapshot using conventional routing mechanisms. At the end, the satellite is able to determine the best route to communicate with another satellite by uploading the different snapshots that conforms the sequence. Nevertheless, the performance of this approach is directly influenced by the number of satellites that compose the network.

Satellite constellations with numerous satellites would generate more snapshots caused by the ISL changes, and a longer sequence period due to the combination of different orbit periods. The resulting sequence size may difficult the upload of the different topology information—i.e. the routing tables—and the necessary resources to compute the entire sequence. This situation was discussed in (Gounder et al., 1999). The proposed solution to address this issue was the execution of a constant sampling of the network topology, instead of defining a snapshot by the natural change of an ISL. This rigorous snapshot definition entails the generation of a bounded sequence delimited by the snapshot sampling rate. This memory-bounded sequence could be used to compute the routing tables of each satellite, and upload them during their ground station contacts. Nevertheless, the resulting sequence may not identify the creation of snapshots caused by frequent changes of ISL that are not perceived by large sampling periods. Therefore,

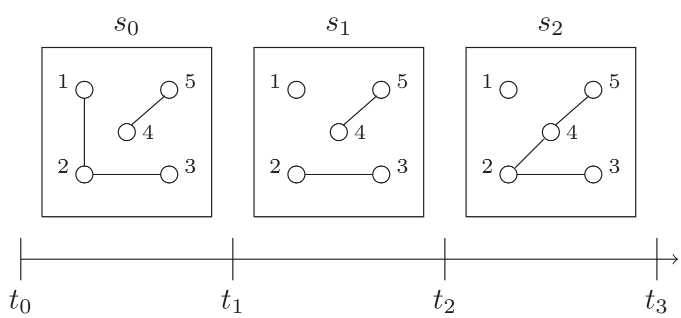


Figure 1.6: Representation of different snapshots (s_k) over time (t_i) associated to five satellites

the definition of the snapshot sampling rate becomes crucial in this solution. Although the size of a sequence may nowadays not be a limitation, this proposal remarks the necessity to conceive communications protocols that take in consideration the consumption of the satellite resources in their own definition.

The creation of the snapshot sequence became a discussion topic that promoted the proliferation of new techniques from lately investigations. Tang et al. proposed a method to optimize the definition of a snapshot taking in consideration the satellite constellation (Tang et al., 2015). Including in the sequence computation the characteristics of the satellite architecture allows determine regions in which the snapshot rate may be different. As an illustration, a polar satellite constellation could benefit from having a larger snapshot rate when satellites are close to the poles, than when they are on the equator. Regions that causes more changes in the network may require a better accuracy to determine the snapshots. This work reflects the efforts performed in the last years to reduce the size of the snapshot sequence.

The snapshot sequence changes depending on the satellites that participates in the network. When new satellites are deployed in this network, the sequence must be recomputed and uploaded in order to update the network information. This regeneration of the snapshot sequence needs to be conducted also for unpredictable events that change the network composition (e.g. satellite failure, satellite re-entry, orbit manoeuvre). The Predictable Link-State Routing (PLSR) protocol (Fischer et al., 2013) was formulated to manage these unpredictable events. The protocol is based on a ground infrastructure to compute the entire sequence, which is notified about the different unpredictable events. When a satellite detects an event that may change the network, it notifies this condition to the ground computing unit during its ground station contact. The ground facility recomputes the entire sequence which is then uploaded in the satellites. This strategy highlights the condition of integrating a central unit in the network that computes the snapshot sequence, making the satellite network dependent of this infrastructure. This feature may limit the flexibility and scalability of the resulting network.

The criteria to define the best route in a snapshot was also discussed in the following researches. The definition of the best route as the one that would have less number of hops or links was proposed in (Werner et al., 1997). The performance of this traditional approach in a dynamic environment like a satellite network was discussed in (He et al., 2008). The presented results suggest that this approach provides communications interfaces with low end-to-end latency, but the stability of the routes and the traffic congestion degrade its performance. Therefore, He et al. encourage the use of traffic state to determine the best route between two satellites. Furthermore, the computation of a ranked number of routes may provide the possibility to in-advance switch in case that one of them is no longer available. This multi-path capability seems powerful in the dynamic environment that a satellite network is.

The abruptly change from a snapshot to another may provoke a transition phase in which satellites must reconfigure themselves to the new scenario. The consequence of this transition could be the loss of communications during a lapse of time. Song et al. proposed a solution to overcome this situation by defining routes along the snapshots. The proposed Snapshot Integration Routing (SIR) protocol (Song et al., 2015) defines end-to-end routes inside and between the snapshots thanks to the integration of the entire sequence in a graph representation. This abstraction of the sequence allows the identification of the connections between the satellites over time, and facilitates the route definition. Despite the proposed approach founds on the snapshot concept, it presents lots of similarities with the DTN proposals. The SIR protocol promotes the data store in the intermediate nodes, which is forwarded in the corresponding snapshot. This remarks that snapshot networks, and DTN are similar concepts with a subtle difference centered on the definition of the network topology.

In summary, a snapshot network presents a satellite system as a sequence of stable topologies concatenated over time. This representation simplifies the challenge to define an end-to-end route in a dynamic environment by computing them in each snapshot independently. Nevertheless, the transition between snapshots may provoke the reconfiguration of the entire network, which could result with the loss of the communications during a lapse of time. Some proposals mitigated this transition effects by proposing techniques applied from snapshot to snapshot, or in the entire sequence. Furthermore, the snapshot sequence is determined by the number of satellites, resulting in a longer sequence when larger is the satellite system. This large dimensions may require large storage capacity in satellite dedicated to keep the information of the network, deallocating part for the main satellite payload.

1.3.4 Delay and Disruptive tolerant networks

The intermittent connectivity experienced in satellite networks encourage to define an environment in which network partitions are frequent, and satellites must leverage on opportunistic and sporadic satellite contacts to communicate. The understanding of this temporal nature of satellite contacts become crucial in the definition of end-to-end routes in this scenario. DTN approaches envision the establishment of communications interfaces in this disruptive environment. Therefore, a store-and-forward mechanism to store messages from a satellite to lately propagate them to another satellite was conceived to leverage the opportunistic and sporadic satellite contacts. Nevertheless, the definition mechanisms to identify end-to-end routes in this disruptive environment were largely discussed.

Among the characteristics of routing protocols, a classification was conducted in (Balasubramanian et al., 2007) based on the generation of replicas of the messages. In this regard, a protocol that replicates messages is known as a replication-based protocol, whereas a forwarding-based protocol is used in the other cases. This section presents the advantages and drawbacks of each approach, and multiple implementations of each class. The vast majority of the presented protocols are heuristic-based, and non-optimal. Balasubramanian et al. (2007) discussed this sub-optimal condition and concluded that the balance between future knowledge of the network evolution, and the computational capacity was the main cause. To put in his own words:

”online algorithms without complete future knowledge and with unlimited computational power, or computationally limited algorithms with complete future knowledge, can be arbitrarily far from optimal.” (Balasubramanian et al., 2007)

The proposed routing protocols may not guarantee the optimal solution for each case, but they employ practical mechanisms that are sufficient to reach a short-term goal.

Replication-based routing protocols

The delivery of the transmitted messages may not be guaranteed in a disruptive environment. Therefore, replication-based protocols allow the replication of messages by each intermediate satellite with the purpose of increasing the probability to correctly deliver the original message. The Epidemic routing protocol (Vahdat and Becker, 2000) was the first protocol based on this replication mechanism. Following a traditional flooding approach, this protocol continuously generates copies of a received message that are forwarded to each satellite neighbor. Despite its simplicity, the generated replicas may waste the network capacity and satellite resources because the majority may not be used. Therefore, more sophisticated techniques were conceived to limit the number of message transfers. The n-Epidemic routing protocol (Lu and Hui, 2010) was an example of this limited replication process by defining a minimum number of neighbors to transmit. A satellite in this scenario would not replicate a message if it is not surrounded by *n* neighbors. The protocol ensures with this constraint that a minimum of nodes receive the message, and thus increase the probability that the message reaches the destination. Furthermore, it mitigates the unfavorable situations in which the replica probably would not reach the destination. However, this approach is limited by the density of the network, being possible to not transmit any message if not enough neighbors are available. A more sophisticated extension of the original epidemic protocol is the Energy Aware Epidemic (EAEpidemic) routing protocol (Bista, 2016). This protocol is based on exchanging the node energy and reception buffer level states to evaluate if the transmission of the message should be done. Using this additional information, the messages are copied to only those neighbors which have energy enough to avoid the depletion. Using this mechanism, the overall network life is increased and thus the probability of message delivery.

An alternative approach to message replication was presented in the Spray and Wait routing protocol (Spyropoulos et al., 2005). This protocol ensures that a maximum number of copies coexist in the network. The source of the original message generates a limited number of copies that are one-by-one transferred to distinct satellites. After receiving the corresponding copy, the relay satellite waits until a contact with the destination is available. An additional mechanism to spray the copies was also presented in the original protocol definition called binary spray and wait (Spyropoulos et al., 2005). This proposal promotes that a satellite transfers half the total number of copies it has to each satellite encounter. When the satellite has a single copy of the message, it enters in the wait phase to contact with the destination. This binary spray mechanism enhances the possibility to deliver one of the copies to the destination by leveraging on the contacts from other satellites. The number of copies sprayed drives the performance of the protocol in terms of message delivery. The larger the number of copies are sprayed, the more probable to deliver one of them in the destination. Nevertheless, the spraying phase would require more time, and thus reaching the wait phase—in which the packet is indeed delivered to the destination—would require more time. Therefore, there is a trade-off between the probability to correctly deliver the message and the time to perform the delivery. Another spray strategy pursued the possibility to have a better message delivery while the time to reach the wait phase is acceptable. The Fibonary Spray and Wait protocol (Das et al., 2016) extends the original approach to achieve this goal with a spray phase following the Fibonacci sequence in mathematics. The satellite transmits different replications to another satellite depending on the number of copies that it has. Specifically, if the stored copies are a Fibonacci number in the sequence, then the spray mechanism is like the binary one. Otherwise, the satellite transmits the immediate predecessor in the Fibonacci sequence. This spray mechanism allows reaching the wait phase sooner than the binary case.

The waste of network resources was also investigated from another perspective rather than limit the number of copies: replicate just the messages that are essential. The MaxProp routing protocol (Burgess et al., 2006)—originally proposed for vehicular networks—optimizes the consumption of network resources by selectively replicating the messages. A satellite that establishes a contact with another satellite would identify all the messages that are not held by this satellite, and it would attempt to replicate them. The determination of which messages should be transmitted, and the ones to be dropped becomes an important challenge. The proposed solution is based on an ordered-queue per destination ordered by the estimated likelihood of a future transitive path to that destination. This probability is recomputed in each satellite contact, and according to its new value the packets in the queue are dropped or maintained.

The mechanisms to deliver messages presented in the previous routing protocols are suitable for scenarios in which encounters are purely random. Nevertheless, as stated previously, satellite contacts are rarely random, being some of them more probable than others. Therefore, some protocols attempted to exploit these non-randomness encounters by maintaining a set of probabilities for successful delivery. The representation of this probabilistic behavior was defined as a MobiSpace (Leguay et al., 2005) which corresponds to an Euclidean space where nodes can be either mobile or static, and they can communicate within their transmission range. The decision to forward or replicate a message to a specific node is conducted according to its probability to deliver it to the destination. The mechanism that determines this probability drives the correct delivery of the messages. Therefore, great efforts were centered on identifying the best approach. The original definition (Leguay et al., 2005) considered the encounters sequence to determine the probability. Nevertheless, the status of the intermediate satellites was not included in the algorithm. This results in large end-to-end delays due to the saturation of intermediate satellites. The Motion VEctor (MOVE) routing protocol (LeBrun et al., 2005) combines two probability matrices to overcome this situation: one that represents the probability of delivery, and the other that is the message sojourn at each satellite. The probability to deliver a message in conjunction to the transmission delay can be estimated with these two matrices. The route decision can be adapted according to the network congestion.

An extension of the MobiSpace was defined for those scenarios in which mobility is cyclic. In this cyclic MobiSpace, nodes are often in contact at a particular time in previous cycles, then the probability that they will be in contact at the same time in the next cycle is high. Considering this features, the Routing in Cyclic Mobility (RCM) proposal (Liu and Wu, 2008) uses an historical of the different encounter at each cycle to quantify the probability to encounter a specific node. Using this information, each node can map the network as a probabilistic state graph enabling the routing decision. Although presented in 2003, the Probabilistic Routing Protocol using Historic of Encounters and Transitivity (PROPHET) (Lindgren et al., 2012) follows a similar strategy than the RCM approach. This implementation was based on the satellite mules, which are nodes that may carry data from other satellites because they have greater probability to encounter other satellites. Nodes using the PROPHET would replicate a message copy to a satellite mule—that does not have it—only if this one appears to have a better chance to deliver it. The computation of this likelihood is conducted using an historic of the different encounters, with which each satellite can recompute the probability that a mule would encounter to a specific satellite.

The prior routing protocols incidentally affect performance metrics, such as average delay and message delivery ratio. Nevertheless, the possibility to affect a single performance metric was not feasible with these approaches. The Resource Allocation Protocol for Intentional DTN (RAPID) proposal (Balasubramanian et al., 2007) enabled to adapt the performance of the routing protocol according to the envisioned service. The original proposal was instrumented to minimize one of the three metrics: maximum and average end-to-end delay, and missed deadlines. The RAPID protocol is founded on the definition of a utility function, which assigns a value to each received message depending on the selected metric to be optimized. The transmission of a message among the others is performed for those that have the highest increase in the utility—i.e. the transmission is useful. As in the other cases, the protocol is based on the replication of the messages.

These Replication-based routing protocols allow reaching larger message delivery rates, since multiple copies exist in the network and only one must reach the destination (Burgess et al., 2006). However, these replicas waste valuable network resources. Therefore, novel forwarding-based protocols proliferated to provide an alternative that waste less resources, but require certain level of computation.

Forwarding-based routing protocols

Forwarding-based routing protocols envision to minimize the waste of the network resources by propagating a single message along the network until its corresponding destination. This transmission strategy—commonly used in ground networks—becomes a great challenge in disruptive environments in which an instantaneous end-to-end route may not exist. As in the Replication-based case, a set of forwarding-based routing protocols leveraged on the deterministic movement of satellites to estimate the succession of satellite contacts. The Contact Graph Routing (CGR) protocol was presented in 2010 by Burleigh as a mechanism that benefits on to the determinism of satellite motion to define end-to-end routes over time (Burleigh, 2010). This CGR protocol proposes the composition of a plan that schedules all the satellite contacts to provide a global view and future knowledge of the network evolution. An end-to-end route is defined as a sequence of ordered satellite contacts that may not occur at the same moment. The CGR protocol was designed to establish communications in the NASA Interplanetary Overlay Network (ION) (Burleigh, 2007), but it has gained interest recently for LEO satellite constellations. The presented protocol standard does not specify, however, the procedure to generate the contact plan.

Traditionally, a centralized solution in which the ground facilities propagate orbit trajectories and compute the corresponding plan—as in the snapshot network case—was preferred. The use of a central unit called Contact Plan Computation Element (CPCE) that supports the generation of the contact plan and the route definition was presented in (Fraire and Finochietto, 2015). This CPCE pre-computes the contact plan and the routes on-ground to periodically upload them into the satellites using the ground station network. Although this centralized approach can generate accurate contact plans for complex systems, the upload process of the plan needs still to be further investigated. In particular, if a new satellite is included in the system, the recomputation and upload of the entire contact plan must be conducted, making the solution difficult to be scalable. Moreover, this centralized facility entails an important operations cost, and makes the satellite system ground-dependent. Alternatively, other prediction mechanisms were proposed. The Opportunistic CGR (OCGR) (Burleigh et al., 2016) proposed that each node estimates satellite contacts and their duration using a historical register. This register accumulates information of contacts that have already achieved, which is updated according to a confidence parameter. This parameter represents how reliable is the new predicted contact with a satellite, according to the possibility to correctly deliver message. This mechanism is characterized by having an important learning curve, which as compared to others that use determinism to estimate contacts (like CGR) degrades its performance. The amount of data that can be exchanged during a contact, known as contact capacity, is another parameter used by the CGR to determine the routes. Walter and Feldmann proposed a mechanism to estimate this capacity between satellites and ground stations, modeling the features of the transceiver and the communications loss (Walter and Feldmann, 2018).

Last researches pursuit the unification of both strategies—with a contact plan and with a historical record—in a unique routing framework that enables to define routes if the network is predictable or not. The outcome was formalized as a new Consultative Committee for Space Data Systems (CCSDS) standard called Schedule-Aware Bundle Routing (SABR) protocol (Book, 2019). This approach provides the necessary means to the satellites in order to adopt the preferred routing strategy according to each scenario.

An alternative forwarding-based routing protocol is presented in the Delay Tolerant Link State Routing (DTLSR) protocol (Demmer and Fall, 2007). This proposal extends the traditional link-state routing protocol in which nodes periodically forward the information about the network topology. This information allows the nodes to define the available end-to-end routes of the entire network. Moreover, the periodic transmission of this information enhances the possibility to adapt these routes with the corresponding changes of the network links. The DTLSR, however, proposes a peculiarity regarding the links that are considered broken. These links are tagged as 'down', and they are not directly discarded. Instead, a metric that indicates the age of the last connection is associated. When the link is old enough, the register is then removed. This mechanism intends to cause data to continue to flow along paths that used to be supported in the hope that they will be supported again in the near-future.

The application of a DTN solution for LEO satellite network is presented in (Diana et al., 2017). In particular, authors expose the benefits of the novel DTN routing protocol, called DQN, in quasi-deterministic networks. A LEO satellite network follows a deterministic dynamism, because it is ruled by satellite mobility. However, due to traffic generation and link outages this scenario cannot be considered totally deterministic, and thus quasi-deterministic. In this case, DQN does not use a predefined routing policy, indeed it uses the exchange of node contacts to determine the node with the closest destination. Although this approach is really specific for this satellite architecture, it demonstrates the flexibility of DTN routing protocols to be used in different satellite scenarios

The forwarding-based routing protocols enhances the consumption of the network resources in exchange of requiring a processing unit to compute the future knowledge of the network. In this regard, centralized processing units seem to stood out as a feasible solution. Nevertheless, satellites require constant communications with this infrastructure to update the available routes previously defined. This requirement may be difficult to be achieved in a heterogeneous context like FSS, which require certain level of autonomy to deploy federations.

1.3.5 LEO satellite networks

In the early 1990s, missions to deploy a near-polar satellite constellation in which spacecraft orbit in the LEO region proliferated to provide broadband telecommunications services in remote areas. This satellite system gained popularity with the development of the Orbcomm and Globalstar satellite constellations (Ilcev, 2011; Wiedeman and Viterbi, 1993). These solutions deployed a satellite constellation to relay data from ground users using spacecraft as bent-pipe systems. A user was able to communicate with another user that would be in the spot of the same satellite. The Iridium corporation proposed another satellite constellation to compete with these previous architectures. This novel satellite system included a distinctive feature: the establishment of satellite-to-satellite communications interfaces to relay data. This interconnected architecture was lately referred to the term of LEO satellite networks.

This kind of network was defined to mitigate the disruption provoked by the satellite motion. The constellation is designed with an orchestrated and fixed architecture in which satellite are specifically located on purpose. This constellation builds a mesh architecture in which each satellite has four satellite-to-satellite interfaces to communicate with its neighbors. The resulting topology of the network is characterized by nodes located in a grid with a set of rows and columns. Despite this design to mitigate the disruption, satellites are always in motion. How-

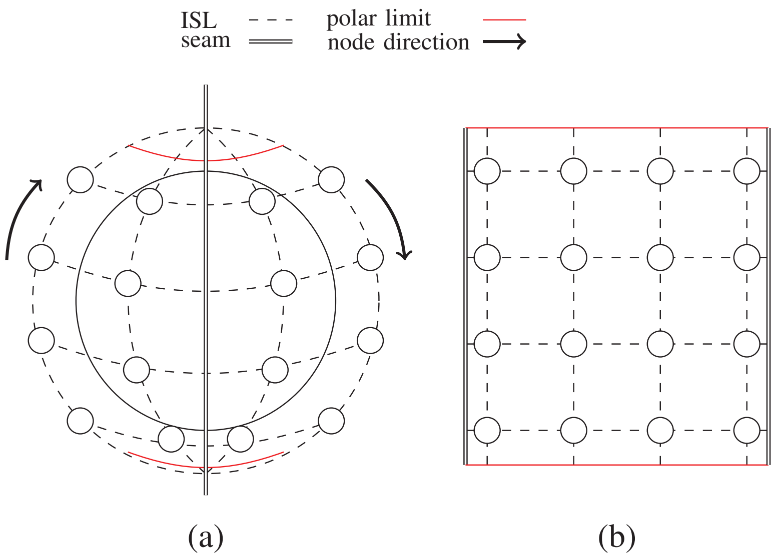


Figure 1.7: Representation of (a) the constellation design that represents a LEO satellite network, and (b) its resulting map to a mesh topology.

ever, the movement of the satellites in this constellation results on a continuous shift of the satellites in the column axis of the mesh. This coordinated movement ensures that from the local view of a satellite the connections with its neighbors remains unaltered. This condition is satisfied in the most populated latitudes, because when the satellites pass over the polar region the formation cannot be respected. Moreover, an abstract line represents a seam in this mesh that separates the direction of the satellites. On one side of this seam, the satellites move from South to the North, while on the other side they travel in the opposite direction. Traditionally, the communications through this seam was forbidden. Figure 1.7 presents this satellite constellation with its corresponding mesh topology.

This network design was firstly presented in (Ekici et al., 2001), and called these satellite-to-satellite connections ISL. Two classes of ISL were defined according to the vicinity of the neighbor. The intra-plane ISL allow a satellite to communicate with its two neighbors that are located in the same orbit plane. Meanwhile, the inter-plane ISL allow a satellite to communicate with its two neighbors located in adjacent planes. This differentiation was conducted because the nature of both ISL types differs: intra-plane ISL are always stable and feasible, while inter-plane ISL may be disconnected in the polar region. The goal to relay data from ground users was satisfied by defining the concept of virtual node. This kind of node is associated to a logical location that corresponds to a square of an entire grid that covers the entire Earth surface. Each satellite is then associated to a logical location when it passes over this surface square, being responsible to serve the users allocated in this area. Due to the satellite movement, the satellite changes over time their logical location when they bypass the corresponding square. Furthermore, the logical location is also mapped in the mesh topology by a vertical and horizontal coordinates that correspond to the column and row numbers.

Ekici et al. based their research on this architecture, and defined the Datagram Routing Algorithm (DRA) (Ekici et al., 2001). This algorithm allows the definition of end-to-end routes between virtual nodes by directly comparing its logical location. This comparison results in the number of rows and columns that separate the nodes in the mesh. Therefore, the route between two remote satellites corresponds to a sequence of satellites or virtual nodes following the defined columns and rows. This simple route computation makes this approach really fast, and suitable for low-latency communications. The DRA also selects the route that is composed by the minimum number of hops. However, due to the mesh formation, multiple routes with the same minimum hops exist simultaneously. Therefore, the transmission of the messages is conducted as in a datagram network: node-by-node. When a satellite receives a message, it evaluates the route to reach the corresponding destination. After determining the number of rows and columns that the message has to be forwarded, the satellite must decide if the next hop is conducted in the row or column direction. Additional link metrics may be used to determine the next hop in the route.

Traditionally, the criteria to select the next hop of the route was the one that ensures the shorter route. As multiple feasible routes satisfy this requirement in this network, additional metrics that quantify the ISL status are used. The DRA estimates the next hop to achieve the lowest transmission delay. However, users located in different Earth regions generate distinct traffic volume (Taleb et al., 2005). This makes that some ISLs congest more frequently, increasing the end-to-end delay. Therefore, DRA includes in the definition of the next hop the load of its own ISL queue load to determine the congestion state. If this state is unacceptable, the node redirects the traffic through an alternative path. At the end, the next hop is determined node-by-node to achieve the minimum-hop route, the lowest transmission delay, and the least congested route. Figure 1.8 shows an example of packet transmission between the source (black node) and the destination (gray node). At the very beginning, the source identifies the six paths with four hops (the minimum value). After evaluating the state of its links, the source decides to transmit the packet to its left neighbor. Successively, the intermediate node performs the same algorithm to define the following hop until the message reaches the destination.

The DRA proposal sets the basis for the next developments which envisioned to improve the original concept. (Taleb et al., 2009) identified that the congestion mechanism used in the DRA could not avoid the apparition of this undesired condition. The original algorithm evaluates the local queue status as a proxy of the network congestion, which may not be representative enough of the entire system. This lack of information provokes a slow reaction against the node congestion which could unleash to the successive failure of all the nodes. Investigations were conducted to address this situation in LEO satellite networks.

The Explicit Load Balancing (ELB) protocol (Taleb et al., 2009) proposed a proactive scheme in which congested satellites notify the neighbors to decrease transmission data rates. This

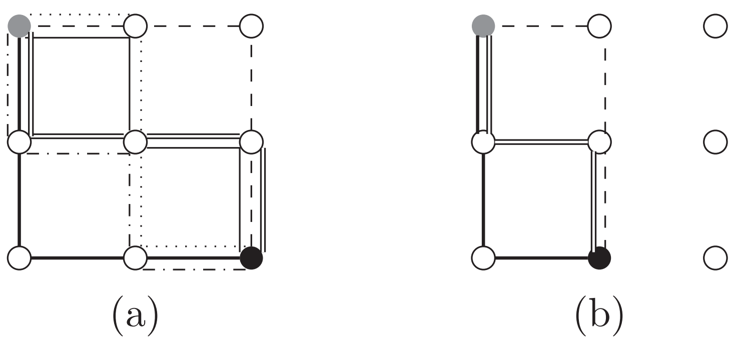


Figure 1.8: Illustration of the minimum-hop route during the transmission of a packet. (a) First the packet is placed in the source (black node) and shall be transmitted to the destination (gray node). When the packet is transmitted to the left neighbor, (b) the path determination is computed again.

notification allows effectively redirecting the traffic to avoid perpetuating the saturation of the satellites. Nevertheless, no solution is presented in the case that the alternative paths would also be congested. Alternatively, the Priority-based Adaptive Routing (PAR) protocol (Korçak et al., 2007) pursued the estimation of congestion trends depending on the queue loads. A historic of the drop/transmission rates and the current queue usage allow a satellite to understand if the congestion is happening, and redirect the traffic accordingly. Another approach that follows a similar strategy was the Traffic-Light-based Routing (TLR) protocol (Song et al., 2014). This protocol proposed the use of a traffic light in each ISL queue that represents the congestion state of the link. Depending on the "color" of the traffic light the data transmission over this link would be enabled. This "color" is computed with the local information of the queue load, and the information of the adjacent neighbors. Combining both metrics, the congestion status of the ISL can be estimated, and the traffic redirected.

All these proposals founded on the local link state to avoid or manage a congested scenario. This limits the capability to manage the situation of a global network congestion. The Agent-based Load Balancing Routing (ALBR) protocol (Rao and Wang, 2010) was proposed to manage the congestion of the entire network. This distributed protocol promotes the integration of a stationary agent—a software module—in the satellite that periodically computes the path cost, and also updates the routing table. This static agent is complemented by a mobile agent that travels through each satellite, and retrieves the congestion state of each one. Aggregating the information collected satellite by satellite, the mobile agent is able to generate a global view of the entire network. Therefore, this agent may prevent the congestion by suggesting in each satellite to redirect the data. Nevertheless, the amount of agents is related to the amount of satellites in the network, which could difficult its deployment in large satellite constellations.

As previously stated, the original LEO satellite network architecture delimits the operation of the satellites in the region over which the mesh formation is respected. On the polar regions, the ISL are disconnected because the same mechanism to define a route cannot be properly used. The existence of these regions makes that a triggered message transmission may reach a deadlock dropping the corresponding message. This deadlock represents a satellite that cannot forward the message because the next hop or destination is located in this forbidden region. Figure 1.9 illustrates this deadlock condition when a satellite source (black node) transmits a packet to the destination (gray node) located in the polar region. As in this region the communication is forbidden, the packet never reaches its destination, and it only arrives to the deadlock node (double-line node). Network resources are wasted if this condition happens. Therefore, the Distributed Load-Aware Routing (DLAR) protocol (Papapetrou and Pavlidou, 2008) leveraged on the deterministic movement of satellites to estimate if the destination is unreachable before performing the transmission.

LEO satellite networks emerged as a potential architecture to provide near-real-time services centered on broadband telecommunications applications. Its organized structure conforms an optimal mesh network that facilitates the definition of a route, and the fast forwarding of the messages in the network. This design promotes the emergence of multiple routes that satisfy the selection criteria, enabling to apply multi-path techniques. Therefore, the investigations were centered on how leverage this characteristic to avoid or manage traffic congestion. Nevertheless, this fixed architecture—that may be a virtue—also represents its major drawback in a heterogeneous context like FSS, in which no predefined architecture can be considered a priori. Furthermore, this custom architecture needs to be always respected, limiting the flexibility of the satellite network. This requires a set of mechanisms to always ensure this architecture. Just as an illustrative example, satellite Iridium 33 collided with a Russian satellite in 2009 which

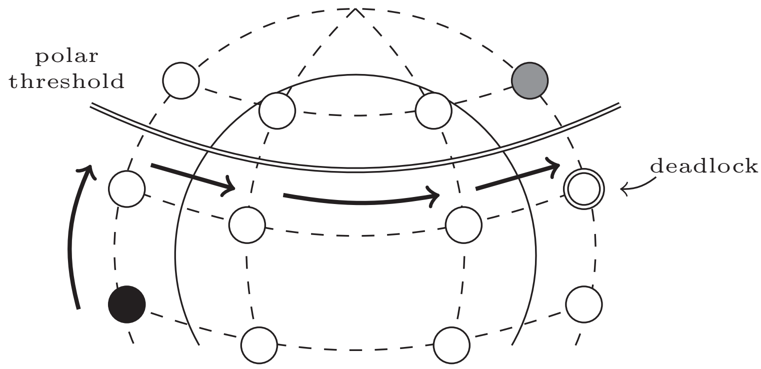


Figure 1.9: Deadlock representation between a source (black node) and a destination (gray node).

provoked—among the multiple space debris—the rupture of the constellation. A spare satellite was required to replace the missing satellite.

1.3.6 Multi-layered satellite networks

The combination of multiple satellite systems stood out as a potential architecture to offer new capabilities or improve the capacity achieved by their own. Multi-Layered Satellite Networks (MLSN) are a system-of-systems architecture (Maier, 1998) compounded of distinct satellite constellations deployed at different altitudes, which corresponds to the layers in this system. This architecture was proposed in (Akyildiz et al., 2002) to enhance the traffic capacity, and the stability of a satellite network. The proposal leverages on the visibility of the satellites located at higher altitudes that can orchestrate a group of satellites deployed in lower altitudes. A hierarchical structure in which successively upper layers always gather lower layers is designed under previous premise. Despite the original proposal did not specifically define the type and the number of layers, the LEO, MEO, and GEO were typically the three main layers associated to this network. In this configuration, GEO satellites would manage a group of MEO satellites, which at the same time each one would gather an ensemble of LEO satellites. Satellites located in the same layer can communicate among them using Intra-Orbital Links (IOL), while they are also able to interact with satellites in adjacent layers using ISL. Figure 1.10 illustrates this multi-layered architecture with the three main altitudes.

This hierarchical architecture enhances the stability of the network thanks to the large visibility of upper-layer satellites. Despite the connections between the satellites still changes over time, the topology changes correspond to fluctuations of the low-layer satellites that belong to the group of an upper-layer satellite. This feature mitigates the influence of the satellite mobility on the network dynamism. Nevertheless, the architecture design of the low-layer satellite system may still provoke irregular changes on the topology. (Lu et al., 2013) discussed this behavior, and suggested the use of a LEO satellite network—which ensures the mesh formation—as a lowest-layer satellite system. This satellite constellation simplifies the computation of end-to-end routes among the layers, and in the same layer. The integration of a LEO satellite network into the MLSN demonstrates the potential of this heterogeneous architecture, that may accept including multiple distinct satellite systems.

Despite the enhancement in the definition of routes thanks to the deployment of an organized structure, numerous possibilities can be defined that compounded by inter- and intra-layer communications. Therefore, large efforts were dedicated to determine a routing protocol that

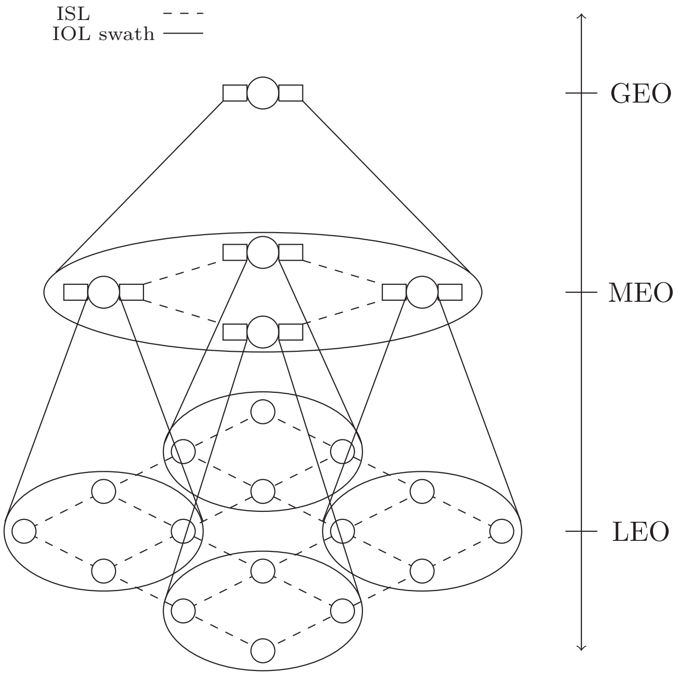


Figure 1.10: Illustration of a MLSN with three layers.

could work in this scenario. The first study on this field was presented by Lee et al. to determine an optimal routing protocol in a two-layered satellite network. Although the MLSN was not formally presented yet, the Hierarchical QoS Routing Protocol (HQRP) was designed to define routes in a satellite network constituted by satellite constellations located in LEO and MEO regions (Lee et al., 2000). This approach was based on the distinction of different information related to intra- and inter-layer network topology. The Local Routing Information (LRI) provided the topology of a specific layer, which allowed to determine routes in this layer. Meanwhile, the Global Routing Information (GRI) corresponded to the status of the O-ISL between layers. Each LEO satellite was able to compute the minimum-hop route using both types of information. This route was mainly determined by the associated MEO satellite to forward the data.

This specific case of two-layers was lately generalized to a multi-layered architecture with the Multi-Layered Satellite Routing (MLSR) protocol (Akyildiz et al., 2002). The goal of this protocol is the definition of routes in MLSN with more than two layers by reducing the computational complexity, the communications overhead, and the end-to-end delay. The protocol retrieves delay measurements of the satellite connections in a layer, which is suddenly reported to the corresponding upper-layer responsible. In this regard, the LEO satellites report the connection status to MEO satellites, and these to GEO satellites. Upper-layer satellites forwards this lower-layer information to achieve a global view of the different layers. The satellites belonging to the highest layer are the nodes which determine the different routes in the lower layer, and report them to down. This procedure allows to reduce the signaling overhead, and maintaining the hierarchical structure. This approach also suggest that upper-layers may be used to manage the network, rather than forwarding data. This conclusion was also presented in (Lluch and Golkar, 2014b) were results suggested to use these IOL to transfer small messages—like control packets—thanks to their link stability, and limited throughput conditioned by large distances.

As in the LEO satellite network case, the MLSN proposal envisioned to provide broadband telecommunications services. Therefore, multiple investigations discussed the management of the incoming traffic to avoid the congestion of the network, and keep providing low-latency communications interfaces. The developments in LEO satellite networks may be applicable into intra-layer systems, but the possibility to use the wider visibility of upper-layer may enhance the congestion management. The Tailored Load-Aware Routing (TLAR) protocol (Wang et al., 2015) proposes a mechanism to retrieve the congestion status of the LEO satellite connections, and forward it to the corresponding associated MEO satellite. This periodic reporting allows the upper-layer satellites to achieve a global view of the lower-layer status, and decide accordingly. Therefore, LEO satellites can detour traffic over MEO satellites to decongest the layer. The upper-layer satellites become a support infrastructure in this scenario. Nevertheless, this protocol does not consider the congestion of MEO satellites. An upper-layer satellite may be congested when all its lower-layer satellites relay data over it. This situation was addressed in (Kawamoto et al., 2013) by applying specific constraints in the hierarchical structure in order to have multiple paths for LEO satellites. Kawamoto et al. discussed that satellites located at higher altitudes may provide more IOL with the same LEO satellites, because different upper-layer satellites are in line-of-sight with the single lower-layer satellite. This feature allows having multiple routes between layers through which the traffic can be split, thus reducing the probability to congest upper-layer satellites. Although the presented results demonstrate the enhancement on the congestion, the increment of the altitude influences directly on the propagation delay. This may not be negligible when distances between layers become considerable—like LEO and GEO regions—, which may degrade the desired low-latency performance. Therefore, a trade-off exists in this approach between reducing traffic management, and increasing end-to-end latency.

Another research proposed the use of the definition of a topology snapshot in the MLSN context (Wu et al., 2017). The resulting Hop-Constrained Adaptive Routing (HCAR) protocol uses the snapshot sequence of the lower-layer satellite system to predict the future changes on the group associated to an upper-layer satellite. This prediction is computed on ground to upload it later to the upper-layer satellites, which becomes responsible to forward it to the lower-layer satellites. The proposed protocol was evaluated in a two-layered satellite network composed of LEO and GEO satellites. The presented results suggested that this protocol successfully combines both models, and improves the capacity of the entire system by anticipating topology changes. However, the centralized solution may limit the flexibility and scalability of the entire system.

The association of lower-layer satellite ensemble to a specific upper-layer satellite was investigated to define the most suitable strategy. A mechanism to dynamically determine the members that are associated to an upper-layer satellite was presented in (Yi et al., 2013). Each group of satellites is identified by three distinct roles: the Group Header (GH), the Group Member (GF), and the Group Manager (GM). The GH corresponds to the upper-layer satellite that collects all the information of the lower-layer satellites. The different satellites that are associated to a GH are the members of the group (the GFs). Among the different members, the GM corresponds to the one that holds the best adjacency with the GH. This GM computes the network changes that are forwarded to the GH, which decides the proper routes on the lower layer. The configuration of the satellite group is achieved by conducting two separated phases: one to compute the predictable changes, and another one to manage unpredictable congestion status. This variable grouping allows adding configuration flexibility and easy management into the MLSN model, although it is a solution Earth-dependent which limits the autonomy of the network.

In summary, multi-layered architectures leverages on the integration of different systems to improve the performance reached individually. This heterogeneous system can be interconnected to enhance the network capacity and stability of the topology, as the results presented in (Liu et al., 2015) suggest. This improvement is related to the number of connections between layers, which promotes the availability of multiple routes simultaneously. Heterogeneity in satellite networks is a key characteristic that may be integrated in future satellite missions.

1.3.7 Massive satellite constellations

The emergence of 5G motivated the development of novel infrastructures that provide continuous Internet access. Despite the numerous improvements in the ground facilities, satellite platforms stood out as a potential system to achieve this requirement. LEO satellite constellations are naturally characterized by providing global coverage to the entire planet. Iridium constellation is an illustrative example of how this architecture can provide data access to a widespread group of ground users. Despite this infrastructure, current Iridium services are limited to a poor messages exchange (hundreds of kbps), which may not be sufficient for current and upcoming Internet services (e.g. video streaming, cloud computing, etc.) Therefore, an extension of these traditional LEO satellite constellations has been proposed to cope with this new demand.

Private companies have taken a step forward in the development of massive satellite constellations, which assemble hundreds or thousands of satellites to provide global and seamless Internet coverage with competitive interfaces; i.e. with low latency and high throughput (Pultarova, 2015). These ambitious goals cannot be achieved with traditional architectures nor delay-tolerant solutions. Among the different companies, OneWeb Ltd.—previously named WorldVu—was the first one that announced the development of this macro architecture (Selding, 2014). Joining efforts with Virgin Group and Qualcomm, OneWeb expected to deploy 720 satellites at 1200 km height. This LEO satellite constellation was not designed to include satellite-to-satellite architecture, which require the deployment of further satellites to satisfy current demand (Portillo et al., 2019). Space Exploration Technologies Corp. (SpaceX) publicly announced the development of the Starlink mega-constellation one year later (Gates, 2015). Starlink comprises 4425 satellites that would be distributed across several sets of orbits. Three different layers are distinguished: the main layer at 1150 km, the secondary layer at 1110 km, and the third layer at 1130 km. This macro satellite system corresponds to an MLSN thanks to the use of satellite-to-satellite interfaces, although all the layers are located in the LEO region. More recently, Telesat Canada envisioned to deploy a massive satellite constellation of 117 small-satellites to compete with the previous constellations (Caleb, 2017). These satellites would include a dedicated ISL with high transmission capacity. Other projects proliferated later (Boyle, 2020).

These massive satellite constellations—also known as mega-constellations or satellite Internet constellations—pursuit the eradication of the satellite disruption by creating a satellite architecture with seamless connectivity. End-to-end routes between satellites are always feasible pointing the quandary to how their composition changes over time. Despite no information has been published about specific routing protocols, this characteristic opens the door to integrate Mobile Ad-hoc Networks (MANET) solutions in this kind of architecture. Nevertheless, current solutions seem to envision simplified IP protocols that are currently working on ground networks (Highfield, 2018).

Preliminary studies demonstrated that these massive satellite constellations can provide communications interfaces that can satisfy high-data volumes (up to Tbps), and low-latency communications (Handley, 2018; Portillo et al., 2019). Despite the potential performance of this architecture, its enormous size poses numerous challenges. Among the different ones that have been discussed during the last years (O'Callaghan, 2019), six challenges stand out: (1) The required funds to maintain the development of the entire project (Caleb, 2020b,c); (2) The necessity to develop and construct a satellite manufacturing infrastructure to reduce the production cost (Caleb, 2020a; Foust, 2019); (3) the increase of space debris due to the overpopulation of the space (Harris, 2019; The European Space Agency, 2018); (4) the hoarding of frequency bands due to the necessary wide bands allocations; (5) the complex administrative registry of this large number of satellites (Caleb, 2019b); (6) Impact on other space fields, like astronomy (Foust, 2020). These constraints make that the deployment of this massive satellite constellation feasible to specific companies or entities. Another perspective in which does not require the launch of massive constellations from independent entities needs to be conceived to balance these problems.

1.3.8 Mobile ad-hoc networks in satellites

Previous routing protocols use deterministic satellite movement to predict the network evolution in advance, resulting as a reduction of the signaling overhead and end-to-end latency. Nevertheless, these approaches were designed to work with a central computing unit that may difficult the reaction against unpredictable network events (e.g. satellite failure, satellite manoeuvre, etc.) MANET were conceived with the necessary capabilities to adapt themselves against these unpredictable events, and deploying more autonomous networks. The nodes that compose these networks are capable to adapt their communications depending on the topology changes, emphasizing on those related to node mobility. These nodes autonomously learn the network topology which allows them to define end-to-end routes and their evolution. These routes, however, correspond to concatenations of links that are available at a specific moment. Therefore, these networks cannot overcome the disruption of the network present in the satellite networks.

Numerous routing protocols were designed to define the route evolution in this kind of networks (Boukerche et al., 2011). Multiple metrics were used to categorize them, but a specific taxonomy has persisted during the last years. The developed routing protocols were classified depending on the mechanism that is used to learn about the network topology. Three different classes are identified: reactive protocols, proactive protocols, and hybrid protocols. A reactive protocol learns about the network topology only when a data transmission must be conducted. On the other hand, a proactive routing protocol corresponds to a mechanism that forces a node to periodically evaluate the network topology. The hybrid solution combines mechanisms from both approaches. This section presents the advantages and drawbacks of each type of protocol by introducing different examples. Some of the presented protocols were integrated or developed in satellite missions. Nevertheless, the section details other protocols that were not applied to this context, but they are largely used on ground and present powerful features for satellite networks.

Reactive routing protocols

Reactive routing protocols learn the network topology only when a data transmission is required. This approach allows limiting the required overhead associated to this operation to only when it is needed. Therefore, this solution is suitable for sporadic communications. Nevertheless, the lack of initial information of the network generates a transition phase that results to the increase of the end-to-end latency. Therefore, time-sensitive services may be difficult to be deployed with this protocol family. The most popular reactive routing protocol is the Ad-hoc On-demand Distance Vector (AODV) detailed in (Perkins et al., 2003). When data must be transmitted, the corresponding node triggers a phase to discover the possible routes that are available at that moment. This route construction is accomplished by transmitting control packets which floods all the network. These control packets are propagated node by node until one of them reaches the destination. This one replies with another message that follows the same route, but in the opposite direction. This backward propagation confirms the availability of the route in both directions, and establishes the communication between the source and the destination. If the destination receives multiple discovery control packets, this one can decide to reply the one that considers the best. Despite this approach may be suitable for sporadic communications, if the route is broken during the communications, the source has to execute again the discovery phase. Therefore, the AODV presents large reaction times against unpredictable route variations.

The Ad-hoc On-demand Multi-path Distance Vector (AOMDV) routing protocol (Marina and Das, 2006) extends the original AODV protocol to cope with this situation. The protocol leverages on the flooding mechanism performed in the discovery phase to determine multiple routes between a source and a destination. In this case, the destination replies to all—or an ensemble—the discovery control packets which constitute the different available routes. The source selects one of those routes as a primary one, and keeps the remaining stored as alternatives. Therefore, if the primary route fails, the source can quickly use one of the alternative routes. Although these alternative routes improve the reaction against network changes, these routes may be obsolete after a lapse of time. The source triggers the discovery phase if this condition happens. As previously seen in the MLSN, the feature of having multiple routes benefits the robustness of the network, and improves the stability of the communications.

This reactive or on-demand strategy is suitable for sporadic and opportunistic communications, without requiring to exchange constantly control packets and wasting node resources. Satellites resources are valuable commodities that need to be properly managed to not jeopardize the mission. Therefore, some researchers tried to apply this reactive mechanism to satellite networks. The Location-Assisted On-demand Routing (LAOR) protocol for satellite communications was proposed in (Papapetrou et al., 2007). This solution benefits from the deterministic movement of satellites to bound the flooding of the discovery control packets. Therefore, the LAOR executes a preliminary phase before the discovery one, called the request area formation, which defines the possible interest zone to flood. The presented results suggest that in a high load LEO satellite network, this protocol can provide less end-to-end delay than a predictable centralized routing protocol. This approach demonstrated that the integration of satellite determinism in MANET routing protocol may result to a powerful combination that improves traditional satellite protocols.

Proactive routing protocols

Reactive protocols perfectly suit for sporadic and short data transmissions. However, another kind of strategy needs to be conceived for a more continuous time-sensitive data flow. Proactive routing protocols were conceived to ensure that when the data must be transmitted the source node is aware of all the available routes. Therefore, no discovery phase is designed in these protocols. Instead, a periodic update of the network topology is required to anticipate the network changes. The Optimized Link State Routing (OLSR) protocol (Clausen and Jacquet, 2003) may be the largest proactive protocol used in MANET. The nodes that execute this protocol intentionally exchange control packets that include the status of all the links in the network. This exchange is optimized by selection specific nodes that performs this propagation. Every node in the network identifies which node is its Multipoint Relay (MPR). This MPR node is responsible of flooding the control packets which notify about topology link status. OLSR reduces the signaling overhead by limiting this flooding among only the MPR nodes. This protocol is capable to quickly detect any topology change, and react accordingly. Each node using this protocol is available of the status of the entire network, which may require certain resources if the network becomes large. Other approaches addressed this situation by optimizing this information to just the useful one.

The Destination-Sequenced Distance-Vector (DSDV) routing protocol (Perkins and Bhagwat, 1994) proposed the discovery of a route between a source and a destination by exchanging control packets between direct neighbors. Each node has just local information about its neighbors with which it can estimate the global topology of the network. Specifically, each node knows the destinations that can be reachable by forwarding data to each of its neighbors. According to this information, the node defines a route node by node. Since in this kind of protocol nodes do not have a global view of the topology, the routing loop problem can appear. This problem appears when the path to a destination includes a close-loop, which makes that the destination is never reached. DSDV addresses this situation by using a sequence number for each destination entry in the routing table. This sequence number identifies the creation time, which allows nodes to verify if the received information is new. If it is the case, then the entry is updated; if not, the information is rejected. This mechanism provides enough means to identify routes in MANET while keeping just reduced local information.

Following a similar premise, the Better Approach To Mobile Ad-hoc Networking (BATMAN) protocol (Neumann et al., 2008) was presented to extend the OLSR with local information, instead of the global view of the network. This approach divides the knowledge of the best end-to-end route between the nodes that conforms the network. Each node perceives and maintains only the information about the best next hop towards all other nodes. This information is constructed by broadcasting Originator Messages (OGM). These OGM are generated by each node to notify its neighbors about its existence. These messages are suddenly re-broadcasted according to specific rules to inform their neighbors about the existence of the original initiator of this message. Each node uses the reception of multiple OGM copies from the same initiator to identify alternative routes, and selects the best one. Those irrelevant messages are then discarded to avoid wasting the network. This process is repeated node by node. The BatMan eXperimental version 6 (BMX6) emerged as an independent development from BATMAN routing protocol to explore and test new approaches for routing and context dissemination (Neumann et al., 2012). Its goal was to reduce the signaling overhead achieved by the use of Internet Protocol version 6 (IPv6) addresses, enable node-individual configurations while clarifying the handling of conflicting node announcements (e.g. duplicate address allocations), and allow efficient state dissemination (thus reduced protocol overhead) through the strict distinction between local and global as well as static and dynamic states. Thanks to these characteristics, the use of BMX6 increased in wireless networks compounded of static and mobile nodes.

The integration of these proactive routing protocols in satellite missions were investigated in the last years. The satellite platforms were identified as potential nodes that could improve the dissemination of the routing tables of a ground network (Giraldo Rodriguez et al., 2010). This support infrastructure follows the premise of MLSN in which high altitudes enhances the visibility of the satellites, and thus are ideal to broadcast messages to a large number of nodes. Nevertheless, the satellites are not considered as part of the network to relay data. The possibility to use the OLSR as a routing protocol to conform the network that deploy federations between remote satellites was discussed in (Lluch et al., 2015). The results demonstrated that this protocol improves the delivery of the data than only using monolithic satellites. Nevertheless, the OLSR was not conceived to follow the sporadic and opportunistic nature of federations. Therefore, satellites that do not expect to participate in the federation must still exchange control packets, wasting its resources. Despite this non-optimal solution, this proposal seems promising by using well-known technology into the heterogeneous context of federations, deeper investigations must be performed in the future.

Hybrid routing protocols

Among the reactive and proactive routing protocols, other developments encouraged the definition of mechanisms that merge features of both approaches. These hybrid solutions are characterized to have the discovery phase of reactive protocols, and the maintenance mechanism of proactive ones. This is the case of the Zone Routing Protocol (ZRP) (Haas et al., 2002) which estimates a zone radius to limit and reduce the signaling overhead. This boundary zone determines the nodes that follow a proactive approach by periodically exchanging the routing tables. The nodes that interacts with other nodes located outside of the zone follow a reactive mechanism to determine the route. It suits for those scenarios that continuous communications among nodes that remain close is required, and sporadically interaction with external nodes are performed. Nevertheless, the ZRP defines a unique zone in the network located in a specific place. This definition limits the flexibility of the protocol to include other relevant zones.

The Independent ZRP (IZRP) (Samar et al., 2004) copes with this limitation by computing a specific proactive zone for each node. Thus, each node keeps a proactive communication with its closer nodes and a reactive one with far nodes. This solution becomes more adaptable and flexible to the needs of each node, however the generation of all the zones in a large satellite system may become a challenge. The Fish-eye State Routing (FSR) protocol (Guangyu et al., 2000) proposes the definition of different quality zones for each node to be applicable in large scale network. In particular, each node defines different zones in which the resolution of link state information is reduced proportionally to the zone radius. Each zone has different accuracy of the node status, being the information more accurate in closer zones rather than further ones (like a fish-eye). This technique allows to keep the entire proactive strategy by reducing protocol signaling. However, it considers that the traffic is uniformly distributed over the entire network.

The Two-ZRP (TZRP) proposal (Wang and Olariu, 2004) combines both concepts of IZRP and FSR to provide a unique solution that is able to manage high-mobility scenarios. In particular, each node defines the Crisp Zone and the Fuzzy Zone. Inside the former one, topology updates follow a proactive mechanism as IZRP. However, in the Fuzzy Zone the fish-eye technique is used, reducing thus the accuracy of the network changes (i.e. a vague image). Outside this zone, the communication is purely reactive. These hybrids solutions based on zone becomes a powerful strategy for satellite systems based on flight formation, like satellite clusters or swarms. This differentiation between close and far nodes enables to separate the communications necessity for each node. The nodes that are closer may require fast reaction against node status, because the interaction with them is more probable. Meanwhile, further satellites could accept on-demand discoveries for sporadic interactions. Despite this potential benefits, no hybrid routing protocol has been applied in satellite missions.

1.3.9 Wireless sensor networks in satellites

A Wireless Sensor Network (WSN) addresses a specific scenario in which wireless nodes, that senses the environment, are interconnected to establish an autonomous network. These sensors are minimally designed to accomplish their sensing function, and consequently they are resource-constrained. The deployment of these sensors as a network may suppose a challenge in terms of energy consumption. Therefore, different routing protocol strategies to minimize this consumption have been discussed in the last years.

A hierarchical architecture was proposed to define smart structures—known as clusters—that optimize the global network residual energy. This approach promotes that limited energy nodes forward data to a more capable node that aggregates incoming data, and performs a more costly transmission. The selection of this centralized node—called cluster head—that aggregates the traffic has been the aim of recent investigations. The Low-Energy Adaptive Clustering Hierarchy (LEACH) routing protocol (Heinzelman et al., 2000) promotes the selection of this cluster-head randomly following a uniform distribution. This selection process would be conducted periodically to ensure that the consumption associated to act as the cluster-head is shared among the nodes that conform the cluster. The implementation of this approach is centered on each node publishes the interest to become a cluster-head, and neighbors autonomously decide to communicate with it. Despite this solution ensures that not only a single node is drained, the undesired situation in which a node with not enough energy emerges as a cluster-head is still feasible. In order to address this issue, the Hybrid Energy Efficient Distributed Clustering (HEED) routing protocol (Younis and Fahmy, 2004) uses the energy state and data rate values to decide the cluster-head. Each node computes the probability to become the cluster-head analyzing its own state. If the situation is appropriate, the node publishes the intention to become the cluster-head. This enhancement reduces signaling overhead as well as fairly distributes cluster-head across the network that increases the network lifetime. The HEED protocol defines two kinds of communications: intra-cluster and inter-cluster. The consumption source analysis is presented in (Senouci et al., 2012), concluding that the transmission distance is an important factor of the high consumption. Therefore, the authors propose the Extended HEED (EHEED) which enables the communications between non-cluster-head nodes if, and only if, the transmission is less energy costly. This situation extends the single-hop communication with the cluster-head to a multi-hop scenario.

The benefits of energy-efficient with hierarchical topologies motivated the Internet Engineering Task Force (IETF) to define the Routing Protocol for Low Power and Lossy Networks (RPL) (Winter, 2012) standard. This distance vector routing protocol has the capability to autonomously define a hierarchical structure whose information is distributed to each node. The root of the structure— the main node—proactively maintains the network topology knowledge by exchanging downward/upward control messages. Using this periodic signaling, this protocol has the capability to join new nodes into the structure. The construction of this hierarchical structure is based on a node rank that identifies each node according to the depth level in the structure tree. During the construction, each node determines its preferred parent to forward packets. Furthermore, the standard decouples the route selection mechanism from the routing protocol core, which provides flexibility for heterogeneous scenarios. In particular, the standard defines the Objective Function (OF) as a way to compute the rank and the parent by combining network metrics and constraints.

The RPL standard does not define a specific OF, indeed it promotes the exploration of the differences between metric, constraint, and selection criteria in the OF concept. One of the first proposals is the Objective Function Zero (OF0) standard (Thubert, 2012) which proposes a basic and common mechanism to unify the computation of Rank value. However, additional OF have been conceived, such as the Minimum Rank with Hysteresis Objective Function (MRHOF) case (Gnawali, 2012) which selects a route that minimizes an additive metric using hysteresis to reduce the impact of small metric changes. Both examples do not restrict the use of a specific and unique metric. However, the first metric proposition was the Hop-Count (HC), which favors the path of fewer—but longer—hops. Other link metrics have been evaluated, like the Average Delay (AD) metric (Gonizzi et al., 2013) which uses the end-to-end route delay to compute the rank value. This approach enables to reduce the communication latency, although the performance may vary depending on the link quality. These previous metrics cannot manager network congestion. This situation is addressed by the Queue Utilization based RPL (QU-RPL) (Kim et al., 2015) which enhances an implementation of the RPL with MRHOF and HC metrics to perform load balancing. Specifically, the used metrics are extended with the queue utilization, which can be used to predict local congestion. This protocol is able to define the less congested parent in a set of possible candidates. The OF definition also includes the possibility to specify constraints of a route, providing another level of decision. The Expected Transmission Count (ETX) (Vasseur et al., 2012) is an example of this formulation, which determines the expected number of transmissions to reach the destination. This parameter can be considered as a metric, but it is also related to maximum number of transmissions that can be accepted (constraint). Treating constraints and metrics differently allows selecting a candidate not only by its qualities, but also the one that respects certain conditions.

Using a single metric may not be enough to accomplish the desired performance in some cases. Therefore, the Improved RPL (IRPL) (Wang et al., 2017) proposes the Life Cycle Index (LCI) as a novel OF definition that aggregates multiple metrics (e.g. link quality, node energy, success transmission data rate, and congestion detection factor.) Despite the complexity of the algorithm increases, the results indicate that the combination of multiple metrics allows improving the performance of the network in different aspect, such as reducing end-to-end latency while ensuring a minimal network residual energy.

The integration of the OF definition in satellite networks may promote the use of different metrics based on resource status (e.g. power state, memory usage, etc.) Nevertheless, the RPL standard does not manage mobile nodes which can limit protocol effectiveness. A solution for the mobility of specific nodes was proposed in (Safdar et al., 2012). A hybrid protocol based on reactive discovery limited by broadcast zones—like the ZRP—and the RPL maintenance network is proposed to address mobility following the resource-aware approach. This solution intelligently combines mechanisms from MANET and WSN to improve routing performance. This is not the unique case that MANET solutions have been used to address energy challenge in WSN (Jabbar et al., 2016). In particular, location-based routing protocols provide information related to the node position. This information can be used to predict transmission consumption using predefined energy model. This is the case of Geographical Adaptive Fidelity (GAF) routing protocol (Inagaki and Ishihara, 2009) which uses an internal energy model to define cells in which a single master is active. The other nodes that are in the same cell are turned off, avoiding unnecessary consumption. Another approach is the Geographic and Energy Aware Routing (GEAR) routing protocol (Yu et al., 2001) which uses the combination of geographical position and energy level of neighbors to define the next hop. Using both information, it can manage the residual network energy and thus improve network lifetime.

Similarly to the last proposals, the Kalman Positioning RPL (KP-RPL) (Barcelo et al., 2016) extends the RPL standard using position information. In particular, it computes node position combining the measure of the Receive Signal Strength Indicator (RSSI) and the Kalman filter to refine it. However, this protocol promotes the communication between static nodes (called anchors) and mobile ones, instead of only between mobile nodes. Although further research needs still to be carried out to address the situation of a full-mobile topology, using node position is a powerful information to predict energy consumption.

In this regard, sensors in WSN share some similarities with small-satellite platforms. Both systems have limited processing and energy resources, and have communications capabilities. Therefore, the developed protocols for WSN that consider the resources status as a key parameter in their design may be suitable for satellite systems. Despite these similarities, satellite networks are normally characterized by larger communications distances than in the sensor case. Therefore, Song et al. target the use of this technology mainly for cluster and swarm satellite systems. The proposed satellite sensor network (Song et al., 2015) defines a central satellite that interact with pico- or femto-satellites using wireless sensor network techniques. The in-orbit demonstration of this technology is still pending.

1.3.10 Operative missions with satellite-to-satellite communications

Among the numerous satellite missions, few deployed an interconnected satellite system that would represent one of the previous satellite networks. The most relevant ones seem to be those that integrated two-layer satellite networks. These reduced MLSN are typically compounded of satellites located on the GEO region that work as data relay to lower-altitude satellites, such as in LEO or MEO regions. The National Aeronautics and Space Administration (NASA) started—at the early 1980s—the Tracking Data Relay Satellite System (TDRSS) project which deployed this two-layer architecture to extend the operations coverage of other NASA satellites (Teles et al., 1995). This project was conceived in replace to—or support of—the ground segment that NASA was using. Nowadays, the TDRSS ensemble ten GEO satellites to provide near continuous information relay service to missions like the Hubble Space Telescope or the International Space Station (ISS). The TDRSS satellites—called directly TDRS—experienced three design generations that changed their capabilities. The first generation was equipped two ISL devices that work at S-, and Ku-bands[2], and they are integrated with a reflector antenna, and multi-element antenna. The second generation was designed with the same ISL devices of its predecessor—with an improvement on the performance—, and additional device that works at Ka-band[3]. Finally, the third generation was equipped with the same ISL devices, but it integrated the capability to perform antenna pattern beamforming. All these satellites are operational and enable NASA missions to use RF ISL to extend its operations coverage.

The European Space Agency (ESA) also developed its own program to deploy a two-layer satellite system which shares certain similarities with the TDRSS. This project was called the European Data Relay System (EDRS), and two satellites—named EDRS A, and EDRS C—currently compound the system (Heine et al., 2014). Both satellites are equipped with an Optical ISL (O-ISL) device that is used to communicate with the lower-altitude satellites. Additionally, EDRS A has a RF ISL device at Ka-band. This EDRS mission is one of the first missions that use optical terminals to establish communications interfaces between satellites. Nevertheless, this kind of technology has started to be investigated in reduced satellite platforms, such as the CubeSat standard. FSSCat mission (Camps et al., 2018) integrates a technology demonstrator of this O-ISL that would enable to communicate between two 6U CubeSats (Briatore et al., 2017). Returning to the two-layer system proposals, the different ISL devices enable to communicate two satellites that are considerably separated—around 35000 km. This is achieved thanks of using directive antennas or terminals, which enhances the Equivalent Isotropical Radiated Power (EIRP).

Another remarkable satellite network which is currently operational is the Iridium satellite constellation. Already introduced in the previous sections, this satellite constellation follows the traditional LEO satellite network architecture, characterized by a well-orchestrated and fixed constellation. The former Iridium constellation was deployed between 1997 and 2002, and it was composed by satellites that had RF ISL devices working at Ka-band. A second generation

[2]The S-band comprises the frequency band from 2 GHz to 4 GHz; meanwhile, the Ku-band comprises the frequency band from 12 GHz to 18 GHz.

[3]The Ka-band comprises the frequency band from 26.5 GHz to 40 GHz.

of the satellite constellation, called Iridium NEXT, was launched in 2017. This generation mainly reviewed the design of the satellites to achieve greater user data rates, and to include other hosted payloads. Nevertheless, the ISL devices remained at Ka-band. Iridium NEXT constellation currently offers communications data links up to 134 kbps for mobile and static terminals, which may be suitable for IoT applications. This performance metric, however, motivated the emergence of other architectures that could provide larger data capacity.

Mega-constellations envision to offer seamless Internet coverage on the entire globe. Therefore, they integrate hundreds or thousands of satellites in one, two or several layers. The Starlink constellation is one of those proposals that aims to compete for this market. The 540 satellites that are currently orbiting represent a primary deployment to reach the envisioned 12,000 satellites of the final constellation. Despite the in-orbit satellites do not have ISL devices, the original design included an O-ISL terminal that would enable to achieve high capacity interfaces (Exploration, 2018). Current satellites, however, use a phased array antenna that performs beamforming at Ku- and Ka-bands to establish links with the ground terminals. Another mega-constellation that promotes the use of O-ISL is the Telesat constellation. The satellites are designed to include this kind of optical terminal to also provide high capacity. The former satellite was launched to demonstrate the communications interface between satellites and ground terminals (Russell, 2018). Although the characteristics of those optical terminals remain privative, other researches demonstrated that these terminals may reach data rates up to 5 Gbps with link distances from 1000 km to 5100 km between LEO satellites (Gregory et al., 2012).

The Tandem-X mission (Geyer et al., 2010) proposed the use of two satellites that flight following a train formation. These satellites are equipped with a S-band transmission system. This system is primary designed to perform the downlink and uplink interaction with the ground stations, but it is also used for satellite-to-satellite interactions. This RF ISL is operated at low-power mode enabling a maximum communications range of 9 km, and a data rate of 31.25 kbps (Tiainen, 2017). The GOMX-4 mission also deployed two satellites—in this case, two CubeSats—that conforms a tandem or train. The in-orbit results (Léon et al., 2018) show that communications were established at a distance of 500 km, although further ranges can be achieved—up to 4500 km—combining multiple modulations and different transmission power profiles. Furthermore, the link is established at S-band, and it is able to provide from 2.4 kbps to 1.25 Mbps as transmission rate, depending on the distance between satellites.

Other missions have also integrated a satellite-to-satellite interface for formation flying. The PRISMA mission (Sun et al., 2010) as a technology demonstration for autonomous rendezvous technologies in which two satellites were deployed. Two RF ISL were established working at Ultra High Frequency (UHF)[4] and S-bands. The maximum achieved distance was 30 km when the ISL were set at high-power mode, and providing 12 kbps of data rate. Can-X satellite mission (Bonin et al., 2015) is another formation flying mission with two CubeSats that integrated a S-band ISL device. This link was established with two patch antennas, and achieved a maximum distance of 100km, and a data rate of 10 kbps.

In summary, numerous communications devices have been designed and implemented in the last years. Some of them were used to establish ISL between satellites, presenting different performance. Among the different characteristics, a model of a generic ISL can be defined by three main parameters: the data rate, the maximum communications range, and the directivity. Distinct data rates are achieved depending on each ISL technology is used. O-ISL can reach communications up to 5 Gbps, while RF ISL may work between the order of kbps to Mbps. Further-

[4]The UHF band comprises the frequency band from 300 MHz to 1 GHz.

more, both RF and O-ISL can achieve large distances depending on their terminal features. In LEO and GEO communications the devices can reach up to 35000km, while in LEO and LEO communications the distance can vary from 10 km to 5000 km. O-ISL, however, are characterized by having farther communications ranges and higher data rates, than RF-ISL devices. Nevertheless, they are more directive with narrow beams. Despite some missions presented systems with omnidirectional antennas, the predominant antenna is directive, which require accurate pointing. Furthermore, directive antennas may difficult the connectivity in the network, which degenerates to more network disruptions. The historic of the different devices shows that it exists a tradeoff between maximum communications range, data rate, and connectivity. So far, a communications system that provides high connectivity, high data rate, and large communications range has not been developed.

1.3.11 Satellite network simulation engines

Designers and researchers are faced with the need to propose and verify new communications protocols that enable the deployment of previous satellite network approaches. The novel DSS and FSS concepts require also a certain infrastructure that facilitates the development of federated negotiation and decision-making algorithms. The validation of these software technologies usually entails the simulation of systems and their environment, and the generation of metrics to assess their quality. In the most simplistic case, satellite networks could be modeled as simple nodes that orbit the Earth, ignoring their functions (i.e. sample the surface or atmosphere of the Earth) and reducing their representation to communication devices. Many network simulation frameworks could be used in this pursue, which facilitate the modeling of moving devices (such as those in MANET). However, the de facto tools used by the industry and academia to simulate networks (e.g. QualNet[5], OMNeT++[6], NS-2/3[7]) are tailored to ground applications and ignore scenarios where the communication devices are complex systems per se (i.e. with a different function other than implementing communications).

Network simulators often provide a good collection of assets to model and implement custom protocols, signal propagation models, and physical devices like transceivers and antennae, but they hardly allow the implementation of non-network components (e.g. a battery) or other environmental and state variables that have direct impact upon links and operations. Examples of the latter would be the Earth and its atmosphere, the state-of-charge of batteries, attitude, pointing of ground station antennae, etc. The literature is certainly abundant with works that tackle resource-aware devices and which rely upon network simulation to demonstrate their designs. However, those works—and the tools they rely upon—usually address WSN, MANET, or other systems in which the complexity of the modeled devices and the environment is not comparable to that of a satellite nor a DSS. Likewise, none of the standard network simulation platforms include mobility models for objects that orbit around the Earth or are constrained by line-of-sight effects.

When spacecraft state variables (e.g. orbital position, recorder capacity, battery state-of-charge) need be evaluated in simulation, aerospace practitioners often resort to dedicated software tools that are tailored for space systems and mission analysis. Some of the most common alternatives in this domain are AGI's Satellite Tool Kit (STK), or NASA's General Mission Analysis Tool (GMAT). These software frameworks provide a wide range of orbit propagation models (analytic/high fidelity like J2 or SGP4, semi-analytic/medium fidelity, numerical/high fidelity

[5]https://www.scalable-networks.com/qualnet-network-simulation
[6]https://www.omnetpp.org/
[7]https://www.nsnam.org/

like HPOP), and an extensive set of built-in tools designed to carry out integral mission analysis simulations (e.g. thermal, schedule of operations, instrument modeling, etc.) Due to their broad adoption in multiple contexts, these simulators have sometimes been paired with some of the network simulation tools mentioned above in order to produce software ecosystems with both functionalities; i.e. the simulation of space systems and networks.

The combination of these two components has been implemented in different architectural approaches. A group of researchers at NASA discussed the possibility to bind STK and QualNet simulators to allow them to run in parallel and in a synchronized manner (Barritt et al., 2010; Wong et al., 2017). This approach provides great simulation accuracy given that each simulation tool receives intermediate outputs from the other and mutually complement their emulation of functional and physical variables. Llatser et al. (2017) also followed a similar approach in the implementation of a simulator for a similar domain: an autonomous fleet of networked vehicles. The major drawback of this kind of architectural approach is the inefficiency that draws from synchronizing two independent software toolboxes that have not been designed to run in an integrated manner. The emulation of large-scale systems and/or the need to analyze a system behavior at long-term (e.g. weeks, months) could turn this lack of computational performance into a prohibitive cost for designers.

A second alternative is the sequential execution of aerospace and network simulation tools. This kind of architectures, implemented in (Baranyai et al., 2005; Lluch et al., 2015), run several instances of physics or aerospace simulators (STK, and others) to produce inputs that will be used in subsequent executions of a network-level simulation tool (e.g. NS-3, QualNet). Despite ameliorating the computational performance, the main limitation in these approaches is the total inability to simulate the impact of network or operational processes upon physical variables, since they have been computed previously (e.g. battery state-of-charge, memory usage, etc.) Generally, network simulation engines are event-based, emulate variables in extremely small time scales (down to nano- or pico-seconds) and are asynchronous by nature; i.e. state variables are not advanced for time points where network components are idle. Conversely, physical and aerospace variables could require numerical methods, and often rely upon time-discrete models. When the output of the latter is used in network simulators, most of the state variables need to be interpolated to obtain their corresponding values at the times triggered by network events. Provided that the integrated simulation of spacecraft components (e.g. battery use) is not the focus of the study and the loss in accuracy due to interpolation can be assumed, the limitations of this approach could be deemed acceptable. Grogan et al. (2014) developed a simulation engine to characterize the stakeholder decisions or operations in the deployment of satellite federations. The presented approach founds on abstracting the satellites in models that represent the roles in the federation, resources, orbit characteristics, among others. Nevertheless, the lack of simulating a realistic protocol stack architecture complicates the development of novel algorithms for this context. **The possibility to emulate autonomous operations or decisions—like in FSS context—considering the on-board resources, and the exchange of information among satellite nodes, precludes the adoption of architectural approaches wherein the two domains are not unified**.

The third and most relevant alternative is the development of fully-integrated tools that comprise the simulation of aerospace components and variables, and communications, network devices and protocols. This concept was explored in (Merts and Barnard, 2016), who presented a simulation framework that relied upon Python's SimPy network simulation library. Similarly, the Satellite Network Simulator 3 was proposed in (Puttonen et al., 2015) as an extension to NS-3 that included an orbit propagator and the definition of satellites as nodes in the network.

Although their design did not consider additional spacecraft components than communications systems, this work proved the feasibility of developing integrated tools that tackle the simulation of networked space systems. Another example with similar characteristics is found in (Niehoefer et al., 2013), which presented an Open Source Satellite Simulator implemented in OMNeT++.

1.3.12 Key findings

The list below summarizes the most critical aspects found in literature with regards to satellite motion, satellite networks, and satellite federations, and is aimed at bridging the state-of-the-art of this field.

- Research has emphasized overtly the need of satellite services that could satisfy current user demands that proliferated from the incursion of Internet services, and the continuous evolution of the Earth environment. The numerous demands can be summarized in two main system requirements: (1) **increase of data transfer capacity**, and (2) **reduce the end-to-end communications latency**.
- The value of FSS that satisfy these demands has been explored extensively in literature, where authors have mostly focused on in-orbit data services. This emphasizes the value of **inter-satellite links and networked constellation designs**, which could also improve some of the performance metrics in new designs.
- The challenges to design a proper satellite network gravitate around the disruption of the network caused by satellite motion. Delay-tolerant solutions cannot satisfy current low-latency demands, while custom-architecture-based proposals do not suit the sporadic and opportunistic nature of FSS. **The development of another satellite network perspective may be fruitful**.
- The **value of heterogeneous satellite architectures** has been emphasized in the previous studies. These architectures can provide further transfer capacity, and better manage the stability of the network by leveraging on distinct altitudes.
- The network disruption is a phenomenon that will always characterize satellite networks. Other mechanisms than custom architectures may be conceived to **mitigate the disruption of the network**:
  - (i) The development of satellite-to-satellite devices that promotes the omnidirectional connectivity with far communications ranges, while provide high transmission data rates.
  - (ii) The development of algorithms that enables the prediction of satellite contacts by the same satellites, avoiding the centralized and ground-dependent approach that limits the flexibility of the system.
- The quantification and analysis of satellite system performance has been discussed with the integration of different simulation engines. The **development of a unified simulation framework** that allows integrating (1) satellite resources, (2) satellite subsystem behavior, (3) satellite operations and decisions, and (3) inter-satellite communications would promote the design of new approaches.
- Research has highlighted the need to **formally define communications protocols** that allow the establishment of collaborations or federations between satellites.

This corresponds to solutions which enable the notification of available resources, the in-flight negotiation of these resources, the verification of the resource consumption, and the route definition with decision-making satellites.

- The previous studies on FSS emphasized the potential benefits of this satellite system, and the market size that would be interested on deploying federations. This outcome motivates the **development of experiments or proof-of-concepts** that demonstrate the feasibility of this approach, and pave the future investigations to enhance the achieved performance.

1.4 Thesis statement

The compliance of current user demands entails multiple open challenges that are being explored by the research community from many different perspectives. The DSS and FSS emerged as potential satellite architectures to achieve the required performance by envisioning low-latency communications, high downlink capacity and fast access to the generated data from space. These new systems are expected to provide in-orbit data services, enhance global temporal resolutions, minimize costs and risk, and deliver a number of system-level qualities. Several research programmes worldwide have already revealed that satellite federations could have an essential societal impact and revolutionize the traditional space concept. Nevertheless, many technologies still need to mature in order to enable novel FSS concepts, emphasizing the ones associated to satellite-to-satellite communications. This dissertation gravitates on these potential architectures, and contributes to the development of fundamental communications technologies to promote their future inclusion in satellite missions.

1.4.1 Research questions and objectives

In this context, FSS still present unanswered questions to which this thesis aims to contribute: (1) What is the nature of end-to-end routes between remote satellites in sporadic and opportunistic satellite networks, and how can they be defined? (2) How can satellites be aware of the available resources offered by other satellites? and (3) what are the needed mechanisms to deploy a federation in this opportunistic context? Under these questions, we elaborate the following list of objectives of this doctoral research:

(O.1) Analyze the fundamental characteristics of satellite networks to support sporadic and opportunistic collaborations.

(O.2) Propose and formalize the networking environment for this sporadic and opportunistic interactions to become the basis over which develop future technologies.

(O.3) Contribute to the development of a simulation framework that triggers the understanding of satellite networks by means of observing specific and relevant qualities.

(O.4) Conceive the fundamental communications protocols that support the establishment of federations in a satellite network.

(O.5) Quantify the impact of the developed protocols to contribute in satellite federations based on experimental results.

The review of the literature and the formulation of the different goals of this research have emphasized the necessity to define a paradigm in which satellite networks would be created following the sporadic and opportunistic nature of FSS. The Internet of Satellites (IoSat) paradigm (Ruiz-de-Azua et al., 2018c) presents this dynamic environment in which Inter-Satellite Networks (ISN) are created sporadically and maintained during the execution of the federations. This dissertation formulates the description of this paradigm by presenting the details of these new satellite networks. Furthermore, the research performed focused on the development of this paradigm and the corresponding protocols which constitutes the necessary communications means to deploy federations.

1.4.2 Structure

Chapter 1 has introduced the context and presented the objectives of this dissertation. We defined the types of satellite systems of interest, and delimited the applications in the context of the current environmental, societal, industrial, and humanitarian needs. We categorized satellite networks archetypes and described some of the most common structural and functional traits. Section 1.3 has structured the body of knowledge centered on presenting the difficulties to communicate in satellite networks, and the numerous solutions developed for each network type. Furthermore, the review has been complemented with a global view of different missions that have included satellite-to-satellite communications, and a discussion of current simulation technologies for these networks. Section 1.3.12 completed the literature review by summarizing the most important aspects in relation to the context and objectives of this thesis. The remaining of this document is structured in three parts: (1) a theoretical definition of the IoSat concept, and a preliminary study on its challenges; (2) the conception of three protocols that contribute in the development of the paradigm; and (3) the implementation and experimentation of the previous contributions in a stratospheric balloon campaign, and a CubeSat mission. Figure 1.11 aims at clarifying the structure of the entire dissertation, before presenting the details of each chapter.

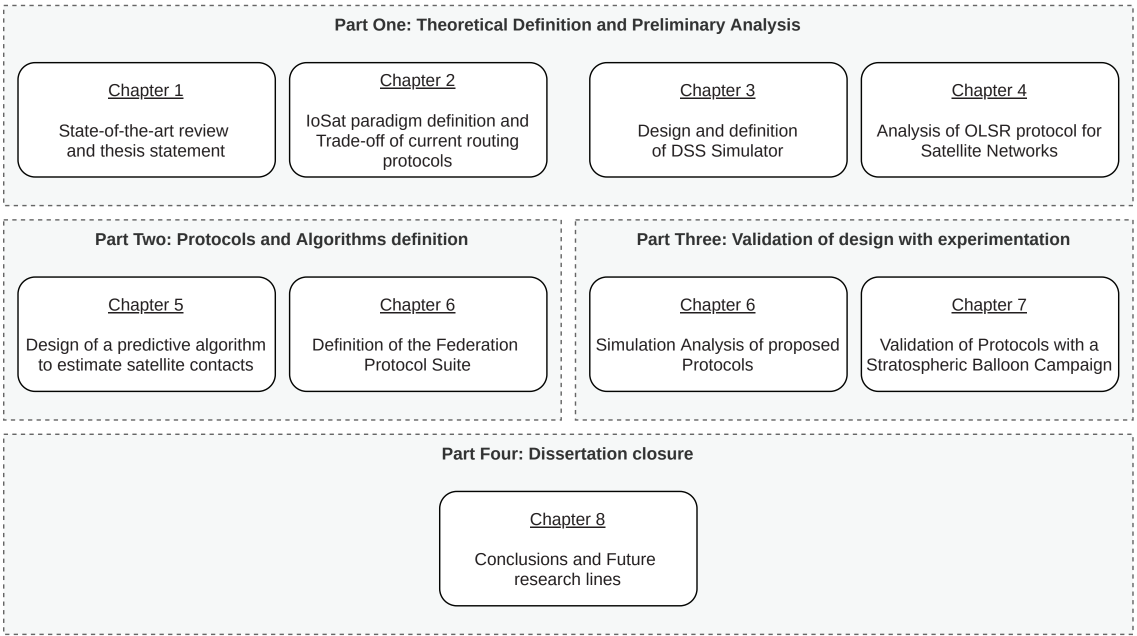


Figure 1.11: Block diagram of the dissertation structure

Chapter 2 formally presents the IoSat paradigm, and details the characteristics of the satellite networks that are generated in this context. Section 2.2 summarizes the different features of the paradigm previously detailed. The comparison of the satellite networks deployed in IoSat with respect to the previous types is conducted in Section 2.4. the applicability of the current protocols in IoSat is discussed in Section 2.5, while a formalization of the properties that a routing protocol may have is presented in Section 2.6. The chapter completes by presenting the features that the routing protocol should have to deploy the desired satellite networks, and leads to evaluate the viability of current technologies in a more sophisticated environment.

Chapter 3 focuses on the description of the simulation engine developed specifically to evaluate future contributions in satellite networks and DSS. The motivation and requirements that this software must satisfy are presented in Section 3.2. These requirements indicate the necessity to deploy a software framework that enables to different users configure the scenario to simulate with their own custom modules. Section 3.3 presents the design of this framework by reviewing all the components that conform it. The chapter is completed with a description of how the performance metrics are generated and which tools are available to visualize them.

Chapter 4 discusses the viability of applying MANET technologies in satellite networks. This discussion is motivated with the simulation of the OLSR protocol in a satellite system. The chapter is initiated in Section 4.2 with a detailed explanation of the protocol. The simulation scenario is detailed in Section 4.3, emphasizing on the components that model satellites, payloads, and communication protocol. The outcome is discussed in Section 4.4, suggesting that current technologies cannot deal with the opportunistic and sporadic nature of ISN. These technologies need to be extended with complementary capabilities developed for other satellite networks, such as predictive algorithms. Chapter 5 goes deeper in this premise and presents an algorithm to predict satellite contacts following a de-centralized approach. Section 5.2 presents an overview of the prediction protocol in which a predictor model is able to estimate future inter-satellite contacts with a single mathematical expression. Different predictors are forumlated in the following sections following distinct strategies. Section 5.3 presents a predictor based on a probabilistic method to estimate with two satellite could experience a closeness situation. Alternatively, Section 5.4 defines another predictor model based on relative orbital motion, which presents better performance. The comparison between the predictors is performed with the performance metrics that are presented in Section 5.5. The chapter is completed showing the performance achieved of the best predictor in a realistic scenario with different satellite systems.

Among the work related to the routing protocol, other questions remain unanswered regarding the establishment of satellite federations. Chapter 6 gravitates on these questions and presents the design of the federation protocol suite. This suite is composed of two protocols that provide the necessary means to establish federations. Specifically, the Opportunistic Service Availability Dissemination Protocol (OSADP) allows notifying the services that are available in a satellite, while the Federation Deployment Control Protocol (FeDeCoP) formalizes the different rules to establish and manage a satellite federation. Both protocols' design are presented in Section 6.2 and Section 6.3 respectively. These protocols are evaluated then in a realistic scenario which is modeled in Section 6.4. The chapter concludes discussing the improvement of the mission performance by applying the federation protocol suite. These results encourage to develop a specific system that materializes them in a device that can deploy satellite federations. Chapter 7 presents the first prototype of this device, called Federated Satellite Systems Experiment (FSSExp) payload. The details of its architecture are introduced in Section 7.4, emphasizing on the RF-ISL board which provides the necessary means to interface—at least—two FSSExp payloads. This payload is evaluated in a stratospheric balloon campaign, detailed in

Section 7.7. The chapter completes with a discussion of the results retrieved from this campaign, which demonstrate the benefits of the payloads, and future improvements.

Chapter 8 concludes by summarizing the main scientific contributions of this doctoral research, and suggesting future strands of work (Section 8.3). The relevant publications associated to this dissertation are listed in Chapter 9, which also include additional publications during the doctorate.

2

The Internet of Satellites paradigm

2.1 Introduction

The current user demands have been presented in the previous chapter, emphasizing the need to develop new services that provide a considerable data transfer capacity with important downlink, uplink, and ISL interfaces; a communications interface with reduced end-to-end latency for time-sensitive data; and a fast access the data generated by the spacecraft. The potential of distributed satellite architectures has been discussed, highlighting the benefits of using FSS in current satellite missions. These new approaches found on the value of inter-satellite communications capabilities and networked constellations designs to achieve the desired performance. These networked architectures suffer the impact of satellite motion, which provoke the fragmentation and isolation of satellite groups. Numerous satellite networks have been proposed over the years to deal with this challenge. Some of those proposals were defined to coexist with this disruption, and provide non-optimal solutions with delay-tolerant services that satisfy specific demand. Other approaches motivated the deployment of custom satellite architectures that would mitigate or eradicate this fragmentation. All these proposals does not assimilate the sporadic, opportunistic, heterogeneous, and decision-making nature of satellite federations, which may make them not optimal for this purpose. Therefore, a new scenario in which satellite networks respect these four axes must be conceived.

Bhasin and Hayden (2001) proposed the deployment of a satellite infrastructure that supports the current and future NASA satellite missions. This dedicated infrastructure represented a backbone that could support the missions by reducing the cost and standardizing the communications. This always-present backbone seems similar to those satellite networks that deploy custom architectures, and presents a well-known and controlled environment to interconnect satellites. Five years later, Raman et al. (2016) extended the idea of deploying a satellite backbone with the Internet of Space (IoS). As in the previous case, a dedicated infrastructure would be deployed to provide communications interfaces to–in this case–ground devices. The concept was motivated by the emergence of IoT and the possibility to use satellites to enhance ground coverage, which then turned into the different use cases suggested for 5G networks. None of those paradigms presented a dynamic environment that could suit the requirements of FSS, instead dedicated infrastructure–which may become expensive–was suggested. As an alternative, the IoSat paradigm promotes the development of this dynamic environment in which satellites decides to establish temporal networks to deploy federations when a benefit is awarded. Following the architecture of current Internet, the IoSat proposes the interaction of satellites from different systems to create a global network, instead of deploying a custom and massive infrastructure–like mega-constellations. This approach makes satellite systems more accessible than in the other case, which may be driven by administrative, cost, and high density issues. This chapter presents the details of this novel satellite networks, and evaluates if previous technologies could be applied in this context.

The main contribution of the present research is a) the presentation of the IoSat paradigm, 2) a discussion of the synergies between this paradigm and other satellite network proposals, 3) the characterization of the features that determine ISN in this paradigm, 4) the required capabilities that a routing protocol must include to deploy this kind of network, and 5) a study of the viability of current technologies to deploy ISN. The definition of the IoSat paradigm and its features has been previously publicated in the peer-reviewed IEEE Access journal (Ruiz-de-Azua et al., 2018c), and in the 5th Federated and Fractionated Satellite Systems Workshop (Ruiz-de-Azua et al., 2017). Furthermore, the participation in the ONION project to define a technology roadmap for satellite networks has crystallized in the formal definition of this paradigm. This work has been publicated in the peer-reviewed IEEE Access and Acta Astronautica journals (Alarcón et al., 2018; Tonetti et al., 2020), the 70th International Astronautical Congress (Tonetti et al., 2019a), and the Living Planet Symposium (Tonetti et al., 2019b). The details of these publications are accessible in Chapter 9 with the identifiers [I], [II], [V], [VII], [XIV], and [XVII].

The remaining of this chapter is structured as follows. First, Section 2.2 introduces the IoSat paradigms and its characteristics. The essential features of the temporal networks that are created in this paradigm are discussed in Section 2.3. The discussion to evaluate the synergies between this new paradigm and previous satellite networks is performed in Section 2.4. Section 2.5 goes deeper and evaluates if current routing protocols from these satellite networks may be suitable for ISN. After reviewing multiple protocols, the features of a generic routing protocol are summarized in Section 2.6. With these features, an ideal routing protocol for ISN is defined in Section 2.7 which motivated its comparison with current technologies. Finally, Section 2.8 concludes the chapter and proposes new research lines to enhance current FSSExp payload design.

2.2 Paradigm concept

The IoSat paradigm cannot be understood without firstly observe the nature of satellite federations. FSS encourage the establishment of sporadic and opportunistic collaborations to share unallocated resources among heterogeneous satellites. It is important to understand concept-by-concept what this statement means. The **sporadic** term refers to the possibility to deploy this collaboration at any moment. This feature makes satellite federations unpredictable events that may occur without notice, and they cannot–normally–be estimated in advance. Despite this randomness, the need to deploy federations is related to the satellite resources and the potential benefit that a satellite can award. This is related to the **opportunistic** term, which suggests that federations are only established if related satellites envision to garner some benefits (e.g. enhancement of mission performance, extension of satellite capabilities, payment for resources shared). This opportunism also refers to the mandatory non-degradation of the original mission. Satellites that establish a federation are designed to perform a specific mission (e.g. observation of the soil moisture, relay data from ground terminals, observe the galaxy), which must remain as its main priority. Therefore, the federation cannot degrade the performance of this mission by the undesired depletion or allocation of resources. If the satellite does not identify a potential benefit, it must be able to **decide** not establishing a federation. This decision-making capacity becomes crucial to deploy federations, and entails the awarding of certain level of autonomy to the satellites. Finally, the satellites that collaborates are equipped with different resources and capacities. This **heterogeneous** configuration poses multiple challenges related to resource sharing, connectivity, among others.

FSS also differ from the other distributed approaches, because they can be conceived as a virtual satellite systems. These systems represent a group of satellites that are part of distinct physical system–like a constellation–and they decide to create a new one which is fictitious. Traditional applications offered to ground users can be achieved with this systems, but new ones proliferate with this virtual group of satellites. Machine-to-machine applications may be deployed among the different satellites that conform the virtual system, like trajectory applications (e.g. flight formation, collision avoidance), applications that require data fusion (e.g. cloud detection, different instruments), among others. This approach becomes relevant when autonomous capabilities are integrated in DSS (Araguz, 2019). In terms of communications, this can be represented as an autonomous satellite application which deploys some services through and for satellites. All these characteristics require solutions that are flexible, adaptable, and scalable that must be reflected in all the development levels, included in the inter-satellite communications ones.

The IoSat paradigm proposes an interconnected space segment that follows these premises from satellite federations. A custom satellite infrastructure that corresponds to a network backbone is not proposed in this paradigm, unlike the previous cases. Instead, it promotes the establishment of networks using peer-to-peer architectures, in which interested satellites are part of the network. Therefore, the paradigm suggests dynamic, sporadic, and opportunistic satellite networks which are temporally established depending on the necessity and choice to deploy federations. These temporal networks has been called ISN following the traditional nomenclature originated in (Ekici et al., 2002). This kind of network is created by the decision to collaborate–not necessarily for free–of satellites, that become the intermediate nodes of the network. In particular, the creation of an ISN is achieved thanks to the combination of point-to-point federations among intermediate nodes that share the possibility to communicate. Note that in terminology of FSS, an ISN can be considered as a distributed federation in which intermediate nodes play the role of suppliers and customers. Figure 2.1 illustrates the paradigm philosophy by showing three ISNs (ISN_1, ISN_2, and ISN_3) which coexist simultaneously. These ISNs are created depending on the FSS requirements and they adapt themselves to manage network dynamism. Note also that there are some nodes that can participate in multiple ISNs at the same time.

A satellite federation is established only when the transaction is required, after that the federation is no longer needed. This temporality is also reflected in the definition ISN. This corresponds that ISN have three phases that characterize their lifetime: (1) the establishment phase, (2) the maintenance phase, and (3) the destruction phase. The establishment of an ISN is the negotiation process in which intermediate federations are created to configure the network. During this phase, its members can decide not accepting this interaction due to their state or strategy interests. Indeed, a probability that a proposed federation would be accepted or at least negotiated is also taken in consideration in (Grogan et al., 2016a). Moreover, the establishment phase ensures that the ISN is able to satisfy FSS requirements by providing the required services. For instance, if a security level is required, intermediate nodes should have secure mechanisms to provide it. This implies that during the ISN establishment, nodes shall indicate which services they can provide. Once the ISN is established, the maintenance phase ensures that the network adapts to different events. In particular, as a satellite network is a dynamic environment in which

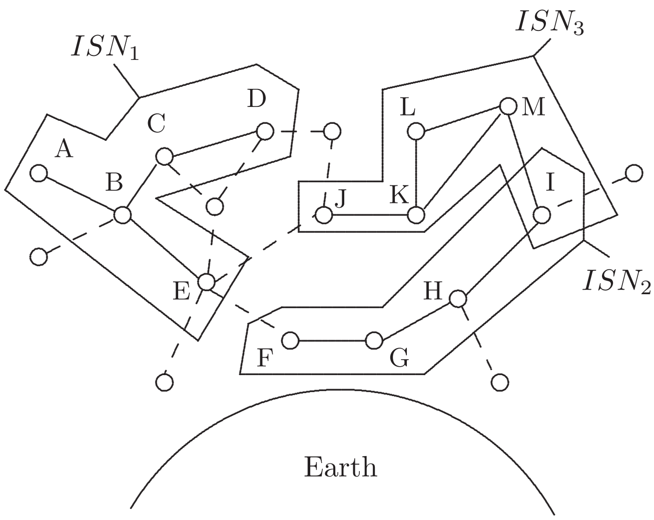


Figure 2.1: IoSat space segment representation.

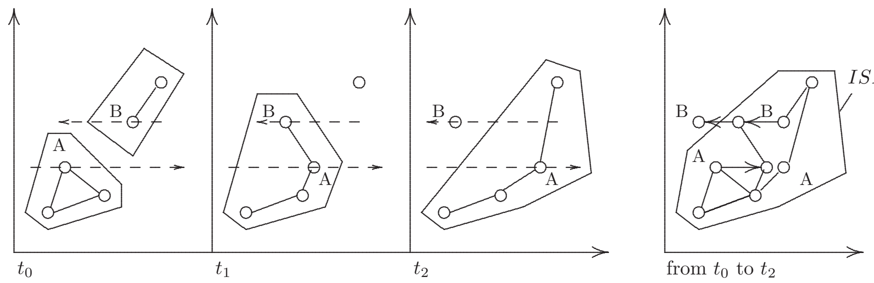


Figure 2.2: Representation of the ISN evolution

nodes are in constant movement, this phase is responsible to update network connections when intermediate links are broken. Therefore, it should be able to replace old intermediate nodes by adding new ones. Moreover, some satellites could request to participate in an existing federation that would need to add more intermediate nodes to increase the current ISN. Thus, the ISN should be able to adhere new satellite nodes as per their request, or by the need to keep the topology stable. Figure 2.2 presents an example of how the maintenance phase should address the ISN dynamism. In particular, it shows how two partitions of an ISN evolve through time (from t_0 to t_2) in which node B and node A are moving establishing new links. Finally, in the destruction phase (once the ISN is no longer required) all the nodes that have participated in the network should perform the destruction process which cleans their internal state and recovers their usual activity. This is an important phase because the resources shall be released when they are no more needed.

There is a common need that should be respected in an ISN. Satellites are embedded systems with severe limitations in terms of energy, computation, and data storage resources, which means that additional inter-satellite communications capabilities could jeopardize the mission. This could appear because satellites are normally conceived to accomplish a specific mission, and the integration of these new capabilities could suppose an additional resource consumption which could deplete the satellite. In other words, the deployment of an ISN shall not impact the mission of intermediate satellites. Therefore, this network is deployed using a resource-aware strategy while trying to satisfy application requirements. Moreover, if a satellite decides that its participation in the network compromises the accomplishment of its mission, it can decide to leave the network. Therefore, satellites require a certain level of intelligence to autonomously take this decision. An ISN is a completely dynamic and constant changing scenario, due to satellite mobility, node participation, and node resource state.

To provide an overview of the ISN concept, Figure 2.3 presents an example of a centralized FSS in the IoSat context. In particular, it can be seen the physical layer which represents the ensemble of all satellites that are physically placed in this region. Some of them accept to participate in the network and can provide the required services to deploy the FSS. Therefore, they create an ISN with intermediate federations. Through this temporal network the end-to-end FSS is then accomplished.

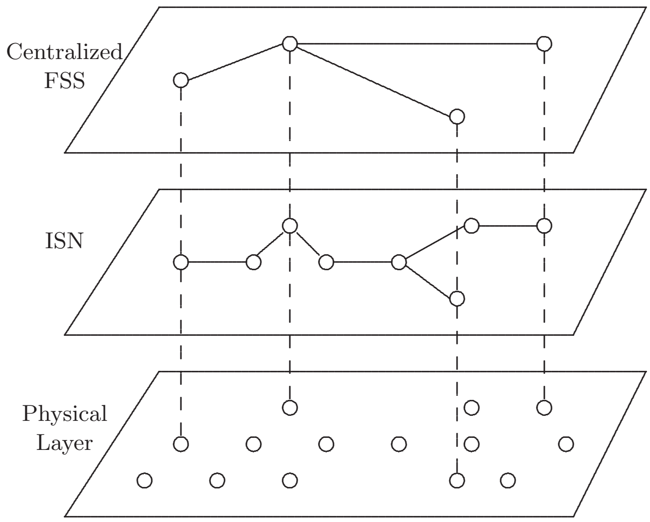


Figure 2.3: Layered representation of a centralized FSS

2.3 Key features of inter-satellite networks

Table 2.1 assembles and summarizes the different features that characterize ISN. The main characteristic of ISN is related to be satellite network. These networks naturally present **a dynamic and variable topology** over time. The fluctuation is mainly driven by the fragmentation of the network, because ISN do not include the deployment of a pre-established architecture that mitigates this phenomenon. This implies that satellite connections are intermittent and not always available. This disruption would condition the protocols to coexist with this undesired events as in the DTN. Nevertheless, the interconnection of satellites from different systems may promote the decrease of the network fragmentation by naturally increasing the number of satellites in the network. Furthermore, space is an aggressive environment that commonly provokes the failure of the entire satellite or its components (e.g. single-events, total ionizing dose, battery damage by temperature variations, battery power cycles, etc.), primarily the small-satellites which may not include hardware redundancy. In conjunction to the variable resource availability, this characteristics make satellites become unreliable node which could revoke its participation in ISN due to lack of resources or node failure. This unreliability is also extended to the ISL that connect satellites. These wireless interfaces are normally characterized by being error prone, and their features may change over time. As an example, a contact between satellites can start with a large distance that is over time reduced to lately increase again. This closeness process generates a fluctuation on the features of the link, such as error bit rate, throughput, etc. This dynamism of the network provokes changes in the topology that may be caused by fragmentation of the network, intermittent connections, unreliable nodes, and unreliable links.

Among these features that entail a dynamic environment, satellites are in constant movement following the **trajectories determined by their orbits**. This kind of motion, however, has unique properties that communications can benefit from. Nowadays, orbits are well-known trajectories, and numerous models have defined them with simple set of parameters (e.g. Keplerian orbit model, SGP model family, etc.) Some of these models include also external perturbations that may change the orbit over time, ensuring an acceptable accuracy of a satellite trajectory. Therefore, a satellite follows a deterministic and well-known trajectory that can be used in-advance. This facilitates the prediction of encounters between satellites, and thus their corresponding links. The topology of a satellite networks is indeed deterministic and predictable. Despite this physical condition, satellite may not be available to perform a contact when it is feasible; i.e. the satellite is unreliable to establish a connection. Among the previous causes of node unreliability, this can be also caused by the nature of satellite operations. A satellite orbit is a periodic movement that enables the satellite to repeatedly pass over the same ground track. Therefore, traditional satellites may have periods of the orbit that it is operative, and remaining the others in standby. This operations mode conforms the duty cycle of the satellite. Satellite orbits ensures a motion pattern that is deterministic, predictable, and periodic.

The dynamic and predictable topology of ISN is a relevant feature that protocols to deploy this kind of network must address. Therefore, the solutions must provide certain level of **autonomy** to the network, which has to be able to coexist with this dynamism. The network must be self-configurable by managing the decisions of each satellite that expect to participate in the network. This also is associated to correctly balance all the capabilities of the satellites. Furthermore, the network must adapt–and in some times predict–itself to possible events, like node failure or network fragmentation. This autonomy level is crucial in the development of ISN, because the necessity to rely on a central ground facility may entail the loss of opportunities to share unallocated resources.

Satellite are designed according to a set of requirements that ensure the accomplish of the mission goals. Since satellite is manufactured and placed in-orbit, its hardware is no longer accessible and its capacities can only be degraded over time which would compromise the mission. Some research has tried to evaluate the possibility to perform in-space manufacturing (Trujillo et al., 2017), and previous experiences demonstrated that on-orbit satellite repairs represents a great cost (Hastings and Joppin, 2006). Therefore, these conditions determine a satellite as an embedded system with bounded and valuable resources that need to be–as much as possible–preserved over time; i.e. **a resource-constrained system**. Typically resources of a satellites can be illustrated as processing capacity, energy availability, storage space, and downlink opportunities. The addition of new communication capabilities can impact the mission performance, which cannot be accepted. Therefore, proposed solutions for ISN have to include be aware of resources consumption, and minimize their corresponding use.

A cause of the custom design of each satellite associated to its mission, the space is an environment in which satellites with different capabilities and goals coexist. This makes ISN composed of **heterogeneous nodes** which differ at hardware, software, and operational levels. Let review the implications of this large heterogeneity. Satellites are owned by entities that not necessarily are the same ones that operates them. This provokes that each satellite has its own goals, and its own behave. The interaction of those satellites require a common interface not only limited to communications mechanisms, but also at operational level. Furthermore, satellites from distinct missions are equipped with different capacities according to their components. This causes that satellite with different resource availability must interact to deploy ISN. Moreover, the communication technology of each satellite can differ, increasing the complexity of the system. Some flexible and adaptable solutions have started to be investigated in the physical level of the communications (Akhtyamov et al., 2016). This heterogeneity appears in other levels of communications capabilities. For instance, a satellite which offers a resource may require certain level of confidentiality that an intermediate satellite cannot satisfy or respect. This unification of the constraints in satellite federations is crucial in the deployment of ISN.

The expected performance in a **federation** is related to the type of resource that is being shared. As an illustrative example, a federation that is intended to share data storage does not require the same performance that one that shares downlink opportunities. This entails that the type of traffic that is generated in a federation depends on the resource that is expected to be shared. Therefore, solutions to deploy ISN must be flexible to address the different traffic natures. So far, we can summarize these different traffics to the following requirements: (1) reliable data transmission from customer to provider (e.g. store data in provider); (2) unreliable data transmission from customer to provider (e.g. download data stream); (3) small-commands generated from customer to provider (e.g. formation flight or notifications); (4) reliable data exchange between customer and provider (e.g. processing of raw data); and (5) small-commands from customers that trigger reliable data transmissions from providers (e.g. execution of payloads). Sometimes, the same resource may be requested by different satellites, this implies that ISN would manage multiple traffic flows from different satellites that would reach a common point. Other cases can propose the integration of multiple providers in the same federation, provoking a traffic that follows a many-to-many transmission manner. Finally, federations are sporadic and opportunistic which implies that ISN must accordingly react to catch these temporal events.

For all these features, the IoSat paradigm, and in particular the ISN concept, becomes a challenging research field. In terms of communications, it is difficult to be implemented using traditional solutions. Therefore, an analysis in depth related to the different options is presented in the following sections.

2.4 Synergies with other satellite networks

So far, the IoSat paradigm concept and the particular features of ISN have been presented. The dynamic and temporal nature of ISN makes difficult the definition of end-to-end routes that must be determined when the network is constructed. This challenging environment requires the proper definition of a routing protocol that is able to comply with this task while it suits the different features of these sporadic and opportunistic networks. As part of this custom definition, a preliminary discussion regarding the suitability of current technologies may provide a global overview of the specific features that this routing protocol should have. For this reason, we start evaluating the similarities and synergies that ISN can share with other satellite networks. Chapter 1 presented a taxonomy of the current satellite networks that were proposed in previous investigations. Therefore, this section does not focus on their defintion, instead the work has been centered on comparing them with respect to the ISN.

As stated previously, the satellite networks that are able to coexist with the disruption of the network propose mechanisms that include the predictive and determinism satellite motion in its own definition. This approach is relevant, because satellite networks are indeed characterized by this unique motion which distinguish them among other networks. Snapshot networks (Werner, 1997) were conceived to leverage on this feature to generate a well-known and periodic sequence of snapshots, which helps to estimate satellite connections changes in the network. These prediction mechanisms provide a powerful capacity to satellites that could estimate the near-future network connections, and decide when to deploy a federation or anticipate possible fragmentation of the network. Nevertheless, this prediction capacity is related to the construction of the snapshot sequence. Typically, this sequence is beforehand generated in a central infrastructure located on ground. This centralized approach may be efficient in terms of processing capacity and prediction accuracy, but it limits the autonomy of the network which its route definition depends on the continuous communications with ground segment. This solution constraints the development of autonomous satellite applications, and requires the deployment of a robust infrastructure that must never be inoperative. Furthermore, the centralized approach also requires to recompute and upload the entire snapshot sequence if a satellite is added or removed to the network. This poses some limitations in the scalability of the solution, which could not be suitable for large scenarios. Therefore, this capacity to predict satellite contacts evol-

Table 2.1: Summary of ISN features

Dynamic Topology	
Intermittent connectivity	Due to node movement, node connections change. This makes that satellites are not constantly in the line-of-sight.
Network Partition	Due to its connectivity, an ISN can be partitioned and then merged (e.g. Figure 2.2).
Node Failures	Although there is component redundancy, space environment can provoke the failure of a spacecraft subsystem, which triggers the death of the node.
Unreliable nodes	The variation of satellite state (e.g. energy) provokes that nodes may not longer be able to participate in an ISN, and thus its withdrawal.
Unreliable channel	Communication is performed through a wireless medium, which is error prone. Furthermore, medium characteristics are time-varying.
Orbital Movement	
Deterministic trajectory	Nodes are satellites which follow well-defined orbital trajectories, determined by specific parameters.
Predictable topology	Due to its nature, orbital trajectories can be predicted with a good level of accuracy, which makes the global topology predictable too.
Duty cycle activity	Orbit movement makes that satellites periodically pass over a target region. Thus, satellite bypass between an operational and standby mode.
Resource-constrained nodes	
Power-limited nodes	Satellites are usually powered by the combination of solar panels and batteries, which makes the node energy limited and its level time-variable.
Memory-limited nodes	A satellite has a limited storage capacity, usually composed of persistent and volatile memories. These memories stores internal house-keeping and mission (science and/or communications) data.
Embedded systems	An in-orbit satellite is an embedded system with limited physical access. This makes impossible its direct maintenance, and thus a control is needed to avoid undesirable behaviors (e.g. maximum depth of discharge).
Custom designed	A spacecraft is designed to accomplish a mission, this implies that additional resource consumption shall not jeopardize its accomplishment.
Autonomous network	
Self-configurable network	Satellites shall be able to autonomously configure themselves to create an ISN.
Adaptive network	Due to the dynamic topology, an ISN shall autonomously react against network events, such as link disconnection or node failure.

Heterogeneous nodes	
Different objectives	Each satellite has a different mission to accomplish.
Different state definition	Due to spacecraft diversity, each satellite defines its state differently.
Different bandwidth capacity	Large, medium, and small satellites coexist in the space. Due to its characteristics, each one has different communication capabilities.
Limited data security	As an ISN is composed of satellites from different entities, forwarded data can be read by undesirable agents. A security level shall be provided to ensure data privacy.
Traffic Dependence	
Federation-dependent	As a federation is deployed upon an ISN, each federation has different requirements that impacts on the network management.
Many-to-many traffic	Depending on the application, multiple spacecrafts can communicate with multiple ones. In this case, it cannot be predefined a specific traffic model as in other networks.
Opportunistic and Sporadic	As a federation is opportunistic, an sporadic ISN is deployed depending on the opportunity to create this federation. It is thus a sporadic network.

ution, and to generate the corresponding snapshot sequence following a distributed strategy seem suitable for ISN.

Although the generation of the snapshot sequence corresponding to the entire network may not be necessary, this prediction capacity could be used for the foreseen of local connection changes. This is also a feature that some DTN solutions (Vinton et al., 2007) proposed in the past. The forwarding-based routing protocols follows this strategy of predicting network evolution to determine routes over time, while the replication-based protocols are based on probabilistic delivery through these time-forward routes. The integral application of these solutions would only support the deployment of delay-tolerant applications in ISN, which is characterized by having different types of data traffics. Nevertheless, This capability to determine routes in this new dimension is a potential feature that may be suitable in heterogeneous scenarios if they can be complemented with typical instantaneous routes. Depending on the type of required communications interface, the routing protocol could select an instantaneous end-to-end route or a route determined by a succession of satellite contacts. In this regard, time-sensitive data may be routed to instantaneous routes, while the other data could be forwarded to a time-forward route which would experience larger delay. This approach can only be possible if the satellite integrates store-and-forward mechanisms developed specifically for DTN. Therefore, the integration of this technology in the heterogeneous context of ISN becomes a powerful capacity.

Other satellite networks were conceived to mitigate the disruption of the network with custom satellite architectures. LEO satellite networks are an example of this static architecture which proposes the deployment of a satellite constellation that is equal to a mesh network over certain regions. This specific composition enables to simplify the routing protocol which practically becomes a forward process based on rows and columns of the mesh. Additionally, each node takes the decision of how the route is defined, which may provide flexibility to the network in case that changes appears. This decision is supported by the availability of multiple routes that connect the same source and destination pair. This multiple possibilities makes the solution flexible to react against different events by switching among the different options. The research on this topic has also been focused on the congestion avoidance, and how define routes according to this congestion state. All these features are suitable for ISN in which the flexibility and adaptability are required. Nevertheless, the custom architecture that is required to deploy these mechanisms does not match with the heterogeneous and dynamic scenario that represents the ISN. Therefore, the application of entire solutions developed in LEO satellite networks may not be feasible in the context of IoSat. Instead, algorithms or parts of the communications mechanisms may be interesting to integrate for specific purposes, like the congestion avoidance mechanisms using multiple routes.

This heterogeneity on satellite orbits is a key feature of MLSN. These networks leverage on the increased view of satellites located in high altitudes to coordinate and forward data from lower altitude satellites. A high bandwidth communication and an enhancement on links stability can be achieved thanks to this hierarchical structure based on altitude layers. This powerful performance may mitigate the disruption of the network that could degraded some communications that require stable routes during a lapse of time. This capacity to remain stable routes using higher-altitude satellites seems a potential feature to manage dynamism in the ISN. Nevertheless, a tradeoff between route stability and distance between layers. A satellite that is located in a higher altitude (e.g. GEO region) has a wider area where low-layer satellites can move and communicate. Therefore, a route between two low-layer satellites that includes this high-layer satellite becomes always more stable that directly the ones composed of only low-layer spacecraft. Nevertheless, the communications distance with a higher altitude satellite requires more transmission power or directive antennas which would consume space and mass of the satellite. Therefore, a specific layer configuration may balance these features. The different orbits that can coexist in ISN can difficult this optimal configuration. Therefore, a mechanism that is able to detect when the inter-layer communications is beneficial could integrate this concept in the ISN. Satellites in these networks are unreliable nodes that may (or not) collaborate to generate the network. This uncertainty entails that those stable routes using high-layer satellites cannot always be available. Therefore, routing protocols that only are based on this layered architecture may not be suitable for ISN dynamism. Nevertheless, a solution that integrates this feature as a complement or support to other kind of routing may enhance the resulting performance.

Other proposals have tried to model a satellite network as MANET (Macker, 1999). The nodes that compound this kind of network are able to collect network metrics (e.g. connections, queue status, etc.), and determine routes according this information. This capability makes the network be self-organized and self-configurable which can determine the ad-hoc routes to each topology status. This a powerful capacity for dynamic environments that provides certain level of autonomy to the network. Nevertheless, the protocols developed in these networks define end-to-end routes to an instantaneous topology configuration, which does not integrate mechanisms to manage network disruption. These solutions are not able to conceive routes over time, and thus apply store-and-forward mechanisms. Therefore, DTN and MANET protocols are opposed in this aspect. The heterogeneity and temporality of ISN require certain level of autonomy to leverage the opportunities to deploy federations. Therefore, the application of MANET solutions seems relevant for ISN, although they may be integrated with store-and-forward mechanisms to coexist with network disruptions.

The integration of WSN in satellite systems were also discussed in the satellite sensor network concept (Song et al., 2015). These networks are compounded of resource-constrained systems which share similarities to satellites. The major difference is the farther communications ranges that satellite network experience. Nevertheless, the WSN routing protocols have integrated mechanisms to introduce the resource status in the definition of routes (e.g. power level, memory usage, etc.) These resource-aware mechanisms are crucial for ISN in which satellites that participate in the network must balance the shared resources to still develop their own mission. For this reason, the use of OF that merges different metrics related to the resources status is a potential feature to be included in the route definition. Nevertheless, most of WSN protocols leverage on hierarchical clusters to reduce the energy consumption due to communications. These clusters may not be feasible in the dynamic and mobile environment of ISN, in which a satellite that works as the cluster manager may not always be accessible. Therefore, a distributed approach should be conceived.

In summary, all the previous satellite networks presents features that are suitable to deploy ISN. However, there is no one that can satisfy alone the requirements of these temporal and opportunistic networks. Table 2.2 summarizes the different interesting features of each satellite network to be included in a routing protocol that deploys ISN, and also their drawbacks. It seems difficult to find a current routing protocol from these satellite networks that can construct and manage ISN. Instead, a conglomerate of features from different protocols may be an interesting strategy to construct the corresponding routing protocol for ISN. For this reason, the following section discusses about which are the common properties of a routing protocol to suddenly evaluate the different implementations conceived for previous satellite networks.

Table 2.2: Summary of the advantages and drawbacks to use specific solutions of each satellite network to deploy ISN

Satellite network	Interesting for ISN	Detrimental for ISN
Snapshot networks	Snapshot concept Flexible and adaptable model	Difficult to be scalable
DTN	Study of network partitions Store-and-forward mechanism	Extreme environments Only for delay tolerant applications
LEO satellite networks	Node-by-node route decision Simple routing mechanism	Custom satellite architecture
MLSN	Hybrid constellations Increase of network capacity	Expected to have high-layer nodes available
MANET	Adaptive to topology events Self-organizing	Centered on random mobility Unable to manage network disruption
WSN	Same node architecture Energy-efficient model	Low data rate and small distances hierarchical clusters

2.5 Applicability of nowadays routing protocols

Each type of satellite networks has features that are suitable to include in a routing protocol definition to deploy ISN. Therefore, the definition of this protocol could be performed by the conglomerate of different mechanisms deployed for each satellite network. This section goes deeper in this reflection by discussing the feature of the routing protocols developed previously for each satellite network class. The details of these protocols are not presented in this section, because Chapter 1 has already introduced them. The discussion that is performed in the following paragraphs is focused on which capability of these known protocols is suitable for ISN, and which one is inappropriate. The section is structured by blocks which correspond to each satellite network type.

2.5.1 Routing protocols from snapshot networks

Among the different protocols, the DT-DVTR protocol is the first proposal developed for snapshot networks. This protocol computes the routes that are available in each snapshot of the sequence. As stated previously, this snapshot sequence is an interesting concept to model the dynamism in a satellite network. The protocol suggest the use of the minimum hop criterion to define the routes in each snapshot, but it can be extended to other criteria. Furthermore, the protocol discusses the possibility to determine the changes between snapshots to anticipate possible changes of a route. The original protocol was developed for the ATM standard, in which a virtual circuit between two remote nodes is established. This circuit is a connection that has to firstly be established to define the intermediate nodes, which allocate resources to maintain available the entire connection. This connection-oriented approach may not be suitable for mobile nodes, which have variable links. Therefore, the DT-DVTR protocol cannot be applied literally for ISN, but its snapshot concept may be useful.

The SIR protocol suggest the integration of all the snapshot sequence in a single graph representation. Using conventional mechanisms, a route could be defined in this new representation. This simplifies the challenge to determine the changes of a route due to snapshot transition, and enables to determine a route considering already these transitions. Nevertheless, the need to compose the entire representation of the sequence require certain level of resources that may be conducted in satellites. Additionally, this unified representation can become really huge depending on the number of snapshots, which intrinsically is determined by the number of satellites. Therefore, it is not a scalable solution, which complicates its use for ISN.

The PLSR protocol extends the SIR protocol by formally define a ground infrastructure that computes the entire snapshot sequence, and the corresponding routing tables for each satellite. Furthermore, the model stored in the infrastructure can be updated with unpredictable events by the corresponding notification from the satellites. This approach may be efficient to optimize the resources of satellites which could not be capable to compute the entire snapshot. However, the network performance is bounded to this infrastructure, which limits its autonomy. If the communications with this infrastructure is not available, then no route can be defined in the network. Therefore, the PLSR cannot directly be applied for ISN, although the possibility to update the predictable model with unpredictable events seems a powerful capacity.

Despite this dependence of the ground infrastructure, the snapshot sequence is a concept that could enhance the performance in ISN. The definition of the sequence that corresponds to the entire network may require large memory allocation, and probably a satellite would not use the entire information. The snapshot concept could be applied in local communications, and for short forward time which would enable the detection of immediate predictable events. This requires the development of a dedicated prediction mechanism that should be executed in each satellite to estimate its satellite contacts. Once we have identified this potential synergy, there is still the question of how to generate a snapshot. The original DT-DVTR suggest the natural change of a ISL; i.e. creation or destruction of a satellite-to-satellite link. This could result with a large amount of snapshots in some parts of the sequence due to the orbit nature (e.g. the polar region due to polar orbits). Therefore, investigations suggested the use of a constant sampling, although some link may be lost between topology sample (Gounder et al., 1999). Furthermore, a custom optimized method for the Iridium constellation was also conceived to address this problem of the polar regions. The current solution is entirely founded on the constellation architecture which makes it not applicable to more heterogeneous cases. Nevertheless, this technique could be considered as a basis to develop an extension for this heterogeneous cases.

Table 2.3 summarizes the particular feature of the different mechanisms developed in the context of snapshot networks. In regard to the IoSat context, the snapshot technique is a mechanism that allows simplifying the complexity of satellite network mobility in a periodic sequence. However, the amount of snapshots is directly related to the number of satellites that compose the network. This could impact the storage consumption of each satellite, when indeed this satellite could not be actively working in the network. Therefore, its direct application in ISN is quite limited. Due to the deterministic nature of satellite movement, this mechanism should not be simply discarded. Indeed, a promising approach could be the use of this mechanism in a local region where the number of satellites can be acceptable, instead of the entire network. This would require the development of a prediction mechanism that should be executed in each satellite. Furthermore, the extension of the optimized mechanism to avoid the large generation of snapshots in polar regions could be developed for heterogeneous scenarios.

Table 2.3: Summary of Routing Protocols in Snapshot Networks

	Interesting for ISN	**Detrimental for ISN**
Routing Protocols		
DT-DVTR	Management by snapshots	Connection-oriented
PLSR	Manage unpredictable events	Earth-dependent
SIR	Predict future connections	Non-scalable
Creation of the snapshot sequence		
Constant sampling	Reduction of memory consumption	Less accuracy on broken links
Optimized method	Managing high latitude links	Specific of Iridium constellation
Route definition in the snapshot		
Minimum hop	Reduction of broken links	Not aware of other events
Multi-path	More reactive against broken links	Memory consumption
Traffic concerned	Manages congestion	Only per a snapshot

2.5.2 Routing protocols from DTN

Replication-based routing protocols from DTN are simple solutions that found on the probability to reach the destination is related to the number of replicas performed of an original message. The Epidemic and n-Epidemic are similar approaches in which continuous replication or a number (n) of copies are transmitted in the network. These simple solutions seems suitable for low-computation devices, which could be satellites. Nevertheless, the amount of network capacity wasted make them totally unsuitable for ISN. This unnecessary consumption of resource could provoke the failure of the main mission, which is unacceptable in the IoSat paradigm. Therefore, the original epidemic approaches cannot be applied for ISN. The EAEpidemic aims at bounding this resource consumption by forwarding depending on the energy state of each node. Despite this is an enhancement for ISN, the satellite resources comprise more than only energy (e.g. memory usage, computation, etc.) Nevertheless, this approach could also be extended to include further status information.

The approach that balances better the replication with the probability to reach the correct delivery is the spread and wait protocol. This protocol proposes different mechanisms to replicate the message in intermediate satellites. An extension of this less resource consumption is the MaxProp approach, which provides mechanisms to replicate just the essential messages. Specifically, the messages that are different from one node to another are the ones that are replicated. Although the consumption of the resources is optimized in both cases, the typical replication-based protocol cannot ensure that a message is correctly delivered. This delivery is associated to a probability which may not be acceptable to certain applications. As an example, if a satellite wants to create a federation to store data, it requires that the data is correctly delivered to its destination. Therefore, these solutions cannot directly be applied for ISN.

The probability to correctly deliver a message may be conditioned if information of the network is used. MobiSpace approaches leverage on determinism movement to create contact history to decide if a satellite may be reliable to deliver a message. The original concept does not include a delay constraint to deliver the message, which some applications could not accept. Therefore, the MOVE extended the concept by including multiple metrics in the contact history. The RCM proposed to also include the fact that satellite orbits are periodic trajectories, which can influence in the probability to contact with a specific satellite. All these features were also developed in the PROPHET protocol which follows a similar behavior. The contact history technique is a potential feature that flexibly is able to estimate a confidence metric of each satellite neighbor. Nevertheless, this cannot ensure that the message has been delivered, neither to detect when a message is not delivered. A return mechanisms that confirms the correct delivery must be implemented.

Forwarding-based protocols identifies an entire route before transmitting a message. Thanks to deterministic satellite motion, these protocols generate topology models in which network changes are integrated. According to these estimated changes, the protocols are able to define a route over time. The CGR is the main implementation that follows this approach, and generates a plan that corresponds a schedule of all the satellite contacts that are available in a satellite network. As in the PLSR, this contact plan is generated by the CPCE which is a central node located on ground with large computing resources. These plans are then used to define routes over time, which are uploaded in each satellite. This capacity to define routes over time is a powerful feature that is suitable in ISN. Nevertheless, as in the case of snapshot networks, a centralize architecture may limit the autonomy of the network. The OCGR extends this original protocol by promoting the estimation of satellite contacts by each satellite using an contact register. Despite the prediction of these contacts in distributed manner is suitable for ISN, the historic approach works with a probabilistic definition. This implies that the contact plan is constructed over contact probabilities which may entail wrong defined routes. Nevertheless, this prediction capability in each satellite is a suitable feature that could be investigated for ISN.

The DTLSR protocol generates this contact plan by the combination of information from different satellites. This information includes satellite status, and is periodically propagated in the network. Therefore, all the satellites that conforms the network has the information of the entire network. Nevertheless, this information may be outdated due to the network disruption. Therefore, when a route is considered broken, the satellite keeps the route during a lapse of time, because the information of the network may not be accurate. This approach quickly react against network changes, but the periodic transmission from each satellite entails a consumption that may not be affordable, specifically for those satellites that are not interested to participate in the network. Therefore, the use of the DTLSR for ISN seems not be suitable.

Table 2.4 summarizes the different features of the routing protocols. DTN solutions applicable to satellite networks provide capabilities to estimate routes over time. These routes are feasible thanks to the store-and-forward approach, which forces to each intermediate satellite to be responsible of the received messages. Despite the forwarding-based protocols seems

Table 2.4: Summary of Routing Protocols in DTN

Routing protocol	**Interesting for ISN**	**Detrimental for ISN**
Replication-based protocols		
Epidemic	Simple to address network disruption	Always flooding the network
		Large energy consumption
n-Epidemic	Reducing flooding mechanism	Not energy aware
EAEpidemic	Adapting flooding to energy state	Reaching the destination is not ensured
Spray and wait	Not flooding, just a group of messages	No consideration of neighbor state
	Less energy consumption	
MaxProp	Replicate just what is essential	Delivery cannot be ensured
MobiSpace	Usage of deterministic satellite movement	Not delay aware
		Scalability issue
MOVE	Path prediction using multiple metrics	Large computation
		Scalability issue
RCM	Concept of the encounter historic	Impossible to manage failures
	Simple method	Scalability issue
PROPHET	Combination of multiple metrics	Scalability issue
	Prediction of future encounters	
RAPID	Adapt the protocol to specific criteria	Delivery cannot be ensured
Forwarding-based protocols		
CGR	Similar to snapshot approaches	central unit to generate the plan
	Estimation of routes over time	
OCGR	Satellite predicts contacts	Prediction based on probability
DTSLR	Routes over time in each satellite	Large resource consumption
	Reaction against unpredictable events	Participation of all the satellites

more suitable for ISN, the centralized architecture to generate the contact plan may limit the autonomy of the network. The other proposals that founds on the contact prediction in each satellite are more adapted to the nature of ISN. Nevertheless, the mechanisms to predict these contacts can be investigated to not be based on probabilistic historic, which could entail wrong route definitions.

2.5.3 Routing protocols from LEO satellite networks

The different routing protocols developed for LEO satellite networks are founded on the custom architecture of this satellite network. The DRA proposes a multi-hop and connectionless routing protocol by which each node decides the best route in the mesh network to reach the satellite destination. The generated route is defined as an ensemble of rows and columns that the message must forward in the mesh. The mesh architecture enables to define this protocol which simplifies the route definition and computation. The signaling necessary to define the route is no longer needed, because each satellite knows in advance the architecture of the entire network. Despite this performance is suitable for satellite platforms, this solution is really bounded to the custom constellation architecture which makes it difficult to be applied in an heterogeneous context like IoSat. Moreover, network events like node congestion are not detected in the original definition of the DRA. Therefore, the ELB protocol suggest the use of small control packets to detect the congestion of certain links. This new capability allows the nodes to redirect the traffic to another route to reduce the congestion. This multi-path approach is suitable to manage this congestion. Nevertheless, the ELB always detect the congestion once it has already appeared, which entails lapses of times during which the messages are lost. Therefore, the PAR protocol predicts the future congestion of the network by evaluating the status of local links. The redirection of the traffic may also be performed in another link that is also congested. The TLR protocol proposes the use of a traffic-light system to represent the congestion status of each link. The ALBR proposes also a mechanism to manage the network congestion using an agent-based approach that enables to retrieve a global view of the network. Despite all these solutions that manages congestion are interesting for ISN, they are founded on the mesh architecture in which satellites have always for ISL. However, all of them propose the use of multiple routes to quickly redirect the traffic in case that one of them are congested. This feature is feasible in heterogeneous scenarios, and would enhance the capability of the network to mitigate node congestion.

The DLAR protocol estimates when a satellite is located in a forbidden region, where the communication is not established. This approach enables to avoid the deadlock problem, that conforms the transmission of messages that would never be delivered. Using this protocol, the waste of resources due to useless transmissions is mitigated, and thus the entire system is optimized. Nevertheless, these forbidden regions is specific of this type of satellite network, and may not be applicable to ISN.

Table 2.5 summarizes the benefits and drawbacks of previous protocols. The protocols developed for LEO satellite networks are dedicated to this custom architecture. Therefore, their use for ISN which is an heterogeneous context is not suitable. Specifically, ISN are compounded by satellites that follow different orbit, and the ensemble may not conform a mesh architecture. Therefore, routes defined as rows and columns are not coherent with this scenario. However, the multi-path approach provides to the satellites the capability to quickly react against network events. These capabilities are interesting to apply in ISN which is compounded by unreliable nodes.

2.5.4 Routing protocols from MLSN

MLSN proposes the use of different altitudes to generate a hierarchical structure between satellites. The satellites that are located in higher altitudes are able to interact with a group of satellites located in lower-altitudes. Approaches that enables the interaction of LEO and MEO satellites have been proposed. HQRP proposes the use of inter- and intra-layer information to define routes that may be composed of satellites in the same layer or among the layers. This protocol, however, does not take in consideration the congestion of the MEO satellites. This congestion may occur because MEO satellites offers the possibility to deploy stable routes between satellites located at LEO layer. Therefore, the routes passing over the MEO layer would be preferred. TLAR protocol takes in consideration this congestion of upper-layer satellites, and proposes the use of multiple satellites to redirect the traffic.

The MLSR protocol proposes a mechanism to define routes in a generic multi-layered architecture. The satellites located in a certain layer estimate the network topology by retrieving different metrics. This information is suddenly forwarded to upper-layer satellites which integ-

Table 2.5: Summary of Routing Protocols in LEO Networks

Routing protocol	Interesting for ISN	Detrimental for ISN
Queue state approach		
DRA	Distributed packet forwarding	Specific for custom constellations
	Simple and low signaling	
ELB	Quick reaction against congestion	Large energy consumption
	Flow redirection through alternative paths	Alternative paths congested
PAR	Prediction of congestion	Local congestion information
	Priority-based next-hop selection	
TLR	Different levels of congestion	Local congestion information
Agent-based approach		
ALBR	Global congestion information	Non-scalable
Prediction-based approach		
DLAR	Use of deterministic behavior	Specific for custom constellations

rate it with their own layer information. After this aggregation, the satellite keeps forwarding this information to upper-layers, until the last one is reached. The aggregation of this information enables an upper-layer satellite to have a global view of all the lower layers, and thus define the routes between the satellites located in these layers. This approach optimizes the routes thanks to the global view of the satellite that are located in high altitude. Nevertheless, these satellites must be there to perform the route definition. If an intermediate satellite located in high altitude decides to not participate in the network, this approach does not work. Therefore, the MLSR protocol cannot be directly applied in ISN. Nevertheless, the heterogeneity on orbit altitudes enhances the capability of the network, and the stability of the routes. Therefore, the concept of layers–not necessarily always available–could be used for ISN.

HCAR protocol proposes the use of the snapshot concept of the network that conforms the low layer to compute the routes. As in the snapshot network case, the routes would be generated in a ground infrastructure that would propagate to GEO satellites. These satellites would suddenly transfer the corresponding routing tables to each low-layer satellite. This approach is another example of how altitude can benefit to manage the network, and in this case GEO satellite is always accessible from ground. Nevertheless, the communications between GEO and LEO require a high power transmission capacity which is not available in all satellite platforms.

Table 2.6 summarizes the different features that may be suitable for ISN, as well as the drawbacks of the previous protocols. In summary, the concept of having multiple satellites in different layers is a promising architecture thanks to the enhancement achieved with the altitude difference. The combination of multiple satellite constellations enhance the global capacity of the network, and the stability of the satellite connections. Nevertheless, all the protocols considers that the upper-layer satellite is always available to manage or participate in the network. This condition cannot be ensured in the IoSat paradigm, which makes inappropriate the use of these solutions. However, the possibility to distinguish between route in the same layer or between layer is a powerful capability to manage different traffic types. Time-sensitive data would be routed through satellites in the same layer, while traffic that require more stable routes could be forwarded to upper-layer satellites.

Table 2.6: Summary of Routing Protocols in MLSN

Routing protocol	**Interesting for ISN**	**Detrimental for ISN**
LEO-MEO approach		
HQRP	Combination of multiple satellite layers	Distinction between inter and intra layer flow
TLAR	Using multiple layers to manage congestion	Assuming that MEO satellites are available
LEO-GEO approach		
HCAR	Topology changes prediction	Large energy consumption Unique control satellite layer Ground infrastructure dependence
Multiple-layers approach		
MLSR	Combination of multiple heterogeneous satellites	High-layer satellites are not always available

2.5.5 Routing protocols from MANET

Reactive routing protocols developed for MANET aims at defining routes only when the data must be transmitted. This approach is suitable for sporadic data transmission, rather than constant data streams. Among the different protocols, AODV is the most used reactive protocol. This protocol triggers a discovery phase to identify a route between a source and a destination. A proper exchange of messages between both nodes is ensured by propagating a control packet to the destination, and with its replication that follows the same route. This solution does not consumes considerable resources, because it just propagates control packets when it is required. Nevertheless, the discovery phase provokes an initial delay on the delivery of the data, which may not be acceptable for certain applications. Furthermore, if the route is broken during the data transmission, the source must to start again the discovery phase. This could generate a scenario in which the source is constantly discovering the network due to the mobility of the nodes. An extension of the AODV addressed this situation by following a multi-path strategy. The AOMDV enables the source to determine a group of routes to the destination. Therefore, if a route is broken, the source can quickly redirect the traffic to another route. Again, the multi-path approach seems a suitable strategy to quickly react against network events.

LAOR protocol tried to integrate the AODV in a satellite network. The discovery phase has been optimized to spend less time, and thus to reduce the initial delay. This optimization is based on using deterministic satellite motion to detect which could be the appropriate region to propagate the discovery control packet. This bounded area can be estimated by the satellite because a ground infrastructure uploads a precomputed plan of the different satellite contacts. LAOR protocol is great example of how MANET solutions can be enhanced thanks to integrating the prediction of satellite movement, which is a unique characteristic of satellite networks. Nevertheless, the centralized architecture–which is the same than in snapshot networks and DTN–limits the autonomy of the network.

The proactive routing protocols are characterized by constantly learning the topology of the network, and define accordingly the different routes among nodes. Depending on how this information is forwarded, the network capacity can be largely wasted. OLSR protocol is an optimized approach to minimize this consumption by defining specific nodes which are the ones that share the status information of its local area. Moreover, this protocol retrieve the routing table which is available in each node to define the routes that connect to all the nodes. This enables the node to detect changes in the network just after they happens, and not wasting time, which is suitable for time-sensitive data. Nevertheless, this operation may not be a scalable solution when satellite systems large. Distance-vector protocols addresses this situation by aggregating information which is at the end delivered by the neighbors of a node. Therefore, the node knows that over a neighbor it can reach a destination. DSDV protocol is an example of this approach. A similar implementation is performed in BATMAN protocol, which proposes the periodic messages to update the information of the network. These messages are propagated just to the neighbors which process the information, and forwards it indicating to which node can be reachable over it. Despite these solutions are suitable in terms of memory usage, they require the participation of the entire network. Furthermore, the constant transmission of information can consume considerable resources of satellites.

The hybrid routing protocols combines both strategies to leverage on the low consumption of reactive protocols and the fast reaction of proactive protocols. Zone-based protocols follow this premise by defining different zone in which reactive or proactive transmissions are done. ZRP is the original protocol that follows this strategy by defining a single zone where the proactive operation is conducted. Outside this zone, the communications is performed following a reactive manner. IZRP extends the concept by defining an independent zone to each node. This represents that a node would detect changes close to it, rather than the ones that are located farther. FSR and TZRP optimizes the approach by defining specific set of zones. The combination of both strategies seems a powerful capacity that could be included in the routing protocol that deploy ISN. Instead of defining zones, which could not be suitable for satellite motion, a discovery phase followed by a proactive maintenance is an interesting combination that suits with the different phases of the ISN lifetime.

Nowadays, the application of MANET routing protocols in satellite context is still a research topic of great interest. Thanks to their adaptability, flexibility, and scalability, these protocols are interesting candidates to deploy ISN. Table 2.7 summarizes the different features that could be interesting for ISN, and also the corresponding drawbacks. Some MANET routing protocols could be evaluated in satellite networks thanks to its flexibility and adaptability, as it is the case of the OLSR or the BATMAN. However, these protocols may consume a considerable amount of satellite resources that could compromise the mission. For that reason, an hybrid approach that combines the discovery phase with the proactive maintenance is an interesting combination that suits with the different phases of the ISN lifetime. Moreover, the integration of satellite motion in the protocol demonstrated the potential enhancement of the original MANET protocol.

2.5.6 Routing protocols from WSN

Satellites are embedded systems that may share some similarities with a sensor. Both systems are resource-constrained, and require that the communications protocols were defined to consume the necessary resources. Therefore, the routing protocols developed in WSN are focused to minimize or optimize the energy of the different nodes. Commonly, the protocols found on the definition of a hierarchical structure. This structure can be a cluster of nodes where one of them represents the manager of the cluster. This manager is responsible to aggregate the data

Table 2.7: Summary of Routing Protocols in MANET

Routing protocol	Interesting for ISN	Detrimental for ISN
Reactive approach		
AODV	Low power consumption	Long reaction time
		Re-execution when path is broken
AOMDV	Multi-path capacity	Increase of memory usage
LAOR	Use of satellite information Discovery phase area bounded	Long reaction time Upload of satellite position information
Proactive approach		
OLSR	Quick reaction against events	Huge energy consumption
	Fast transmission	For small scenarios
DSDV	Manage multiple updates	Large overhead
BATMAN	Intelligent propagation of network topology	Cannot manage network disruption
		All the nodes must participate
Hybrid approach		
ZRP	Combination of reactive and proactive techniques	Unique and static zone
	Zone boundary concept	
IZRP	Each node has its own proactive zone	Difficult to manage in a large scenario
FSR	Different quality zones	Assuming uniformly distributed traffic
TZRP	Manage high-mobility with just two zones	Difficult to manage in a large scenario

from the other nodes, and transmit it to another manager in order to be delivered to one of the nodes in another cluster. The definition of this manager is the crucial challenge that needs to be investigated. LEACH protocol proposes a random selection among the different nodes that conform the cluster. This may provoke that some nodes with less resource capacity would become the responsible to aggregate the data, which could entail the entire depletion of this node. Therefore, HEED protocol determines a cluster manager according to the energy state and transmission capacity of each node. In some cases, the communications with a cluster manager may be more resource expensive than directly communicate with a node from another cluster that is closer. Therefore, EHEED extends the HEED protocol to enable the possibility to interact the nodes from different cluster in specific conditions. This cluster approach is suitable for satellite systems like swarms or cluster, but not appropriate for dynamic environments like ISN.

The definition of these clusters is also another research trend that has been conducted. RPL protocol allows defining a ranked hierarchy in which a node becomes the parent of an ensemble of nodes according to the OF. This is a flexible approach to construct this hierarchical architecture depending on a custom and variable criteria, represented by the OF. Depending on the OF definition, and which metrics are included the performance of the hierarchical structure varies.

This definition is not bounded to a single metric, which enables to better adapt the structure to different cases. This OF concept is a powerful tool that enables to aggregate different metrics and constraint to define routes that respect them. Therefore, it is an interesting feature to include in ISN due to its heterogeneity. Nevertheless, the RPL is not defined to manage nodes that are in motion, and needs to be extended to address this situation.

Other approaches leverages on the location of the nodes to estimate the energy consumption of a data transmission. GAF protocol uses this geolocated information to generate models of energy models corresponding to each transmission. With these models, the protocol determines cells where all the nodes are in standby except one. This one is the node that has a lower energy consumption profile. GEAR protocol combines this geographical information with the current energy state of the neighbor nodes. Therefore, this protocol aggregates these energy models with the current status of the node to determine the node that must be operative in the cell. Following the same premise, KP-RPL extends the original RPL protocol with geographical information. This information is computed by the received signal, and the Kalman filter. Although further research needs still to be carried out to address the situation of a full-mobile topology, using node position is a powerful information to predict energy consumption. Despite GNSS receivers are more frequent in satellite design, some satellites may not be equipped with this technology, limiting the possibility to retrieve geographical information.

Table 2.8 summarizes a trade-off between the previous routing protocols. The RPL is a promising routing protocol to deploy ISN because its flexibility and autonomy. In particular, the capability to implement different OFs as well as the freedom to select specific metrics makes this protocol a perfect candidate to manage ISN heterogeneity. Furthermore, the decoupling of metrics and constraints allows the developer to define new conditions that cannot only be related to performance, and it can be more oriented on strategy decisions. However, a large effort to translate this solution to a more mobile ad-hoc environment needs still be done.

2.6 Properties of a routing protocol

The development of a routing protocol that is able to leverage from the dynamism, opportunism, and sporadic behavior of ISN is fundamental to achieve the IoSat paradigm. Among the numerous routing protocols that have been conceived over the years, certain common properties can be detected. These properties classifies the behavior and mechanisms that the routing protocol use to define a route. This section review and present these properties to pave the discussion on the recommended and suitable properties that a routing protocol in IoSat may have.

The routing protocol is the responsible to determine the route between a source and destination node pair in a network. This route may be defined by a group of intermediate nodes that represent the sequences of hops or transmissions that a packet must perform from a source until reaching the destination. The protocols that defines a route following this ensemble of nodes are **multi-hop**. Other alternatives proposes the simple strategy to just transmit the packet without caring about the definition of a route itself. These protocols forward the packet **hop-by-hop**, assuming certain probability to reach the destination. As an illustrative comparison, replication-based protocols of DTN are hop-by-hop solutions while MANET protocols typically defines routes. Multi-hop protocols are more efficient that the hop-by-hop ones because they waste less the network capacity. Nevertheless, these multi-hop solutions requires the additional capabilities that allow to identify routes composed by nodes.

Table 2.8: Summary of Routing Protocols in WSN

Routing protocol	Interesting for ISN	Detrimental for ISN
Cluster approach		
LEACH	Simple - random distribution	Energy state not considered
	Cluster strategy - cannot manage mobility	
HEED	Energy state-based CH distribution	Communication always through CH
		Cluster strategy - cannot manage mobility
EHEED	Intelligent communication between non-CH nodes	Cluster strategy - cannot manage mobility
Dynamic Hierarchical approach		
RPL	Flexible, scalable, and autonomous	Cannot naturally manage mobility
	Topology creation using metrics and constraints	
RPL with OF0	Simple and easy to implement	Variation effect on rank computation
RPL with MRHOF	Hysteresis to avoid variations on rank computation	More complex mechanism
RPL with HC	Minimum hop path	Cannot manage congestion
RPL with AD	Minimum latency path	Cannot manage congestion
QU-RPL	Manage congestion scenarios	Aware of local congestion only
RPL-ETX	Path with less than a maximum of transmissions	Cannot manage congestion
IRPL	Combination of multiple metrics	A set of constraints are not used
Location-based approach		
GAF	Use of Energy model	Dynamic node activation
	Communication depending on energy consumption	
GEAR	Combination of position and energy state	Inaccuracy of node position
KP-RPL	Position improvement using Kalman filter	Important computation cost
	Extension of RPL	Mobile-to-static nodes communication only

Those multi-hop protocols can define a route by observing and reacting according to the current network topology. These **adaptive** protocols have the necessary capability to retrieve and understand different network metrics (e.g. link congestion, energy status, etc.) that enables to determines routes. Furthermore, these protocols are typically associated to a selection criterion which enables to choose the best route according to a specific condition (e.g. minimize the number of hops, or energy wasted). With this information and criteria, each node can construct the routing table which indicates how to reach each node with the best route. Other solution would be the installation of **static** routing tables in the nodes. This solution cannot react against network changes, but it can be suitable for small and controlled scenarios. As an example, MANET protocols are adaptive routing solutions.

Numerous adaptive protocols differ on how the routing tables are constructed. **Proactive** protocols are constantly observing the topology changes, and thus updating the routing table accordingly. Alternatively, **reactive** protocols only observes the network topology when data must be transmitted. An **hybrid** solution that combines both techniques has also been defined. These hybrid protocols perform the reactive discovery phase to suddenly keep observing the network as the proactive approach. Another hybrid approach is the definition of areas in which a reactive or a proactive behavior is conducted.

The adaptive routing protocols also presents differences on the amount of network information that is retrieved. **Link state** protocols retrieve details of all the nodes that conform the network. A global view of the entire scenario can be represented in each node using this kind of protocol. This enhances the possibility to identify network events (e.g. node congestion) in each node, and decide to change the route accordingly. Nevertheless, this solution may not be suitable for large-scale networks, which would require to store in each node large amount of information. As an alternative, **distance vector** protocols aims at retrieving global information by implicitly including in it from the local status. This approach enables a node to know how to reach a destination by using a neighbor node. This solution reduces the amount of information that needs to be shared, but the route definition depends on the neighbor information.

A feature of a routing protocol represents in which node the route is defined. **Connection-oriented** routing protocols are characterized by allowing the source to determine the entire route that a packet must follow. In this regard, when the packet exits the source, all intermediate nodes are already determined, and they just forward the packet. On the other hand, **connectionless** routing protocols allow each node to decide which is the best route to achieve a destination. The former type of protocols are faster than the connectionless ones, because they do not require to determine the route at each node. Nevertheless, they can react slowly to any network changes, instead of the connectionless protocols that may change the original route during the packet propagation.

A routing protocol can also enable the definition of routes in each node following a **distributed** manner. This approach enables to distribute the computation among the different nodes, and also include the definition of a route from the point of view of each node. Alternatively, a **centralized** protocol would promote the computation of the different available routes in a central processing node that have the information of all the network. This central node can be located in the same network, or it can be an external infrastructure. This approach enables to perform demanding computational operations with a dedicated infrastructure, which could not be performed in each node separately. Nevertheless, all the network depends on this central node, which could impact the operations in case of failure. A combination of strategies can also be defined as a **hierarchical** protocol, which defines different groups of nodes that can compute the routes of others. This approach enables to define hierarchy layers (e.g. clusters, altitude-layers) which are used to easily reach other nodes.

Some routing protocols are able to anticipate network changes over time, and thus determine routes that are succession of temporal links. These protocols use **predictable** mechanisms that allow estimate these network changes. These time-forward definition opens the door to a new concept of routes which are determined in the time domain, and can complement the common **instantaneous** end-to-end routes. These routes are commonly defined in a specific topology configuration. If this topology changes, new instantaneous routes would be defined in this new configuration. Unlike the previous routes, these routes are defined in the space domain.

Finally, the routing protocols define a unique route that is the best according to a specific criteria. This **single-path** protocol is commonly used in networks, but it cannot quickly react against network events. If this best route becomes unfeasible or its status changes over time, a recomputation of the best route must be conducted. Other approaches addressed this challenge by defining **multi-path** routing protocols. These protocols compute an ensemble of routes that may be ranked according to a criteria. This rank defines a priority usage, but ensures that if some route becomes unavailable there are still others that can be used. This approach ensures a quickly reaction, although a recomputation of the routes may be conducted lately.

These properties of a routing protocol have been identified after the review of numerous implementations, and helps to define a suitable protocol configuration for ISN. Table 2.9 summarizes these properties emphasizing the key concepts.

2.7 Recommendations for a future design

The review of the routing protocols developed for each satellite network type has exhibited numerous features and capabilities that may be appropriate to deploy ISN in the IoSat paradigm. The question falls on the existence of a current routing protocol that could directly be applied to deploy ISN. The discussion of the different capabilities has not stood out a single solution that could cover all the characteristics of ISN. Instead, a routing protocol that merges distinct capabilities from each current solution seems to be the optimal strategy. The identification of the protocol properties has been conducted in the previous section, without discussing their suitability for ISN. This discussion is conducted in the following paragraphs, making a recommendation of the best routing protocol that would include the necessary properties to deploy ISN in the IoSat paradigm. Before starting with the corresponding discussion, let's briefly review the features of ISN. This kind of satellite network is characterized by (1) having a dynamic topology due to satellite motion, (2) having nodes that follow the deterministic orbit trajectory, (3) being compounded of unreliable nodes that may decide to not participate in the network, (4) interconnecting nodes that are resource-constrained, (5) coexisting distinct technologies and operation strategies conforming an heterogeneous scenario, (6) being deployed sporadically and opportunistically, and (7) deploying distinct federations type which differ on traffic nature. The selected routing protocol must integrate all these features in its design.

Despite the ISN has three different phases–being the first one the establishment phase,–a **connectionless** protocol can better manage network mobility than a connection-oriented one. At the very beginning, it seems that a connection-oriented protocol follows the same concept of establishing the ISN by defining fixed paths. Nevertheless, since after the establishment of the path all the packets are strictly forwarded over the same sequence of nodes, this protocol is more influenced by the network changes. The route between a source and a destination may

Table 2.9: Relevant features of a generic routing protocol

Property	Description
Depending on the necessity to define a route	
Hop-by-Hop	It does not define a route, the packet is forwarded node-by-node.
Multi-hop	It defines a route which is composed of intermediate nodes.
Depending on the flexibility to forward a packet	
Connection-oriented	It defines a specific path between a pair of nodes that all packets follow.
Connectionless	It does not determine a specific route, different packets can follow different paths.
Depending on the interest in the network topology	
Static	It does not retrieve information about network topology and just applies a predefined routing policy.
Adaptive	It adapts the routing table depending on network topology.
Depending on the observation of the network topology	
Proactive	It periodically retrieves network topology in order to quickly act against a network change.
Reactive	It discovers the network topology if, and only if, data transmission shall be done.
Hybrid	It discovers the network and then periodically evaluates its topology (i.e. reactive and proactive).
Depending on the span of the network knowledge	
Link State	It retrieves information about the entire network topology.
Distance Vector	It provides global information implicitly through local nodes.
Depending on the topology structure	
Distributed	Each node computes independently the path to a destination.
Hierarchical	It defines routes depending on node ranks, creating a hierarchy.
Centralized	A central entity computes the entire routing tables of each network node.
Depending on being time aware	
Instantaneous	It defines routes in the current topology of the network (at specific moment).
Predictable	It uses node models to predict the topology over time (resources, position, etc.)
Depending on the alternatives to route	
Single-path	It defines a single route between a pair of nodes.
Multi-path	It defines a set of routes between a pair of nodes.

change over time, which would require the execution again of the algorithm to establish the connection. Therefore, the protocol would be more flexible and adaptable if the communication is performed without establishing a specific end-to-end connection. The establishment concept of ISN is related to the configuration of the satellites that would participate in the network to perceive this interconnected structure, rather than strictly a connection.

Addressing also the dynamism of the network, an **adaptive** routing protocol is able to learn the network topology, and reacts against events. Despite the satellite motion is deterministic, unpredictable events may still appear, due to the decision of a satellite to not participate in the network, its disassociation of the network once established, or the depletion of satellite resources. Therefore, the adaptation of the protocol would provide capabilities to work with unreliable nodes. Moreover, a **distributed** protocol would provide a more flexible solution than a centralized architecture. Those protocols that are ground dependent are optimized in terms of computation capacity, but this approach could limit network autonomy by requiring constant interaction with this ground infrastructure. Due to satellite motion, satellites have lapses of time where the communications with the ground is not established, and they operate–partially–as an autonomous node. Furthermore, different stakeholders may require to interact through a unique ground interface to deploy the network, which could limit the operations of a satellite. Hierarchical structures are more energy-efficient, but it requires that the parent or manager of the structure must be accessible. This accessibility cannot be ensured in a mobile environment in which network disruption is a common phenomenon. Therefore, the distributed architecture is the best option for ISN.

Both reactive and proactive protocols have their own benefits which makes them interesting depending on each scenario. Despite reactive protocols could be more energy-efficient, proactive protocols manage better the network changes. A balance between these two strategies is required in the context of satellite networks. Therefore, a **hybrid** routing protocol could integrate both approaches by deploying the network following a reactive manner, and maintain the network with a proactive approach once established. As ISN size is variable being possible to reach large number of satellites, a **distance vector** protocol consumes less memory, computing, and power resources. A link state protocol would require the replication of the routing tables from each node, which could be difficult if the size of the network becomes considerable. Nevertheless, the link state approach could be used in a local and small part of the network. A tradeoff between resource requirements and knowledge of the network exists.

A **multi-path** routing protocol can always better react against path failures or congestion scenarios. Therefore, this feature improves considerably the network performance, although the complexity and memory consumption will also increase. Furthermore, protocols that predict network topology allow having low signaling overhead, but they cannot quickly react well against congestion scenarios or node failures. The use of these predictable techniques could also anticipate network disruptions, and avoid the wast of resources. On the other hand, instantaneous protocols are able to detect unpredictable events, and adapt themselves to these events. ISN comprises this kind of event, therefore instantaneous protocols are also necessary to deploy this kind of network. A routing protocol that can integrated **predicted with instantaneous** routes is the ideal approach for ISN, which is the case of the LAOR protocol. Nevertheless, these predictions cannot be conducted by a central node,–as previously suggested–it has to be performed in a distributed manner.

The metrics that could be used to determine a route in ISN must be adaptable, and aggregate different satellite status metrics. Therefore, an approach like the OF from the RPL seems to be the best solution, because it performs an **integration of the metrics** with the possibility to include route constraints. This capability would properly manage the heterogeneity of the network by identifying routes that all the nodes are common capabilities required by federations. Finally, as it has been suggesting, the **multi-hop** approach is more efficient in terms of network capacity consumption, rather than the hop-by-hop propagation. Therefore, the routing protocol must have this property.

The identification of the required properties–that the routing protocol should include in its design to deploy ISN–allows to evaluate if previous solutions can directly be applied in the IoSat paradigm. Table 2.10 presents the comparison of the previous routing protocols according to the generic properties. The table also highlights in gray the recommended properties that this routing protocol should have. The suitability metric is a Figure of Merit (FoM) that corresponds to the number of desired properties are implemented by the protocol with respect to all the required properties (gray cells). This percentage allows to compare each protocol, and to determine which could be more suitable for ISN. The MANET and WSN protocols are the ones that are—in mean—more suitable to deploy ISN. Specifically, all the hybrid solutions prevail against the other reactive and proactive protocols. Nevertheless, these hybrid approaches are zone-based, which could not be entirely integrated in satellite networks. Only the AOMDV stands out in the reactive group, because the multi-path capability is a promising capability for ISN. All the protocols of the WSN have similar FoM, except the IRPL which has an 80 % as FoM. The capability to integrate different metrics in a single OF, which includes the status of the node resources, makes this protocol a relevant approach for ISN. However, the RPL family is not able to manage node mobility by default. Therefore, the protocol should be extended in order to be integrated in the IoSat paradigm.

The third group of protocols that could deploy ISN are the ones from the LEO satellite networks. Their distance-vector and multi-path properties make the PAR, PAR-MD and TLR protocol interesting for ISN. Nevertheless, this multi-path approach is achieved thanks to the custom satellite architecture. Therefore, this architecture limits the possibility to apply them in the IoSat paradigm which is characterized by being completely heterogeneous. DTN routing protocols are the fourth group with a 52.9 % of suitability. Among them, the replication-based approaches are more appropriate for ISN. This result is because forwarding-based solutions performs the prediction of the nodes following a centralized architecture. If this strategy is changed to a distributed architecture, the capability to predict network changes would increase the FoM of those protocols. Another improvement for ISN would be the integration of other metrics which include satellite resource status in the protocol. The solutions proposed in MLSN are close to the DTN ones, but they also lack certain features. The hierarchical structure of these systems makes that routing protocols are only suitable in these cases. If the protocols are enhanced with multi-path, and prediction capabilities, they would be more suitable for ISN. Finally, protocols developed for snapshot networks are strictly dependent on a central node to compute the snapshot sequence, which makes them completely inappropriate for ISN.

Among the different protocols, the MANET and the WSN are the most suitable ones to deploy ISN. Nevertheless, the maximum averaged FoM is 65 % which indicates that they are not perfect for satellite networks, and its performance may be impacted by this environment. Therefore, further investigations can be conducted to evaluate the performance of these protocols, and identify potential improvements to overcome these limitations.

Table 2.10: Tradeoff of rotuing protocols with respect to the desired protocol (gray properties).

Routing protocol	**Suitable [%]**	**Connection**	**Connectionless**	**Multi-hop**	**Hop-by-hop**	**Static**	**Adaptive**	**Proactive**	**Reactive**	**Hybrid**	**Link-State**	**Distance Vector**	**Distributed**	**Centralized**	**Hierarchical**	**Instantaneous**	**Time-forward**	**Single-path**	**Multi-path**	**Hop metric**	**Delay metric**	**Load metric**	**Resources**
Snapshot Networks (36.7 % of averaged suitability)																							
DT-DVTR	40	X		X			X	X			X		X				X	X		X			
PLSR	40		X	X			X	X			X			X			X	X		X			
SIR	30		X	X		X					X			X			X	X				X	
LEO Satellite Networks (62.9 % of averaged suitability)																							
DRA	60		X		X		X		X			X	X			X			X	X	X	X	
ELB	60		X		X		X	X				X	X			X			X			X	
PAR	70		X	X			X	X				X	X			X			X	X		X	
PAR-MD	70		X	X			X	X				X	X			X			X		X	X	
TLR	70		X	X			X	X				X	X			X			X		X	X	
ALBR	50		X	X			X	X			X		X			X		X			X		
DLAR	60		X		X		X		X			X	X				X		X	X		X	
Multi-Layered Satellite Networks (52.5 % of averaged suitability)																							
HQRP	40	X		X			X	X			X		X		X	X		X				X	
MLSR	60		X	X			X	X			X		X	X		X		X			X		
TLAR	60		X	X			X	X			X		X		X	X			X		X	X	
HCAR	50		X	X			X	X				X		X	X		X	X				X	
Mobile Ad-hoc Networks (65.0 % of averaged suitability)																							
AODV	60		X	X			X		X			X	X			X		X		X			
AOMDV	70		X	X			X		X			X	X			X			X	X			
OLSR	50		X	X			X	X			X		X			X		X		X			
DSDV	60		X	X			X	X				X	X			X		X		X			
BATMAN	60		X	X			X	X				X	X			X		X		X	X	X	
ZRP	70		X	X			X			X	X	X	X			X		X		X			
IZRP	70		X	X			X			X	X	X	X			X		X		X			
TZRP	70		X	X			X			X	X	X	X				X	X		X			
FSR	70		X	X			X			X	X	X	X				X	X		X			
LAOR	70		X	X			X		X			X	X			X	X	X			X		
Wireless Sensor Networks (65.0 % of averaged suitability)																							
GAF	50		X	X			X	X			X		X			X		X		X			
GEAR	70		X	X			X	X				X	X			X		X		X			X
LEACH	60		X	X			X	X				X	X		X	X		X					
HEED	70		X	X			X	X				X	X		X	X		X				X	X
EHEED	80		X	X			X	X				X	X		X	X			X			X	X
RPL	60		X	X			X	X				X	X		X	X		X					
RPL-HC	60		X	X			X	X				X	X		X	X		X		X			
RPL-AD	60		X	X			X	X				X	X		X	X		X			X		
RPL-ETX	70		X	X			X	X				X	X		X	X		X		X			X
QU-RPL	60		X	X			X	X				X	X		X	X		X				X	
IRPL	80		X	X			X	X				X	X		X	X			X	X	X	X	X
KP-RPL	60		X	X			X	X				X	X		X	X		X			X		
Delay and Disruption Tolerant Networks (52.9 % of averaged suitability)																							
Epidemic	40		X		X	X							X				X		X				
n-Epidemic	70		X		X		X			X		X	X				X		X				
EAEpidemic	70		X		X		X		X			X	X				X		X			X	X
MaxProp	70		X		X		X			X		X	X				X		X				
SaW	40		X		X	X							X				X		X				
MOVE	40		X		X		X	X			X		X				X	X		X	X		
MobiSpace	40		X		X		X	X			X		X				X	X		X			
RCM	40		X		X		X	X			X		X				X	X			X		
PROPHET	50		X		X		X	X				X		X			X	X		X		X	
DQN	60		X	X			X	X				X	X			X		X		X			
CGR	50		X	X			X	X				X		X			X	X			X		
OCGR	60		X	X			X	X				X	X				X	X			X		
DLSTR	60		X	X			X	X				X	X				X	X			X		
RAPID	50		X		X		X	X				X		X			X	X		X	X	X	

2.8 Summary

The current user demands require the development of new services that provide a considerable data transfer capacity with important downlink, uplink, and ISL interfaces; a communications interface with reduced end-to-end latency for time-sensitive data; and a fast access the data generated by the spacecraft. The distributed satellite architectures may satisfy this demand by the deployment of federations among satellites. These new approaches found on the value of inter-satellite communications capabilities and networked constellations designs to achieve the desired performance. These networked architectures suffer the impact of satellite motion, which provoke the fragmentation and isolation of satellite groups. Numerous satellite networks have been proposed over the years to deal with this challenge. Some of those proposals were defined to coexist with this disruption, and provide non-optimal solutions with delay-tolerant services that satisfy specific demand. Other approaches motivated the deployment of custom satellite architectures that would mitigate or eradicate this fragmentation. All these proposals does not assimilate the sporadic, opportunistic, heterogeneous, and decision-making nature of satellite federations, which may make them not optimal for this purpose.

Therefore, the IoSat paradigm has been presented in this chapter, emphasizing the deployment of sporadic and temporal satellite networks–the ISN–that enables the establishment of federations. These networks are compounded of satellites that decide to participate in the interconnected structure because they can retrieve a benefit; i.e. they are opportunistic. The details of ISN have also been presented, emphasizing the following characteristics: (1) a dynamic topology due to satellite motion, (2) nodes that follow the deterministic orbit trajectory, (3) compounded of unreliable nodes that may decide to not participate in the network, (4) interconnected nodes that are resource-constrained, (5) coexistence of distinct technologies and operation strategies conforming an heterogeneous scenario, (6) sporadic and opportunistic deployments, and (7) the establishment of distinct federations which differ on traffic nature. The definition of a route in this environment becomes a considerable challenge. Instead of starting the development of a custom routing protocol, a review of the similarities and synergies with other satellite networks–and their protocols—has been conducted. The outcome has indicated that none of the proposed protocols can directly be applied to deploy ISN.

To extend the discussion, Section 2.7 has suggested the necessary properties that a routing protocol should have to be appropriate for ISN. A mapping of the protocols developed for the other satellite networks to these routing protocol properties has also performed. This representation enabled to confirm that none of the protocols can directly used in the IoSat paradigm. Nevertheless, MANET and WSN solutions stood as a promising technologies that could be further investigated. The integration of prediction capabilities from DTN protocols, and multi-path strategy from LEO satellite networks could enhance the performance of the previous protocols in satellite networks. However, non of the routing protocols includes the capability to decide the participation in the network. This feature is crucial for the IoSat paradigm, and must be investigated in future solutions. The presented results correspond to a preliminary discussion that should be extended with the evaluation of the performance of certain protocols in satellite networks. The development of a simulation environment is required to achieve the following analysis.

3

Distributed Satellite Systems simulation engine

3.1 Introduction

FSS has emerged as a potential distributed satellite architecture to satisfy current user demands. This kind of DSS leverages on inter-satellite communications to establish satellite collaborations that share unallocated resources. By trading with and sharing underutilized spaceborne capacities, FSS concepts envision the formation of virtual systems that agglomerate multiple flying assets. Such innovative mission concepts suggest that satellite missions can be composed of multiple networked satellites that provide or consume services from a space infrastructure. Thus, satellites embarking excess capacities (e.g. computational power, downlink bandwidth) could share them with other federated nodes. These sporadic and opportunistic satellites interactions require the design of proper communications capabilities that enables the deployment of networked satellite systems. The networked architectures suffer the impact of satellite motion, which provoke the fragmentation and isolation of satellite groups. Therefore, the design of the system at the network-level is crucial to overcome this situation.

Chapter 2 presented the IoSat paradigm which conceives the space as an ensemble of sporadic and temporal networks that are configured to establish satellite federations. These networks—known as ISN—are composed by satellites that decide to participate in the establishment of the federation because they can retrieve a benefit. This opportunistic behavior poses additional challenges that may not be addressed in other satellite networks. In the same chapter, the similarities and synergies that other satellite networks (e.g. snapshot networks, LEO satellite networks, etc.) share with ISN have been discussed. This study has been performed at protocol level by discussing the suitable features of the protocols developed for each satellite network. The results suggest that no specific protocol can be directly applied in the IoSat paradigm to deploy ISN. Instead, a routing protocol that integrates distinct features from these current designs seems to be the best approach. However, among the different routing protocols, the ones developed for MANET and WSN have certain features that would be appropiate to satellite networks. This outcome motivates the extension of the study with the simulation of these protocols.

In general, designers of FSS are faced with the need to propose and verify new network protocols, federated negotiation approaches, or decision-making algorithms. The validation of these software technologies usually entails the simulation of systems and their environment and the generation of metrics to assess their quality. In the most simplistic case, satellite networks could be modeled as simple nodes that orbit the Earth, ignoring their functions and reducing their representation to communication devices. Many network simulation frameworks could be used in this pursue, which facilitate the modelling of moving devices. However, the de facto tools used by the industry and academia to simulate networks are tailored to ground applications and ignore scenarios where the communication devices are complex systems per se (i.e. with a different function other than those implementing the communications). Network simulators often provide a good collection of assets to model and implement custom protocols, signal propagation models, and physical devices like transceivers and antennae, but they hardly allow the implementation of non-network components (e.g. a battery) or other environmental and state variables that have direct impact upon links. Examples of the latter would be the Earth and its atmosphere, the state-of-charge of the batteries, attitude, pointing of ground station antennae, etc. The literature is certainly abundant with works that tackle resource-aware devices and which rely upon network simulation tools to demonstrate their designs (see Chapter 1, Section 1.3.11). However, these works–and the tools they rely upon–usually address systems in which the complexity of the modeled devices and the environment is not comparable to that of a satellite nor FSS. Likewise, none of the standard network simulation platforms include mobility models for objects that orbit around the Earth or are constrained by line-of-sight effects.

The development of fully-integrated tools that comprise the simulation of aerospace components and variables, and communications, network devices and protocols becomes essential for the design of FSS, and the IoSat paradigm. This work proved the feasibility of developing integrated tools that tackle the simulation of networked space systems. In this context, similar approaches should also be adopted in order to simulate DSS with the aforementioned level of detail. To the best of our knowledge, none of the previous works addressed the design of an integral simulation platform that facilitates the emulation of spacecraft components in a generic, extensible manner, nor considered a layer for autonomous, decentralized decision-making. None of the works have explored heterogeneous networks of satellites nor have provided modelling environments to define small-satellite platforms and their limitations. All these features are insofar lacking in any of the existing tools that could be used to study and validate FSS concepts, ISN protocols, and autonomous operations. This chapter introduces the design of this simulation engine oriented to DSS that requires inter-satellite communications.

The remaining of the chapter is structured as follows. First, Section 3.2 presents a global view of the goal and requirements of this software framework. The design of the framework is presented component-by-component in Section 3.3. Section 5.4 formulates the predictor based on relative orbit motion. The different metrics generated during the simulation, and the tools to visualize them are presented in Section 3.4. Finally, Section 3.5 concludes the chapter and remarks future research aspects on this topic.

The design presented in this chapter has been conceived in collaboration with Dr. Carles Araguz, which also conformed part of his Ph.D. dissertation (Araguz, 2019). Nowadays, the simulator is encompassed in an on-going project developed at UPC that was briefly introduced in (Ruiz-de-Azua et al., 2018d) and (Araguz et al., 2019). The details of these publications are accessible in Chapter 9 with the identifiers [IX], and [XV].

Domain	System layer	Emulation of	Time scale	Simulation span
Spacecraft operations	Autonomous decision-making and mission planning.	Management and allocation of resources, modes of operation, exchange of Application-level information.	Minutes to days	Weeks
Space systems	Spacecraft states and subsystems.	Orbit propagation, physical variables, emulation of subsystems and their interplay.	Seconds	Hours to days
Networking and comms.	In-orbit data services, ISL and downlink, network protocols.	Network-, Link-, and Physical-layer packets and control, propagation delays, data exchange.	Milliseconds to nano-seconds	Some orbital periods

Figure 3.1: Time scales in integral simulation of DSS networks. Figure from (Araguz, 2019).

3.2 Requirements for a Distributed Satellite Systems simulator

A software tool that allows the simulation of DSS missions and satellite communications is affected by two aspects that need be carefully considered: the combination of processes that evolve in different time scales, and the ability to simulate a very large number of components (i.e. satellites and their internal subsystems). In our case, three independent time scales have been identified according to the three groups of processes involved in the simulation: (1) networking and communications (in the order of milliseconds), (2) physical variables associated to subsystems (with changes occurring in seconds), and (3) spacecraft operations and mission planning (comprising operations that entail minutes to days). Therefore, the emulation of such system may also require different time spans (Figure 3.1). While some number of orbital periods may suffice in the simulation of routing protocols, extracting mission-level metrics may demand larger simulation spans. Combining the need to simulate long time periods and potentially large-scale DSS, undoubtedly forces us to propose software architectures in which performance and computational efficiency are two important factors. On the other hand, a simulation environment needs to allow for the definition of multiple system cases. As a result, the software should rather be designed as a generic platform wherein future researchers define their specific components and behavioral models.

The design has deliberately centered on the development of platform in which the complexity of spacecraft components is not a priori defined, nor the simulation scenario. This software allows the definition of custom models that represent a satellite—and its components—with the fidelity level that the experiment or test requires. In some cases, researchers may need the definition of a number of components to understand the low-level effects of some ISN protocols, whereas in other cases, defining a couple of simplified resource models would probably suffice for the evaluation of long-term mission performance of an autonomous constellation. Furthermore, a generic simulation platform should allow defining multiple scenarios that could be simulated with the same implemented models. As a consequence, we identify two types of users that could leverage from this platform:

1. Developers of custom models — they are users that require the verification and evaluation of custom models developed by themselves (e.g. spacecraft components, subsystems, physical variables, network protocols, or behavioral controllers).

2. Executors of satellite scenarios — they are users that leverages on the above-mentioned spacecraft components, which may not necessarily be implemented by themselves, to configure and simulate different scenarios.

The design of this software has tried to address both needs equally and proposed different design and configuration environments to allow the flexible definition of a system and a simulation case. Finally, two additional aspects are considered essential in a simulation environment oriented to satellite missions: (1) the ability to represent the system and its state in a graphical manner, and (2) the generation of reports that quantify the performance metrics. Therefore, Section 3.4 briefly presents the generation of these metrics, and the corresponding visualization.

The resulting software represents a programming framework that facilitates the extension of the simulation engine core with custom components to simulate DSS missions. With this architecture, the software is able to provide the main three goals: (1) the simulation of satellite-to-satellite communications, and its impact on satellite resources, (2) the capability to implement high-level spacecraft interactions (e.g. satellite operations, satellite federations), and (3) the development of an adaptable and highly extensible tool where users can tailor discrete physical models that represent spacecraft components and instruments.

3.3 Software design

As stated above, the simulation of DSS implies the combination of different processes that evolve over time with distinct time scales. This aspect conditions the use of another simulation execution approach than the common fixed-increment time progression. This progression proposes the fragmentation of the time in different slices after which the new state of the simulation is computed. This approach periodically updates the simulation with all the events that may happen between time slices. The existence of small and large time scales—from nanoseconds to days—makes the use of this progression inappropriate. If the time slice is defined in the order of nanoseconds, an entire simulation of a DSS scenario may require considerable execution time or it would become impossible to be executed. Meanwhile, large time slices may provoke the loss of fast events, resulting on inaccurate performance metrics. For that reason, the Network Simulator 3 (NS-3) has been adopted as the simulation engine core that will manage the simulation and schedule the execution of functionality. NS-3 is a discrete-event network simulator for Internet systems, targeted primarily for research and educational use. The use of discrete-event progression enables to efficiently manage different time scales by the execution of custom events during the simulation. The NS-3 is provided as an open source platform written in C++ that encompasses multiple libraries and components to emulate networks of different kind. The DSS Simulator is built on top of this core to extend the library of components and provide an environment to represent networks of satellites.

3.3.1 Software architecture

The software framework architecture is defined by three blocks that are sequentially executed: (1) the definition of the configuration parameters that represents the scenario to be simulated, (2) the execution of this scenario, and (3) the processing of the simulation results to generate the performance metrics. Figure 3.2 presents these blocks, and the flux of information generated during their execution. Let's review each of these blocks separately to better understand their functionality.

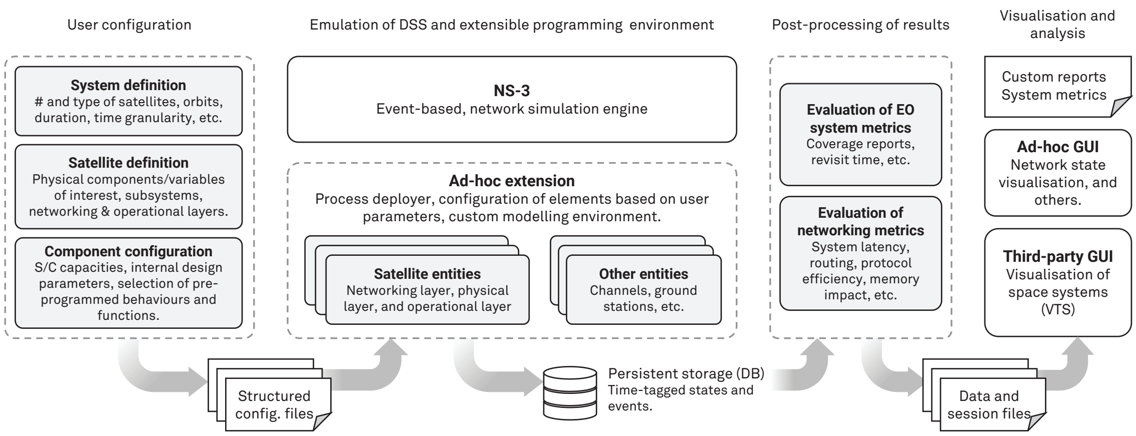


Figure 3.2: Simulator software architecture

Users are expected to provide the definition of the system in structured configuration files. These configuration files enumerate the components that will be generated and linked at runtime by the simulation software in order to represent the desired scenario. Three different file types are defined depending on the configuration level of the simulation. Some files detail the specific components or subsystems and configure their internal parametric variables (e.g. the capacity of a battery, the number of cells, etc). These components are associated to a satellite platform type by means of dedicated files. With the definition of these satellite types, a global configuration file determines the configuration of the satellite systems by providing orbital characteristics for each node and the definition of the network protocol stack. This modular configuration design enables to easily modify or extend a simulation scenario.

The software deploys the virtual DSS and hands the control over to the NS-3 event-driven engine. The emulation of satellite states, communication, and operations are individually tackled in three modules (Figure 3.3): (1) the physical module, (2) the networking module, and (3) the operations module. The architecture revolves around the definition of a network node (i.e. a satellite). Each satellite object encompasses a set of protocol instances—the network stack—that are paired with the definition of on-board capacities and can modify them during operations. This allows for memory-intensive services (e.g. like those of a FSS) to actually produce an impact upon satellite components and state. Likewise, communications of any kind (inter-satellite links or ground station links) are also simulated functionally through network devices and transceivers that belong to the networking module, but which do influence energy reservoir devices of the physical module. The operations module implements user-defined behavior (e.g. the self-organization rules, on-board scheduling of activities, flight software control). Leveraging on the built-in features of NS-3, these components are modeled as applications (i.e. the final entities in common TCP/IP protocol stacks). Their interface to communication devices is performed through emulated sockets, that encode the messages and data within the chosen transport protocol. Simultaneously, the operations layer has direct access to variables from the physical module with which subsystem commands can be emulated and critical spacecraft states can be read. The following sections presents further details about each of these modules.

The emulation of communications and satellite states generates data that is stored in indexed database registers for later access and exploitation. Once the simulation is completed, a post-processing process is launched which has two fundamental goals, namely, to prepare session files for the GUI, and to generate performance reports both for network processes, and functional aspects of the mission (i.e. Earth observation performance metrics).

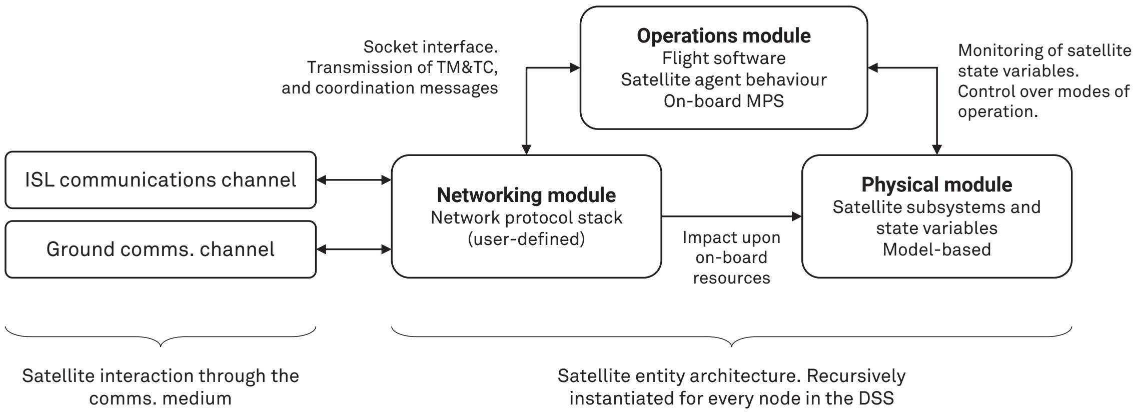


Figure 3.3: Three different modules define the behavior of satellite nodes.

3.3.2 Physical module

Users are provided with a flexible and extensible modeling environment in which physical spacecraft components are defined. These components emulate the functionality of subsystems of a spacecraft (e.g. satellite bus, payloads, etc.) or essentially any relevant physical variable (e.g. state-of-charge). In order to implement these components, the a generic model is defined as the basis class over which all the physical components must be extended. This model provides the necessary interfaces to interconnect a sequence of extended models. Models are thus the main wrappers for the implementation of custom devices, subsystem functionalities, and physical components. Two characteristics define a model, and the behavior of its derived sub-classes. The former is the definition of its interface as a set of input, output, and state variables. All of them are parameters that represents the state of the model, and only the inputs and outputs are externally accessible from other models. The other characteristic is the process to update the state of the model. This process is determined by the loop four operations: (1) the initialization of the parameters with the user-defined original value and the model associations; (2) the update loop of a model starts by verifying that all input variables are set; (3) the update of the state is implemented by the proper behavior of the model, which should read the input values, process them, and generate the corresponding output; and (4) once NS-3 triggers the next model update cycle, this function transitions all the outputs in order to flag possible changes. This allows subsequent models to identify the need to update their own states (since some of their inputs might have changed). Figure 3.4 presents a diagram of the model with its interface, and the execution loop.

The execution of each model is asynchronous and depends upon implementation. Some models may encode the behavior of physical variables with slower time dynamics (e.g. a battery state-of-charge) whereas others may require faster updates (e.g. attitude). However, models are linked in static networks that define provider-consumer dependencies. Changing the outputs of a provider may trigger the update of its consumers. This dependency can be represented as Directed Acyclic Graphs (DAG). The control mechanisms of the physical module ensure that the updating of model objects is carried out in the correct sequential order. In order to do so, the network of model dependencies is inspected—on a satellite-by-satellite basis—right after the linking process. Furthermore, model-derived entities can explicitly register their input variables as triggers of their update cycle. In such cases, a change in their linked output forces the update during the same absolute time. Conversely, models can also register their inputs as non-critical values. This set-up does not force their update upon changes in the corresponding output.

Further details of the models and their functionality are presented in the dissertation of Araguz (Araguz, 2019).

3.3.3 Networking module

The NS-3[8] is a discrete-event simulation engine specifically tailored to emulate communication systems. By means of linking multiple protocols, a custom protocol stack can be installed in a node as part of its configuration. The NS-3 core provides a set of libraries that implement the most common and standard Internet protocols, such as the User Datagram Protocol (UDP) or the IPv6. This engine also allows a user to customize these implemented protocols by adjusting their parameters (e.g. transmission windows, timeouts values, etc.) The potential of its design falls on the possibility to develop custom protocol implementations that can easily be integrated in the stack, and linking them to common Internet protocols. A developer can conceive a protocol with the desired level of fidelity to evaluate its performance. This modular and extensible architecture facilitates the research and development of new solutions for different scenarios. Therefore, NS-3 has largely been used to evaluate new wireless network solutions, including those in which nodes are characterized by a mobility pattern. These capabilities make NS-3 an extremely appropiate engine to design and assess inter-satellite communications in the context of DSS and satellite networks.

Among the different protocol modules that may conform a complete stack, we may mention the ones that emulate the physical communication environment: the `NetDevice` and `Channel` modules. These abstract classes represent the communications technology integrated in a node, and the corresponding medium over which the communications among devices is performed. From the literature of personal computer, a `NetDevice` emulates a hardware component of a node that allows communicating with other nodes (e.g. Ethernet card, WiFi board, USB module, etc.) This module should integrate the functionalities corresponding to this low-level of the communications, such as signal modulation, encoding, among others. Traditionally, the devices have also provided functions to manage the link interaction between two nodes, such as medium access, link-buffer congestion, among others. As a global view, this module may provide a merge of the physical and link layers' functionalities defined in the Open System Interconnection (OSI) model. These devices are interconnected over the `Channel` module, which represents the communications environment. This module emulates the physical aspects of the communications, such as signal propagation, mobility impact, the share—or not—of the

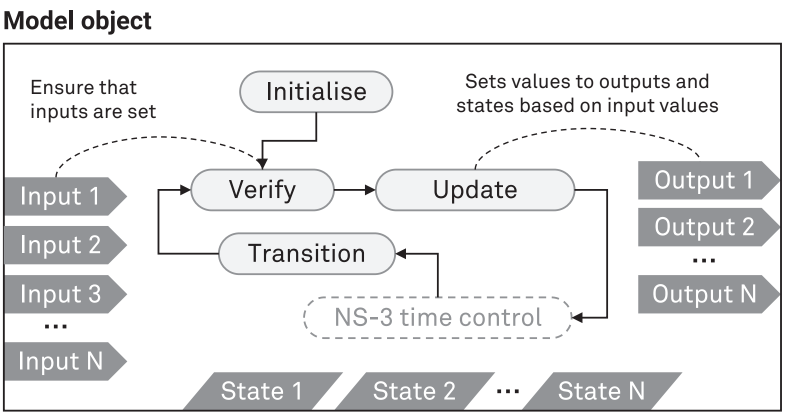


Figure 3.4: Model object

[8] `https://www.nsnam.org`

medium, etc. Despite NS-3 presents these generic and abstract modules, the emulation of the physical elements in the communications are bounded to each scenario that is intended to simulate. While personal computers may communicate over a static and wired infrastructure, mobile nodes interact over a wireless environment. Both environments presents different characteristics that cannot be unified in a single and generic module. Therefore, these two modules may be extended with the dedicated functionalities of the expected environment.

The emulation of DSS and satellite networks requires the development of a custom device and channel that represent the dynamic environment of satellite-to-satellite and satellite-to-ground communications. A dedicated project was presented in the ESA Summer of Code in Space (SOCIS) to develop a mobility pattern for satellites (Nsnam, 2019), but—to the best of our knowledge—the project has never started. Therefore, we decided to design the corresponding device and channel that represent this satellite mobility and its corresponding effects in the communications. The resulting modules are named `SpaceNetDevice` and `SpaceChannel`.

The `SpaceNetDevice` extends the original device module with the mobility pattern of a satellite. This device has specific functionalities related to the transmission and reception of a packet. The packets that are forwarded from upper protocols in the stack are sequentially processed according to their arrival time at the device; i.e. a first in, first out (FIFO) strategy is applied. The device is characterized by a transmission data rate which determines the lapse of time that a certain packet requires to be transmitted—according to the size of the packet. If another packet is forwarded to the device while a transmission is being conducted, this packet is stored in a queue waiting to be transmitted later in time. This queue is indeed the one that ensures that the FIFO process is accomplished. The transmission of a packet of the communications channel is performed with a configurable EIRP, and at certain central frequency. When a packet is received through the channel into a device, the sensitivity parameters determines if the received signal is enough to be correctly processed. In case that the packet has enough power, the next step corresponds to the characterization of erroneous bits due to this received power. This process emulates the Signal to Noise Ration (SNR) effects on the bits that conform the packet. After this, the packet is processed byte level to lately be forwarded to upper protocols. An address that identifies the node is inserted in all the packets that are transmitted to the channel. By default, this address corresponds to a 48 bits IEEE standard address (EUI-48), but it can be replaced to custom addresses. The use of a distinctive address enables the device to discard those packets that are not destined to it before forward them to upper protocols. Despite unicast addresses are used to identify the node, broadcast addresses are also accepted by the implementation of the `SpaceNetDevice`.

These devices are interconnected over the `SpaceChannel` that emulates the communications environment related to the space and the atmosphere. This channel has been designed taking in consideration the heterogeneity of DSS and FSS which may be reflected in different communications subsystems. Satellites working with different physical parameters (e.g. central frequency, transmission power, etc.) may coexist in this environment. Only those satellites that have similar communications features may communicate among them. Therefore, the `SpaceChannel` is designed considering different planes that are determined by the central frequency, and data rate pair. A combination of both parameters generate a `ContactChannel` module, which encompasses all the devices that work at the same frequency and data rate. This plane ensures that only the satellite that belong to the plane can communicate among them; i.e. the communication between planes is forbidden per se. The `ContactChannel` is composed of numerous communications links among the different satellites that belongs to the plane, which each one corresponds to a `Contact`. This contact emulates the communications medium of an

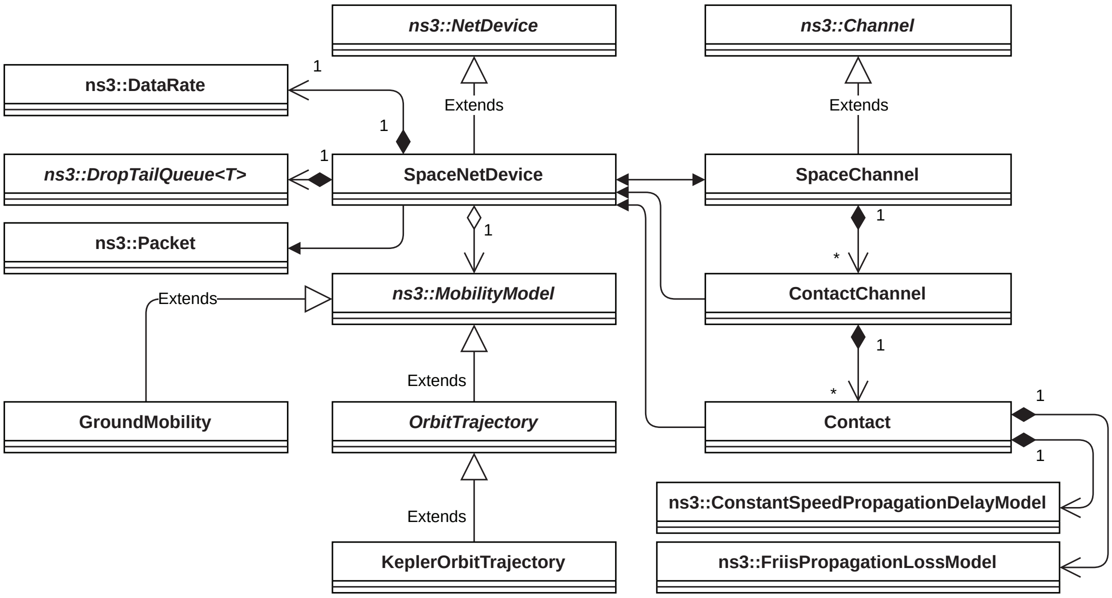


Figure 3.5: Simplified UML diagram of the channel representation in the DSS simulator

unidirectional link between a satellite transmitter and another satellite receiver. These modules are responsible for the characterization of the propagation of the signal with the corresponding power loss, and propagation time. The propagation can be conducted following a Free Space Path Loss (FSPL) model for a satellite-to-satellite communications, or an atmospheric model for satellite-to-ground communications.

The propagation loss is also determined by the distance between the satellites of the contact. Therefore, the contact is able to extract the orbital pattern from each device that conforms the contact. This pattern is emulated with the `OrbitTrajectory` which is an abstract module that represents a generic orbit. By extending this class, a user can implement different orbit models with the required fidelity. The default orbit model is the Keplerian representation, but other propagators can be used (e.g. SGP4). This module ensures that any orbit extension respects the interface required by the `Contact` module, such as the retrieval of the satellite position and velocity in the Earth-Centered Inertial (ECI) frame. With this channel representation (Figure 3.5), the intermittent communications of two satellites is naturally emulated thanks to the evolution of the distance. The corresponding line-of-sight of a satellite is modeled directly by the received power level at the device, instead of evaluating specific conditions to detect this situation–as in other simulation cases. This approach literally emulates the proper propagation, and represents the nature of ISL per se. Ionospheric and atmospheric effects are not currently included in this communications channel, but they can be easily integrated in future developments. Nevertheless, these effects may occur in large communication distances, such as greater than 5000 km.

Figure 3.6 represents an illustrative case with four satellites that generate three distinct planes in the channel (green, red, and yellow, in the figure). Each satellite has a contact—represented in boxes—which enables to communicate with the other satellites if they share the same plane. Otherwise, the communications is not feasible, like satellites 1 and 4. When a satellite transmits a packet, this one travels through the `SpaceChannel` which forwards it to the corresponding plane. In this plane, the packet is copied in each contact in which the communications is feasible; i.e. the Earth is not in the middle between two satellites. The packet is then received by the satellites which perform the corresponding processing–previously presented. If

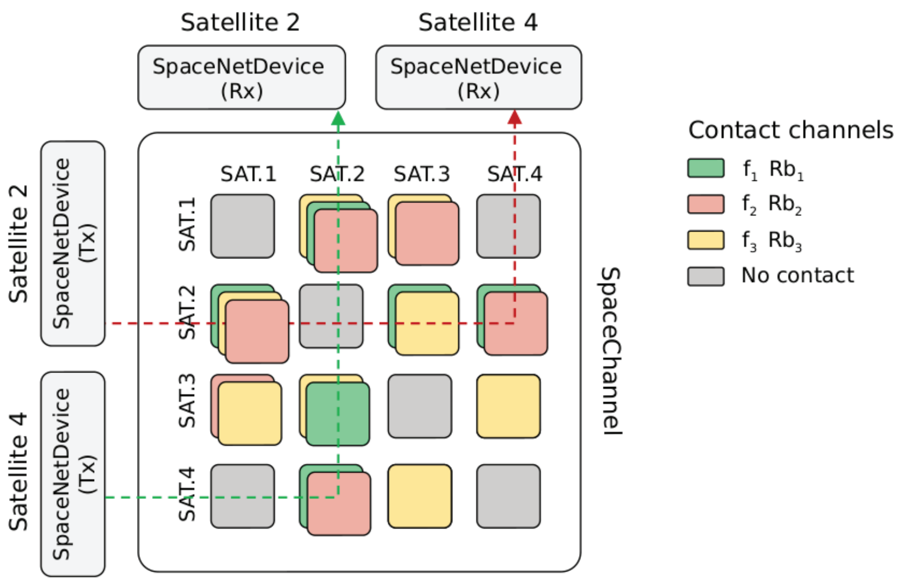


Figure 3.6: Contact objects are generated per each frequency (f_i) and data-rate (Rb_i) if they are supported by the satellite's devices (*SpaceNetDevice)*.

one of the satellites decides to reply, the transmission is performed again from this satellite. This design enables a satellite to have numerous devices that work at different frequency and data rate, like satellite 2 in the example. With this capacity, the upper protocols can select with which technology the transmission is performed. The figure also illustrates this situation by a transmission from satellite 2 to satellite 4 using the technology represented in the red plane, and the corresponding reply from satellite 4 over the green plane. This approach allows extending the satellite with new communications technologies, like optical communications.

The proposed design to emulate the satellite-to-satellite and satellite-to-ground communications opens the door to investigate custom protocols over this environment. As the following chapters will present, this is a powerful capability of the developed software which allows evaluate protocols that were not intended to work in this satellite environment. Furthermore, the software enables the possibility to extend the different modules that represents the communications medium with custom propagation models that can include additional effects (e.g. ionosphere, rains, etc.) In summary, this extensible and modular approach is ideal to keep investigating on satellite networks.

3.3.4 Operations module

The operations of a satellite correspond to the tasks or system functions that are executed to achieve the mission. These tasks may be defined by the user to indicate the behavior of a satellite (e.g. self-organization rules, on-board scheduling of activities, flight software control). This behavior is emulated following the same premise than in NS-3: they correspond to application software in the OSI model. These applications are modeled as final entities in common TCP/IP protocol stacks. The functionality of each application may be implemented specifically to the behavior that is expected to achieve, and by extending the module provided by NS-3. As some tasks in satellites follow a periodic execution, we have developed a specific application that periodically executes a custom function. This application triggers an event to process the corresponding function at every time period. This abstract module may be extended to achieve the desired periodicity.

The applications interact with the protocol stack–the networking module–by means of dedicated sockets. These sockets emulates current computer implementations to ensure a high fidelity. Therefore, the sockets from NS-3 offer the traditional and well-known UNIX interface (e.g. open, write, read, etc.) Thanks to this approach the implementation of the application can be easily integrated in any UNIX-like system, like Linux. These sockets are interfaces to one of the protocols in the stack. Typically, the protocols that implements the functionalities related to the transport layer are the ones that interact with the sockets (e.g. UDP and TCP). Nevertheless, sockets to transfer raw data are also available for custom designs.

Simultaneously, the applications have direct access to the models that compound the physical module of the satellite. This interaction is achieved by reading the variables of each model–the outputs–, and by modifying the inputs. With this approach subsystem commands can be emulate, and critical spacecraft states can be read to address the corresponding action.

3.4 Data processing and visualization

The execution of the simulation scenario generates data related to the satellites status, and communications performance. This data is generated by each of the modules configured in the corresponding user files. During the execution, a module can store data into a persistent relational database. This database organizes the incoming data into a single table with columns that represent the type of information stored, and rows that are an instance of the simulation status. These rows correspond to table entries which are identified by a unique number, an agent, a keyname, and the simulation time when the entry is recorded. The agent parameter represents the satellite identifier that is storing the information, while the keyname corresponds to the module identifier. The combination of both parameters enables to know which module being executed in a satellite is storing the corresponding information. This information is stored in a JSON string that provides flexibility on the kind of data that is recorded. using this format a complex data objects is represented in a single field. For the sake of illustration, consider a battery model that computes the state-of-charge as well as its temperature. These values can be stored in a single JSON field, as follows: `{"soc":0.5, "temp":27.5}`. This format makes the system flexible to the generation of new models with arbitrary data fields without the need to know their structures beforehand.

The resulting database of the simulation is used to generate the relevant performance metrics. The database stores thus raw data of the simulation, which needs to be post-processed to retrieve mission and system metrics. These metrics can be computed after the simulation with scripts that access to the database. The resulting metrics can represent status of the satellite (e.g. battery state-of-charge, memory usage, satellite positions), performance of the system (e.g. revisit time, coverage area), and communications behavior (e.g. percentage of delivered packets, topology representation, latency). Despite some scripts are implemented in the same software framework, the design is extensible with custom implementations. The use of a persistent database enables to easily export the results in case that it is desired.

Simultaneously to this post-processing, the database is also accessible to a Graphical User Interface (GUI) to visualize the relevant behavior of the simulation (Figure 3.7). The implementation of this GUI is realized with VTS Timeloop[9], a third-party software developed by the French National Centre for Space Studies (CNES) and used in multiple missions and software frameworks both at CNES and the European Space Agency. VTS provides a configurable interface that

[9] `https://timeloop.fr/vts/`

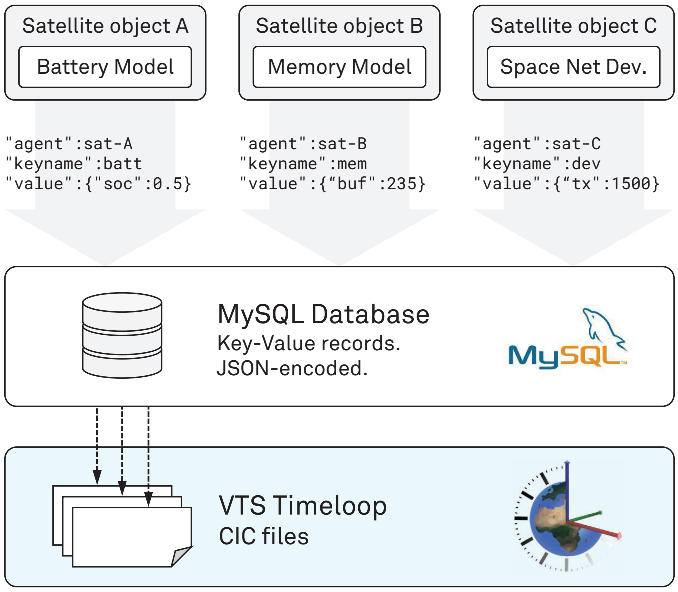


Figure 3.7: Data flow of the generated data

can be used to display multiple kinds of space systems, state variables, and projections. The software defines a custom data protocol based on plain text files with which users can provide the data to visualize. Some of the benefits of VTS include, but are not limited to:

- Flexibility and reliability to display any kind of systems (be it EO DSS, single-satellite missions, extraplanetary probe missions, etc) and their variables of interest. The software has been used in some areas of the operations for ESA programmes and missions, including SMOS, or Rosetta.
- It provides built-in visualization modules (e.g. Celestia, SurfaceView) that facilitate the representation of systems in 3D or planar Earth projections.
- It is well documented and extensible. VTS can be paired with custom visualization engines that implement their communication interface. Thus, multiple views of the system are seamlessly synchronized and controlled from VTS.
- Visualization can be either in real-time (i.e. from a constant stream of data) or from previously computed data stored in files. The DSS implementation leverages this second mode of operations (i.e. offline) to let users observe the results of a simulation with full control of time.
- At the time of writing this document VTS was an actively maintained project, receiving periodic updates and new features.
- It is freeware and endorsed by competent actors in the space industry (ESA, CNES).

As a brief example of the type of visualization that can be performed with this tool, Figure 3.8 presents the trajectory of the Telesat mega-constellation in a surface view. This trajectory has been retrieved by the execution of one simulated day. Further views are available in the VTS which enables to evaluate the performance of the satellites or the system.

3.5 Summary

This chapter has motivated the design and implementation of ad-hoc simulation software frameworks that can represent and emulate complex DSS which require inter-satellite com-

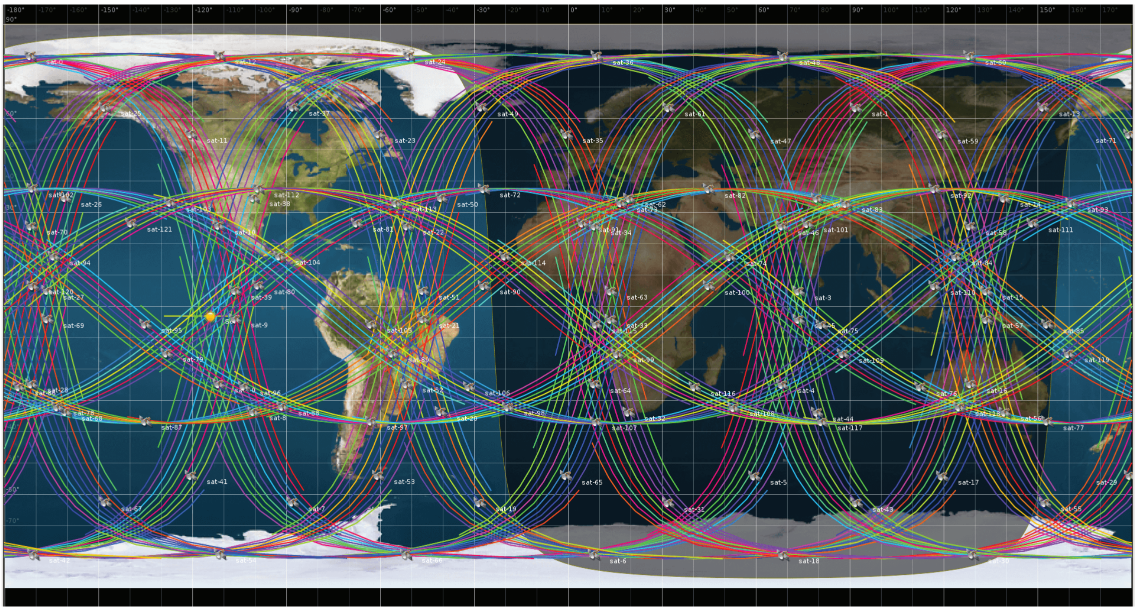


Figure 3.8: Surface view with the Telesat mega-constellation from the VTS viewer.

munications. The purpose of the presented design is twofold: (1) to enable the simulation of autonomous DSS, and (2) to emulate ISL communications with fidelity. This software is an ongoing research project at UPC that is briefly introduced in (Araguz et al., 2019; Ruiz-de-Azua et al., 2018d), and is expected to be released in the future. The software is aimed at becoming an extensible support tool for future research groups.

Implemented on top of the NS-3 core, the simulator encompasses a programming environment through which custom satellite components can be modeled and controlled, as well as a user configuration area that facilitates the definition of multiple different DSS contexts. We have briefly presented how the architectural definition of a satellite is composed of three unique modules: operations, networking, and physical. Satellite behavior (e.g. modes of operations, execution of tasks) and flight software is emulated in the operations module, which leverages NS-3 components and is internally implemented as the application layer of a network node. The physical layer encompasses as many self-contained representations of subsystems, devices, or abstract models provided by the user. Finally, the networking module allows the user to define their own custom network stack, enabling the design, implementation, and verification of new ISN protocols. By means of a dedicated design, the software framework is able to emulate satellite-to-satellite and satellite-to-ground communications which are characterized by intermittent contacts. This framework encompasses thus the three main modules that could be critical technological enablers for future research on networked satellite systems. Therefore, this framework has been used in the investigations conducted during this thesis research, which are presented in the following chapters.

4

Assessment of the OLSR protocol performance to deploy Inter Satellite Networks

4.1 Introduction

The IoSat paradigm proposes the deployment of opportunistic and temporal satellite networks—the ISN—to conform the necessary infrastructure to establish satellite federations. The definition of the routing protocol—that determines the routes among the satellites that conform the federations—becomes an important challenge due to its sporadic, opportunistic and temporal nature. We have discussed in Chapter 1 the desired properties that a generic routing protocol should integrate in its design to deploy such satellite networks. Furthermore, the discussion has been extended to compare this ideal protocol with current implementations for other satellite networks. Among the different protocols presented in the literature, the WSN and MANET ones seem to present key features for the deployment of ISN.

The routing protocols developed for WSN has the ability to differentiate routes according to a group of metrics and constraints. These metrics may be the commonly used in routing protocols—i.e. number of hops, delay—, but they can be extended to other metrics related to the status of the nodes. These last metrics represents the status of the resources that are allocated in a node, such as batter state-of-charge, memory usage, among others. This concern about the node resources is essential in a network composed of embedded systems, like sensors or satellites. Is this wider perspective to determine a route what makes WSN routing protocols appropiate for ISN. Nevertheless, these protocols were not originally conceived for movable nodes, which complicates its direct application in satellite networks. On the oter hand, the flexibility and adaptability of the protocols conceived for MANET are suitable capabilities to address the dynamic environment of ISN. The possibility to observe the topology changes would allow reacting to topology changes due to satellite motion. Among the different protocols, the hybrids routing protocols has been the most appropiate for ISN by combining reactive and proactive mechanisms. The hybridization of these protocols is founded on the definition of a zone which differentiates the area where proactive mechanisms are used from the one where the reactive techniques are applied. This approach may not be suitable for satellite networks, which would be preferable to combine the discovery from reactive approaches with the proactive maintenance of the network. Therefore, proactive routing protocols seem to be more appropiate for this kind of networks.

This chapter extends the tradeoff performed previously by simulating a scenario where satellites deploy a network using the OLSR protocol (Clausen and Jacquet, 2003). This protocol has largely been used in other connected infrastructures, such as in networks composed of ground or flying vehicles (Haerri et al., 2006; Leonov and Litvinov, 2018; Singh and Verma, 2014). The protocol was also used in (Lluch et al., 2015) to deploy federations among satellites. Despite the results demonstrated the possibility to establish these federations, a discussion of the implications of using this protocol was not conducted. Therefore, a study of the viability of the protocol to deploy networks in satellite system would provide be a step forward. Furthermore, the outcome generated will identify the advantages of this protocol to address satellite motion, and the drawbacks related to the resource consumption and network disruption. This assessment is performed by simulating the OLSR protocol in a realistic scenario. Considering the current demanded new services (see Chapter 1, Section 1.1), a satellite system composed of heterogeneous spacecraft platforms that monitor the Arctic region has been selected as a representative scenario. The simulation is conducted with the software tool presented in Chapter 3 which has been designed to evaluate the performance of DSS missions.

The main contribution of the present research is 1) a study of the viability of OLSR protocol in satellite systems, 2) a topology study of a satellite network by indetinfying the kye parameters that characterize its nature, and 3) recommendations to improve current technologies to be integrated in satellite networks. The research presented in this chapter has been published in the peer-reviewed IEEE Access journal (Ruiz-de-Azua et al., 2018b). The details of this publication are accessible in Chapter 9 with the identifier [III].

The remainder of the chapter is structured as follows. Despite the OLSR protocol has previously introduced, Section 4.2 presents in details the protocol and its behavior. The simulation scenario and the satellite models are then presented in Section 4.3. Characteristics of the simulation engine are detailed in Section 6.5. The analysis and results are discussed in Section 7.7. Finally, Section 4.5 summarizes the work, and identifies future research lines.

4.2 Optimized Link-State Routing Protocol

The OLSR protocol (Clausen and Jacquet, 2003) was conceived to define routes among a source and a destination node pair in a network compounded of mobile nodes. This kind of network is characterized by having a variable topology, where routes changes constantly. Despite these changes, routes that connect the nodes always exist, being thus not impacted by network disruption. Therefore, only the composition of these routes is what varies; i.e. the sequence of intermediate nodes that connects two remote satellites. The challenge in this environment is the definition of these routes over time, which are stored into the routing table of each satellite. The OLSR protocol is provided with the capability to learn about the network topology, and adapt itself to the topology changes. This observation is conducted proactively by periodically transmitting control messages. Specifically, each node in the network periodically sends to its neighbors a `hello` message, which contains information to determine the MPR nodes. The MPR are specific nodes that are willing to carry or aggregate data from other nodes. With this kind of node, the network capacity is optimized by just propagating data among MPR nodes.

The transmission of the hello messages allows thus to accomplish three goals: (1) the observation of the link status that a node has with its neighbors, (2) the detection of neighbors that a node has around, and (3) the selection of a MPR node among its neighbors. A node that is executing the OLSR protocol keeps three different data sets in memory that corresponds to distinct levels of network information: (1) the Link Set contains the information of all the links, (2) the Neighbor Set contains the information of the neighbor nodes, and (3) the MPR Set contains a list of all the neighbors that have been MPR. The hello message is generated with the information of these three different sets. Let's first review of these sets are constructed.

The process of populating the Link Set is called link sensing, and allows detecting bidirectional links established between a node and its neighbors. This process is performed by exchanging hello messages between adjacent nodes. Upon receiving a hello message, a node processes the message to evaluate if the link with the originator of the message is new or not. In case that it is new, the node updates the Link Set with a new entry that includes the neighbor address, and sets a limit time after which the record would be expired. If the link was already stored in the set, the node just updates the limit time that indicates when to expire the entry. This expiration of an entry indicates that the link is no longer available. Instead of removing the entry, this is propagated in the hello message, to notify other nodes. The reception of the hello messages also enables to identify if a link is symmetric or asymmetric—bidirectional or unidirectional—, which is crucial to define the network topology.

The process of populating the Neighbor Set is called neighbor detection, and it is executed after the link sensing. This set is a list of the different neighbors that a node has active links. Therefore, the set is created with the information of the Link Set. When a new entry is included in the Link Set, a new entry with the corresponding neighbor address is also included in the Neighbor Set. If an entry in the Link Set is updated, the corresponding entry in the Neighbor Set must be updated. These updates corresponds to a change on the status which indicate also if the communications with the node is symmetric or not. In case that a link entry is removed, the corresponding neighbor entry is removed if no more links with this node are active. The differentiation of these two sets allows to quantify the different links with the same neighbor. In addition to this Neighbor Set, a node also keeps a 2-hop Neighbor Set which corresponds to a list of the nodes that have a symmetric link with the neighbors. These nodes are thus–at least–reachable over the neighbors of a node; i.e. after 2 hops.

The MPR Set is populated with the information that is included in the hello message. In addition to the type of link, this message also identifies a neighbor that is working as a MPR node. Each node is able to select the MPR nodes that ensures the bidirectional communications with all the 2-hop neighbors. The selection of thes MPR nodes conforms the MPR Set. The selection process is founded on the willingness of each node–propagated also in the hello message—and is defined with an heuristic method. This process starts by considering all the—1 hop—neighbors as a candidates to be MPR. The filtering of those nodes according to their connectivity with the 2-hop Neighbor Set allows to determine the final MPR Set. This MPR Set corresponds to the neighbor list that ensures that a node can communicate with all the nodes located at 2 hops. The nodes that are considered MPR are tagged to be notified with the hello message. Apart of the MPR Set, each node also keeps a MPR Selector Set which includes the list of neighbors that considers the node as a MPR. The set is populated when a hello message is received, and contains the address of the node—that has received the message—with the tag of MPR. This MPR Selector Set allows a node to be aware of which nodes it is responsible to relay data.

Once all the nodes identify the possible MPR, and are aware of those that consider them as MPR, the different connections of the entire network is collected for every node. Therefore,

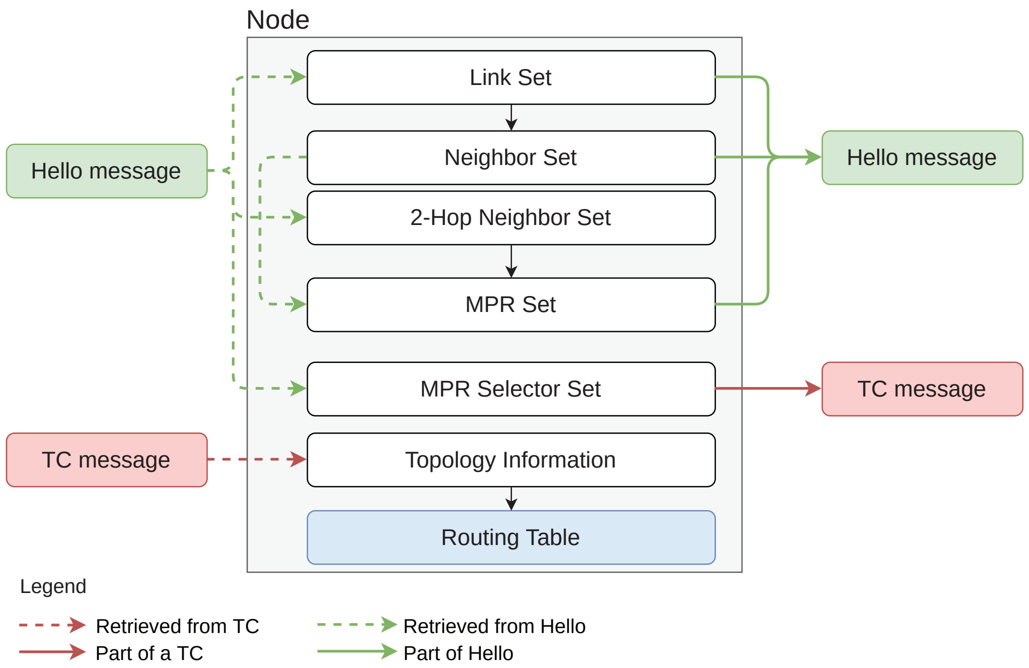


Figure 4.1: Representation of the different sets generated in a node that executes the OLSR protocol

the topology with the MPR nodes can be constructed, which is performed with the exchange of Topology Control (TC) messages. Using these messages, MPR nodes advertises the symmetric links with—at least—the MPR Selector Set. These TC messages are the ones that are broadcasted to the entire network following a flooding approach. The information included in each TC message helps to calculate in each node the routing table. This table is composed of entries that corresponds to routes to all the nodes that conforms the network. These entries indicates how long—number of hops—a destination can be reached among the corresponding neighbor. The definition of these entries is performed using a shorter path algorithm. Any time that a link changes, all the sets are updated with the corresponding message, and also the routing table. Figure 4.1 aims at clarifying the different sets of a node with a dedicated diagram.

The broadcast of TC messages by only the MPR nodes reduces—partially—the consumption of the network capacity. Therefore, this protocol optimizes the original Link State Routing protocol with which all the nodes broadcast its local link with its neighbors. The periodic transmission of these TC messages allows to notify changes in the network, and recompute the routing table accordingly. Therefore, the nodes with this proactive protocol know and maintain the routes to all the destinations before use them. However, this protocol does only considers that links are bi-modal—symmetric or asymmetric—instead of using link quality. The implementation OLSRd for Linux includes this link quality sensing. For all these features, OLSR has become one of the most deployed portocols in MANET. Its capacity to quickly react against network changes, make it interesting to evaluate in satellite networks.

4.3 Polar satellite mission scenario

The viability of the OLSR protocol to interconnect satellite systems must be studied in a realistic scenario. Therefore, a satellite mission to observe the Arctic region has been defined. The deployment of new services that monitor this regions are the most demanded ones by the EO community (see Chapter 1). Therefore, the simulation of the OLSR protocol in this context would motivate the discussion of its benefits and limitations for these missions. The modeling of the scenario becomes crucial to retrieve faithful results. This section presents the different elements that has been decided to conform the simulation scenario. This scenario is compounded of operative satellites which are equipped with different payloads designed to measure the ice status in the Arctic region. The resulting mission configuration is a heterogeneous satellite system, in which satellites have different capabilities depending on their platform. Section 4.3.1 presents the different details of this spacecraft platform model, while the payload and downlink system models are introduced in Section 6.4.1. Section 4.3.3 presents the set of current operational satellites, and their orbit parameters.

4.3.1 Platform Model

The ONION project (Alarcón et al., 2018) aimed to investigate the capabilities of DSS to enhance current and future missions that generate services for the Copernicus programme. Among the different outcomes, the project emphasized the potential of combining satellites with different capabilities to improve current mission performance. This heterogeneous scenario is characterized by distinct spacecraft platforms which may be categorized by different parameters. Guerra et al. (2016) suggest the use of the satellite mass to perform a taxonomy of spacecraft platforms and their capabilities, resulting in three main classes:

- The **heavy platform** is characterized by having more than 1000 kg of satellite mass. This class corresponds to the *Conventional* type in (Guerra et al., 2016).
- The **medium platform** is characterized by having more than 100 kg of satellite mass, but less than a heavy one. This class corresponds to the *Mini* type in (Guerra et al., 2016).
- The **small platform** is characterized by having less than 100 kg of satellite mass. This class corresponds to the *Micro* type in (Guerra et al., 2016).

In this classification, Heavy platforms are furnished with further resources than Small platforms. These resources may correspond to hardware characteristics, such as memory size, battery dimensions, or ISL capacity. The last resource is important in the simulation of satellite networks, which is determined by the ISL device. The same ONION project proposed a model of this device defined by a transmission data rate at a certain maximum communications range. Both parameters are bounded, and are determined by the characteristics of the ISL device (e.g. EIRP, sensitivity, reception gain, etc.) Following a similar classification process for the satellite platforms, different device classes have been determined depending on these two parameters. With the objective to reduce the variability, the maximum communications range has been fixed to 1500 km which corresponds to a feasible range using current technologies (see Chapter 1, Section 1.3.10). By determining the maximum range, the data rate is the parameter that classifies the different communications subsystems:

- The **heavy ISL device** is characterized by providing 4000 kbps at 1500 km of range.
- The **medium ISL device** is characterized by providing 750 kbps at 1500 km of range.
- The **small ISL device** is characterized by providing 100 kbps at 1500 km of range.

Commonly, more capable device consumes further power, and mass of the available in the satellite. Therefore, not all the previous platforms can be equipped with these ISL devices. The mass of a satellite has traditionally been used as a proxy metric of the satellite resources (Lluch and Golkar, 2014a). Therefore, heavy platforms can carry more subsystems than small plat-

Table 4.1: Transceiver configuration per platform type

Satellite platforms	Small ISL device	Medium ISL device	Heavy ISL device
Heavy platform	X	X	X
Medium platform	X	X	
Small platform	X		

forms. Table 4.1 presents the configuration of each platform according to the number of ISL devices and type that are equipped. Small platforms can only afford a small transceiver, while heavy platforms have enough resources to incorporate one transceiver of each type. Medium platforms balances the resources with small and medium devices. The possibility to have different devices allows to interact satellites that are conformed with different platforms. In this case, the least restrictive ISL device determines the common interface over which satellites communications. For the sake of illustration, a satellite with a heavy platform would communicate with another satellite with a small platform through the small device (at 100 kbps), while the same heavy satellite would communicate with a medium one using the medium device (at 750 kbps). The interactions between satellites from the same platform class is performed at its maximum data rate; e.g. heavy satellites communicate with heavy devices at 4 Mbps. 'Note that the negotiation process to determine at which transmission rate the communication is established has not been included in the current study.

These satellite-to-satellite connections are achieved by means of dedicated antennas located in differents parts of the spacecraft. Following the perliminary design proposed in the ONION project, four of these antennas would be distributed in different faces of the spacecraft to accomplish an omnidirectional radiation pattern. This kind of pattern enhances the connectivity of the network, and increases the probability to have a path between a source and the desired destination. Reviewing the literature of ISL devices, the S-band be an experimental frequency band to evaluate this kind of interactions. Thanks to its low propagation loss, this frequency band correctly balances the link capacity with the maximum communications range. Despite the ITU-R sector does not consider ISL frequency bands bellow 22 GHz, some satellite missions have already used these previous bands for this purpose (Léon et al., 2018). The continued interest in this field should encourage the discussion of the corresponding coordination in next World Radiocommunication Conferences (WRC).

Table 4.2 summarizes the characteristics of each satellite platform class with its ISL capabilities. Apart of these ISL devices, each platform is equipped with an EO payload to observe the Arctic region. The following section details these payloads, and the data traffic nature.

Table 4.2: Configuration of each satellite platform type with their corresponding ISL capabilities

Platform type	Mass (m)	ISL data rate	ISL max. range
Heavy	$m > 1000$ kg	4000 kbps 750 kbps 100 kbps	1500 km
Medium	$1000 \text{ kg} \geq m > 100$ kg	750 kbps 100 kbps	1500 km
Small	$100 \text{ kg} \geq m$	100 kbps	1500 km

4.3.2 Payload and Traffic Models

EO payloads are fundamental subsystems of satellites, which generates data when the satellite overpasses a specific target area. In this scenario, the payload is modeled as a Constant Bit Rate (CBR) application that is executed only over the Arctic region; i.e. latitudes larger than 66º. Therefore, a satellite periodically generates data according to a bit rate when its ground track is over this region. The ONION project conducted a survey and analysis of the different payloads that can be used to observe the ice status in the Arctic region (Alarcón et al., 2018). The investigation concluded that the set of instruments that shall be used are a Synthetic Aperture Radar (SAR) at C- or X-band, a SAR Interferometer Altimeter (SIRAL), a Microwave Radiometer (MWR), a Microwave Radiation Imager (MWRI), an Advanced Microwave Scanning Radiometer (AMSR), Global Navigation Satellite System Reflectometer (GNSS-R), and a Radar Altimeter. Table 4.3 presents the payload data rates retrieved from the Observing Systems Capability Analysis and Review (OSCAR)[10] database (Organization, 2011). Lancheros et al. (2018) emphasized that the instruments require a minimum spatial resolution of 10 km to perform ice monitoring. Therefore, a low resolution mode of the SAR-C has been considered to provide a spatial resolution of 150 m with a swath of 1500 km. The resulting data rate of the payload is 12 Mbps. A summary of the payload data rate values is presented in Table 4.3.

Table 4.3: Payload model characteristics

Payload	Data rate
SAR-C (low resolution)	12.0 Mbps
SAR-X	35.5 Mbps
SIRAL	10.1 Mbps
AMSR	87.4 kbps
Rad. Altimeter	35.0 kbps
MWR	10.6 kbps
MWRI	100.0 kbps
GNSS-R	16.8 kbps

A dedicated network of ground stations are located close to the Arctic region (e.g. Inuvik, Svalbard, Esrange, etc.) These ground stations offer the capability to interact with satellites that follow polar orbits. Despite this ground station network, the present study aims at evaluating the application of OLSR protocol to deploy satellite networks. Therefore, these ground stations have been omitted for the sake of the analysis. Another list of ground stations has been defined to generate a reception area over which the satellites can download data. Table 4.4 presents this list of ground stations that conform this reception area. The selected ground stations conforms this wide reception area thanks to their distinct locations in different countries. The reception area is delimited by two regions:

- The American region characterized by longitudes between 120 ºW and 45 ºW, and latitudes between 0 ºN and 55 ºN.
- The European-Asian region characterized by longitudes between 15 ºW and 150 ºE, and latitudes between 0 ºN and 55 ºN.

An illustration to summarize the behavior of a satellite according to its location is presented in Figure 4.2. When the satellite is placed in the Arctic region, it generates data (represented in

[10] OSCAR database has mission information and satellite characteristics.

Table 4.4: List of ground stations that conform the reception area

Ground Station	Country	Latitude	Longitude
Beijing	China	40.5º N	116.9º E
KaShi	China	39.5º N	76.0º E
Kumamoto	Japan	32.8º N	130.9º E
Libreville	Gabon	0.4º N	9.6º E
Sioux Falls	USA	43.7º N	96.6º W
Matera	Italy	40.7º N	16.7º E
Neustrelitz	Germany	53.4º N	13.1º E
Prince Albert	Canada	53.2º N	105.9º W
Shadnagar	India	17º N	78.2º E
SanYa	China	18.3º N	109.3º E
Si Racha	Thailand	13.1º N	100.9º E
Ulsan	South Korea	35.6º N	129.3º E
Chilton	United Kingdom	51.6º N	1.3º W
Redu	Belgium	50.0º N	5.1º E

the figure by a red line). This data is stored into the satellite until it can establish a downlink contact with one of the grounds stations located in the reception area. When the satellite is over this area, it triggers the download of the data (represented by the blue line). If a satellite is located over this area and does not have data to download, then it becomes a potential node to download data from other satellites. For those satellites that are not placed in the target area nor in the reception region, they can stay in two operational modes. Specifically, if the satellite has still stored data, it transmits the data if a route exists with a satellite that can download it; otherwise, the satellite remains in standby, which makes important this satellite to compose the network. Therefore, these standby satellites are the potential candidates to forward messages to other satellites.

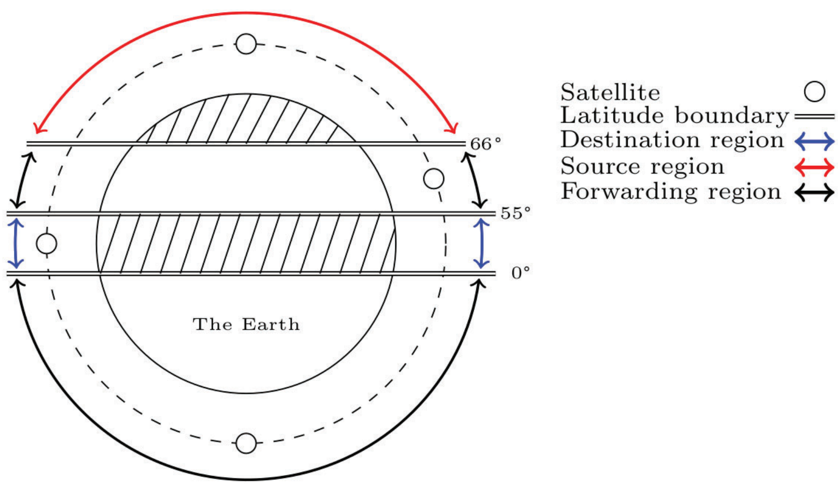


Figure 4.2: Representation of the transmission and reception regions.

4.3.3 Satellite candidates

The selection of which spacecrafts should compose the scenario has been deeply studied. The assessment of the OLSR protocol for satellite networks require a realistic scenario in which satellites follow coherent orbits. Therefore, only satellites that are currently in-orbit operative to observe the ice status has been selected. Considering the heterogeneity of the IoSat paradigm, spacecraft operated by different entities from distinct nationalities (e.g. European, American, and Asian satellites). Moreover, satellites with different objectives have been combined, such as EO and Telecommunications ones. Among the different satellites, only those that has an EO payload produce data. The remaining satellites would help to conform the network, and to download data from the others. Furthermore, the ISS has been included in the candidate list due to the different projects that are expected to be performed (Jahnke et al., 2017). Among them, the Bartolomeo platform enables the host of external payloads in the ISS, such as a subsystem that interacts with satellites. The ISS trajectory becomes interesting to interconnect satellites in Arctic region with other satellites located around the globe. Specifically, its inclination of roughly 51.6º makes it an ideal intermediate node that is close to the poles and the equatorial areas. Table 5.3 presents the list and the characteristics of each satellite. The table also details the orbital elements of each satellites, being a the semi-major axis, e the eccentricity, i the inclination, Ω the longitude of the ascending node, ω the argument of periapsis, and M the initial mean anomaly of the satellite. These orbital parameters have been extracted from the Celestrack database[11]. With this configuration, it is possible to determine if the combination of multiple spacecraft with different features and objectives is beneficial for the observation and monitoring of the Arctic zone, and evaluate if the OLSR protocol is a potential solution for these missions.

4.3.4 Protocol stack configuration

The simulation is conducted with the DSS software framework presented in Chapter 3. This simulator has been designed to easily configure a satellite system that has inter-satellite communications capabilities to create networked architectures. This framework defines a spacecraft model with different modules that represent its physical components, execution tasks, and communications protocols. Thanks to its modularity, the framework enables to select a custom protocol stack with different implementations available from the NS-3 libraries. For the scenario that we have presented, the protocol stack has also been defined to properly evaluate the performance of the OLSR for satellite networks.

A dedicated application has been designed to emulate the behavior of a satellite in this scenario. This behavior corresponds to the data generation when the satellite is over the Arctic region, and the download of the data when it is over the reception area. These satellites integrate in its protocol stack the version 4 of the IP, which interacts with the OLSR protocol. The selection of this IP version is conditioned to the incompatibility of the OLSR implementation—from NS-3—with the current version 6 of the IP. With this protocol, the routing tables can be defined, and thus each satellite can communicate in the network. The communications between a satellites is conducted over the UDP. This connectionless transport protocol does not guarantee the reception of the messages, which makes it unreliable. Nevertheless, this protocol is ideal to evaluate the performance of a routing protocol, because it does not perform additional processes; i.e. if a packet is loss during the transmission, the cause is related to the route formation. Following the same premise, the translation between IP addresses and link-layer addresses is performed by a pre-configured Address Resolution Protocol (ARP). When the scenario is configured, the map-

[11]Celestrack website: https://celestrak.com (last access at July 24, 2018)

Table 4.5: Satellite features used to perform the analysis

Satellite	Mass	Type	Payload	a	e	i	Ω	ω	M
European satellites									
Sentinel-1A	2300 kg	Heavy	SAR-C	7064 km	0	98º	57º	0º	287º
Sentinel-3A	1250 kg	Heavy	MWR	7181 km	0	99º	117º	0º	261º
CryoSat-2	720 kg	Medium	SAR-Alt	7088 km	0	92º	261º	0º	147º
TanDEM-X	1340 kg	Heavy	SAR-X	6886 km	0	97º	62º	0º	271º
TerraSAR-X	1230 kg	Heavy	SAR-X	6886 km	0	97º	58º	0º	46º
SEOSAR/Paz	1341 kg	Heavy	SAR-X	6885 km	0	97º	66º	0º	121º
SWARM-A	473 kg	Medium	-	6831 km	0	87º	72º	0º	279º
SWARM-B	473 kg	Medium	-	6901 km	0	88º	173º	0º	252º
SWARM-C	473 kg	Medium	-	6831 km	0	87º	81º	0º	279º
Metop-A	4085 kg	Heavy	-	7198 km	0	99º	114º	0º	0º
Metop-B	4085 kg	Heavy	-	7198 km	0	99º	115º	0º	316º
PROBA-V	160 kg	Medium	-	7191 km	0	99º	127º	0º	127º
American satellites									
Aqua	2934 kg	Heavy	AMSR-E	7076 km	0	98º	351º	0º	293º
RadarSat-2	2200 kg	Heavy	SAR-C	7169 km	0	98º	56º	0º	27º
Terra	5190 kg	Heavy	-	7076 km	0	98º	141º	0º	285º
Aura	2967 kg	Heavy	-	7076 km	0	98º	10º	0º	303º
SCISAT-1	152 kg	Medium	-	7021 km	0	74º	32º	0º	291º
CASSIOPE	490 kg	Medium	-	7041 km	0.07	81º	104º	212º	140º
ODIN	250 kg	Medium	-	6923 km	0	98º	84º	0º	125º
CYGNSS-1	25 kg	Small	-	6881 km	0	35º	312º	0º	210º
CYGNSS-3	25 kg	Small	-	6881 km	0	35º	304º	0º	212º
CYGNSS-4	25 kg	Small	-	6881 km	0	35º	310º	0º	211º
CYGNSS-8	25 kg	Small	-	6881 km	0	35º	210º	0º	212º
Asian satellites									
FY-3D	2300 kg	Heavy	MWRI	7207 km	0	99º	351º	0º	348º
HY-2A	1500 kg	Heavy	Radar Alt	7335 km	0	99º	60º	0º	293º
SARAL	630 kg	Medium	Radar Alt	7171 km	0	99º	237º	0º	214º
RISAT-1	1858 kg	Heavy	SAR-C	6917 km	0	98º	58º	0º	219º
GaoFen-3	2950 kg	Heavy	SAR-C	7129 km	0	98º	58º	0º	303º
RISAT-2	300 kg	Medium	-	6811 km	0	41º	259º	0º	25º
GaoFen-1	1080 kg	Heavy	-	7016 km	0	98º	149º	0º	64º
GaoFen-2	2100 kg	Heavy	-	7002 km	0	98º	148º	0º	350º
GaoFen-8	2100 kg	Heavy	-	6861 km	0	97º	184º	0º	278º
GaoFen-9	2100 kg	Heavy	-	7021 km	0	98º	151º	0º	144º
CartoSat-2C	727 kg	Medium	-	6876 km	0	97º	128º	0º	208º
CartoSat-2D	714 kg	Medium	-	6871 km	0	97º	128º	0º	320º
CartoSat-2E	712 kg	Medium	-	6876 km	0	97º	126º	0º	288º
CartoSat-2F	710 kg	Medium	-	6871 km	0	97º	127º	0º	274º
Additional Spacecraft									
ISS	419455 kg	Heavy	-	6782 km	0	52º	233º	0º	49º

ping between these addresses are also installed in each satellites. With this approach, the ARP does not require to perform the request of the IP address because a static entry is integrated in its cache. Therefore, the execution of the ARP does not adulterate the analysis of the OLSR protocol performance. Lastly, the satellite application interacts directly with the UDP using a dedicated socket interface, like in a UNIX systems.

4.4 Results and Discussion

The OLSR protocol is able to determine an existing path between a source and a destination at certain moment. If this path is broken, then the communications between the satellites is unfeasible until the path is restored. Unlike DTN solutions, this protocol is not able to predict future contacts between satellites, which may affect its performance. Therefore, the characterization of the satellite mobility effects on the path creation and destruction becomes fundamental. The first section addresses this study to probabilistically estimate the path characteristics. With this information, it is possible to evaluate if an ideal case can ensure certain quality of the communication. This represents the reference of the future results—when the simulation is executed. Once the study of the topology is finished, the analysis on the performance of the OLSR protocol to deploy satellite networks that improve polar missions is conducted. A satellite system is composed of nodes that are in movement following a periodic trajectory. Considering a Keplerian orbit, this means that the entire system is also periodic, with a global period determined by all the orbital periods of satellites. If we take into consideration the list of satellites presented in Table 5.3, the system period is around 43910 weeks. Computationally, this cannot be simulated with the previous tool. Therefore, it has been considered to simulate less time, but long enough to perform a representative analysis. The connectivity analysis has been performed with two days of simulation. To characterize the benefits of applying OLSR to polar satellite missions, it has been simulated nine hours. The following sections present the results of the simulation, as well as the corresponding discussion.

4.4.1 Connectivity Analysis

This study aims at evaluating the feasibility of using MANET routing protocols in the satellite context. These protocols can only identify routes between a source and a destination at certain moment. Therefore, if the destination is not reachable, the communications cannot be established. This situation could be caused by not being any satellite on the reception area to download data, or because the satellite that wants to transmit data is isolated. This last condition is the result of the network disruption, which fragments the network in different isolated parts. This phenomenon may provoke that no instantaneous routes are available. Therefore, some solutions predict future satellite contacts to define a route over time. This is not the case of the OLSR protocol, which can only determine instantaneous routes. Therefore, the characterization of the network disruption and—in general—the topology dynamism becomes important.

The main cause of network disruption is the limited number of satellites and their mobility. A snapshot is the temporal representation of a satellite network in which all the links remain stable depending on the node position and mobility. When a link between two satellites changes, the remaining topology generates a new snapshot. Is in each of these snapshots that the OLSR protocol will be able to identify routes between a satellite that has data with another satellite that is located over the reception area. If the number of satellites that compose the satellite system is small, the definition of a desired route at certain snapshot becomes improbable. To

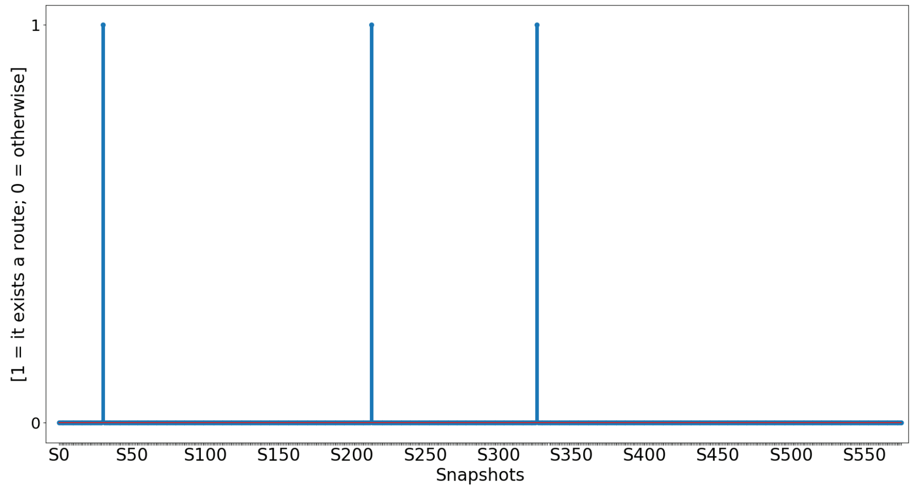


Figure 4.3: Snapshots that contain a feasible route with only European satellites.

illustrate this situation, Figure 4.3 presents the snapshots that contain a desired route with the value 1. If the snpashot does not contain a desired route, then the figure plots a 0 value. This figure presents the results of a simulation in which only the European satellites have been considered; i.e. twelve satellites. This configuration makes that only three snapshots have feasible routes between the Arctic region and the reception area. A lack of connectivity is clearly detected with this configuration, where the network disruption predominates. Therefore, using the satellites of a single nationality are not enough to mitigate this phenomenon, which prevent the establishment of any end-to-end connection. Nevertheless, the expansion of the satellites can enhance this situation, as Figure 4.4 presents. In this case, all the satellites have been selected to conform the satellite system, being a total of 38 satellites. Unlike the previous case, more snapshots contains feasible routes that can be leveraged to download data. Is in this kind of scenario where OLSR protocol can enhance the system by defining these routes.

This brief introduction of the network disruption and its effect on the routes opens the door to deeply characterize the nature of these routes. The former parameter that may determine a route is the number of hops that compose it. Figure 4.5 presents a histogram of the number of hops of a route in this scenario. The 70.29 % of routes are composed of—least—seven hops to

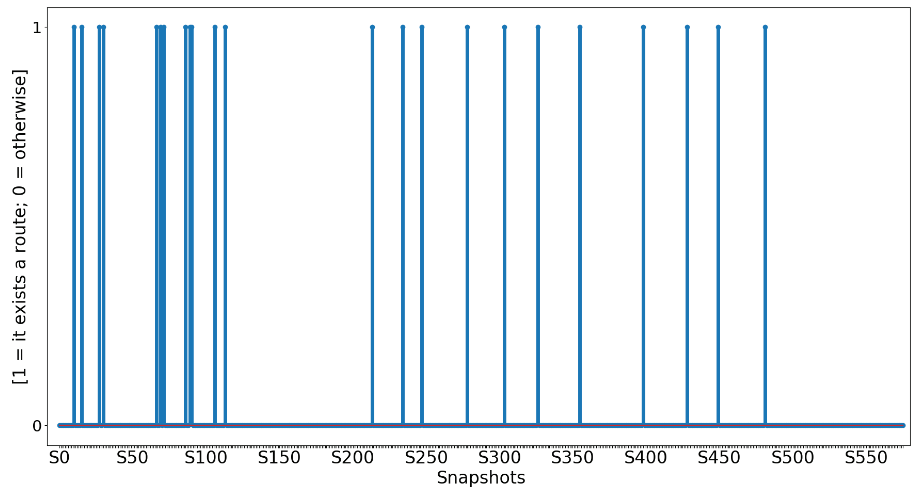


Figure 4.4: Snapshots that contain a feasible route with all the satellites.

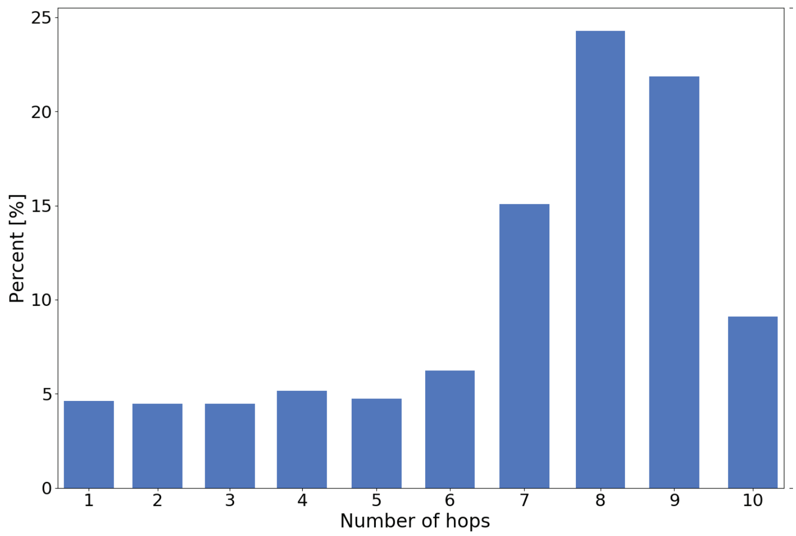


Figure 4.5: Histogram of the amount of hops per routes.

connect a source with a destination. This is caused by the trajectory of polar orbits, which promotes the closeness of the satellites in this region. Therefore, numerous routes may be feasible with the combination of the different satellites that sporadically meet in this region. Despite this connectivity, larger routes are more sensitive to any change rather than the smaller ones, because if a single intermediate link is broken the entire route also. For this reason, in a disruptive environment, routes with less hops are preferred—although in this case they are less probable.

The stability of a route can be measured with its lifetime, which corresponds to the time that a route remains unaltered. As part of this route characterization, this metric becomes an important feature to evaluate the dynamism of the network. Figure 4.6 shows the percentage of routes with respect to their minimum lifetime. With this representation, we can identify that all the routes have a lifetime larger than 1 second, but less than the 5 % of routes have a lifetime larger than 130 seconds. The shape of this figure shows a logarithmic curve which decrements while the minimum lifetime increases. This decrease is noticeable with small values of lifetime, while it becomes smooth with large values. Specifically, the 75 % of routes have at least 5 seconds of lifetime, while approximately the 50 % of routes have a lifetime larger than 13 seconds. This behavior indicates that dynamic routes are frequent in this scenario, where routes are created and destroyed quickly. Nevertheless, there is a region in which the 20 % of the routes are stable at least 40 seconds. Note that the maximum lifetime of a route is 136 seconds. The short-time routes may be used to send sporadic data blocks, while the more stable routes should be used to keep a data stream.

As previously stated, the routes that are larger may experience shorter lifetime. However, larger routes may provide farther communications range to reach other satellites. Therefore, a tradeoff between length of the route and its lifetime needs to be performed. Specifically, the utility of a route could be considered as the relationship between the route length and its lifetime that ensures that enough data can be transmitted. For example, if a route has a small lifetime and it is composed of one single hop, a useful communication can be established. However, if the lifetime is small and the route is large, the communication is not possible because the route is not stable enough regarding its composition. Therefore, it is important to correlate the lifetime of a route with its length. This information is represented using the statistical box plot

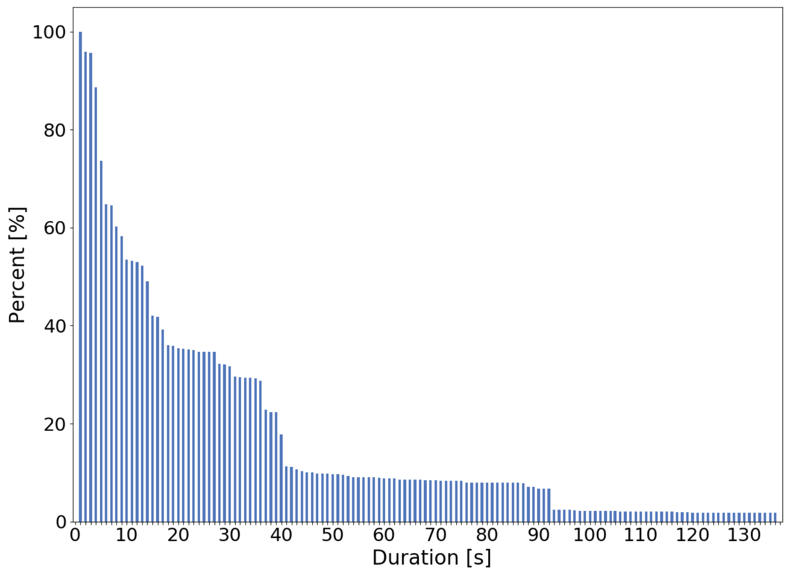


Figure 4.6: Percentage of routes with respect to their lifetimes.

(Figure 4.7), in which a box corresponds to a distribution function that the Inter-Quantile Range (IRQ) is delimited by the third quartile (Q3) and the first quartile (Q1). The median is also represented in the IRQ, and it enables to understand which of these quartiles are more important. The different dots placed outside the box represent outliers (sporadic values).

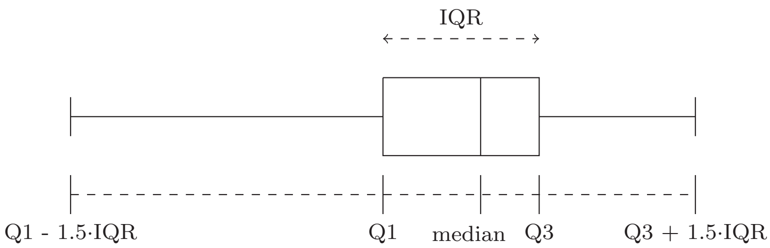


Figure 4.7: Explanation of box parameters in a box plot.

The corresponding box plot with the correlation between route lifetime and length is presented in Figure 4.8. As stated previously, the larger routes experience shorter lifetime median values being less stable than others. Nevertheless, the smallest routes are not the most stable ones. Indeed, the routes that have four and five hops are the most stable. This indicates that these routes should be used to stream data, while the others should be used to sporadically send small blocks of data. However, the routes with four and five hops are not the most frequent in this scenario (Figure 4.5). Specifically, only a 9.9 % of the routes have this length, which may not be comparable to the probability achieve in routes with seven, eight, and nine hops. In this regards, the most probable routes are also the shortest ones—with a lifetime shorter than 15 seconds. This results suggest again that the present scenario is considerably dynamic with numerous fluctuations on the routes. Nevertheless, the utility of a route does not only fall on its length and lifetime, but also upon the amount of data that is expected to be transmitted. As previously introduced, the short routes may be appropiate for small data blocks, while data streams could be transfered over longer routes. Therefore, it is necessary to put these values in data context: a route is useful if its lifetime is long enough to forward a block of payload data. In other words, a route is useful if the end-to-end transmission time t_{E2E} is less or equal to its

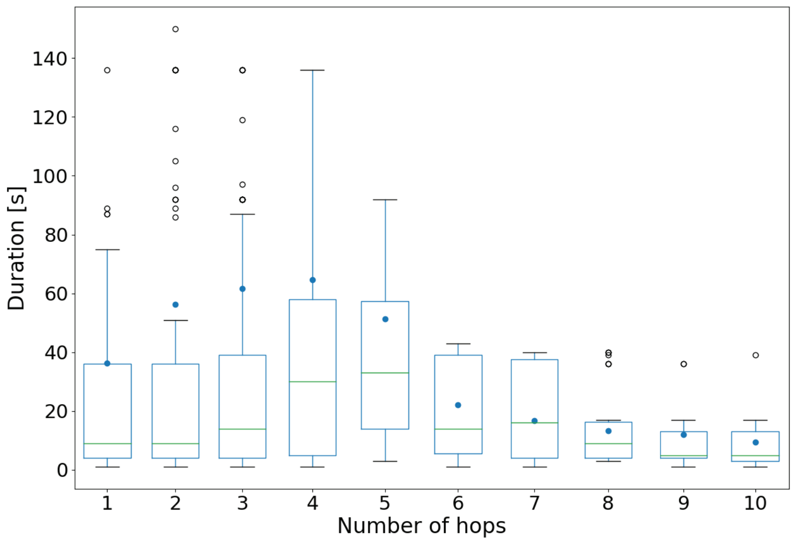


Figure 4.8: Box plot of the route lifetime depending on the number of hops with the averaged values represented by blue dots and the median one with a green line.

lifetime t_{LT}:

$$t_{E2E} \leq t_{LT} \tag{4.1}$$

The end-to-end transmission time depends on different parameters. One of them is the transmission time of a packet t_{TX}, which represents the amount of time that a node needs to send a packet. Thus, it depends on the size of the packet l and the transmission capacity C of the node. Considering that all the transmitted packets have the same size, the transmission time in a node i is determined by the capacity of its link C_i:

$$t_{TX_i} = \frac{l}{C_i} \tag{4.2}$$

For a specific link j, the propagation time t_P is characterized by the distance of the link d_j and the speed of light c:

$$t_{P_j} = \frac{d_j}{c} \tag{4.3}$$

Using both concepts, the amount of end-to-end time needed to transmit a payload data block, composed of N_p packets, through a route is defined by the following equation:

$$t_{E2E} = \sum_i t_{TX_i} + \sum_j t_{P_j} + (N_p - 1) \cdot \max_i\{t_{TX_i}\} \tag{4.4}$$

where i is the index that represents the nodes and j the index that represents the hops that compose the route. Note also that the following relationship applies: $j = i - 1$. With this definition, the amount of packets that can be transmitted during the lifetime of each route t_{LT} is determined as follows:

$$N_p = \left\lfloor \frac{t_{LT} - \sum_i t_{TX_i} - \sum_j t_{P_j}}{\max_i\{t_{TX_i}\}} \right\rfloor + 1 \tag{4.5}$$

The size of the payload data block L that can be transmitted during the path lifetime is characterized as follows:

$$L = N_p \cdot (l - h), \tag{4.6}$$

where h represents all the header bytes of a packet.

Using this definition, the amount of payload data that can be transmitted during the path lifetime is represented in Figure 4.9. This representation allows to evaluate this amount of data

Table 4.6: Amount of time (in seconds) of payload execution that the resulting data can be transmitted through each route type while it is active

Payload	Number of route hops									
	1	2	3	4	5	6	7	8	9	10
SAR-C	0.8	0.8	0.9	2.3	2.5	0.9	0.8	0.6	0.4	0.4
SAR-X	0.3	0.3	0.3	0.8	0.8	0.3	0.3	0.2	0.1	0.1
SIRAL	1.0	0.9	1.1	2.8	2.9	1.0	1.0	0.7	0.5	0.4
AMSR-E	112.5	103.1	121.9	318.7	337.4	117.2	112.5	84.4	53.2	46.9
Radar Altimeter	280.9	257.5	304.3	795.8	842.6	292.6	280.9	210.7	140.4	117.0
MWR	927.4	850.1	1004.7	2627.6	2782.2	966.0	927.4	695.6	463.7	386.4
MWRI	98.3	90.1	106.5	278.5	294.9	102.4	98.3	73.7	49.1	41.0

A/N: All the values in the table are in seconds

according to the length of the route, by combining the formulation with the results previously presented in Figure 4.8. As expected, the shape of the figure remains similar as before the processing. In this case, the routes four and five could transport a total of 3000 kB per instance of the route (median values), while the others are bellow the 1000 kB. These results suggest that statistically every time that a route of four or five hops is generated roughly 2.93 MB can be forwarded. This capacity of each route instance may correspond to the maximum amount of data that can be downloaded over the network. The correlation of the generated data per payload must be conducted to identify if the routes may be useful. Table 4.6 presents the execution time of each payload that can be transmitted through each type of route. The value has been computed using the median of each box in Figure 4.9, and the generation data rate of each payload. This table indicates that the payloads with higher data rate (e.g. SAR-C, SAR-X, etc.) cannot deliver a huge amount of data, while less demanding payloads can use the network to transmit significant data. In both cases, however, the deployment of a network makes it possible to deliver certain amount of payload data, and thus to improve the mission performance.

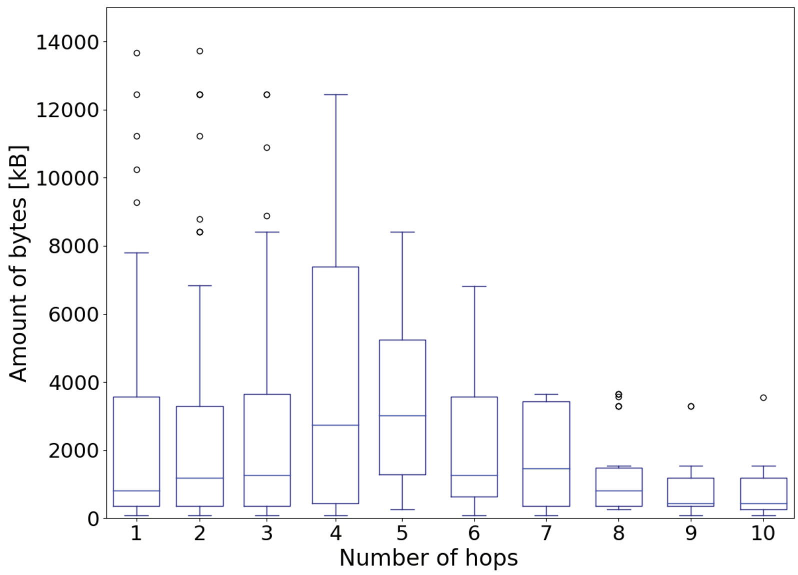


Figure 4.9: Maximum amount of bytes that can be transmitted through active routes.

At the end of this connectivity analysis, the results have demonstrated that the deployment of a network would enable the possibility to transfer data (no matter which kind of payload) from satellites placed in the Arctic area to other satellites in the reception zone. However, the network becomes extremely dynamic, having the shortest routes the ones that are also the most probable. This would provoke that a defined route may change quickly, and a routing protocol that cannot react accordingly would loss the opportunities to use these sporadic routes. As a proactive protocol, the OLSR protocol seems to be an interesting candidate that could leverage from these intermittent connections to deliver certain amount of data. The following section presents the results achieved by simulating this protocol in this dynamic environment.

4.4.2 OLSR performance analysis

Previous section has evaluated the feasibility of the scenario without considering a specific implementation of a routing protocol. Therefore, this section aims at quantifying the enhancement or deterioration in the system performance by applying the OLSR protocol. A group of performance metrics have been defined accordingly:

- Packet Delivery Ratio (PDR) – Fraction of the data packets that have been correctly delivered to the destination nodes. This fraction is computed with respect to the total of packets that are transmitted.
- Local Packet Loss Ratio (LPLR) – Fraction of data packets that cannot be transmitted because the source does not have a valid route to the destination. This fraction is computed with respect to the total of packets that are generated. This parameter characterizes the disruption of the network at each snapshot, because it quantifies the impossibility of a node to transmit data due to the absence of a route. It is thus an interesting parameter to consider for instantaneous routings, such as the OLSR protocol.
- Packet Delivery In Route Ratio (PDIRR) – Fraction of data packets that has been correctly delivered to the destinations, when a valid route exists. By definition, the following relationship is established:

$$PDIRR = \frac{PDR}{1 - LPLR} \tag{4.7}$$

- Average end-to-end delay of data packets – Mean time of a packet to travel from a source to a destination. This delay metric corresponds to the aggregation of the propagation, transmission and queue times.
- Average access time of data packets – Mean time that a satellite needs to reach a zone over which it can download a packet. This metric corresponds to the case that no inter-satellite communications are performed. The comparison of the average end-to-end delay with the average access time enables to evaluate how beneficial is the satellite network in terms of time enhancement.

The results that are presented in the following paragraphs correspond to the simulation of 33496 seconds; i.e. 9.3 hours. This time has been split in different slots of random size to evaluate the metrics values in different moments rather than a global average value. This enables to retrieve statistics at different conditions and thus to identify possible transmission windows. A transmission window is a time slot in which the conditions to transmit data are better than in other slot, e.g. a time slot with a snapshot that has an active route. As stated in the previous section, the satellites with the payloads that generate more data may not properly leverage on the establishment of routes. To characterize this effect, two scenarios with different data generators are considered for the following analysis. The former scenario is composed of all the satellites, but only the ones that have a SAR-C, SAR-X, or SIRAL payload would generate data. This scenario is known as ”high data rate payloads” in the results. In the other scenario, the satellites that are equipped with a MWR, MWRI, Radar Altimeter, or AMSR-E payload would be the ones that generate data. This scenario is known as ”low data rate payloads” in the results.

Table 4.7 presents the metrics related to the transmission and reception of packets after the simulation; i.e. PDR, PLRL, and PDIRR. In all the times slots and scenarios, the PDR is always less than 10.5 %. This result indicates that the delivery of the packets is not always accomplished, lossing more than the 90 % of the transmitted packets. This undesired behavior may be conducted by the protocol implementation or the disruption of the network. The LPLR metric helps to clarify this situation, which in all the cases its value is always larger than 89 %. These results suggest that more than the 89 % of the generated packets—which are intended to be transmitted—are not transmitted because the routing protocol does not detect an available route, and remains in the satellite. This is mainly caused by the network disruption, which clearly predominates in this scenario. Therefore, it is indicating that during the major time of the simulation the network is disrupted and the protocol cannot identify a route between the Arctic area and the reception zone. In this case, solutions that could perform store-and-forward or predict routes over time could enhance the performance. Despite this disruption, the delivery ratio when a route between a source and a destination is available is still an interesting parameter. The PDIRR clarifies this by representing the probability that a packet can be correctly delivered in an active route. In the configuration with instruments of low data rate, the PDIRR value is always larger than 90 %, indicating that almost all the packets that are transmitted are correctly received. In this case, the OLSR protocol is able to manage changes between snapshots when in each adjacent snapshot contains a feasible route. On the other hand, the payloads with high data rate have a lower ratio, because the network cannot support the input flow. It is important to highlight that for this configuration, certain time intervals exist in which the PDIRR reaches values higher than 90 % (last two rows of the table). This result indicates that there are two transmission windows that, if the transmissions can be scheduled in specific intervals, the data reception can be optimized. These results demonstrate the potential of applying prediction mechanisms in OLSR protocol which could leverage from these transmission windows.

Another performance aspect is the evaluation of the time needed to deliver the payload data. The comparison of average end-to-end delay of transmitted payload data with the average access time that a satellite with no ISL would need to download the data. Table 4.8 presents this comparison, emphzising that in both scenarios the average end-to-end delay is lower then the average access time. This means that when a route is established, the delivery time of the payload data is always enhanced when a satellite network is established with the OLSR protocol.

Table 4.7: Results of the packet metrics according to the received packets (Rx. packets).

Time slot	Low data rate payloads				High data rate payloads			
	Rx. Packets	PDR	LPLR	PDIRR	Rx. Packets	PDR	LPLR	PDIRR
0 - 3679	1073	5.4%	94.4%	96.4%	96382	1.12%	97.5%	44.8%
3679 - 11335	864	4.0%	95.6%	92.2%	135996	0.82%	97.7%	35.0%
11335 - 18843	929	4.2%	95.8%	100.0%	11861	0.08%	99.7%	22.9%
18843 - 24505	1466	6.6%	93.0%	93.6%	14687	0.12%	99.8%	66.7%
24505 - 28616	2265	10.2%	89.8%	100.0%	451139	4.61%	95.2%	95.8%
28616 - 33496	1021	6.4%	93.6%	100.0%	161364	1.59%	98.4%	100.0%

Table 4.8: Results of the time metrics

Time slot	Low data rate payloads		High data rate payloads	
	Average Delay	Average Access Time	Average Delay	Average Access Time
0 - 3679	24.42 ms	279.35 s	89.56 s	818.64 s
3679 - 11335	22.31 ms	335.55 s	177.00 s	569.60 s
11335 - 18843	17.92 ms	131.98 s	76.90 s	994.36 s
18843 - 24505	34.87 ms	219.11 s	67.94 s	316.47 s
24505 - 28616	21.32 ms	187.57 s	157.07 s	530.19 s
28616 - 33496	13.87 ms	119.84 s	75.64 s	535.17 s

The delay time can be reduced from 279.35 seconds to 24.42 milliseconds. Note also that in the high data rate case the end-to-end average delay is larger than in the low data rate case. This is mainly due to the fact that node queues are saturated and that a packet spends more time in the queue before to be processed. The achieved average delays are acceptable for near-real-time services, which are currently demanded for satellite missions in Arctic region.

The performance analysis concludes that using the well-known OLSR protocol it is possible to establish certain routes between remote satellites. The ratio of packets delivered when a route is feasible remains always greater than 90 % for low data rate payloads, while for high data rate payloads the ratio is reduced to 66.7 %. These packets are forwarded over the network achieving delay times acceptable for the current user demands. However, the network disruption provokes that more than 89 % of packets cannot be forwarded because a route is not feasible. Consequently, we can conclude that the scenario is totally driven by this disruption. However, certain time windows present better performance than others. This indicates that the snapshots presents better connectivity in these slots. Therefore, a predicitive mechanism than enables to detect in-advance these windows could optimize the transmission during only these slots.

4.5 Summary

The IoSat paradigm proposes the deployment of opportunistic and temporal satellite networks to conform the necessary infrastructure to establish satellite federations. The definition of the routing protocol becomes an important challenge due to its sporadic, opportunistic and temporal nature. This chapter has evaluated the performance of the OLSR protocol to deploy this kind of network. This protocol has largely been used in other connected infrastructures commonly located on ground, such as in networks composed of ground or flying vehicles. Furthermore, a preliminary study on deploying satellite federations used this protocol (Lluch et al., 2015). This chapter has extended these investigations by performing a discussion of the implications of using this protocol for satellite networks. The protocol design has been reviewed at the beginning of the chapter. This review introduced more details about the behavior of the protocol and its implementations. Then, a realistic scenario has been conceived to evaluate the protocol. The configured scenario corresponds to a satellite mission to observe the ice status of the Arctic region. The selected satellites that participated in the mission followed a polar orbit trajectory, and were from different nationalities. This allowed to determine a heterogeneous scenario where satellites with different communications capabilities interacted.

The study has started by analyzing the dynamism of the network topology that could be conformed in this scenario. The results suggested that the number of satellites is a key parameter that influences on the apparition of the network disruption. Furthermore, the results demonstrate that longer routes are more sensitives to changes. Therefore, these routes typically have less lifetime. The correlation between route length and its lifetime indicated that the routes with four and five hops are more stables, but they are less frequent. Therefore, the utility of a route is the important metric in this dynamic scenario. Specifically, the utility corresponds to the amount of data that can be transmitted in a feasible route with respect to the generated data. The results indicate that short-time routes may be used for sporadic transmissions, while long-time routes may be used for data streams. In this sense, satellites with low data rate payloads would leverage on intermittent communications, while satellites with larger data rates payloads may require more stable connections. As this scenario is predominated by intermittent connections, the performance achieved with the OLSR protocol is better for those satellites that have low data rate payloads. Specifically, the PDIRR of these satellites remains always larger than 90 % indicating that almost of the packets are correctly delivered when a route is available. Nevertheless, the network disruption is driven the performance of the OLSR protocol, because more than 89 % of packets cannot be transmitted because no route has been identified. These packets remain stored in the satellites.

These results suggest that intermittent communications perdominate in this scenario. Therefore, an interesting approach for this scenario would be the use of store-and-forward mechanisms. Specifically, spraying small blocks of data over different contacts could enable to reduce the LPLR metrics, and thus mitigate the disruption of the network. Furthermore, the results suggest that some time windows are interesting to transmit data than others. Therefore, the deployment of prediction mechanisms that anticipate these windows could enhance the performance of the protocol. In summary, the OLSR protocol can be used in satellite context, because data has been downloaded thanks to it. Nevertheless, its design can be improved if store-and-forward and predicition mechanisms are integrated in its design. This conclusion is coherent with the discussion performed in Chapter 1 which emphizes the necessity to create a routing protocol that integrates capabilities from different implementations. The following chapter goes deeper in this premise by conceiving a prediction mechanism that can be executed in each satellite to estimate satellite contacts.

5

Assessment of predictive algorithms to estimate inter-satellite contacts

5.1 Introduction

So far, the IoSat paradigm has been presented as a proposal to deploy sporadic and temporal satellite networks that enable the establishment of satellite federations. The discussion conducted in Chapter 1 suggests that a current routing protocols may not manage the definition of routes in this dynamic and intermittent network. A dedicated protocol that integrates capabilities from different implementations may be defined. Chapter 4 has reinforced this approach by evaluating the performance of a MANET routing protocol to deploy satellite networks. The selected protocol has been the OLSR protocol, which may quickly detect changes in the network and adapt the defined routes accordingly thanks to its proactive approach. By means of different simulations, the protocol can identify available routes that are contained in a snapshot of the topology, and propagate data over it. Nevertheless, the disruption of the network—a relevant characteristic in satellite systems—affects the performance of the OLSR protocol, making that more than 89 % of packets remain in the source node. This performance is induced by the protocol itself, which is only capable to detect instantaneous end-to-end routes. The capability to define routes over time—which represents a sequence of scheduled satellite contacts—may enhance its current performance in this satellite context. The approach is the integration of distinct capabilities from other protocols to conform the desired protocol. This chapter presents the discussion of the possibility to integrate a predictive algorithm that may complement protocols like OLSR.

The capability to predict the topology of the network in advance has been investigated in the scope of DTN, proposing solutions that take advantage of the satellite orbit trajectory determinism. This is the case of the CGR protocol (Araniti et al., 2015; Book, 2019) which determines a route between a source and a destination over time using a contact plan. This plan is a scheduled sequence of contacts between all the satellites that compose the system. The CGR protocol was designed to establish communications in the NASA ION (Burleigh, 2007), but it has recently gained interest for LEO systems. The presented protocol standard draft does not clarify however the procedure to generate the contact plan. Traditionally, a centralized solution in which the ground facilities propagate orbit trajectories and compute the corresponding plan is used. The concept of the CPCE to support the generation of the contact plan and the route definition is presented in (Fraire and Finochietto, 2015). This CPCE pre-computes the contact plan and the routes on-ground to periodically upload them into the satellites using the ground station network. Example of protocols that use this centralized architecture are also presented in snapshot networks, such as the PLSR protocol (Fischer et al., 2013). Although this centralized architecture can generate accurate contact plans for complex systems, the upload process of the plan needs still to be further investigated. In particular, if a new satellite is included in the system, the re-computation and upload of the entire contact plan must be conducted, making the solution not scalable. Moreover, this centralized facility entails an important operations cost, and makes the satellite system ground-dependent. Due to these features, the integration of this architecture in the IoSat paradigm, where heterogeneous systems with satellites from different operators interact, becomes not suitable. Additionally, this centralized solution would require large resources for massive satellite constellations with thousands of satellites, such as Starlink (Exploration, 2018) or Telesat (Canada, 2018).

To overcome these limitations, new investigations aim at providing autonomy to satellite systems by enabling the capability to identify route changes. A routing protocol was proposed in (Su et al., 2019) that selects routes according to its stability, characterized by its Route Expiration Time (RET). This RET is determined by the minimum lifetime of the links that compose a route. This minimum lifetime of the links is also called Link Expiration Time (LET). Featuring the protocol with predictive capabilities reduces the overhead as well as mitigates the disruption. Although a model to compute the LET is presented, its accuracy and the estimation procedure is not detailed. Additionally, the LET is determined by propagating the orbit trajectory of the satellites over time, which means more processing capacity for the satellites. Furthermore, other prediction mechanisms have been proposed from the DTN research community. The CGR (Burleigh et al., 2016) proposes that each node estimates contacts and their duration using a history log. This log accumulates information of contacts that have already achieved, which is updated according to the confidence parameter. This parameter represents how reliable is the new predicted contact with a satellite, according to its history log. This mechanism is characterized by having an important learning curve, which—as compared to others that use determinism to estimate contacts (like CGR)—degrades its performance. The amount of data that can be exchanged during a contact, well-known as contact capacity, is another parameter used by the CGR to determine the routes. Authors in (Walter and Feldmann, 2018) propose a mechanism to estimate this capacity between satellites and ground stations, modelling the features of the transceiver and the communications loss.

As opposed to the previous proposals, this chapter presents an autonomous and decentralized protocol by which a satellite identifies other neighbor satellites, and predicts when a contact happens assuming an omnidirectional antenna pattern. The proposed solution uses a *predictor* which estimates the contacts leveraging from the determinism of the satellite trajectory. Therefore, it uses the satellite orbital elements to formulate simple model, without propagating the entire trajectory. To define the prediction algorithm, a contact between two satellites has been modeled as a satellite proximity or closeness. This scenario has largely been studied for satellite or debris collisions (Alfano and Oltrogge, 2018b; Richey, 1986), defining the PCA of two satellites. As the following sections demonstrate, a predictor based on the PCA is impacted by the initial satellite conditions, which could entail the schedule of wrong contacts. For that reason, an additional predictor is defined using the relative orbital motion theory (Schaub, 2004), which has normally been used in flight formation and spacecraft rendezvous. With this relative orbital motion, the position of a *deputy* satellite with respect to a *chief* satellite can be determined and modelled. To the best of our knowledge, these models have not been used to estimate the time interval of a satellite contact before. Instead, they have been commonly used to estimate a relative position at certain time. Note that the contact estimation is achieved considering a maximum communications range, which is determined by the link budget of the satellites.

Leveraging from this predictor, a protocol to share the predictions among the satellites has also been defined. This protocol enables a satellite to estimate the contact sequence with its neighbors, and share it to them. The entire contact plan can be constructed by means of sharing all these sequences. This chapter also evaluates the performance of this proposed protocol in two different scenarios. The former with a hybrid satellite system composed of tropical and polar constellations that demonstrates the correctness of the contact plan construction procedure. The last scenario evaluates the time of the satellites to receive the contact plan from all the other satellites in a massive satellite constellation.

The main contribution of the present research is 1) a predictor model based on close approach theory, 2) an alternative predictor definition based on relative orbit dynamics, 3) an accuracy assessment of the different predictors, 4) a protocol with which the satellites can autonomously generate a contact plan, 5) a feasibility analysis executing the protocol in a hybrid satellite constellation, and 6) an evaluation of the time to construct the contact plan in a mega-constellation. The research presented in this chapter has been published in the peer-reviewed IEEE Access journal (Ruiz-de-Azua et al., 2020b), and the International Astronautical Congress (Ruiz-de-Azua et al., 2019b). The details of these publications are accessible in Chapter 9 with the identifiers [IV], and [XIII].

The remaining of the chapter is structured as follows. First, Section 5.2 presents a global view of the predictive algorithm. Characteristics of the predictor based on the PCA are detailed in Section 5.3. Section 5.4 formulates the predictor based on relative orbital motion. The mechanism to generate the contact plan and the corresponding performance analysis is presented in Section 5.5. Finally, Section 5.6 concludes the chapter and remarks future research aspects on this topic.

5.2 Predictive Algorithm Overview

In a satellite network, the nodes are constantly moving, which modifies the topology over time. However, during a lapse of time the topology remains stable generating the well-known snapshot. The topology evolution is represented thus as a sequence of snapshots, not evenly spaced in time. Due to the periodicity of satellite motion, this sequence becomes periodic repeating the same snapshots defined in m time segments. Note that this periodicity is achieved in short integration periods, because of secular effects (e.g. J2). Despite these effects, the sequence model is valid due to their slow impact on the orbit. Each snapshot can be modeled as a graph G, in which n satellites—denoted by $V = \{\mathrm{v}_1,\mathrm{v}_2,\cdots,\mathrm{v}_n\}$—are connected by a set of edges E. An edge $(\mathrm{v}_j,\mathrm{v}_k) \in E$ represents an active ISL from v_j to v_k satellites. The graph of a snapshot is thus determined by the satellites, and the different edges: $G = (V, E)$. In this dynamic environment, the definition of a route between distant satellites is a considerable challenge. Some solutions computed feasible routes in each snapshot graph independently (see Section 1.3.3). Nevertheless, a route can also exist between snapshots, being possible to define routes in the time-domain. The space-time graph model (He et al., 2016) presents a succession of graphs in the snapshot sequence $\{G(t_i)|i = 1, \cdots, m\}$. Two types of edges cohabit in this model. The former is the temporal edge (v$_j$,v$_j$)$_{t_i}$ that connects the same node v$_j$ between snapshots at t_i and t_{i+1}, storing any possible data. The other edge is the spatial one (v$_j$,v$_k$)$_{t_i}$ (where $j \neq k$) that connects different nodes in the time segment $[t_i, t_{i+1})$. Using both models it is possible to define the routes in space and time domains.

The CGR protocol implements an algorithm based on these graph models. In particular, it defines routes over graphs using the contact plan. A contact between satellites is represented as a time interval $[t_j, t_k)$ in which an ISL remains feasible. The schedule of all the contacts between satellites conforms the contact plan, referred in this work as the *global contact plan*. This protocol assumes that each satellite holds this plan, which normally is pre-computed and distributed through a centralized ground entity; i.e. the CPCE. Although this strategy is suitable for planned and scheduled networks, it cannot be applied to heterogeneous and autonomous scenarios, such as the IoSat paradigm. To overcome this limitation, this work presents a predictive algorithm that is executed in each satellite to self-learn and self-construct the contact plan. As it is presented in the following sections, this de-centralized approach is suitable for heterogeneous satellite systems (e.g. hybrid satellite constellations) or massive satellite constellations. In order to understand the proposed algorithm, different definitions need to be presented first.

Definition 1 (Contact Sequence)**.** *Considering a node v_j has a sub-set of nodes that are its neighbours $N_{v_j} \subset V$, the contact sequence of node v_j with a neighbor $v_k \in N_{v_j}$ is the schedule of contacts with this neighbor*

$$C_{v_j}(v_k) = \{[t_1, t_2),\ [t_3, t_4),\ \cdots,\ [t_m, t_{m+1})\} \tag{5.1}$$

Definition 2 (Local Contact Plan)**.** *Considering a node v_j, the union of its contact sequences represents the local contact plan (i.e. only with its neighbors N_{v_j}).*

$$C_{v_j} = \bigcup_{\forall v_k \in N_{v_j}} C_{v_j}(v_k) \tag{5.2}$$

Definition 3 (Global Contact Plan)**.** *Considering the nodes that compose the entire network $v_j \in V$, the union of their local contact plans C_{v_j} represents the global contact plan C.*

$$C = \bigcup_{\forall v_j \in V} C_{v_j} \tag{5.3}$$

The proposed algorithm is based on periodically predicting the different contact sequences to generate a local contact plan which is shared through the network to construct a global contact plan. Figure 5.1 presents a block diagram of the modules that compose the algorithm executed in each satellite. The former component is the *predictor* which iteratively computes the local contact plan of the satellite from time t_m until time t_{m+1} (where $T = t_{m+1} - t_m$ is the prediction period) using the orbital elements vector $\mathbf{e} = [a, e, i, \omega, \Omega, M]$, where a is the semi-major axis,

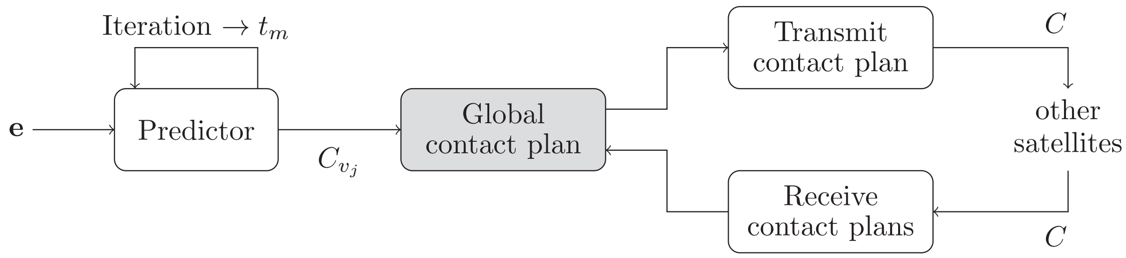


Figure 5.1: Block diagram of the predictive algorithm

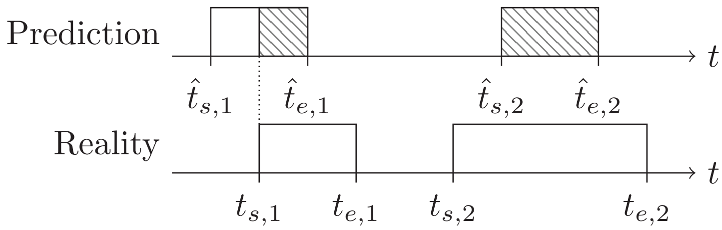


Figure 5.2: Example of a real contact and its estimation, being the dashed area the one that is correctly predicted.

e is the eccentricity, i is the inclination, ω is the argument of periapsis, Ω the argument of the ascending node, and M the mean anomaly of the satellite position[12]. The goal of this predictor is to estimate future contacts without propagating an orbit trajectory model, just with a simple and direct formulation. The predictor constructs the local contact plan with the predicted contacts. After computing the local contact plan, the global contact plan is updated in each satellite. As this global plan is uncompleted, the satellite exchanges it with its neighbors (and vice-versa) when a contact starts. At the end, the satellite is able to have a complete contact plan, which can be used by a routing protocol.

The definition of the predictor becomes crucial to ensure accurate estimations. For this reason, two predictor models have been evaluated in Sections 5.3 and 5.4 respectively. This predictor must be accurate enough to correctly schedule the estimated contacts. For instance, Figure 5.2 illustrates two predicted contacts which both are partially correct with respect to the real contacts. In particular, the striped areas are the time slots in which the prediction and the reality match. This predictor has unsuccessfully estimated the beginning of the former estimation, and the contact duration of the other one. This situation could exist depending on the predictor definition. If the predictor is based on constantly and periodically propagate an orbit trajectory (traditional method), the error between predicted and real contacts would not exist (consuming more resources). However, if the predictor uses a relative trajectory model due to its definition, errors in the predictions can appear, being the predictor not accurate enough. Therefore, two features characterize the accuracy of a contact predictor: the punctuality, and the duration matching.

Definition 4. ***Punctuality of an estimated contact***

It is the probability to estimate a contact inside the real time slot. Considering J contacts, the end time $\hat{t}_{e,j}$ and the start time $\hat{t}_{s,j}$ of the estimated contact $j \in J$ (where $\hat{t}_{s,j} < \hat{t}_{e,j}$), and the end time $t_{e,j}$ and start time $t_{s,j}$ of the real contact $j \in J$ (where $t_{s,j} < t_{e,j}$), the punctuality of the jth estimated contact P_j is defined as follows.

$$P_j = \begin{cases} \left(\dfrac{\hat{t}_{e,j} - t_{s,j}}{\hat{t}_{e,j} - \hat{t}_{s,j}}\right) & \text{if } \hat{t}_{s,j} < t_{s,j} \text{ and } t_{s,j} < \hat{t}_{e,j} < t_{e,j} \\ \left(\dfrac{t_{e,j} - \hat{t}_{s,j}}{\hat{t}_{e,j} - \hat{t}_{s,j}}\right) & \text{if } t_{s,j} < \hat{t}_{s,j} < t_{e,j} \text{ and } \hat{t}_{e,j} > t_{e,j} \\ 0 & \text{if } \hat{t}_{s,j} \geq t_{e,j} \text{ or } \hat{t}_{e,j} \leq t_{s,j} \\ 1 & \text{otherwise} \end{cases} \tag{5.4}$$

Definition 5. ***Predictor punctuality***

[12] The position of a satellite can be defined by the mean anomaly angle M, the eccentric anomaly angle E, the true anomaly angle ν, and the distance from the orbit center r. Further information of those anomaly angles is presented in Section 5.3, and in (Bate et al., 2019)

Considering J contacts, the punctuality P is the average of the punctuality of all the estimated contacts P_j ($\forall j \in J$). Its definition is presented in (5.5).

$$P = \frac{1}{J} \sum_{\forall j \in J} P_j \tag{5.5}$$

Definition 6. ***Duration matching of an estimated contact***

It is the probability to correctly estimate the contact duration. Considering J contacts, the start time $\hat{t}_{s,j}$ and the end time $\hat{t}_{e,j}$ of the estimated contact $j \in J$ (where $\hat{t}_{s,j} < \hat{t}_{e,j}$), the start time $t_{s,j}$ and the end time $t_{e,j}$ of the real contact $j \in J$ (where $t_{s,j} < t_{e,j}$), the duration matching of the jth estimated contact L_j is defined as follows.

$$L_j = \begin{cases} \left(\dfrac{\hat{t}_{e,j} - \hat{t}_{s,j}}{t_{e,j} - t_{s,j}}\right) & \text{if } (\hat{t}_{e,j} - \hat{t}_{s,j}) \leq (t_{e,j} - t_{s,j}) \\ \left(\dfrac{t_{e,j} - t_{s,j}}{\hat{t}_{e,j} - \hat{t}_{s,j}}\right) & otherwise \end{cases} \tag{5.6}$$

Definition 7. ***Predictor duration matching***

Considering J contacts, the duration matching L is the average of the duration matching of all the estimated contacts L_j ($\forall j \in J$). Its definition is presented in (5.7).

$$L = \frac{1}{J} \sum_{\forall j \in J} L_j \tag{5.7}$$

Definition 8. ***Predictor Accuracy***

The accuracy of a predictor Q is the combination of the punctuality P, and the duration matching L.

$$Q = P \cdot L \tag{5.8}$$

Using these definitions, the following sections present the details of the predictors and their accuracy. The predictors have been defined for satellite constellations targeting EO missions. This kind of architecture is characterized by circular orbits at the similar altitudes.

5.3 Prediction with the Probability of Close Approach

A satellite contact can be modeled as a temporal closeness of two satellites. In the course of the past years, space object proximity has concerned the community due to the possibility to experience spacecraft or space debris collisions. Consequently, large efforts intended to statistically appraise these proximities. One of the outcomes of this research has been the definition of the PCA (Richey, 1986) between a *deputy* satellite with respect to a *chief* one. The PCA estimates the possibility of two satellites to be close enough to consider it a nearness, i.e. within a distance threshold. The corresponding formulation of the PCA is detailed in (Richey, 1986), providing the following compact definition:

$$PCA = \int_{-\pi}^{\pi} \frac{1}{4\pi^2} \, \Delta M_d(v_c) \, \frac{dM_c(v_c)}{dv_c} \, dv_c \ , \tag{5.9}$$

where v_c corresponds to the true anomaly of the *chief* satellite, M_d the mean anomaly of the *deputy* satellite, and M_c the mean anomaly of *chief* satellite. To better understand the PCA definition, let's inspect each term that composes its definition. The derivative term of the mean anomaly $\frac{dM_c(v_c)}{dv_c}$ characterizes the deviation of the mean anomaly M with respect to the true anomaly

v due to the orbit eccentricity e:

$$\frac{dM(v)}{dv} = \frac{(1-e)\,\beta}{[\cos^2(v/2) + \beta^2 \sin^2(v/2)]^2} \ , \tag{5.10}$$

where $\beta = \sqrt{(1-e)/(1+e)}$. The other term $\Delta M_d(v_c)$ corresponds to the mean anomaly regions in which the distance between the *deputy* satellite and the *chief* one is less than a distance threshold, so-called close approach regions. This term can be represented by the union of each jth close approach region in the *deputy* satellite orbit, defined by the true anomalies $v_{d/j1}$ and $v_{d/j2}$:

$$\Delta M_d(v_c) = \sum_{j=1}^{J} [\, M_d(v_{d/j2}) - M_d(v_{d/j1}) \,] \ , \tag{5.11}$$

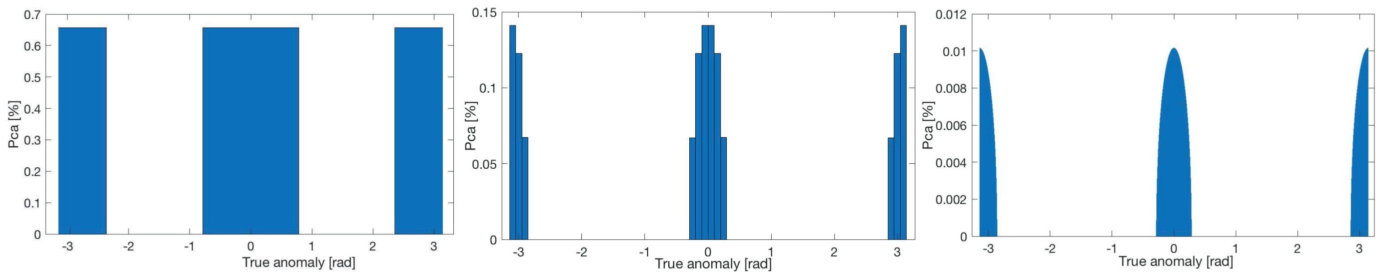
where J is the amount of close approach regions. The method of how to compute these boundaries and regions is highly detailed in (Richey, 1986). EO and broadband communications missions are normally composed of satellites that follow circular orbits (i.e. $e \approx 0$). In these scenarios $\beta \approx 1$, and the derivative term of the mean anomaly $\frac{dM(v)}{dv}$ does not drive the PCA behavior (i.e. $\frac{dM(v)}{dv} \approx 1$). Therefore, the PCA is defined as follows:

$$PCA = \int_{-\pi}^{\pi} \frac{1}{4\pi^2} \, \Delta M_d(v_c) \, dv_c \tag{5.12}$$

This continuous formulation cannot be implemented in a computational algorithm, being necessary to represent it in the discrete domain. Considering the integral step Δv which fragments the space $[-\pi, \pi)$ in regions $R = \{[-\pi, -\pi + \Delta v) \ \ldots \ [\pi - \Delta v, \pi)\}$, the PCA becomes the addition of the probability in each region p:

$$PCA = \sum_{k \in |R|} p_k = \sum_{k \in |R|} \frac{1}{4\pi^2} \, \Delta M_{dk}(v_c) \, \Delta v \ , \tag{5.13}$$

where $\Delta M_{dk}(v_c)$ corresponds to the mean anomaly difference in the kth region of the *chief* true anomaly space. Equation (5.13) highlights the importance of the integral step Δv, which drives the resulting probability of the kth region (i.e. p_k). Therefore, this work explores for different steps the behavior of this probability between two satellites following orbits with 90 of inclination (i), and periapsis argument (w) difference (Figure 5.3). Potential closeness true anomaly regions are identified in all the three cases (i.e. $p_k \neq 0$). However, while the resolution improves decreasing Δv, the probability inside the potential region does not remain homogeneous. This makes some regions more favorable to have a satellite proximity, presented in Figure 5.3 with blue areas. It is thus clearly possible to identify and define these potential regions using this probability. For instance, the close approach regions, in terms of true anomaly, in Figure 5.3c



(a) $\Delta v = \pi/4$ (b) $\Delta v = \pi/32$ (c) $\Delta v = \pi/512$

Figure 5.3: Close approach probability assessment with different true anomaly integral step Δv_c

are $[-0.25,\ 0.25]$ radians, and $[2.75,\ 3.25]$ radians. When the satellite passes through these regions, the probability of having a contact with the other satellite is 0.1%. These regions become the most probable parts of the orbit trajectory to have contacts.

Following this conduct, a selection algorithm has been conceived to provide these true anomaly regions in which a contact is probable. Figure 5.4 presents an example of how this algorithm works. It starts by computing the probability with $\Delta v = 2\pi$, which preliminary evaluates if two satellites could have a closeness (having a non-zero probability). If this condition is not achieved, the algorithm stops indicating that no contact is possible between the satellites. Otherwise, the integration step is decreased, and the probability is computed in each of the new regions. If the resulting probability is zero, the region is discarded. Contrarily, if the region can have a contact, it is explored in the next iteration. This procedure is repeated until reaching the minimum integral step value (configurable). At the end, the true anomaly boundaries are determined with a resolution of the last integral step used.

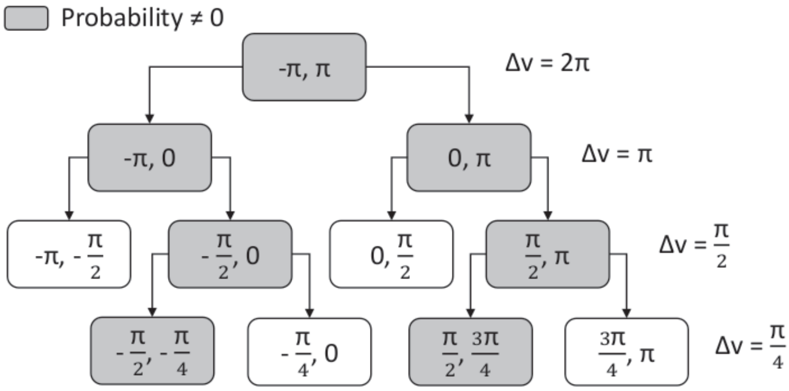


Figure 5.4: Selection algorithm illustration with four levels

After retrieving the true anomaly boundaries, the start and ending times of contacts are computed with an orbit trajectory model. This model presents the formulation to characterize the position of a satellite at each instant time. In our case, the Keplerian orbit (Bate et al., 2019) has been used as the trajectory model, because of its low-computation requirements and enough precision for mission analyses. This orbit model presents the satellite motion considering only the attraction of the Earth body, neglecting any orbital perturbation. Using Kepler's formulation, the true anomaly v, eccentric anomaly E, and mean anomaly M are defined according to the time evolution:

$$M = M_0 + nt \ , \tag{5.14a}$$

$$M = E - e\sin(E) \ , \tag{5.14b}$$

$$v = 2\tan^{-1}\left(\sqrt{\frac{1+e}{1-e}}\tan\frac{E}{2}\right) \ , \tag{5.14c}$$

where M_0 represents the initial mean anomaly, e the orbit eccentricity, and n the mean angular motion.

To evaluate the orbital element difference impact on the predictions, Figure 5.5 presents the estimation of the contact time interval applying the selection algorithm between two satellites with just 30 of inclination difference. The blue line corresponds to the evolution of the distance between the satellites, and the orange one is the distance threshold d_{th} (in this case, 2500 km). The propagation of the Kepler orbit (presented previously) in the ECI frame (Bate et al., 2019) has

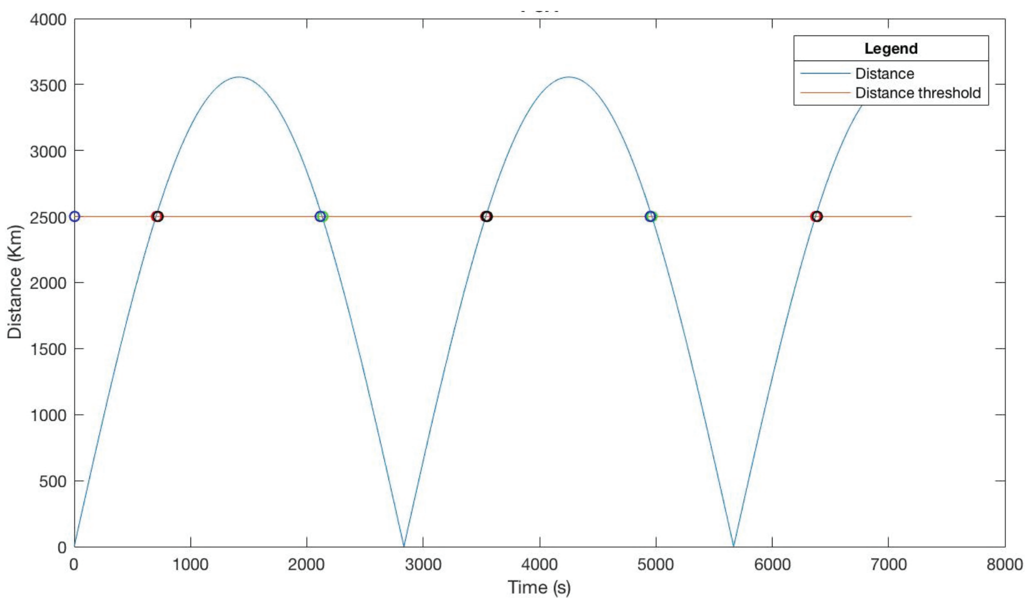


Figure 5.5: Comparison of predicted contacts with respect to satellite distance with 30 of inclination difference

been used to compute just the reference distance between the satellites (blue line)[13]. When the distance is less (or equal) to the threshold value, a contact between the satellites is achieved. Using the presented predictive algorithm, the contacts are estimated and scheduled with blue dots and black dots, indicating the contact start and end respectively. The results indicate that, in this case, this solution estimates the contacts with an accuracy of 98.9%.

Extending the previous results, Table 5.1 presents the performance of the proposed predictor in scenarios with different orbits. The search algorithm has been executed until achieving a resolution of $\frac{\pi}{10^4}$ in the integral step Δv. This predictor is capable to correctly estimate the contact duration of all the contacts (i.e. $L \approx 100\%$), due to the high resolution of the selection algorithm. However, this high resolution provokes more iterations, increasing the processing time. Additionally, the punctuality of the predictor P fluctuates more, from 10.8% to 97.7%. This predictor can thus correctly estimate the duration of each contact, using more computation, but it cannot always correctly schedule them. Moreover, in cases 16, and 17 the predictor schedules perfectly the contacts, because it is able to detect that the satellites are in constant line-of-sight. Cases 5, 10, 12, 15, and 18 are scenarios in which satellites would not have any contact. The predictor cannot deal with this situation, scheduling unrealistic contacts that would never happen. For that reason, the accuracy Q in these scenarios is zero. This behavior appears because of the PCA definition itself, which does not consider the initial distance between the satellites. This initial state is driven by the argument of periapsis w, the argument of the ascending node Ω, and the initial true anomaly M. This algorithm has also been evaluated between three NASA EO satellites: *Calipso*, *Terra*, and *Aqua*. As in previous cases, when the orbit elements difference increases the punctuality of the predictor is degraded. However, in case 19 the accuracy is well enough to be applied in this scenario.

Concluding this section, the predictor based on the PCA cannot correctly estimate satellite contacts. Additionally, due to the search algorithm, it presents an iterative execution which depends on the integration step, that could be computational expensive. For these features, this predictor is not suitable to generate a reliable contact plan. An alternative predictor model has been considered in the section that follows, which presents better features and performance.

[13] Note that the same procedure to compute the distance has been followed for the analyses in the following sections.

Table 5.1: Performance of the Close Approach Probability Predictor with $e = 0.0$

Cases	Chief Satellite					Deputy Satellite					d_{th}	P^*	L^{**}	Q^{***}
	a [km]	i [°]	ω [°]	Ω [°]	M [°]	a [km]	i [°]	ω [°]	Ω [°]	M [°]	[km]	[%]	[%]	[%]
1	6871.0	0.0	0.0	0.0	0.0	6871.0	30.0	0.0	0.0	0.0	2500	97.7	100.0	97.4
2	6871.0	0.0	0.0	0.0	0.0	6871.0	45.0	0.0	0.0	0.0	2500	93.3	100.0	93.3
3	6871.0	0.0	0.0	0.0	0.0	6871.0	60.0	0.0	0.0	0.0	2500	87.3	100.0	87.3
4	6871.0	0.0	0.0	0.0	0.0	6871.0	90.0	0.0	0.0	0.0	2500	70.9	100.0	70.8
5	6871.0	0.0	0.0	0.0	0.0	6871.0	0.0	0.0	30.0	0.0	2500	0.0	0.0	0.0
6	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	0.0	30.0	0.0	2500	97.7	99.9	97.7
7	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	0.0	45.0	0.0	2500	93.5	99.9	93.5
8	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	0.0	60.0	0.0	2500	87.9	99.9	87.9
9	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	0.0	90.0	0.0	2500	72.7	99.9	72.7
10	6871.0	0.0	0.0	0.0	0.0	6871.0	0.0	30.0	0.0	0.0	2500	0.0	0.0	0.0
11	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	10.0	0.0	0.0	2500	86.9	100.0	86.9.0
12	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	30.0	0.0	0.0	2500	0.0	0.0	0.0
13	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	5.0	30.0	0.0	2500	95.6	99.9	95.5
14	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	10.0	30.0	0.0	2500	86.9	99.9	86.8
15	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	30.0	30.0	0.0	2500	0.0	0.0	0.0
16	6871.0	0.0	0.0	0.0	0.0	6871.0	0.0	0.0	0.0	10.0	2500	100.0	100.0	100.0
17	6871.0	0.0	0.0	0.0	0.0	6871.0	0.0	0.0	0.0	20.0	2500	100.0	100.0	100.0
18	6871.0	0.0	0.0	0.0	0.0	6871.0	0.0	0.0	0.0	30.0	2500	0.0	0.0	0.0
19	*Calipso Satellite*					*Terra Satellite*					3500	96.3	99.9	96.3
	7080.7	98.2	80.3	285.6	279.8	7080.0	98.2	83.8	251.3	276.4				
20	*Aqua Satellite*					*Terra Satellite*					3500	13.4	100.0	13.4
	7080.7	98.2	89.5	96.8	270.6	7080.0	98.2	83.8	251.3	276.4				
21	*Aqua Satellite*					*Calipso Satellite*					3500	10.8	100.0	10.8
	7080.7	98.2	89.5	96.8	270.6	7080.7	98.2	80.3	285.6	279.8				

*Punctuality; **Duration matching; ***Accuracy

5.4 Prediction with relative orbital motion

During the last years, different researchers have focused on conceiving flight formation and rendezvous missions. For these missions, the relative position of a *deputy* satellite with respect to a *chief* satellite has been studied generating different relative orbit models. Those models are based on the Hill coordinate frame O (Hill, 1878), which is a Cartesian coordinate system centered in the *chief* position. The orientation of this system is given by the unit vector $\{\hat{o}_r,\, \hat{o}_\theta,\, \hat{o}_h\}$, where $\hat{o}_r$ is in the orbit radius direction, whereas $\hat{o}_\theta$ is parallel to the orbit momentum vector, and $\hat{o}_h$ completes the right-handed coordinate system. The relative position vector $\boldsymbol{\rho}$ of a *deputy* satellite with respect to the *chief* one is expressed on three components.

$$\boldsymbol{\rho} = [x,\ y,\ z]^T \tag{5.15}$$

This vector represents thus the position of the *deputy* satellite from the point of view of the *chief*, determining directly the distance between them.

$$d = \|\boldsymbol{\rho}\|_2 = \sqrt{x^2 + y^2 + z^2} \tag{5.16}$$

The definition of these three components becomes essential to determine the distance. Different researchers have used non-linear models to characterize these components. However, Schaub (2004) presents linear expressions using the orbit element differences $\delta\boldsymbol{e}$ (Alfriend and Schaub, 2000).

$$\delta\boldsymbol{e} = [\delta a,\ \delta e,\ \delta i,\ \delta\omega,\ \delta\Omega,\ \delta M] \tag{5.17}$$

This representation simplifies the definition because only the δM component is time variant. Applying (5.17), the Clohessy-Wiltshire (CW) relative equations (Clohessy and Wiltshire, 1960), which determine the relative motion with respect to a circular *chief* satellite orbit, have the convenient analytic solution:

$$x(t) = A_0 \cos(nt + \alpha) + x_{\text{off}}\ , \tag{5.18a}$$

$$y(t) = -2\, A_0 \sin(nt + \alpha) + y_{\text{off}} + \frac{3}{2}\, n\, x_{\text{off}}\, t\ , \tag{5.18b}$$

$$z(t) = B_0 \cos(nt + \beta)\ , \tag{5.18c}$$

where A_0, B_0, α, β, x_{off}, and y_{off} are the integration constants determined by the initial conditions, n the mean angular motion of the *chief* satellite, and t represents the time. These expressions present the relative motion of a *deputy* satellite with respect to a *chief* satellite, while this *chief* follows a circular orbit. EO and broadband communications missions are normally composed of satellites that follow circular orbit, and same altitude (i.e. $e \approx 0$, $\delta e \approx 0$, and $\delta a \approx 0$). Applying these conditions into the definition of the integration constants presented in (Schaub, 2004), they become as follows:

$$A_0 = 0\ , \tag{5.19a}$$

$$B_0 = a\,\delta_\omega \approx a\,\sqrt{\delta i^2 + sin^2(\delta i)\,\delta\Omega}\ , \tag{5.19b}$$

$$\alpha = M_0\ , \tag{5.19c}$$

$$\beta = \omega - tan^{-1}\left(\frac{\delta i}{-sin(i)\,\delta\Omega}\right)\ , \tag{5.19d}$$

$$x_{\text{off}} = 0\ , \tag{5.19e}$$

$$y_{\text{off}} = a\,[\delta\omega + \delta M + cos(i)\,\delta\Omega]\ , \tag{5.19f}$$

where M_0 corresponds to the initial mean anomaly of the *chief*. As noted previously, this model has largely been used to estimate the distance between satellites. To the best of our knowledge, it has never been used before to estimate the time slot of a satellite contact. For this reason, the following work describes the formulation to estimate the contact time boundaries from the relative orbital motion equations. Considering an omnidirectional antenna pattern, an ISL is feasible if (and only if) the distance d is less (or equal) to the communication range d_{th}.

$$d \le d_{th} \tag{5.20}$$

Applying the values of (5.19) in (5.18), the ISL existence condition, presented in (5.20), becomes

$$y_{\text{off}}^2 + (a\,\delta_\omega)^2\cos^2(nt+\beta) \le d_{\text{th}}^2 \tag{5.21}$$

At the end, a more compact formulation of the ISL existence condition is presented in (5.22).

$$\gamma = \pm\sqrt{\frac{d_{\text{th}}^2 - y_{\text{off}}^2}{a^2\,\delta_\omega^2}} \ge \cos(\sigma)\ , \tag{5.22}$$

where $\sigma = nt + \beta$ is a compressed expression of the time variant component, and γ determines ISL characteristics. Depending on γ, this condition is always or never satisfied.

Theorem 1. ***Existence of an Inter-Satellite Link***

Let d_{th} be the maximum range of an omnidirectional ISL transceiver, and y_{off} the integration constant in the in-track component of the deputy relative position. An ISL with these features exists between a chief satellite and a deputy satellite if the condition (5.23) is satisfied.

$$|d_{\text{th}}| \ge |y_{\text{off}}| \tag{5.23}$$

Proof. From (5.22), γ can be complex $\mathbb{C}$ or real $\mathbb{R}$. An ISL exists if γ belongs to the real set $\gamma \in \mathbb{R}$. To achieve this situation, $d_{\text{th}}^2 - y_{\text{off}}^2 \ge 0$, which concludes to (5.23). □

Theorem 2. ***Sporadic Inter-Satellite Link***

Let d_{th} be the maximum range of an omnidirectional ISL transceiver, y_{off} the integration constant in the in-track component of the deputy relative position, and $a\,\delta_\omega$ the integration constant in the cross-track component of the deputy relative position. An ISL with these features is time-invariant (i.e. infinite duration), if the condition (5.24) is satisfied.

$$d_{\text{th}}^2 \ge (a\,\delta_\omega)^2 + y_{\text{off}}^2 \tag{5.24}$$

Otherwise, it becomes time-variant with a bounded lifetime.

Proof. Equation (5.22) presents the condition in which the distance is less than a distance threshold, having a feasible ISL. The condition is determined by a cosinusoidal behavior, defined in the range $[-1,\,1]$. For this reason, if $\gamma \ge 1$ the condition is always satisfied, and the ISL is feasible. Taking the definition of γ, and this condition, the theorem is proven. □

Theorems 1 and 2 present two conditions that filters those contacts that are feasible. When the corresponding conditions are satisfied, a temporal contact between *chief* and *deputy* satellites exists. The time boundaries of this contact are given by the solution of (5.22). Because γ is defined by a square root, two values exist that can satisfy (5.22). Let's denote γ^+ as the positive solution of the square root (i.e. $\gamma^+ = |\gamma| \geq 0$) and γ^- as the negative one (i.e. $\gamma^- = -\gamma^+$). Following these definitions, (5.22) is presented with different formulations.

$$\gamma^+ \geq \cos(\sigma) \tag{5.25a}$$

$$\gamma^- \leq \cos(\sigma) \tag{5.25b}$$

Figure 5.6 represents the values of a cosine in the axes and the σ space in which both conditions are satisfied. In particular, σ has to be bounded in the following space to satisfy (5.25a):

$$\arccos(\gamma^+) \leq \sigma \leq \arccos(\gamma^-) = \arccos(-\gamma^+) \tag{5.26}$$

Simultaneously, σ must be defined inside the following boundaries to satisfy (5.25b):

$$\pi + \arccos(\gamma^+) \leq \sigma \leq \pi + \arccos(\gamma^-) \tag{5.27}$$

Taking in consideration σ and the γ^- definitions, the contact time is bounded by known limits, shown in (5.28).

$$\frac{\arccos(\gamma^+) - \beta + 2\pi k}{n} \leq t \leq \frac{\arccos(-\gamma^+) - \beta + 2\pi k}{n} \tag{5.28a}$$

$$\frac{\arccos(\gamma^+) + \pi - \beta + 2\pi k}{n} \leq t \leq \frac{\arccos(-\gamma^+) + \pi - \beta + 2\pi k}{n} \tag{5.28b}$$

where k represents the number of turns that a satellite has performed following the orbit. These boundaries determine the beginning and the end of a contact. Equations (5.28a) and (5.28b) determine the model of the predictor based on relative satellite motion, which is used to construct the global contact plan. Figure 5.7 presents the execution of this predictor showing different estimated contacts between two satellites with an inclination difference of 80. The distance (blue line) evolves over time generating different regions in which it is lower than the distance threshold 2500 km (orange line). In this scenario, contacts are correctly estimated and scheduled, with an accuracy of 92.6%.

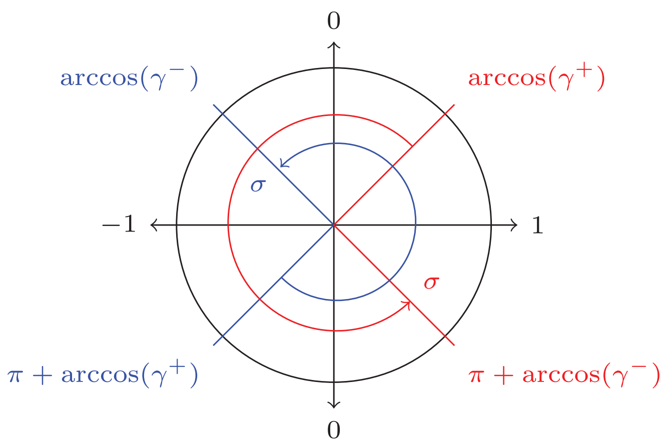


Figure 5.6: Representation of σ value space in which (5.22) is satisfied

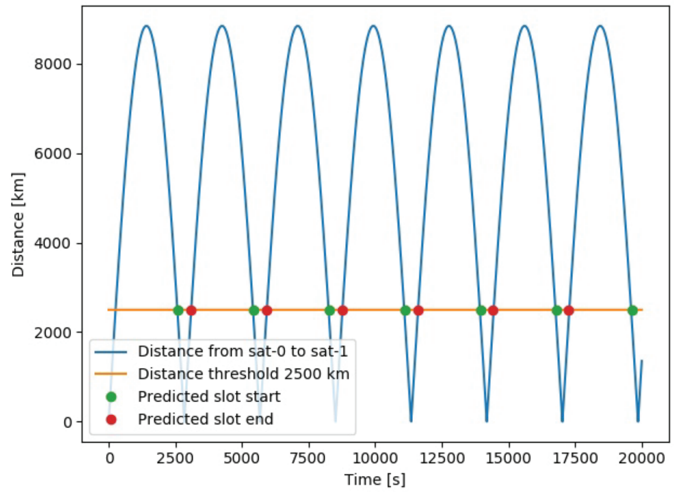


Figure 5.7: Comparison of predicted contacts with respect to satellite distance ($\delta i = 80$)

Extending the results, this predictor has been evaluated in scenarios with different orbits (as previously done in Table 5.1). Table 5.2 presents its performance in terms of punctuality (P), duration matching (L), and accuracy (Q). The results indicate that the accuracy of the predictor is high (i.e. larger than 90%) when the inclination is the main difference. Furthermore, the duration matching metric is the one which determines the accuracy in almost all the cases. This indicates that the predictions are scheduled correctly inside the real contact time slot, however the duration of the estimated contact does not match with the reality. This impacts the communications duration, but it ensures that in all the contacts the ISL is a fact. Additionally, the predictor is able to detect those cases in which a contact never appears, such as cases 5, 10, 12, 15, and 18. However, when the difference is based on the right ascending node, the argument of perigee, and the initial mean anomaly the prediction is less accurate. This behavior is induced by the linearization of the relative motion (i.e. the CW equations), which is valid when the orbit element difference vector is small. Figure 5.8 presents an example of the accuracy improvement while using the non-linear definition of the integration constant δ_ω, presented in (5.29).

$$\cos(\delta_\omega) = \cos(i)\cos(i+\delta i) + \sin(i)\sin(i+\delta i)\cos(\delta\Omega) \tag{5.29}$$

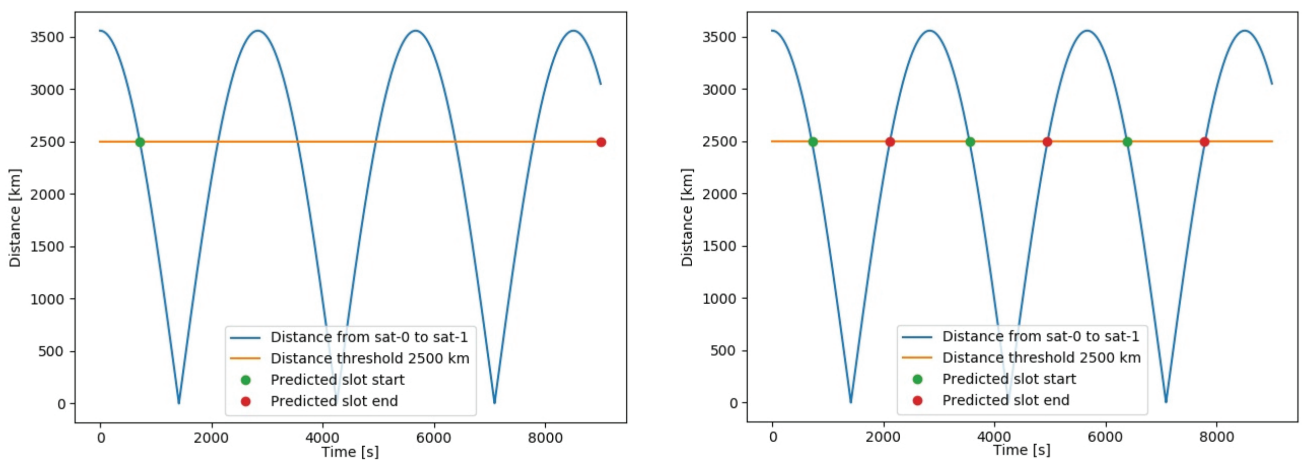


Figure 5.8: Representation of linearity impact, using a linear model (left) and a non-linear one (right)

Despite this limitation, the predictor is still able to estimate contacts between real satellites. Furthermore, the predictor is able to identify those cases in which the satellites are always in line-of-sight, having a constant ISL. For instance, the accuracy of the predictor is 99.9% of cases 16, and 17. This performance is due to the inclusion of the orbital elements in the definition of the predictor, unlike the predictor based on the PCA. In Table 5.2, case 19 presents the accuracy of the predictor while detecting contacts between *Calipso* and *Terra* NASA satellites. As both follow close orbits in terms of orbital elements, the predictor is able to estimate the contacts with 94.3% of accuracy. This accuracy is degraded when the prediction is computed between *Terra* and *Aura* satellites (case 20), because the orbital element difference increases. Finally, the predictor can estimate contacts with an accuracy of 22.2% between *Calipso* and *Aura* satellites (case 21). Although this reduction of the accuracy, all the estimations have 100% of punctuality (P), indicating that they are inside the real time slot (i.e. maximum punctuality).

Comparing the performance of the predictors based on the PCA and the relative satellite motion, the last one is able to estimate and schedule satellite contacts with higher accuracy. However, the linear representation of the predictor decreases the duration matching of the estimations when the orbit element difference is considerable. Although this limitation, all the estimated contacts are punctual, scheduling them when an ISL is feasible, and thus ensuring the communications. For these features, this predictor has been selected to generate the contacts that are used to construct the local contact plan.

5.5 Generating the Contact Plan

Previous section has presented how the contact prediction is computed and the performance of the predictor. Implementing this predictor in each satellite provides the local contact plan, but not the global one. This section describes in details the complementary part of the algorithm to construct a global contact plan in each satellite. The protocol is based on exchanging different information between satellites. The former is the orbital elements which are required by the predictor to estimate the contacts, and to define the local contact plan. In this scenario, each satellite periodically transmits its TLE which are received (in due course) by its neighbors. This transmission is performed using the UDP. After receiving a TLE, the satellite retrieves the orbital elements of the neighbor, and defines the contact sequence of this *deputy* satellite using the predictor. Thus, the predictor iteratively defines future contacts after finishing the last one. The number of contacts predicted per iteration can be configured depending on each scenario. Note that a tradeoff regarding the forward knowledge with respect to the memory usage needs to be conducted in order to define this parameter. This tradeoff has not been considered in the present work because it depends on the spacecraft characteristics, related to each mission. With the accumulation of the different TLEs, the satellite defines the local contact plan. Then, the satellite schedules a request of the local contact plan to each neighbor. To do that, it establishes a TCP connection with the neighbor satellite, and it iteratively updates the internal global contact plan with the received local contact plan. Note that the local contact plan of the neighbor is at the same time incomplete, because it only contains information that it learned. However, the global contact plan is being completed after a lapse of time.

This de-centralized solution becomes suitable for heterogeneous satellite systems, like the ones proposed in the IoSat paradigm. The satellites in this scenario are operated by different operation entities, and follow different orbit trajectories. Subsection 5.5.1 evaluates the performance of the proposed protocol with a hybrid and heterogeneous satellite system. Additionally, the de-centralized solution provides a scalable strategy that is suitable with large satellite con-

Table 5.2: Relative Motion Predictor performance with $e = 0.0$

Cases	Chief Satellite					Deputy Satellite					d_{th}	P^*	L^{**}	Q^{***}
	a [km]	i [°]	ω [°]	Ω [°]	M [°]	a [km]	i [°]	ω [°]	Ω [°]	M [°]	[km]	[%]	[%]	[%]
1	6871.0	0.0	0.0	0.0	0.0	6871.0	30.0	0.0	0.0	0.0	2500	100.0	98.9	98.9
2	6871.0	0.0	0.0	0.0	0.0	6871.0	45.0	0.0	0.0	0.0	2500	100.0	97.4	97.4
3	6871.0	0.0	0.0	0.0	0.0	6871.0	60.0	0.0	0.0	0.0	2500	100.0	95.6	95.6
4	6871.0	0.0	0.0	0.0	0.0	6871.0	90.0	0.0	0.0	0.0	2500	100.0	90.3	90.3
5	6871.0	0.0	0.0	0.0	0.0	6871.0	0.0	0.0	30.0	0.0	2500	100.0	100.0	100.0
6	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	0.0	30.0	0.0	2500	100.0	98.7	98.7
7	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	0.0	45.0	0.0	2500	100.0	97.5	97.5
8	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	0.0	60.0	0.0	2500	100.0	95.8	95.8
9	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	0.0	90.0	0.0	2500	100.0	91.0	91.0
10	6871.0	0.0	0.0	0.0	0.0	6871.0	0.0	30.0	0.0	0.0	2500	100.0	100.0	100.0
11	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	10.0	0.0	0.0	2500	100.0	99.9	99.9
12	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	30.0	0.0	0.0	2500	100.0	100.0	100.0
13	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	5.0	30.0	0.0	2500	98.4	95.8	94.3
14	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	10.0	30.0	0.0	2500	95.9	91.4	87.6
15	6871.0	90.0	0.0	0.0	0.0	6871.0	90.0	30.0	30.0	0.0	2500	100.0	100.0	100.0
16	6871.0	0.0	0.0	0.0	0.0	6871.0	0.0	0.0	0.0	10.0	2500	100.0	99.9	99.9
17	6871.0	0.0	0.0	0.0	0.0	6871.0	0.0	0.0	0.0	20.0	2500	100.0	99.9	99.9
18	6871.0	0.0	0.0	0.0	0.0	6871.0	0.0	0.0	0.0	30.0	2500	100.0	100.0	100
19	*Calipso* Satellite					*Terra* Satellite					3500	100.0	94.3	94.3
	7080.7	98.2	80.3	285.6	279.8	7080	98.2	83.8	251.3	276.4				
20	*Aqua* Satellite					*Terra* Satellite					3500	100.0	54.6	54.6
	7080.7	98.2	89.5	96.8	270.6	7080	98.2	83.8	251.3	276.4				
21	*Aqua* Satellite					*Terra* Satellite					3500	100.0	22.2	22.2
	7080.7	98.2	89.5	96.8	270.6	7080.7	98.2	80.3	285.6	279.8				

*Punctuality; **Duration matching; ***Accuracy stellations, such as the Mega-constellations. For that reason, the performance in terms of construction time has been evaluated in Subsection 5.5.2. Both scenarios have been executed using the event-based simulation engine conceived for DSS architectures—presented in Chapter 3.

5.5.1 Scenario with a hybrid satellite system

The de-centralized nature of the proposed predictive protocol make it suitable for an heterogeneous and hybrid satellite system, which could be composed of different satellite systems. As the ONION project highlights in (Alarcón et al., 2018), this kind of system could provide different features that benefit current EO missions. Therefore, to evaluate the capacity to construct the global contact plan in this context, a case study with a hybrid satellite system is presented in this subsection. Table 5.3 shows the orbit characteristics of the hybrid satellite system, composed of a tropical satellite constellation and a polar one. Five satellites belong to the polar satellite constellation, which follow an orbit trajectory with 90 of inclination. Each satellite plane is equidistant with $\delta\Omega$ of 50, ensuring a total coverage of the equatorial area. Additionally, five more satellites compose the tropical constellation with an orbit of 10 inclination. As in the previous constellation, satellite planes are equidistant placed with $\delta\Omega = 50$. This satellite system is a representation of a hybrid system composed of the Cyclone Global Navigation Satellite System (CYGNSS) (Ruf et al., 2018), and *Planet Labs* satellites (Marshall and Boshuizen, 2013).

Table 5.3: Orbit characteristics of the hybrid system

	Identifier	a [km]	e	i [°]	ω [°]	Ω [°]	M [°]
Tropical	sat-0	6876	0.0	10.0	0.0	0.0	0.0
	sat-1	6876	0.0	10.0	0.0	50.0	0.0
	sat-2	6876	0.0	10.0	0.0	150.0	0.0
	sat-3	6876	0.0	10.0	0.0	200.0	0.0
	sat-4	6876	0.0	10.0	0.0	300.0	0.0
Polar	sat-5	6876	0.0	90.0	0.0	0.0	0.0
	sat-6	6876	0.0	90.0	0.0	50.0	0.0
	sat-7	6876	0.0	90.0	0.0	150.0	0.0
	sat-8	6876	0.0	90.0	0.0	200.0	0.0
	sat-9	6876	0.0	90.0	0.0	300.0	0.0

Figure 5.9 presents the global contact plan generated from the physical trajectory of the satellites (without using any predictive algorithm) after 14000 seconds of simulation. The local contact plan is presented for each satellite, that contains the different contacts over time with the neighbors (represented as colored boxes). Note that each satellite of the tropical constellation has a unique neighbor, which belongs to the polar satellite constellation. In particular, this neighbor is the one that follows an orbit with the same argument of the ascending node Ω. In addition to these contacts, the satellites of the polar constellation have contacts between them in the Earth poles. This kind of constellation is characterized by having high satellite density in these areas, which is interesting to exchange information. Although the satellites of the tropical constellation cannot establish ISLs among them, they can use the contacts with the satellites of the polar constellation to recover their contact plan. This is achieved with the proposed predictive algorithm.

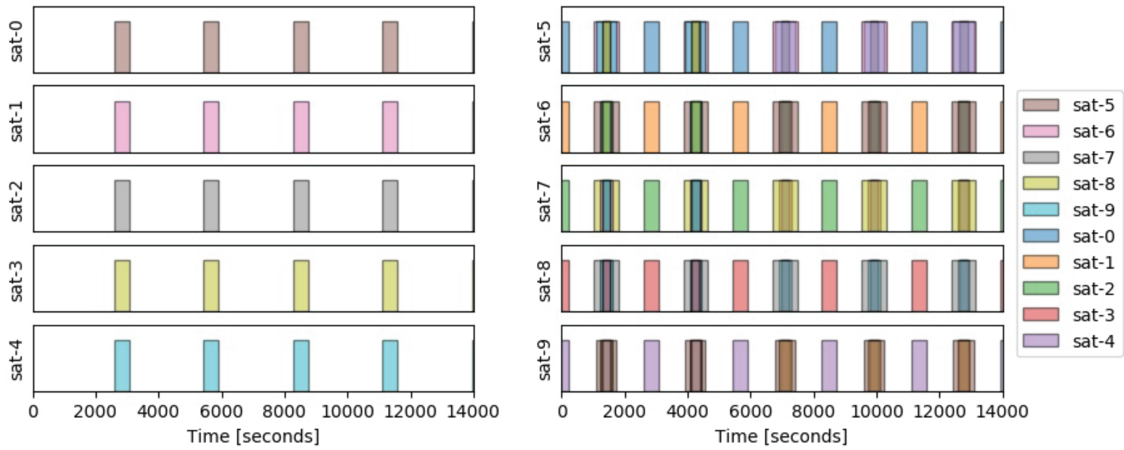


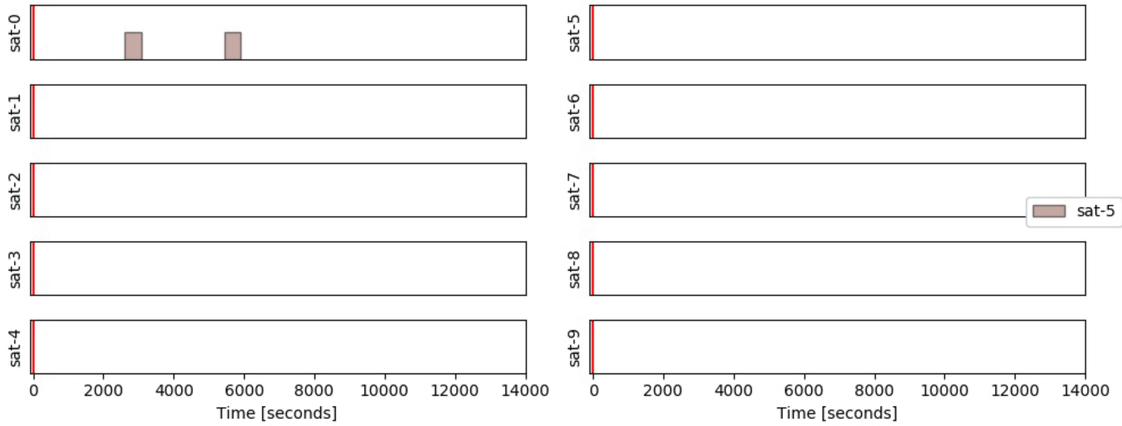
Figure 5.9: Global and complete contact plan of the hybrid satellite system

Figure 5.10 presents three moments (indicated with the red line) of the global contact plan construction from the satellite *sat-0* point of view. At the very beginning, *sat-0* only has information about its neighbor satellite *sat-5*. For that reason, only the contacts with this neighbor are predicted in Figure 5.10a. When this contact is feasible, it exchanges the global contact plan with *sat-5*, and vice versa. Due to the polar constellation nature, *sat-5* has had the contacts with the other satellites of the constellation before 2000 seconds of simulation. This enabled it to learn about its new neighbors (predicting the corresponding contacts), and update its global contact plan with their information. For that reason, when *sat-5* provides its global contact plan to *sat-0*, it can correctly schedule the different contacts of the other satellites, as shown in Figure 5.10b. This behavior is continuously repeated completing the global view of *sat-0*, presented in Figure 5.10c. Note that the number of predicted contacts by the predictor performed at each iteration limits the forward knowledge of the global contact plan. Depending on this knowledge, the performance of the routing protocol (e.g. the CGR) could be impacted. However, increasing this parameter would require more memory storage in each satellite. This study needs to be conducted for each satellite mission, which is out of scope of the presented work.

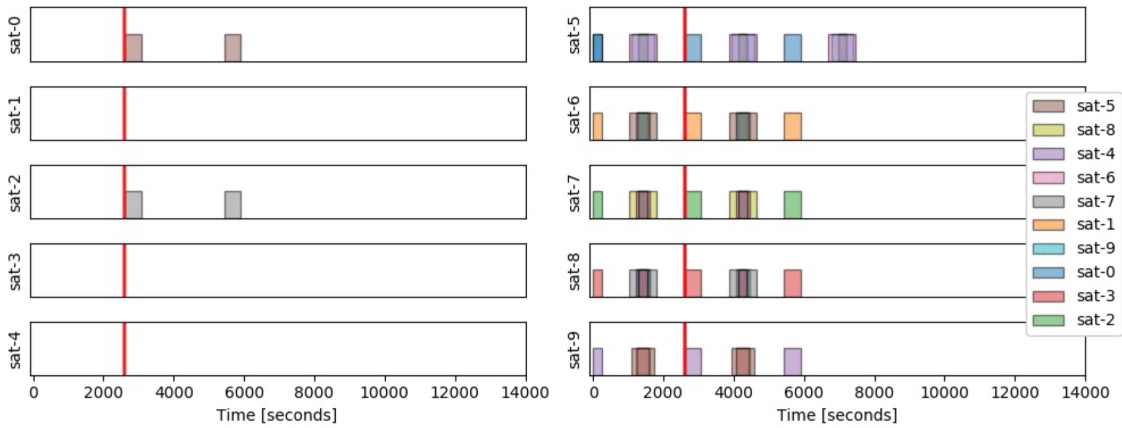
In conclusion, the presented results demonstrate that using the predictive algorithm each satellite can dynamically construct the global contact plan by its own. This de-centralized algorithm is suitable for hybrid and heterogeneous satellite systems, because it does not consider any predefined satellite architecture. The following subsection evaluates the performance of the predictive algorithm in a scenario with a large number of satellites.

5.5.2 Scenario with a massive satellite constellation

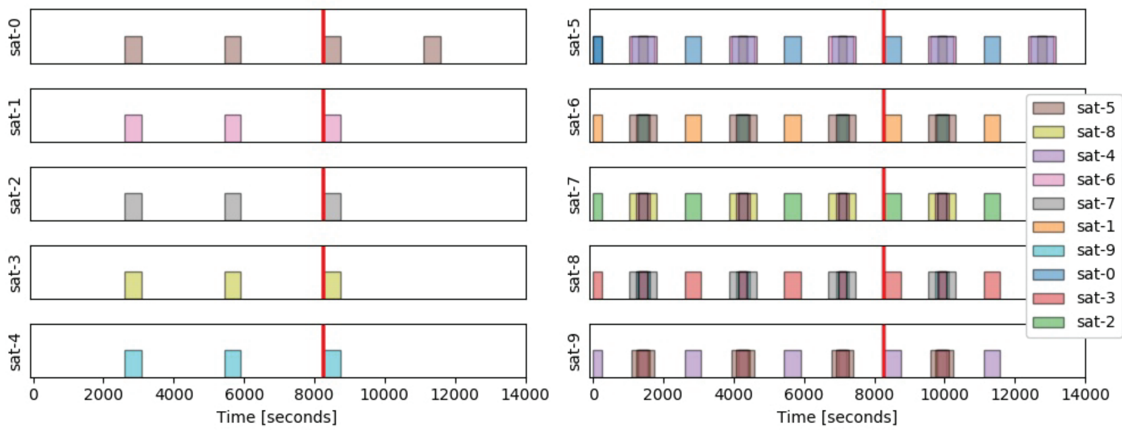
Due to the satellite motion and the algorithm itself, each satellite requires a lapse of time to complete the global contact plan of the network. In the previous analysis, the complete global contact plan has been achieved in a small heterogeneous satellite system. An additional analysis with hundreds of satellites composing a massive satellite constellation has been performed. This analysis highlights the impact of network disruption and network size on the performance of the proposed solution. Table 5.4 presents the characteristics of the Telesat constellation, detailed in (Portillo et al., 2019). It is composed of two groups of satellites: Type 1 and Type 2. The former is composed of 6 polar planes in which 12 satellites per plane follow a circular orbit at 1000 km of altitude. The other group follows the same architecture with planes at 1200 km of altitude and 37.4 of inclination. All the planes are equally spaced in terms of argument of the



(a) Local contact plan generated with the TLE of the neighbor *sat-5* (at 6 seconds of simulation)



(b) Contact plan generated with the collected information from the neighbor *sat-5* (at 2599 seconds of simulation)



(c) Iterative construction of the contact plan (at 8749 seconds of simulation)

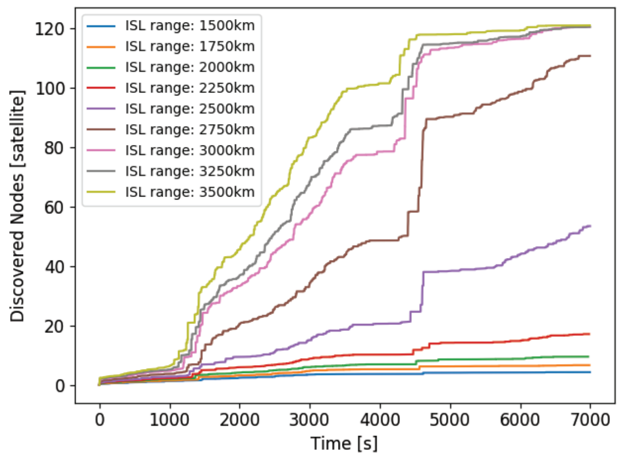
Figure 5.10: Sequential construction of the global contact plan by satellite $sat-0$

ascending node Ω, and the satellites are also equally located in the orbit with an incremental argument of periapsis ω. Note that this architecture is similar to the hybrid constellation presented in Table 5.3, with a larger number of spacecraft. In total, 122 satellites are deployed to achieve global Earth coverage.

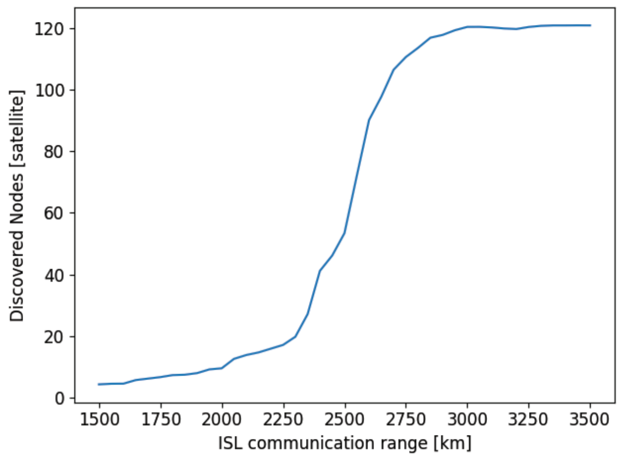
Table 5.4: Telesat Mega-constellation characteristics

Plane type	Identifier	Planes	a [km]	i [º]	e [º]	satellites / plane
Type 1	sat-0	6	7371 km	99.5º	0.0	12
Type 2	sat-72	6	7571 km	37.4º	0.0	9

As explained before, using the predictive algorithm a satellite learns the contacts of other satellites, meanwhile it shares its own knowledge. Thus, the satellite discovers new remote spacecraft with which interact using a scheduled sequence of contacts. Figure 5.11a presents the averaged discovered remote satellites over time, considering all the satellites that compose the constellation. The number of discovered satellites depends on the connectivity of the constellation, which is directly related to the ISL communications range. For that reason, Figure 5.11a shows the results with different ranges. Note that the satellites start discovering only their neighbors, presenting a slow slope in all the cases. After 1000 seconds of simulation, the contact plan of each satellite increases, sharing the predicted contacts with their neighbors. Therefore, each satellite discovers remote satellites that cannot be directly linked (i.e. point-to-point). This behavior remains until reaching an inflection point (at 4500 seconds) in which the number of discovered satellites increases substantially. This event appears in all the scenarios because each satellite collects enough information to entail a considerable benefit to the others in their contact plan construction. The importance of this event is linked to the communications range, having a maximum at 2750 km. After that, each satellite has wide knowledge of the network that the exchange of contact plans does not signify an important improvement. Some differences are detected depending on the connectivity of the network. In particular, Figure 5.11b shows an important correlation: the number of discovered satellites at the end of the

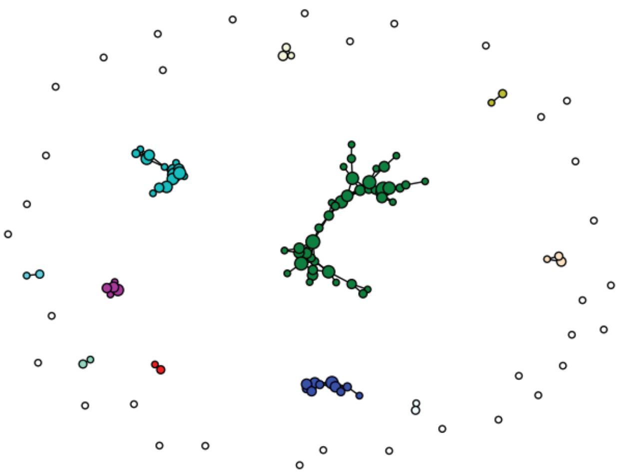


(a) Mean reachable remote nodes over time depending on ISL range

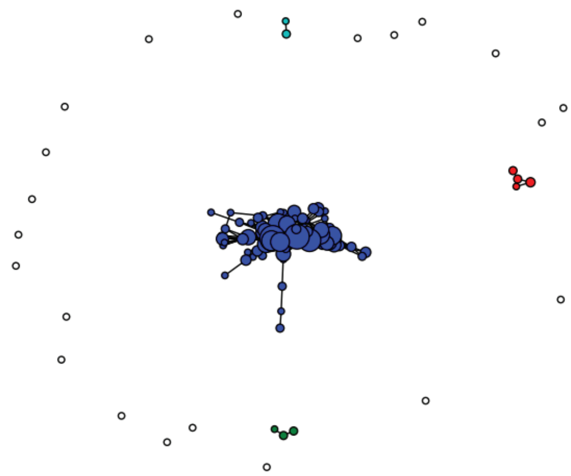


(b) Maximum mean reachable remote nodes depending on ISL range

Figure 5.11: Mean value of the number of remote nodes that can be reached from a satellite of the constellation (left), and maximum number depending on the ISL range (right)



(a) Creation of topology communities with 2000 km of ISL range



(b) Topology communities unification with 2250 km of ISL range

Figure 5.12: Network topology representation generated with the predictive algorithm

simulation varies depending on the communications range. The disruption of the network is the cause of this situation, limiting the possibility to collect information from other satellites. The satellites can discover the 95% of the entire network using ranges larger than 2750 km. However, in those scenarios that the ranges are smaller, the vision of each satellite is limited.

This low range provokes disruption of the network that generates sub-networks, so-called communities, in which all the nodes are connected (at least once). Figure 5.12 presents this existence of communities by plotting the integration of the different topologies over time. The integrated topology with 2000 km of communications range is displayed in Figure 5.12a. Note that a large group of satellites are isolated becoming small white dots in the representation. However, 10 communities (represented by colors) are generated in this scenario. These communities are generated because satellites in adjacent planes with polar orbits have close approaches in polar areas. Additionally, those satellites have a proximity with the satellites with tropical orbits that have the same argument of periapsis ω. When the range increases, a community merges with other ones resulting to a unified and larger community. This is presented in Figure 5.12b, that a large blue community has been created with 2250 km of range. The predictive algorithm is able to learn about these communities, and thus generate the contact plane of them.

5.6 Summary

This chapter has presented a de-centralized predictive protocol that can be used in satellites to estimate contacts with other satellites, and construct the contact plan. This contact plan is a schedule of the contacts between all the satellites that compose the system. The construction of a contact plan allows to predict link status, and thus to anticipate the disruption of the network. Alternatively to centralized solutions, the proposed algorithm is flexible, modular, and scalable which makes it suitable for the IoSat paradigm. The proposed protocol is based on a predictor that iteratively estimates satellite contacts, which are then shared with the other satellites.

Two predictors, and a mechanism to evaluate their prediction accuracy have been defined. The former estimates contacts using the PCA definition, largely used for space object collisions. As highlighted previously, this predictor can correctly estimate the duration of the contacts, but the schedule is not properly done (reaching accuracies of 10% in some cases). For this reason, another predictor using relative orbital motion models has been defined. This new predictor correctly estimates contacts between two satellites following similar orbits. Although it schedules the estimated contact at the correct moment (punctuality larger than 90%), the duration of the estimated contacts between satellites following orbits with important differences is not perfectly estimated. This situation is due to the linearity of the model, which makes it valid for certain scenarios. Therefore, future efforts may be oriented to non-linear relative motion models.

In addition, this work has presented the communication protocol necessary to exchange the contact plan between the satellites, to construct a global one. With a combination of the publication of the TLE and the connection establishment, the system is able to iteratively construct the contact plan in each satellite. A representative hybrid constellation combining CYGNSS and Planet Labs satellites is used to evaluate the performance of the proposed solution. These results demonstrate that each satellite is able to construct the global contact plan, knowing at the end feasible routes to communicate with remote satellites. However, it is highlighted that this final knowledge is achieved after a lapse of time in which the algorithm is discovering. To evaluate the nature of this feature, the algorithm has been evaluated for the Telesat mega-constellation. The results demonstrate that this time is directly related to the ISL communications range, which determines the disruption of the network impact. However, an inflection point in which all the satellites have the needed knowledge appears in all the cases. Additionally, the results also highlight that the proposed algorithm is able to discover those remote nodes with which physically an interaction can be established. Therefore, depending on the disruption of the network, the satellite does not necessarily learn about the entire network, just about its community. This efficient behavior makes it suitable for this kind of massive satellite constellations.

The benefits and the feasibility of applying the proposed algorithm have been demonstrated for future satellite missions. The research presented in this chapter has been focused on the design of the contact predictor and the possibility to autonomously construct the contact plan. As a future step, a study may evaluate how this iterative construction process could be integrated in current routing protocols, such as OLSR or the CGR ones. Furthermore, the achieved results motivate the development of further predictors that could provide more accurate estimations. Due to its non-linear behavior, these predictors may be defined by means of Artificial Intelligence (AI) that could enhance the final accuracy. In any case, future investigations can expand the proposed protocol and algorithms presented in this chapter.

6

Development of protocols to publish available services and establish opportunistic federations

6.1 Introduction

The previous study has demonstrated that a satellite can estimate communications contacts with other satellites. This estimation enables to know important features of the ISL, such as when it is created, and its corresponding lifetime. This new information provides a better understanding of the network dynamics to the satellite, which can be used to identify how connections evolve over time. Therefore, a satellite can predict the establishment of ISLs, and in consequence the composition of the future network. This new capability becomes useful in a satellite system in which no specific and fixed architecture is defined in advance, and it is composed of heterogeneous satellites. Therefore, it has great relevance in the IoSat paradigm, in which opportunistic and temporal networks are created. Although the possibility to predict the contact behavior facilitates the management of these networks, their creation is only performed when federations are established. Therefore, understanding the federation nature is crucial to correctly deploy these networks.

Federations between satellites are established to share underutilized satellite resources, such as memory storage or downlink opportunities. These resources are used by the satellite to perform tasks (e.g. a payload execution generates data that is stored in the memory system). These tasks consume the corresponding resources, fluctuating their usage over time. Having always access to these resources becomes important for the correct execution of the mission. For that reason, the traditional design of a satellite oversizes the required resources to ensure that they are always available for the mission tasks. However, this provokes that commonly not all the resources are consumed in the entire mission. Additionally, the resources are no constantly being used. Indeed, a satellite consumes resources according to the performed tasks, which are scheduled at different moments of the mission. This generates periods of time in which the resources are being consumed, normally represented in percentage of the orbit period. For instance, a satellite would execute an EO payload only over specific target areas, which the duration of this execution could corresponds the 15 % of orbit period. This temporal resource usage generates lapses of time in which the resource is not being consumed, becoming thus available to offer it as services to other satellites. For example, a satellite can offer the possibility to store data from other satellite when its memory is not being used. Following the terminology proposed in (Golkar and Lluch, 2015), the satellite that offers the service is known as a *provider*, while the satellite that consumes it is known as a *customer*. Due to the usage fluctuation of the resources, the associated service is not always available. This temporal availability generates valuable opportunities to consume the service, that should be used by the customers. The federations are thus opportunistic interactions between satellites that are established depending on the temporal availability of the corresponding service. A mechanism that allows to notify the available services in a network would enable to catch these opportunities, and thus to establish the federations.

As previously presented in Chapter 1, some mechanisms have been designed to determine the value of resources in satellite federations following an auction-based manner (Ehsanfar and Grogan, 2020b; Pica and Golkar, 2017). The proposed solution uses a custom communications protocol stack to exchange information about the auctions status between the satellites. In particular, information about the available services is inserted with the topology information, which is periodically propagated to all the satellites that compose the network. Therefore, all the satellites are aware of the available services that can be consumed in the network. With this information, a sealed-bid reverse auction pricing mechanism is conducted in the network to sell the services. In this context, the providers are the bidders, which compete with each other to sell its service to the customer. The final price of the service depends on the provider-issued bids, and when the service is awarded to the customer. This one tries to find the best service candidate among the different ones that are being bidden. At the end, this approach enables to manage the existence of multiple instances of the same service, and to optimize the benefits of the provider to offer the service. However, the proposed mechanism to notify the existence of the available service, which is then used to perform the auction, has some features that may not be suitable for satellite networks. In particular, the proposed stack is based on the OLSR protocol which we have identified that has some limitations in satellite networks (Chapter 4), such as impossibility to manage network disruption, and periodic transmissions could entail the waste of power for those satellites that are isolated, and they do not participate in the network. Furthermore, the proposed protocol stack forces all the satellites to periodically transmit its status while they are not involved in any transaction related to a service. Thus, this approach is not suitable for sporadic and opportunistic scenarios, such as IoSat paradigm. A more efficient mechanism to notify the available services must be conceived.

Another fundamental aspect to deploy federations over ISN is the necessary mechanisms to perform the establishment and the management of the federation itself. Once a satellite knows which satellites have available services, the question falls on which is the protocol that a satellite needs to follow to establish a federation with this provider. Lluch et al. (2015) presented a preliminary approach to establish these federations combining current Internet technologies with DTN ones. Specifically, a complete protocol stack is presented in which the Saratoga protocol (Wood et al., 2007) is used to exchange data during the federation. This file-oriented protocol was originally designed to download data performing hop-by-hop transfers between satellites. The achieved results of this approach demonstrate the benefits of having federations to download data. However, this solution does not take in consideration a key concept of a federation: the service nature. As previously indicated, a service from a satellite is available during a lapse of time, according to the use of the resources. This temporal availability needs to be reflected in the protocol. Moreover, as the satellite resources are valuable and limited, the amount of resources consumed during an established federation needs to be properly managed.

This chapter presents two protocols to address these current challenges in satellite federations and the IoSat paradigm: (1) the proper notification of available services to the network, and (2) the establishment and management of a satellite federation. The Opportunistic Service Availability Dissemination Protocol (OSADP) has been designed to tackle the former challenge—the service notification—, while the Federation Deployment Control Protocol (FeDeCoP) has been designed to deal with the other challenge—the establishment of the federation. The combination of both protocols conform—what we call—the *federation protocol suite*, which is executed as application software in a satellite.

The OSADP has been conceived to provide a service notification mechanism that suits with the opportunistic nature of the federations. This mechanism could be conceived with a broker that corresponds to a central node which receives all the notifications, and then it forwards to the others. However, as Chapter 2 highlights, centralized approaches are not appropriate for the dynamism of satellite networks. Specifically, if the communications with the central node is lost due to the disruption of the network, the performance would be compromised. Instead, the OSADP promotes a distributed publication of the available services only by the providers. To mitigate the usage of the communications channel, these publications are intelligently forwarded node-by-node, and its propagation is bounded. Thanks to this adjustment, unnecessary transmissions from other satellites are avoided, reducing their resource consumption. The details of how this propagation process is conducted are presented in Section 6.2, as part of the protocol definition. As the OSADP is focused on intelligently spreading the available services to the satellites, other protocols can benefit of this information. For instance, the sealed-bid reverse auction pricing mechanism (Pica and Golkar, 2017) can be executed on top of the OSADP.

The publication of the available services may trigger the establishment of a federation using the FeDeCoP. This protocol provides the necessary means to deploy and maintain a federation between two satellites, following a connection-oriented mechanism. Therefore, three different phases are defined: *negotiation*, *consumption*, and *closure* phases. Despite connection-oriented protocols are typically not suitable for satellite networks, the deployment of a federation entails the use of a context which takes into consideration the service status. Nevertheless, this does not imply that the routing protocol has to be connection-oriented. The negotiation phase is the essential part of the federation itself, in which the amount of resources to be reserved (and thus consumed) is determined between the satellites. The following phases corresponds to the consumption of those resources, and the proper termination of the federation. Furthermore, the protocol takes into consideration the variability of the possible services that a satellite can offer. Therefore, the design takes in consideration the consumption of different resources, such as memory storage or computational capacity.

The design of the federation protocol suite leads to evaluate its performance in a realistic scenario. As presented in Chapter 1, one of the current challenges that satellite missions are dealing is the large amount of data generated from EO payloads. Monolithic satellite missions have normally addressed this situation by using more ground stations. Placing these stations in strategic locations increments the number of contacts with the satellite, and thus its capacity to download data. Nevertheless, this approach is limited to transferring data over a set of ground stations, being difficult to achieve a low latency (i.e. near-real-time services). Additionally, the use of a large ground station network considerably rises the mission cost. Despite these difficulties, the improvement of the download capacity seems still the correct strategy. For that reason, the protocol suite has been evaluated to confirm if it can improve the download capacity of the entire satellite system by deploying federations to leverage from downlink opportunities. Following a similar scenario design as in Chapter 4, the protocol suite is executed in a scenario that includes different EO satellites—which are currently in-orbit—that observe the ice status of the Arctic region.

The main contribution of the present research is 1) a detailed description of the OSADP and its features, 2) the corresponding design of the FeDeCoP, 3) a performance analysis evaluating the benefits of the federation protocol suite, 4) a topology study of applying a mega-constellation as a network backbone, and 5) proofs of the benefits of using mega-constellations for satellite-to-satellite applications. The research presented in this chapter has been published in the peer-reviewed IEEE Access journal (Ruiz-de-Azua et al., 2020a). The details of this publication are accessible in Chapter 9 with the identifier [VI].

The remainder of the chapter is structured as follows. First, Section 6.2 discusses the nature of the services and presents the details about the OSADP. The FeDeCoP design is presented in Section 6.3. The simulation scenario and the satellite model are then presented in Section 6.4. The implementation of this scenario in the simulation engine are detailed in Section 6.5. The analysis and results are exposed in Section 7.7. Finally, Section 6.7 summarizes the chapter, and discusses future research topics.

6.2 Opportunistic Service Availability Dissemination Protocol

In all missions, the satellite has to perform different actions that are scheduled over time (e.g. payload execution, communications with the ground station), which conform the well-known *mission plan*. Different researchers have been working on conceiving mechanisms to design the optimum plan (Araguz et al., 2018; Carrel and Palmer, 2005). All of them have remarked the importance to properly model satellite *resources* (Chien et al., 2012). In this context, a resource is an abstract object that encapsulates a system capacity, which is typically associated to a satellite subsystem or device (e.g. the available power is associated to the batteries). However, other less tangible resources have also been considered crucial for the mission. It is the case of the mission plan defined in (Geyer et al., 2010), which also considers ground station contacts as an important satellite resource. These resources are consumed or released by the different actions that the satellite must perform, known as satellite *tasks*. These tasks change the state of the resources over time. Normally, resources are oversized to always ensure its availability. Therefore, satellites have excess capacity at certain moments of the mission, being able to offer this excess to others, like in satellite federations. In these federations, the satellite offers its remaining resources as *services* is known as the service *provider*. These services can be consumed by other satellites to perform their tasks—known as *customers*. These services can be offered only when they are available. For instance, a satellite can offer the service to download data because it is in contact with a ground station.

Due to the fluctuation of the resources usage, the service is not always available. The corresponding service becomes thus time-dependent, being just available during a lapse of time, known as *service lifetime*. This temporal availability generates valuable opportunities to consume the service by the other satellites. Therefore, a notification of the available services must be designed to catch these temporal opportunities. The proposed OSADP aims at addressing this challenge following a decentralized publish mechanism. With this technique, a satellite provider periodically publishes a service only when it is available. The dissemination of the publication is crucial to correctly notify the satellites, and at the same time not saturating the network with unnecessary messages. However, the main purpose of this publication is to notify as many satellites as possible. Therefore, an intelligent dissemination to satisfy both requirements has been considered in the OSADP. The proposed mechanism is based on the OGM propagation of the BATMAN (Neumann et al., 2008) and BMX6 (Neumann et al., 2012) protocols. In particular, the propagation of a publication is conducted node-by-node, in which each node or satellite processes the packet accordingly. When a node receives a publication that contains information about the availability of a service, it evaluates if it is interested in consuming this service or not. If it is the case, it replies requesting a federation. This request is conducted by executing the FeDeCoP, detailed in Section 6.3. Otherwise, it forwards the publication to its neighbors. Note that this propagation not only notifies the neighbors, but it also indicates that the node accepts and commits to be part of a possible route as an intermediate node. If it is not the case, the node does not propagate the publication. Additionally, a publication is discarded if the node detects that it has already been propagated.

Figure 6.1 presents an example of this publication process in a network composed of different satellites (S_0, S_1, S_2, S_3, S_4 and S_5), in which the satellite S_0 is the provider, and the satellite S_5 the customer. In this example, the nodes propagate the publication of the provider (red arrows), blocking those that are duplicated (red arrows with a dot). The dissemination process is thus optimized by only propagating useful notifications. Additionally, in this case satellite S_4 does not propagate the publication because it does not commit to be an intermediate node in a possible route due to its limited resources. Finally, the customer requests the service after receiving the publication.

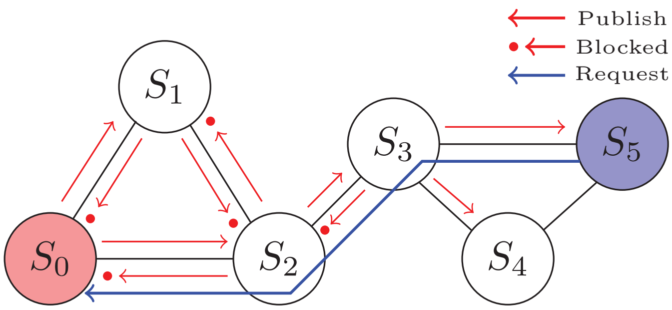
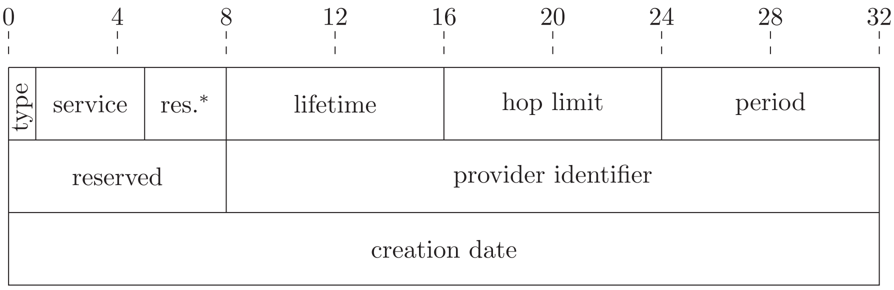


Figure 6.1: Representation of the publication process between the satellite provider (S_0) and the customer (S_5)

All these intermediate decisions are possible because a `publish` packet is propagated. This packet contains information of the available service structured in different fields. Figure 6.2 presents this structure composed of 13 bytes, including 11 bits reserved for future extensions. At the very beginning of the structure, the field *type* identifies that the packet is a publication using a single bit, and the following field indicates the available *service* category with four bits (e.g. data downlink). These two fields, restrained in a byte, are accompanied with three bits reserved for the definitions of future services types. The provider of the service is determined with the *provider identifier*, which enables to distinguish multiple services from the same provider or different providers offering the same kind of service. This identifier is represented with three bytes, allowing it to identify more than 16 million of satellites.

A Service Table (ST) is generated in each node to register in different entries the available services from each provider. A new entry is generated when a new publication is received, storing the corresponding provider and the service. This entry shall be removed when the service is no longer available. Therefore, an expiration date is associated with each entry, computed using the



* reserved

Figure 6.2: Structure of the `publish` packet (bit unit)

estimated *Service Lifetime* (SLT) field of the `publish` structure. Additionally, a service may be not available because the provider is unreachable from the node. Therefore, the corresponding entry is updated with every received publication, periodically transmitted by the provider. Using the *Period* field, a node considers the loss of the provider if no publication is received after three timeouts, then it removes the corresponding entry.

After the entry update, the publication is propagated to the neighbors. As indicated previously, this propagation is performed intelligently, discarding useless publications. In particular, the same `publish` packet can be received from different neighbors. When a node receives a publication of the same provider with the same service, it can detect packet duplicity thanks to the *creation datetime* field. This field corresponds to the timestamp at which the packet has been transmitted by the provider. This time is updated in each periodic transmission. Therefore, if two (or more) received publications have the same date, they are duplicated. Additionally, this parameter enables the detection of old publications that remain in the network. If the received packet has a creation date older than the stored one, it is outdated, and thus discarded.

The propagation of this `publish` packet can be bounded thanks to the *Hop Limit* (HL) field. This parameter has largely been used in the Internet, and it limits the number of hops that a `publish` packet can travel into a network. Therefore, the HL starts with an initial non-negative integer, which is decremented by one before being forwarded by each node. When the HL value becomes zero, the packet is thus discarded. This parameter enables thus to control the diameter of the network in which the publication is disseminated. Note that an additional byte is reserved considering possible extensions of the HL, SLT, and period values.

Figure 6.3 summarizes the previous decision process in each node for a specific service s_k in a flow diagram. Note that an accurate terminology has been used in the diagram, where S corresponds to the state of the OSADP related to service s_k. This state can be PUB for publication state, $PROV$ for provider state, and $CUST$ for customer state in the federation. Moreover, pb corresponds to the `publish` packet of the service current s_k, which is transmitted every T_k period, and at the transmission time t_k^{tx}. The service lifetime STL_k associated to the service s_k enables to compute its expiration date t_k^d. The creation date of pb is represented as t_k^c, being ascribed at the current time t. Additionally, a received packet p_{rx} contains different information always identified with the "rx" subscript, such as the provider identifier P_{rx} or the service type s_{rx}. Finally, the received packet can be a publish one or a request of the service r.

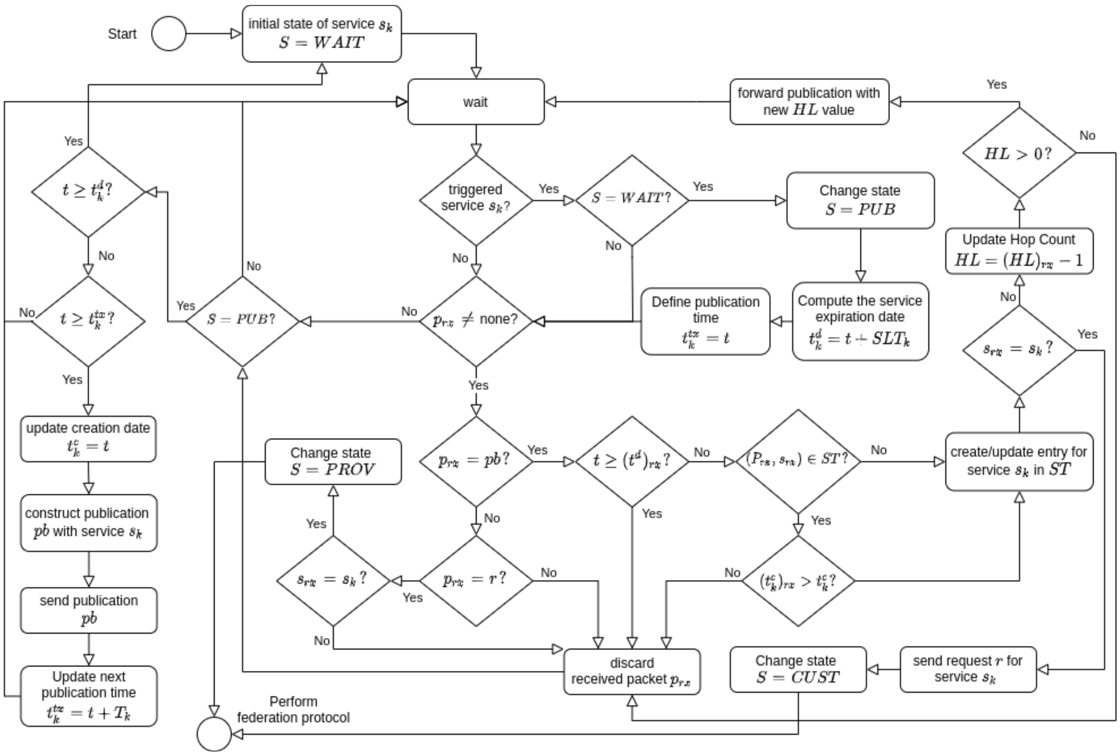


Figure 6.3: Flow diagram of the OSADP executed in a satellite

6.3 Federation Deployment Control Protocol

After the notification of available services, a satellite can decide to establish a federation with the service provider. Therefore, the mechanisms that rule this interaction must be properly designed. The FeDeCoP aims at addressing this challenge by defining a set of directives to establish a federation. This protocol is based on a connection-oriented communication, in which the customer triggers the request of the service (previously published). This communications scheme is necessary in order to keep a stable federation context or session, which is composed of the type and the amount of resources to be consumed, the status of the shared resources, among other parameters. Typically, the connection-oriented protocols are composed of three phases that characterize the entire interaction. In the FeDeCoP case, these phases are known as the *negotiation*, the *consumption*, and the *closure* phases. Figure 6.4 shows the state diagram of the FeDeCoP with the three phases and their transitions according to external events. Note that the *not started* and the *standby* states correspond respectively to the initial boot of the protocol, and the state in which the satellite is waiting for a request or a publication. This last state is thus the default mode of the protocol. The details of each phase are presented in the following sections.

6.3.1 Negotiating the federation

The negotiation phase is triggered when a satellite requests a service. Once a notification of an available service is received by a satellite, this one sends a `request` packet indicating its interest on this service. This packet is transmitted to the service provider—in a unicast manner—, which decides to establish the federation depending on the information included in the packet. This information corresponds to the conditions that the remote satellite aims to have during the

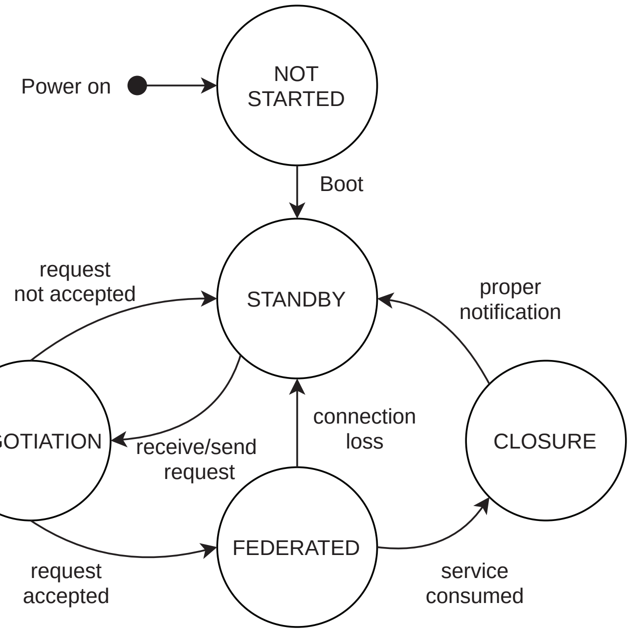


Figure 6.4: States and transitions of the FSS protocol

federation, and it is mapped to a set of fields. Figure 6.5 presents these fields integrated in the `request` packet.

At the very beginning of the structure, three bits identify the packet type, being in this case the `request` one. The *provider identifier* field is included to select the proper satellite if multiple providers are publishing in the network. The following *customer identifier* field determines the satellite which is requesting the service. Note that both identifiers are represented with three bytes, which enables to identify more than 16 million of nodes. This format also follows the one defined in OSADP. After identifying the provider and the candidate customer, the type of *service* is determined with four bits. With this field, it is possible to recognize multiple services from a single provider. Finally, the fields *quantity* and *unit* determines how much of the service is requested. In particular, the three bits of the *unit* field determines the magnitude of the value represented in the *quantity*. With this structure, it is possible to request a specific value of a resource associated to a service, such as bytes or time. For instance, a customer that would like to consume the service to store data, it would request to store 10 GB indicating the value 10 in the *quantity* and the GB unit in the *unit* field (properly coded). Note that six bits are reserved for future extensions related to the number of services or unit levels.

The negotiation process is conducted as follows. Once a satellite receives a publication of an available service, it requests the service indicating the amount of resources that would like to consume. The provider evaluates if this quantity can be satisfied. If this request can be properly satisfied, the provider replies with an `accept` packet, which follows the same structure presented in Figure 6.5 with the corresponding packet type. This packet includes the quantity that will be served. This value can be the requested one, indicating that the provider accepts all the conditions of the customer. Alternatively, the value can be reduced to a magnitude that is considered acceptable for the provider. After receiving the `accept` packet, the candidate customer replies if it accepts the conditions offered by the provider. Otherwise, it discards the packet. When this second `accept` packet is received by the provider, this one considers that the negotiation is finished satisfactorily, and thus the federation established. Note that the packet exchange during the negotiation phase follows the well-known three-way handshake procedure.

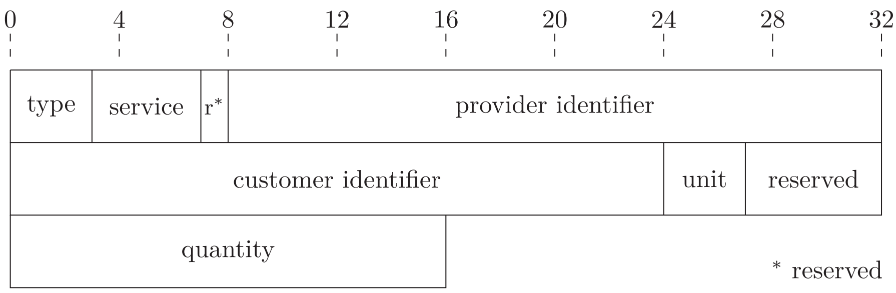


Figure 6.5: `request` and `accept` structures (bit unit)

The negotiation of the federation can be abruptly stopped if one of the `accept` packets is lost. Therefore, a retransmission process is conducted if no reply is received after a configurable timeout. After three retransmissions, the satellite considers that the link is broken. In this case, the negotiation phase is finished unsatisfactorily, and thus the satellite returns to the *standby* mode.

6.3.2 Consuming the service

The completion of the negotiation phase leads to the proper consumption of the service, which corresponds to the consumption phase. The behavior of this phase depends on the type of service which is served. For instance, a federation for downloading data would be based on only exchanging data, while another for computational capacity would exchange a task or software code, and then wait until it is computed. The definition of the procedure to serve resources related to data and time is presented in the following paragraphs.

Those federations based on exchanging data are characterized by constantly transmitting this data from the customer to the provider. This unidirectional data flow is evident in federations for downloading or storing data. In this case, the customer transmits a `data` packet to the provider. This packet is structured as Figure 6.6 shows, having the same initial seven bytes as the `request` packet. After these initial fields, a *length* field of two bytes anticipates the amount of bytes that includes the *data* field. A *checksum* is included at the end of the packet to verify the integrity of the packet itself. This checksum corresponds to a Cyclic Redundancy Check (CRC) of two bytes, known as CRC-16-CCITT (Williams, 1993).

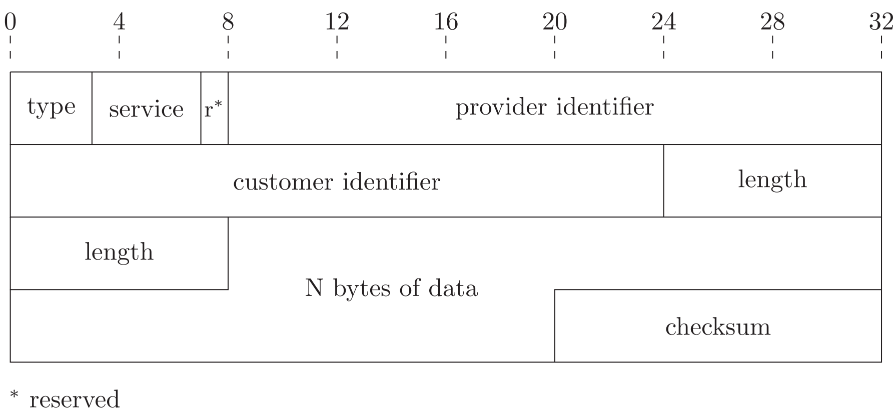


Figure 6.6: `data` structure where N corresponds to the data length (bit unit)

When the provider receives this `data` packet, it performs the pertinent action associated to the service, such as download or store. When this action is correctly achieved, the provider notifies the customer with an `acknowledgement` packet. This packet is composed by only the initial seven bytes of the structure presented in Figure 6.6. Note that this transmission mechanism follows the stop and wait Automatic Repeat-Request (ARQ) procedure (Moeneclaey et al., 1986). Although this mechanism does not optimize the communications capacity of the connection (i.e. bandwidth-delay product), this process ensures that packet-by-packet the customer is aware of the completion of the service. Furthermore, the provider is always verifying that the amount of data received from the customer does not exceed the negotiation one. If it is the case, the provider would start the closure phase, detailed in Section 6.3.3.

The interaction between the satellites changes a little bit for those federations that are based on resources of time, such as computational capacity. In this case, the customer sends a `data` packet in which the *data* field corresponds to the task or the software code that needs to be executed. After the proper acknowledgement from the provider, the customer should wait the corresponding time previously negotiated. In order to evaluate the proper status of the service, a `keep-alive` packet is periodically transmitted by the customer. This mechanism follows the same strategy presented in the TCP, in which the receiver node of the `keep-alive` packet (i.e. the service provider) replies with an `acknowledgement` packet. In this case, however, the packet structure follows the initial seven bytes of the one presented in Figure 6.6. With this mechanisms, the federation can remain alive during all the time that the service is being consumed in the provider (e.g. executing the code). Once this is done, the provider replies with a `data` packet including the results of the execution.

In both cases, the completion of the federated phase triggers the execution of the closure one. This phase can thus be started by the customer indicating that it has been satisfied, or by the provider indicating that the service is no longer available.

6.3.3 Federation closure

The closure phase aims at returning the satellite to its initial state, releasing the services that were used during the federation. As this phase is triggered by one of the nodes of the federation, a three-way handshake is performed to close the federation. The exchange of `close` packets indicates the intentions to terminate the federation. This packet follows a similar structure as the previous ones (see Figure 6.7). The initial fields are the same as the other packets, changing the *type* value according to a `close` packet. Additionally, this packet includes an *acknowledgement* bit if it is required to acknowledge a previous message, such as a `data` packet. Therefore, the satellite that is initiating the closure phase can also notify the proper reception of the last packet with this bit. Seven reserved bits have been included in further information is needed in future extensions.

The packet exchange in the closure phase is performed as follows. A satellite can decide to close the federation because the service is no longer available—i.e. the provider decides—or the customer has already been satisfied—i.e. the customer decides. In both cases, the satellite that wants to terminate the federation transmits a `close` packet highlighting this intention to the other one. When the packet is received, the satellite starts releasing the resources allocated for the federation, and replies with another `close` packet which has the *acknowledgement* flag activated. With this flag, the originator of the federation closure understands that the received `close` packet is at the same time the reply of first one. Finally, it replies with the last `acknowledgement` packet (previously presented). As remarked previously, this process follows a connection termination based on a three-way handshake. This robust mechanism to close the

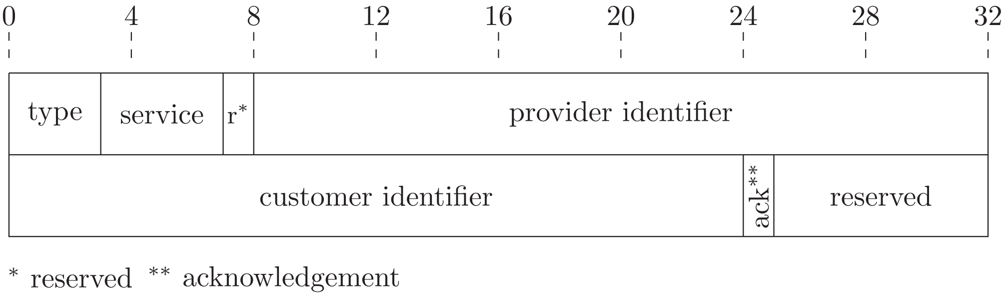


Figure 6.7: `close` packet structure (bit unit)

federation ensures that both satellites can clean the internal context associated to the federation. If the communications are lost during all the three phases that composes the federation, each satellite will perform the process of releasing the resources and cleaning the context after detecting that the connection is broken.

6.4 Mission scenario and satellite model

The evaluation of the federation protocol suite performance is crucial to consider if the protocol is suitable for ISN. For that reason, a scenario that can provide relevant results needs to be conceived. As previously presented, the EO community has a great interest in developing satellite missions to monitor the Arctic region (Alarcón et al., 2018; Camps et al., 2018). Therefore, a scenario based on this type of mission would demonstrate the benefits of this protocol suite for future satellite missions. The scenario has been defined following an approach similar of the one presented in Chapter 4. Specifically, instruments and satellites that typically are used for polar satellite missions have been selected. In this scenario, a satellite is conceived as a platform that integrates an EO payload with other subsystems. Note that this work does not aim at evaluating the hardware impact of using federations, which was already discussed in (Lluch and Golkar, 2015a). Instead, it expects to evaluate its benefits in terms of increased download capacity. Therefore, the satellite model has focused only on three of those subsystems: the satellite-to-ground communication subsystem, the ISL subsystem, and the payload. Details on how these three subsystems have been modeled are presented in the following sections.

6.4.1 Payload traffic model

The payload is the fundamental subsystem in a satellite, as it is all designed to achieve the mission goals. A payload generates data when the satellite ground track is over a target area. Therefore, it can be modeled as a CBR application that periodically generates data over certain target areas. This case study has been focused on monitoring the ice status in the Arctic region. This region is delimited by the Arctic Circle, located at 66º of latitude (approximately). Thus, a satellite over the Arctic will generate data according to the payload that it is carrying. Figure 6.8 presents the Arctic region and the circle that delimits it (red line).

The ONION project conducted a survey and analysis of the different payloads that can be used for this type of mission (Alarcón et al., 2018). The investigation concluded that the set of instruments that shall be used are a SAR at C- or X-band, a SIRAL, a MWR, a MWRI, an AMSR, GNSS-R, and a Radar Altimeter. As in the analysis conducted in Chapter 4, the characteristics of each payload has been retrieved from the OSCAR database (Organization, 2011). Lancheros

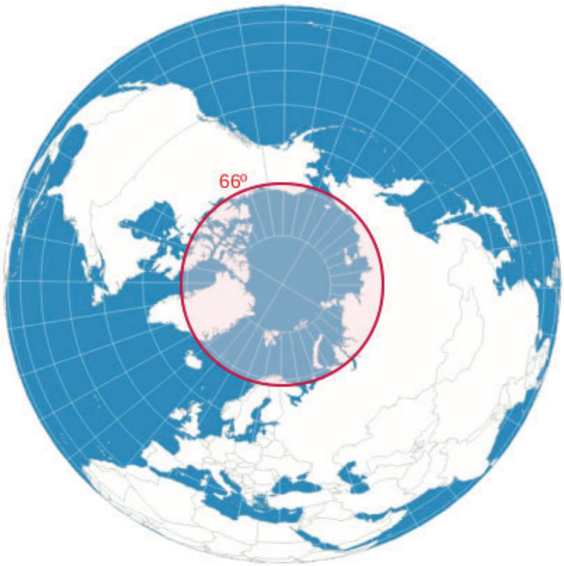


Figure 6.8: Map of the Arctic region, highlighted with red.

et al. (2018) emphasized that the instruments require a minimum spatial resolution of 10 km to perform ice monitoring. Therefore, a low resolution mode of the SAR-C has been considered to provide a spatial resolution of 150 m with a swath of 1500 km. The resulting data rate of the payload is 12 Mbps. A summary of the payload data rate values is presented in Table 6.1.

Table 6.1: Payload model parameters

Payload types	**Data rate**
SAR-C (low resolution)	12.0 Mbps
SAR-X	35.5 Mbps
SIRAL	10.1 Mbps
AMSR	87.4 kbps
Radar Altimeter	35.0 kbps
MWR	10.6 kbps
MWRI	100.0 kbps
GNSS-R	16.8 kbps

6.4.2 Satellite candidates

The selection of the spacecraft that should compose the scenario has been deeply studied. In order to demonstrate the benefits of the federation protocol suite, only satellites that are currently in-orbit have been selected. Moreover, these satellites must have been conceived to observe the ice status in the Arctic region, and thus being able to carry some of the previous payloads. These satellites must follow a polar orbit (i.e. roughly 98º orbit inclination). Considering the heterogeneity of federation approaches, spacecraft from different national agencies and dimensions have been considered. Table 6.2 presents the list of those EO satellites that have been selected using again the OSCAR database (Organization, 2011). Note that this table also presents the orbital elements of each satellite, which have been extracted from the CelesTrak[14] database (Kelso, 1985).

[14]Values retrieved at 24th July 2018.

Table 6.2: Earth Observation Satellite features used to perform the analysis

Identifier	Satellite	Operator	Payload	a[km]	e	i[º]	Ω[º]	ω[º]	M[º]
Ice Observation Satellites									
sat-0	TanDEM-X	DLR	SAR-X	6886	0.0002	97.5	62.0	0.0	271.0
sat-1	TerraSAR-X	DLR	SAR-X	6886	0.0002	97.4	58.0	0.0	46.0
sat-2	SEOSAR/Paz	Hisdesat	SAR-X	6885	0.0002	97.4	66.0	0.0	121.1
sat-3	Sentinel-1A	ESA	SAR-C	7064	0.0001	98.2	57.0	0.0	287.0
sat-4	RadarSat-2	MDA	SAR-C	7169	0.0001	98.6	56.5	0.0	27.3
sat-5	RISAT-1	ISRO	SAR-C	6917	0.0005	97.6	58.1	0.0	218.8
sat-6	GaoFen-3	CNSA	SAR-C	7129	0.00007	98.4	58.5	0.0	303.0
sat-7	Sentinel-3A	ESA	MWR	7181	0.0001	98.6	117.0	0.0	261.7
sat-8	CryoSat-2	ESA	SIRAL	7088	0.0009	92.0	261.2	0.0	147.5
sat-9	Aqua	NASA	AMSR-E	7076	0.00001	98.2	351.3	0.0	293.0
sat-10	FY-3D	CNSA	MWRI	7207	0.0002	98.8	351.6	0.0	348.0
sat-11	HY-2A	CNSA	Radar Alt	7335	0.00003	99.3	60.0	0.0	293.2
sat-12	SARAL	CNES/ISRO	Radar Alt	7171	0.0002	98.5	237.3	0.0	213.8

Chapter 4 has discussed the impact of the network disruption caused by the use of a reduced number of satellites. This typical phenomenon of satellite networks provokes that a connection between two remote satellites is not always feasible. Therefore, a mega-constellation has been included as a network backbone to mitigate this phenomenon. Using this massive satellite system, the connectivity of the network is improved. A detailed analysis of different mega-constellations proposals is presented in (Portillo et al., 2019). Reviewing the performance of each architecture, the Telesat approach has been selected because it provides the highest connectivity with less number of satellites. Details about the constellation architecture are presented in Table 6.3.

Table 6.3: Telesat Mega-constellation characteristics

Plane type	Planes	a	i	e	Satellite / plane
Type 1	6	7371 km	99.5º	0.0	12
Type 2	6	7571 km	37.4º	0.0	9

6.4.3 Download traffic model

The ground segment provides the communications interface with the satellite using a network of ground stations. This infrastructure enables the transmission of telecommands, and the reception of data from the satellites. This interface is available only when the satellite is in line-of-sight of the ground station (i.e. a *satellite pass*). The duration of this pass drives the amount of data that can be downloaded, and thus the entire downloaded traffic of the satellite system. Therefore, the ground segment must be modeled properly. A proper configuration of the ground segment that promotes satellite passes—and thus the possibility to publish downlink opportunities—would facilitate this analysis. For this reason, current operative ground stations have been selected considering their location. Table 6.4 details the list of the selected ground stations, and Figure 6.9 places them on the surface of the Earth, showing a large ground segment around the globe.

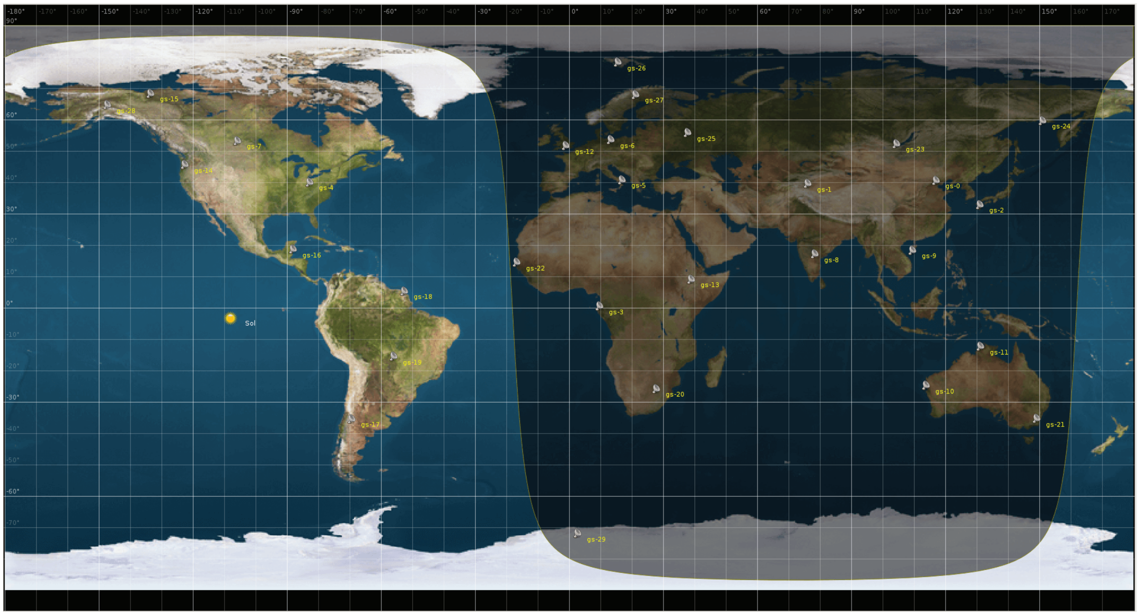


Figure 6.9: Location of the ground stations on the Earth

An area around the ground station represents the region over which a satellite pass is feasible. Thus, when a satellite ground track passes over this ground station area, the satellite downloads data following a CBR transmission. The dimensions of this downlink region directly depend on the minimum elevation angle (α) above the horizon at which the ground station can communicate with the satellite. The minimum value of this parameter is achieved at the maximum distance (r_{max}) that satisfies the link budget between the satellite and the ground station. Figure 6.10 presents a plot illustrating these two parameters, and the area over which the satellite downloads data (gray surface).

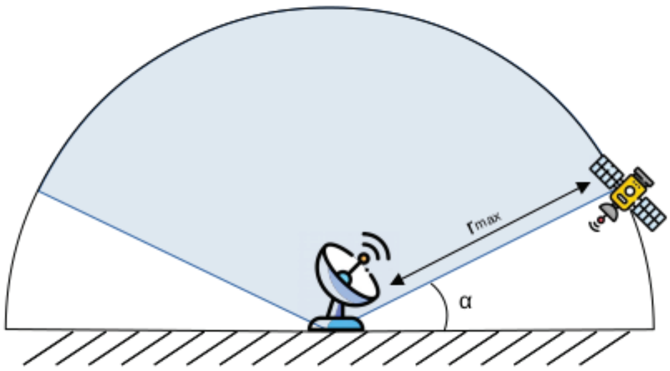


Figure 6.10: Ground station model with the corresponding communication area (blue surface)

The definition of the link budget is determined by the transmission and reception features of each node (i.e. the satellite and the ground station). Transceivers working at multiple frequencies have been used in the past missions. From all of them, the S-band transceiver has been founded as a primary solution for satellite platforms. Although each COTS transceiver has its own characteristics, all of them can provide at least 1 Mbps of data rate (Aps, 2018; *TerraSAR-X Mission*; Tyvak Nano-Satellite Systems, 2019; *Landsat Missions - Landsat 8*). A link budget

Table 6.4: Ground stations locations that conforms the ground segment in the simulation, and the World Geodetic System 84 (WGS84) as the Earth model.

Ground Station	Country	Latitude	Longitude
Beijing	China	40.5º N	116.9º E
KaShi	China	39.5º N	76.0º E
Kumamoto	Japan	32.8º N	130.9º E
Libreville	Gabon	0.4º N	9.6º E
Ohio	USA	40.0º N	83.0º W
Matera	Italy	40.7º N	16.7º E
Neustrelitz	Germany	53.4º N	13.1º E
Prince Albert	Canada	53.2º N	105.9º W
Shadnagar	India	17.0º N	78.2º E
SanYa	China	18.3º N	109.3º E
Carnavaron	Australia	24.9º S	113.7º E
Shoal Bay	Australia	12.4º S	131.0º E
Chilton	United Kingdom	51.6º N	1.3º W
Addis Ababa	Ethiopia	9.0º N	38.8º E
Oregon	USA	45.5º N	122.7º W
Inuvik	Canada	68.3º N	133.6º W
Chetumal	Mexico	18.5º N	88.3º W
Malargüe	Argentina	35.5º S	69.6º W
Kourou	French Guyana	5.2º N	52.8º W
Cuidaba	Brazil	15.6º S	56.1º W
Hartebeesthoek	South Africa	25.9º S	27.7º E
Tidbinbilla	Australia	35.4º S	149.0º E
M’Bour	Senegal	14.4º N	16.9º W
Irkutsk	Russia	52.2º N	104.3º E
Magadan	Russia	59.6º N	150.8º E
Moscow	Russia	55.7º N	37.6º E
Svalbard	Norwegia	78.2º N	15.4º E
Kiruna	Svergie	67.9º N	21.0º E
Alaska	USA	64.6º N	147.5º W
Troll	Antarctica	72.0º S	2.5º E

analysis with a S-band transceiver is presented in (Barbarić et al., 2018). Applying specific coding and modulation techniques and considering atmospheric effects, the achieved maximum distance and data rate are 1392 km and 3.4 Mbps respectively. Considering that the minimum COTS transceiver provides a smaller data rate, the maximum distance of this analysis can be incremented. Therefore, following the formulation of (Barbarić et al., 2018), a common S-band transceiver for all the satellites is selected, which transmits at 1 Mbps and has a maximum distance of 2500 km. Note that although Telesat satellites work at Ka-band (Portillo et al., 2019), it has been preferred to "virtually" equip them with the S-band transceiver to unify the downlink capabilities. Considering this maximum distance value, Table 6.5 presents the corresponding minimum elevation angles of each satellite, selected in Section 6.4.2.

Table 6.5: Minimum elevation angle of each satellite

Satellite	Altitude [km]	Maximum distance [km]	α [°]*
TanDEM-X	515	2500	11.89
TerraSAR-X	515		11.89
SEOSAR/Paz	514		11.86
Sentinell-1A	693		16.09
RadarSat-2	798		18.61
RISAT-1	546		12.62
GaoFen-3	758		17.65
Sentinel-3A	810		18.91
CryoSat-2	717		16.67
Aqua	705		16.38
FY-3D	836		19.54
HY-2A	964		22.68
SARAL	800		18.66
Telesat Type-1	1000		23.58
Telesat Type-2	1200		28.69

* Minimum elevation angle

The downlink regions for each satellite are computed with these angle thresholds. Figure 6.11 presents these regions for the SEOSAR/Paz satellite as an example. For those satellites that are not placed on these downlink regions, they can still transmit data using the ISN or become part as intermediate nodes.

6.4.4 Inter-Satellite Link model

Satellite platforms have inter-satellite communication capabilities in this scenario. This technology has already been deployed in large satellite missions (e.g. Iridium constellation), and currently is being demonstrated for smaller platforms, such as CubeSats. We encourage the reader to review Section 1.3.10 (Chapter 1) for further investigations about the deployment of missions that integrates ISL devices. An ISL subsystem model can thus be characterized mainly with three parameters: the maximum communication distance d_{max}[15], the transmission data rate achieved at this distance Rb, and the radiation pattern or direction. The ideal ISL subsystem would provide large data rates, large communication ranges, and transmitting through

[15] Note that r_{max} has been used for satellite-to-ground communications, and d_{max} for satellite-to-satellite communications.

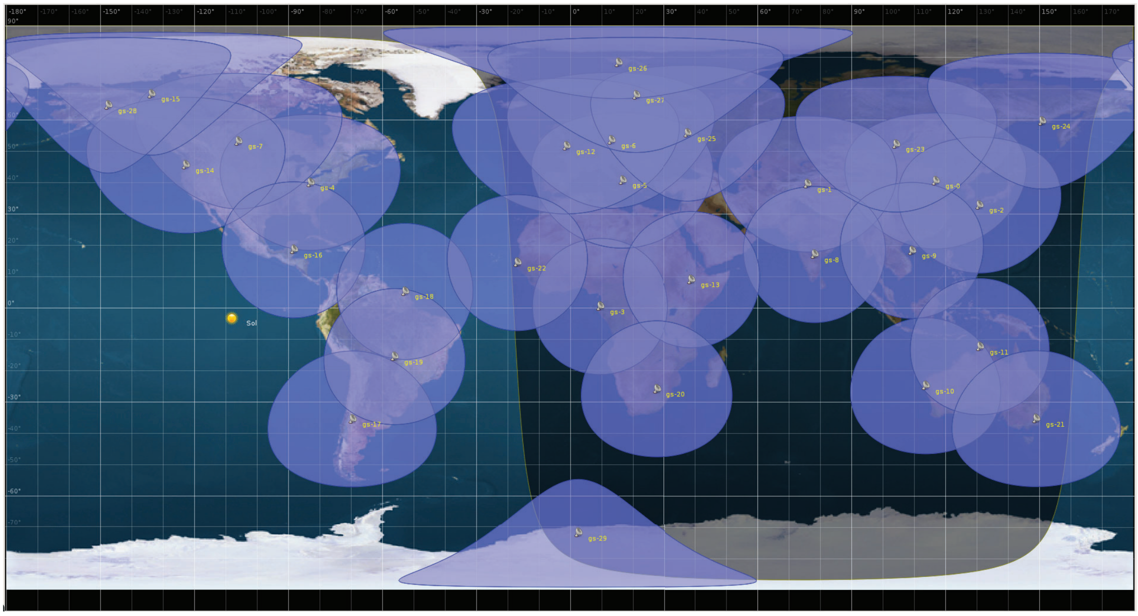


Figure 6.11: Downlink regions for SEOSAR/Paz satellite

an omnidirectional pattern. With these characteristics, it would be possible to have a system that provides fast transmission and high connectivity. Unfortunately, these characteristics cannot be achieved using a single subsystem. Nevertheless, the combination of multiple ones can address this challenge. This approach is considered in this scenario. In particular, different ISL subsystems are combined to create two communications planes: the control, and the data planes. Software Defined Networks (SDN) and 5G networks have been working on this concept some years ago (Yazıcı et al., 2014) highlighting the benefits of using these two planes. The combination of decoupled and independent planes allows to isolate network managing tasks and traffic, which optimizes the data transmission on the other plane. Note also that in (Lluch and Golkar, 2015a) FSS commands and scientific data are differentiated, implicitly working with these two planes.

The control plane provides a communications channel for those packets that manage the network. Typically, these kinds of packets are composed of small byte structures, and they do not require high transmission data rates. Additionally, a subsystem that provides a large communications distance is necessary to increase the connectivity. In this direction, transceivers that work at Very High Frequency (VHF) and UHF would suffer less attenuation due to the signal propagation. Although the ITU-R sector does not consider ISL frequency bands below 22 GHz, some satellite missions have already used these previous bands for this purpose (Radhakrishnan et al., 2016). The continued interest in this field should encourage the discussion of the corresponding coordination in upcoming WRC.

Taking into consideration these requirements, the ISL for the control plane can be implemented with a monopole antenna, and modulation techniques applied in Low-Power Wide-Area Networks (LPWAN) (Raza et al., 2017). A large family of technologies have been developed for LPWAN, standing out the Narrow-Band Internet of Things (NB-IoT) (Ratasuk et al., 2016). The NB-IoT is conceived to provide considerable data rate with low power consumption, and for indoor communications with high density nodes. Its standardization—performed by the 3GPP (Generation Partnership Project, 2020)—has expanded its use in current IoT ground networks. However, this technology has not been yet evaluated in space missions. On the other hands, the Long Range (LoRa) technology (Vangelista et al., 2015) has been evaluated and studied for CubeSats missions (Caleb, 2019a; Doroshkin et al., 2019; Fernandez et al., 2020; Systems, 2020). LoRa works at unlicensed UHF band, and it is designed to provide different low data profiles at long ranges. It seems thus suitable for the ISL subsystem responsible of the control plane.

The data plane provides a communications channel to transmit the generated data from the payloads. This data flow requires a communication link with a high data rate. In order to satisfy this requirement, an ISL subsystem based on a directive antenna would provide enough power at large distances, by specifically pointing to the neighbor satellite. In this case, the coordinated ISL bands of ITU-R suit perfectly for this purpose, such as the Ka-band. This band has started to be used in spacecraft configurations, such as the ones of the Starlink mega-constellation (Exploration, 2018). These spacecraft use a four phased array antenna which generate different beams of 2.5º beamwidth (Sayin et al., 2019).

The combination of both planes provides a satellite-to-satellite communications interface with high connectivity, large range, and great data rate. In particular, a satellite uses the control plane to manage the network and also to detect the pointing direction of the data plane ISL subsystem. This would be feasible if the satellite is able to predict the relative position of its neighbors by means of predictive algorithms like the one presented in Chapter 5. Additionally, authors in (Lluch and Golkar, 2015a) remarks that current attitude actuators are accurate and fast enough to achieve this system. The link budgets of both planes are presented in Tables 6.7 and 6.6. These link budgets have been performed considering the previous possible configurations, and for two cases: one in which both satellites are close (1500 km), and another on where they are far away (3000 km). The link budgets of the control plane have been based on the LoRa SX-1261X chip characteristics (Sensing and Division, 2015). However, due to private information, the Forward Correction Code (FEC) gain has not been considered in the link budgets. The link budgets related to the data planes have been based on the downlink and uplink transceiver characteristics of Telesat satellite, provided in (Portillo et al., 2019). In particular, the EIRP, the modulation, and other aspects have been considered to compute the link budgets. These budgets remark that satellite-to-satellite communications can be established at 1500 km and at 3000 km, modifying properly the modulation and bandwidth. Specifically, the ISL subsystem for the control plane offers 5.5 kbps at 1500 km, and 1.8 kbps at 3000 km, while it offers for the data plane 104.1 Mbps at 1500 km and 1.1 Mbps at 3000 km. Taking into consideration these values, a type of ISL subsystem with a maximum distance of 1500 km and different rates of 1 Mbps, 10 Mbps, and 100 Mbps has been considered for this study. Moreover, another subsystem providing 1 Mbps at 1500 km, 2500 km, and 3000 km has also been considered.

6.5 Protocol stack definition

The previous scenario has been executed using the simulation engine presented in Chapter 3. A simulation structure has been conceived according to the different models detailed in the previous sections. Figure 6.12 presents the Unified Modeling Language (UML) diagram of the different classes that represent a satellite. The NS-3 simulator defines a `Node` as an entity that has a set of `Applications` and protocols. In this case, the main application is the `SatelliteApplication`. This one manages the operations of the satellite, deciding when an action has to be executed. A `Scheduler` provides support notifying when the satellite ground track is over a specific area. Therefore, thanks to these notifications, the satellite generates data with the `Payload` application or downloads data with the `TelemetryAndTelecommand` one. The exchange of data between those two applications is performed using a `DataBuffer`.

Table 6.6: Link budgets for close and far cases of the ISL for the data plane

Parameter	Close case	Far case	Unit
Frequency **	23.3	23.3	GHz
Bandwidth	75	0.8	MHz
EIRP *	36	36	dBW
Modulation *	16APSK	16APSK	
Coding Rate *	28/45	28/45	
Spectral efficiency *	2.23	2.23	bps/Hz
Path Distance	1500	3000	km
Free Space Loss	183.3	189.3	dB
Rx. Antenna Gain *	31.8	31.8	dB
Rx. System Temperature *	285.3	285.3	K
Rx. Power	-115.5	-121.5	dBW
Rx. SNR	9.8	23.5	dB
Req. Eb/N0 *	4.6	4.6	dB
Req. SNR	6.0	6.0	dB
Link Margin	3.8	17.5	dB
Data Rate	104.1	1.1	Mbps

* Retrieved from Portillo et al., 2019; ** Inter-Satellite band coordinated by ITU

Table 6.7: Link budgets for close and far cases of the ISL for the control plane

Parameter	Close case	Far case	Unit
Frequency *	868	868	MHz
Bandwidth *	125	125	kHz
Transmitted Power	30	30	dBm
Tx. Antenna Gain	1	1	dB
EIRP	31	31	dBm
Spreading Factor *	7	9	
Coding Rate *	4/5	4/5	bps/Hz
Path Distance	1500	3000	km
Free Space Loss	154.7	160.75	dB
Rx. Antenna Gain	1	1	dB
Rx. Power	-122.7	-128.8	dB
Sensitivity *	-123	-129	dB
Margin **	0.3	0.3	dB
Data Rate *	5.5	1.8	kbps

* Retrieved from LoRa SX-1261X Sensing and Division, 2015; ** FEC gain is not considered

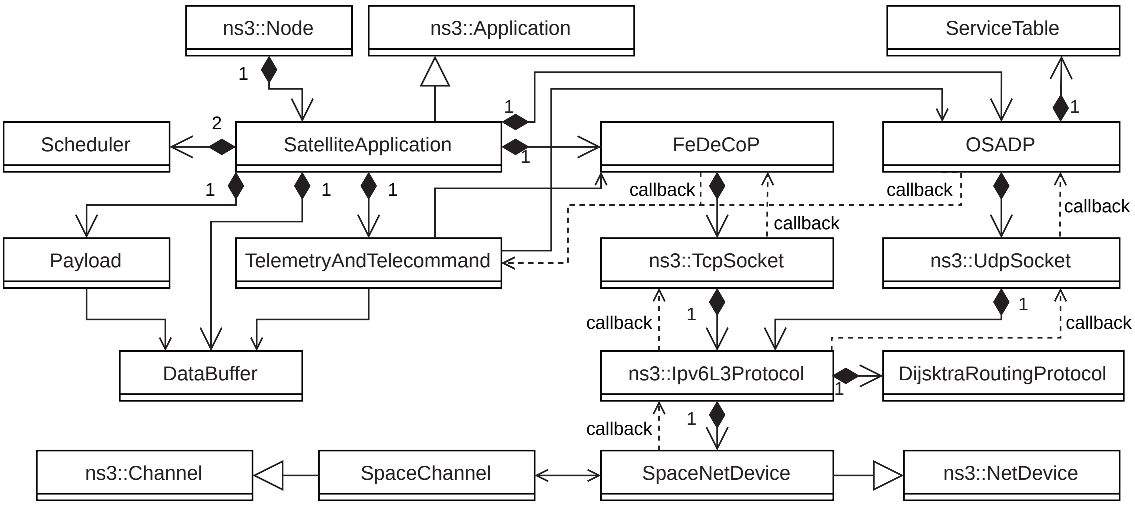


Figure 6.12: Simplified UML diagram of a satellite in the simulation

Note that these applications are implemented according to the models presented in Sections 6.4.3 and 6.4.1.

The OSADP is implemented in the OSADP class, with its corresponding ServiceTable. This protocol uses a UDP socket because the reliability is not required during the publication of the service. Additionally, the FeDeCoP implements the proposed protocol to establish federations, and it uses a TCP socket to ensure that the data has been correctly delivered to the destination. Both types of sockets directly interact with the IPv6, which has a custom routing protocol based on the Dijkstra algorithm (Dijkstra, 1959). This analysis does not aim at evaluating the consequences of the selected routing protocol, and thus an ideal one has been considered. This protocol, however, may be evaluated with another routing protocol in future research, which may suit the requirements presented in Chapter 2. Note that in this case, the provider identifier has been defined with the last three bytes of the IPv6 address.

The TelemetryandTelecommand not only manages the data download, it also triggers the publication in the OSADP class when the service is available (see previous Figure 6.3). A callback function is used when a publication is received by the OSADP class. In this last case, the TelemetryandTelecommand class deploys the federation using the FeDeCoP, which notifies about its status using different callback functions. The SpaceNetDevice and the SpaceChannel classes have been presented in Chapter 3, and they implement the discontinuity of the ISL due to satellite motions.

This class structure is replicated for each satellite in the simulation scenario. Note that the Telesat satellites do not have an EO payload, and neither the corresponding Payload class. Also the execution of the OSADP and FeDeCoP classes is configurable, enabling to select specific satellites that participate in the service publication and federations. With these features, the scenario has been simulated retrieving different metrics about the protocols' performance.

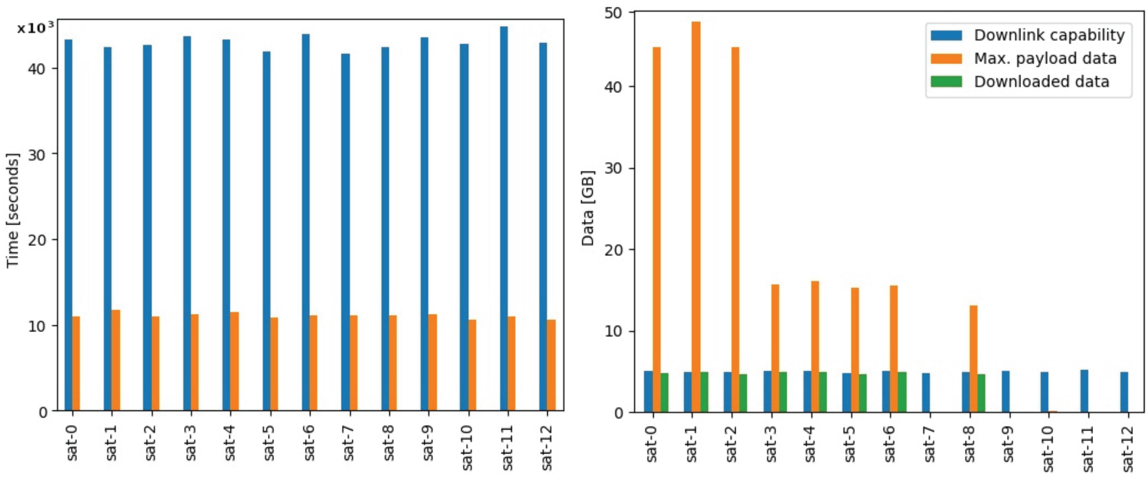
6.6 Results and discussion

The protocol suite performance is evaluated with the execution of the previous scenario. This scenario is executed during a simulated day—86400 seconds—, and different parameters related to inter-satellite communications are configured. Additionally, satellites in this scenario that execute the OSADP transmit publications every 5 seconds, with 17 as HL. These parameters have been defined according to the dynamics of the networks, presented in Section 6.6.2. The present analysis helps to understand how the federation protocol suite improves the download capacity of the original EO satellites. For this reason, Section 6.6.1 discusses the initial capacity of only those satellites, highlighting an interesting progress margin. The stability of the topology is greatly impacted by the number of satellites. Therefore, a discussion of how the download capacity changes when the mega-constellation is used as a network backbone is addressed in Section 6.6.2. After demonstrating the benefits of using this backbone, Section 6.6.3 goes deeper in evaluating the performance considering that satellites of the mega-constellation also publish the availability of their services.

6.6.1 Scenario study

As it has been presented previously, EO satellites generate data when they are over the Arctic region, and download them when passing over a ground station. A data budget between the generated and downloaded data determines the storage capacity of the satellite. This budget is driven by the amount of time that each satellite passes over the target areas. Figure 7.19a presents the accumulated time of each EO satellite over those areas after one simulated day. All of them pass around 12 hours per day over a ground station (blue bar), and have thus the opportunity to download up to 5.29 GB approximately. Note that this amount of time is very large because a large ground segment has been considered to promote downlink opportunities. Additionally, each satellite passes 3 hours per day (roughly) over the Arctic region, generating data according to its EO instrument.

Figure 7.19c shows the difference between time and data budgets. Satellites that carry payloads with greater data rate (i.e. from sat-0 to sat-6, and sat-8) generate more data (orange bar) than the one that they can download (blue bar). Although they are using all the downlink time to download its own data (green bar), this is not enough to avoid overloading their storage system. This result highlights that these SAR-based payloads cannot be executed over the entire Arctic zone, as it is the case in traditional monolithic satellite missions. Note that the satellites that



(a) Time over downlink (blue) and Arctic regions (orange)

(b) Downloaded (green) and generated data (orange) with respect the download capacity (blue)

Figure 6.13: Time and data budgets of each EO satellite without using satellite-to-satellite communications

carry these payloads are called "memory-overloaded EO satellites" during the rest of the results discussion. However, some satellites do not suffer from this problem (i.e. sat-7, and from sat-9 to sat-12), because they can download all the generated data. Additionally, these satellites have surplus downlink time that could be shared. These ones can thus offer this remaining capacity as a service that the memory-overloaded EO satellites could benefit.

The opportunity to share this unused time is achieved by enabling the satellite-to-satellite communications, and using the federation protocol suite. Figure 6.14 presents a growth of the downloaded data when this interaction between satellites is enabled. In particular, this figure compares the system performance with respect to different ISL subsystems. These subsystems provide 1 Mbps of data rate at communications ranges (d) of 1500 km, 2500 km, and 3000 km. Clearly, the use of those systems with larger range facilitates the propagation of the service publication, proliferating the federations. With these federations, the satellites download 5.24 GB per day more of the memory-overloaded EO satellites (in the 3000 km case).

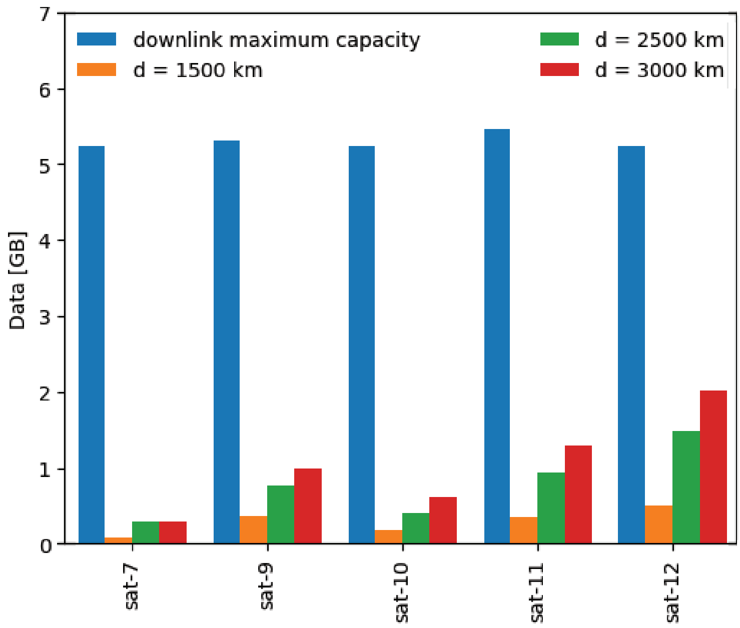


Figure 6.14: Data budgets of EO satellites that provide downlink service with ISL subsystems of 1 Mbps data rate and different maximum ranges d.

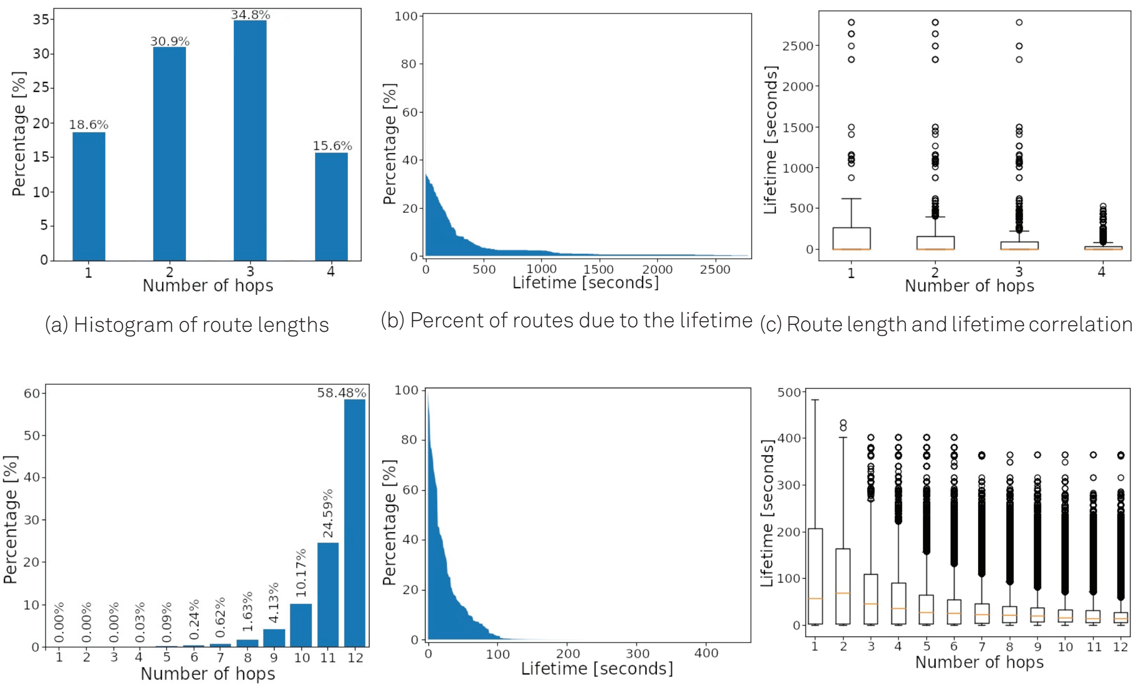
These results demonstrate that the proposed protocol suite helps to establish satellite federations. Satellite resources are properly leveraged with these federations, and more data is downloaded. However, the previous figure also remarks that although using a range of 3000 km there is still unused downlink time. The network disruption is the cause of this behavior, which provokes that not all the customers can communicate with the service providers. As in previous analysis, this situation is mainly caused by the network disruption which isolates parts of the network. As previously suggested, the integration of predictive algorithms that could anticipate this situation would enhance the performance. Despite these possible solutions, the use of a network backbone can also mitigate this phenomenon thanks to promote the connectivity with more satellites. This last approach has been used in the following simulations. This work does not aim at evaluating the consequences of network disruption in the proposed service notification, which may be discussed in future investigations. The remaining download capacity of the EO satellites may be leveraged by using a satellite backbone.

6.6.2 Topology analysis

The integration of the Telesat mega-constellation as a network backbone increases considerably the number of satellites that participates in the network. As a result, the possibility to have multiple routes between a service provider and a customer becomes more probable. These routes are extended with additional intermediate satellites. Note that larger routes are susceptible to suffer more changes, making the entire topology more dynamic. Therefore, it is crucial to understand how the inclusion of the mega-constellation impacts on the topology dynamism.

Figure 6.15 presents the behavior of the topology before and after integrating the mega-constellation with the EO satellites (top and bottom figures respectively). Figures 6.15a and 6.15d present a histogram of the routes that are feasible according to the number of satellites that compose them (i.e. the route length). Figures 6.15b and 6.15e indicate the probability that a route is stable during (at least) a specific lapse of time, well-known as route lifetime. Finally, Figures 6.15c and 6.15f correlate this lifetime information with respect to the route length, represented in box plots[16]. Note that these topology metrics are the same ones used in Chapter 4.

Figures 6.15a, 6.15b, and 6.15c present the topology behavior when only EO satellites compose the satellite system (i.e. the one analyzed in Section 6.6.1). In this scenario, the ISL subsystem has a maximum communications range of 1500 km. Figure 6.15a shows that only routes with less than four hops are feasible, being the most probable the routes with three hops (i.e. four satellites). This performance is related to the reduced number of satellites, just 12 spacecraft in this case. Additionally, some of these routes are quite stable reaching lifetime values up to 2500 seconds, as Figure 6.15b shows. Nevertheless, the majority have a short lifetime. This result is



(a) Histogram of route lengths (b) Percent of routes due to the lifetime (c) Route length and lifetime correlation

(d) Histogram of route lengths (e) Percent of routes due to the lifetime (f) Route length and lifetime correlation

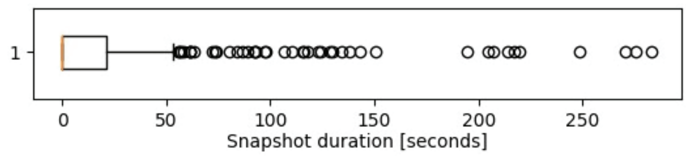
Figure 6.15: Topology behavior without the Telesat mega-constellation (a, b, and c), and with it (d, e, and f)

[16] A box plot represents a distribution function as a box delimited by the third quartile and the first quartile. Additionally, the median is represented as a line in this box.

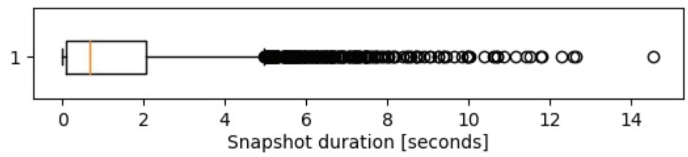
also presented in Figure 6.15c, where all the routes have commonly a short lifetime with some exceptional cases (represented in dots). This topology is thus driven by the network disruption, which produces short and unstable routes. As a result, producer and customer satellites cannot establish federations to download data.

This behavior changes when the mega-constellation is considered as part of the satellite system. Figure 6.15d shows that using the same ISL subsystem (i.e. 1500 km of range) the routes are more variable, but larger (up to 12 hops). Thanks to the increased number of satellites, the largest routes are more probable. This implies that the communications range is extended, being able to reach further satellites. Figure 6.15e highlights that the stability of a route has been homogenized with respect to the previous case, reducing the abrupt differences in some routes. However, this is still a high dynamic scenario, as Figure 6.15f remarks, because the most probable routes are the ones that have short lifetime. In summary, the use of the mega-constellation helps to expand the network diameter with larger routes, although they are more volatile.

The duration of a stable snapshot is also another metric that helps to understand how dynamic a network is. Figure 6.16 shows the statistics of this snapshot duration in each of the previous cases (i.e. without and with mega-constellation). As a result, the use of the mega-constellation makes the topology more dynamic, generating more snapshots than in the case with only EO. Although the established federations could be broken due to the disappearance of a route, the possibility to expand the network becomes promising to establish federations.



(a) Boxplot (top) and samples (bottom) with EO satellites only



(b) Boxplot (top) and samples (bottom) with mega-constellation

Figure 6.16: Duration between snapshots without Telesat mega-constellation (top) and with it (bottom)

It is important, however, to clarify if this improvement of the connectivity prevails over the variability of the network. For that reason, this scenario in which the EO satellites use the mega-constellation to forward publications is evaluated. The corresponding results are compared to the scenario in which only the EO satellite composes the satellite system, presented in Section 6.6.1. Figure 6.17 shows the downloaded data of those EO satellites that are not memory-overloaded, comparing both scenarios. In particular, the EO satellites that can provide the service download more data using the mega-constellation as backbone (green bar) than in the original case (orange bar). This remarks that although the network changes quickly, the possibility of having more and larger routes helps to download more data. Consequently, the use of the mega-constellation is beneficial. However, it remains still unused capacity (blue bar), which could be leveraged.

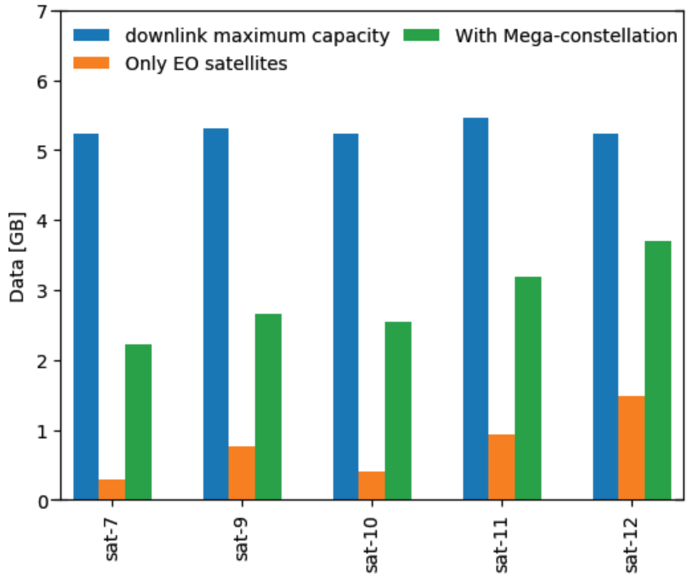


Figure 6.17: Data budgets of EO satellites that provide downlink service with ISL of $Rb = 1$ Mbps and $d_{max} =$ 2500 km, with and without the mega-constellation.

These results pose the question on what drives the use of this surplus capacity. For that purpose, the different actions that a satellite performs during a ground station pass are evaluated. During a pass, a satellite can perform three different actions: download its own data, publish the opportunity to download, or provide this downlink service to download data from others (i.e. it is federated). Figure 6.18 presents these three concepts in the form of time percent. These percent values correspond to the average value computed with the time percents of each EO satellite. The study is thus focused on analyzing these averaged values which represent the behavior of an "average EO satellite". Specifically, how this average EO satellite uses its ground station pass depending on its ISL subsystem features is discussed. As indicated previously, the maximum range determines the expansion of the network, and thus the possibility to establish federations. For that reason, the time percent is plotted with respect to this feature.

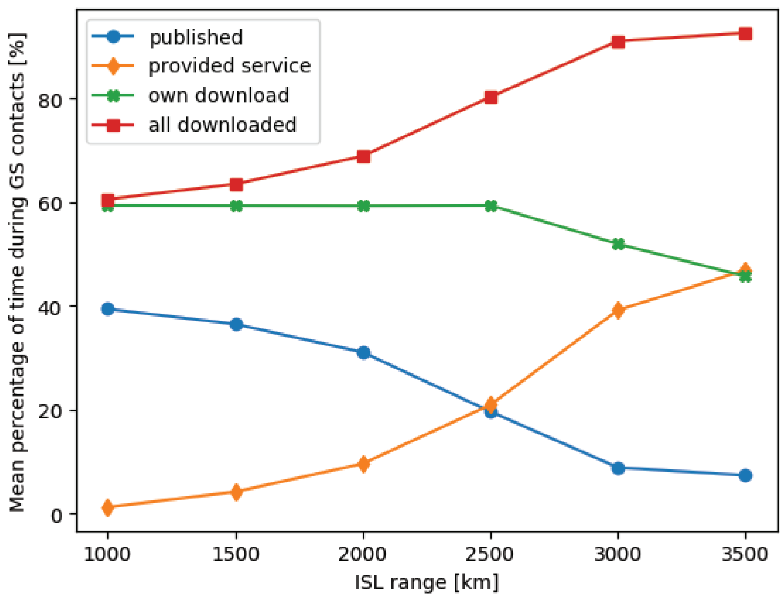


Figure 6.18: Average time percentages related to the downlink pass actions of all the EO satellite

Regarding the time used publishing downlink opportunities (dot-blue line in Figure 6.18), it decrements while the range increases. Increasing this range facilitates the establishment of federations, as diamond-orange line remarks with this growth of the time providing the service. Note that the EO satellites keep using the same time of a ground station pass to download its own data (cross-green line), until the range is 2500 km. At this moment, less time is used for this purpose, because the satellites have already downloaded it using federations. Although this decreases, the total time used downloading data (square-red line), its own and from the others, increases with the range. This result remarks that the ISL range allows leveraging the pass time by reducing the time during which the satellite is publishing, and thus downloading more data.

Considering that the saturated EO satellites should download data over federations, their activity would also drive the amounts of data that the entire system downloads. Specifically, if those satellites remain federated during all the time, this would signify that all data, susceptible to be downloaded, is indeed transferred. Therefore, the percentage of time during which an average satellite of those saturated does not transfer data is considered as a resulting metrics. The optimization of those satellites is thus characterized with these metrics. Figure 6.19 presents the evolution of this time percentage (dot-blue line) with respect to different ISL ranges. Note that this average satellite becomes more active when the range increases, due to the different federations that are established. In particular, this satellite passes from 12 hours per day (i.e. around 50% of a day) to 7 hours of inactivity (i.e. 29% of a day). Although in the best case (range 3500 km) the activity is considerably incremented, this average satellite still has some time that could use transferring data with federations. This result remarks that the download capacity of unsaturated EO satellites (i.e. sat-7 and form sat-9 to sat-12) is not enough to download all the generated data from the saturated ones. Therefore, this work goes deeper evaluating how this capacity increases while the mega-constellation satellites participate sharing their own downlink opportunities.

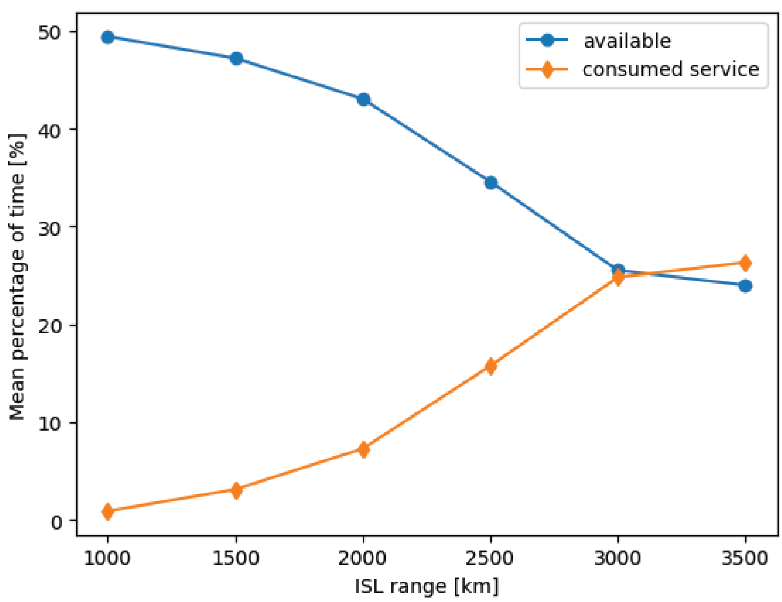


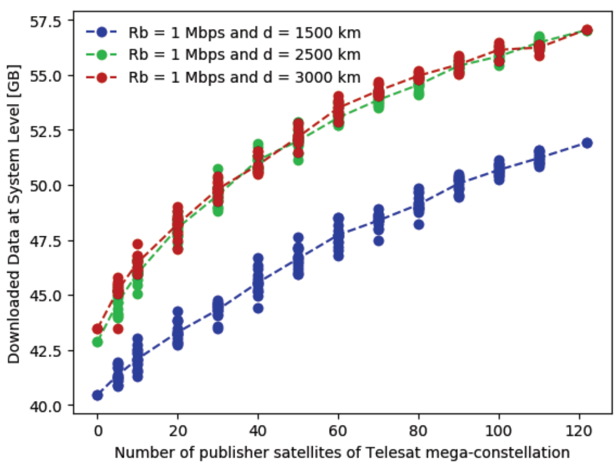
Figure 6.19: Average time percentage of inactivity (blue) and federated (orange) of the memory-overloaded EO satellite

6.6.3 Participation analysis

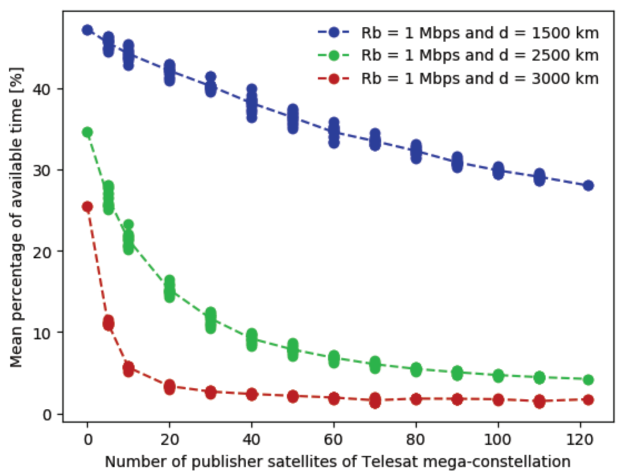
The participation of mega-constellation satellites by sharing downlink opportunities would provide more download capacity to the entire satellite system. With this approach, those saturated EO satellites could leverage the unused time transferring data with new federations. However, it is important to clarify how many of those mega-constellation satellites are needed to reach the maximum activity of the saturated ones. Thus, this section presents a study of the download capacity evolution with respect to the number of satellites from the mega-constellation sharing the service. Different groups of those satellites have been analyzed in different simulations, selected randomly of the entire constellation. Therefore, multiple simulations per group have been performed, retrieving at the end an average behavior (represented in all the figures as a slashed line). Note that those satellites that are not included as publishers of the service, they take part of the network backbone to keep the same connectivity as before.

Figure 6.20a plots the bytes downloaded per day of the entire system depending on the number of mega-constellation satellites that publish the service. As remarked in the previous results, the communications range of the ISL subsystem is one of the parameters that drives the data download. Therefore, this metric is compared to ranges of 1500 km (blue line), 2500 km (green line), and 3000 km (red line). In those cases, the data rate of the ISL subsystem remains equal to 1 Mbps.

If no satellite from the mega-constellation publishes the service, they conform the network backbone as in Section 6.6.2. Consequently, the resulting downloaded data is the same as in Figure 6.17, growing with larger ranges. While the more publishers start to participate, the downloaded data increases following a logarithmic growth in the three cases. This behavior is correlated with Figure 6.20b which presents the time percentage per day of the average saturated satellite that is not being used to transfer data (as in Figure 6.19). As expected, this performance metric decreases while the number of publishers increases. In particular, when the range



(a) Downloaded data depending on the publisher satellites



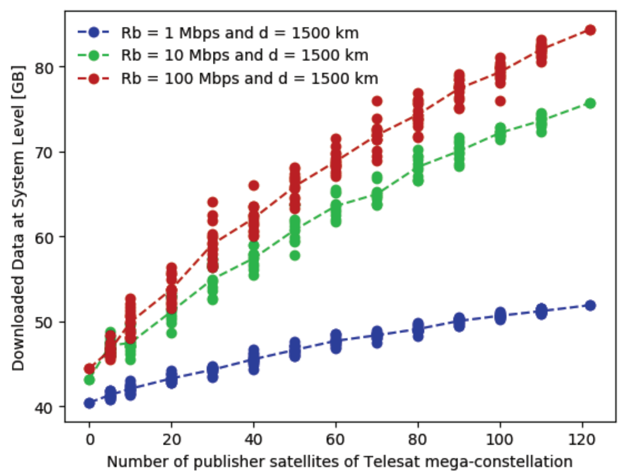
(b) Average available time percent per day of memory-overloaded EO satellites with respect to the publisher satellites

Figure 6.20: Downloaded data and average available time percent of the memory-overloaded EO satellites per day according to the publisher satellites and ISL subsystems with $Rb = 1$ Mbps and different maximum ranges d_{max}. Simulations conducted in an Intel-i7 computer with 16 GB of DDR4 memory, and spending between 30 minutes up to 40 hours per simulation.

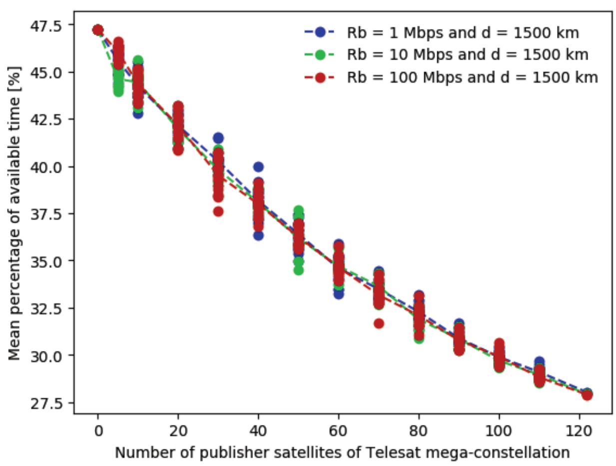
is 1500 km the unused time is reduced from 11 hours to 8 hours per day (approximately), having still some unused time. This curve changes considerably for larger ISL ranges, such as 2500 and 3000 km. In particular, the value trend follows a logarithmic curve, reaching a minimum of 45 minutes per day of inactivity (best case). This is also reflected in previous Figure 6.20a where the amount of downloaded data is bounded until a maximum value, because the saturated satellites do not have more time to keep downloading data with federations. In this case, the entire satellite system can download 51.92 GB, 57.04 GB, and 56.11 GB (respectively). This corresponds to an increase of 34%, 47%, and 45% of the number of downloaded bytes with respect to the original case (only EO without satellite-to-satellite communications).

One of the resulting features of this topology (presented in Section 6.6.2) is that the routes are abundant and large, but also short in lifetime. Therefore, federations are established also shortly, being necessary to leverage the time while they are stable. Therefore, the data rate of the ISL subsystem can play an important role. Specifically, more data can be exchanged during the same lapse of time with greater data rates. Figure 6.21 presents thus a study on how the data download is impacted with three ISL subsystems providing 1 Mbps, 10 Mbps, and 100 Mbps at 1500 km of range. At the first sight, the amount of downloaded data has increased considerably, reaching greater values than in the previous case (modifying the distances). Additionally, the improvement between the different data rates is not linear. Indeed, there is less improvement passing from 10 Mbps to 100Mbps, rather than from 1 Mbps to 10 Mbps. This situation appears because although the ISL data rate grows, the downlink data rate remains always to 1 Mbps. Thus, the bottleneck would be this downlink interface. There is, however, an important conclusion in these results: spreading the data over the entire satellite system is beneficial, because each satellite will then download the data during its downlink passes.

Note that the inactive time percentage of the saturated satellites follows in all the cases the same curve in Figure 6.21b, which is indeed the same as the blue line in Figure 6.20b. This reinforces the concept that **the distance is the unique parameter of the ISL subsystem that**



(a) Downloaded data with respect to the number of publishers and ISL maximum ranges



(b) ISL of 1 Mbps and 2500 km as range

Figure 6.21: Downloaded data and time percentage of saturated EO satellites per day according to the publisher satellites and ISL subsystems with $d_{max} = 1500$ km Mbps and different data rates Rb. Simulations conducted in an Intel-i7 computer with 16 GB of DDR4 memory, and spending between 30 minutes up to 40 hours per simulation.

can modify the activity of those saturated satellites. It is not the case of the downloaded data, which is affected by the range and the data rate. Note that this also indicates that there is still margin to improve if an ISL that provides high data rate and large range is installed in the satellite. These configurations download a maximum of 51.92 GB, 75.74 GB, and 84.36 GB respectively. This corresponds to a growth of 34%, 95%, and 118% with respect to the original case.

In conclusion, the use of the OSADP enables downloading more data in all cases, being evident that the improvement is larger when more satellites participate sharing the service. Additionally, the results demonstrate that increasing the ISL range enables to improve the connectivity, and thus the use of time that saturated satellites remain inactive. However, ISL subsystems with greater data rates better leverages the temporal connections than those with larger ranges. **To increase the amount of downloaded data, the enhancement of ISL data rate seems to be more beneficial**. This is the case for those satellite missions that are limited in the downlink capacity, and it is intended to improve it.

6.7 Summary

The establishment of the IoSat paradigm where satellites can establish temporal networks requires the development of different technologies. Previous chapters have presented a discussion of which capabilities an appropriate routing protocol should have to deploy ISN, and novel predictive techniques that could enhance this protocol by anticipating the disruption of the network. This chapter has presented the development of other required technologies. One of the challenges of this paradigm is the notification of the available services, and the corresponding establishment of satellite federations. This chapter has gravitated around these challenges, and presented the federation protocol suite. This suite is composed of the OSADP and the FeDeCoP protocols. The OSADP enables to publish the available services following a proper dissemination in the network which bounds the regions of the network where the publication is forwarded. A satellite that receives this publication retrieves information about the available service, and it can decide to request the establishment of the federation. This establishment is conducted with the FeDeCoP which proposes the execution of three phases: the negotiation, the consumption, and the closure phases. With these three phases, different types of federations can be established in a satellite system. The design of both protocols has been detailed in this chapter, presenting all its features, such as packet format, and their flow diagrams.

To evaluate the protocols' design a realistic scenario composed of EO satellites that monitor the Arctic region has been defined. These satellites are equipped with payloads commonly used to observe the ice status, such as SAR and MWRs. A large ground segment has been defined to provide enough downlink opportunities to be published, and additionally the Telesat mega-constellation has been integrated as a network backbone. The downlink areas have been properly modeled according to current technologies used in satellite missions. Furthermore, different ISL subsystems have been considered to evaluate its impact on the protocol suite.

The results presented in this study demonstrate that the proposed protocols enable the establishment of federations, and greatly increase the amount of downloaded data of the entire system. Specifically, executing the protocol suite with only the EO satellites allows downloading 7% more than not using it. In this case, the potential benefits of applying this solution are limited by the network disruption associated to the small number of satellites deployed. Despite this limitation, the use of the federation protocol suite in current satellite missions improves the downlink capacity. To mitigate this situation and fully evaluate the potential of the proposed protocols, a satellite backbone has been integrated. With a Telesat like mega-constellation–with maximum communications ranges from 1500 km to 3000 km, and a data rate from 1 Mbps to 100 Mbps–working as a network backbone, the system can download up to 15% (depending on the ISL characteristics) than the original case. Finally, if the mega-constellation publishes its downlink opportunities, the downloaded data growth is 118%; i.e. the overall downloaded data is more than doubled.

As ISL subsystems with different communications ranges and data rates have been used, the study also provides information about how these characteristics suit satellite networks. These networks are characterized by being greatly variable, having considerable changes on the different links that compose the network. The retrieved results remark that an ISL with greater data rate than larger range better matches this kind of network. In particular, large data rates allow optimizing the temporal links by exchanging more data, and thus downloading more data for the entire system. Therefore, ISLs with large data rates should be encouraged if the goal is to increase the download capacity. For other goals (e.g. to minimize latency), a different analysis must be conducted.

All these results demonstrate that the proposed protocol can improve current and future EO missions by downloading more data. As compared to the traditional solution of increasing the number of ground stations, the proposed protocols reduce the cost of the downloaded data with the same ground segment; i.e. cost per byte. Furthermore, the growth on the downlink opportunities enables to retrieve the generated data faster than in the traditional case. Finally, the protocol suite enables to optimize the satellite resources by notifying the available ones to establish federations, as the FSS proposal.

Despite all these benefits, the potential of the federation protocol suite is limited by the network disruption. As previously discussed, the protocol is originally conceived to provide mechanisms to leverage the opportunity to consume services that remain available during lapses of time. Therefore, the protocol definition does not take in consideration the network disruption. This fragmentation could miss the opportunity to consume an available service due to the lack of a route. In this regard, the application of disruptive-tolerant mechanisms in the OSADP and FeDeCoP may be interesting protocol extensions to be addressed in future investigations. These improvements would enhance its features and allow it to achieve its maximum performance, which has been presented with the satellite backbone case.

7

Federated Satellite System Experiment: A proof-of-concept with a stratospheric balloon campaign

7.1 Introduction

The last chapter has presented the federation protocol suite which allows a satellite to notify available services and establish federations. These protocols are part of the necessary technology development in order to the IoSat paradigm to be able to become a reality. This paradigm promotes the deployment of temporal and opportunistic satellite networks over which satellites federations are established. These federations allows to share unallocated resources among different satellites, leveraging from these opportunities to improve the mission performance. The federation protocol suite is composed of two application-layer protocols: the OSADP and the FeDeCoP. The former allows a satellite to notify its available services to other satellites. These services corresponds to the unallocated resources that a satellite has, and are not continuously available, existing opportunities to consume them only during lapses of time. With the OSADP, a satellite can use these opportunities by discovering those published services. Once the service is notified, a satellite can request it to the provider of the service. The FeDeCoP presents the different regulations to establish and maintain a federation between two satellites. By following three phases, the provider can be certain that the amount of resources that has been offered are correctly consumed by the customer.

The presented results suggest that using this protocol suite, satellites can largely increase their downlink capability by means of additional downlink contacts. Despite this performance depends on the disruption of the network, these promising results motivate pursuing the development of this suite that could satisfy certain requirements of EO and broadband telecommunications communities. With this premise in mind, we have developed the Federated Satellite System Experiment (FSSExp) payload. This payload is a system which includes an implementation of the federation protocol suite, and the necessary hardware components to establish federations. This payload is composed of different subsystems, where the RF-ISL board stands out among the others. The use of multiple RF-ISL boards allows the creation of a wireless communications interface over which data can be exchanged. This board inherits the design of the communication system developed in the UPC NanoSat Lab for the ^{3}Cat-4 mission (Ruiz-de-Azua et al., 2019a). This 1U CubeSat mission aims at demonstrating the benefits of these small platforms for observing the Earth. The details of this mission have been excluded of this dissertation, but—due to its relevance—the design of the RF-ISL board is presented in the following sections. The FSSExp payload enables to verify the proposed protocol suite on a new environment, and taking in consideration the hardware limitations. This payload has been included in the FSSCat mission (Camps et al., 2018) which aims at demonstrating the benefits of applying the FSS concept to EO missions. This mission promotes the execution of multiple payloads, which are distributed in two 6U CubeSats following a train formation. The FSSExp execution in this mission corresponds to a proof-of-concept of FSS, which would demonstrate the feasibility of deploying future satellite federations.

This chapter presents the design of this payload by reviewing all its components and the philosophy of the experiment. The development of this payload has been conducted over different phases. The design and implementation allowed the verification of the final system in the laboratory. This verification consisted of different executions in a controlled location, with a far distance configuration, and in the facilities that allowed to emulate space conditions. All these tests were conducted satisfactorily, demonstrating that the FSSExp payload could be embedded in the spacecraft. Nevertheless, another scenario where the payloads could experience real mobility was also planned to be conducted. Therefore, a stratospheric balloon campaign has conducted as part of this verification plan. In this campaign, three stratospheric balloons established different federations sharing downlink opportunities and storage capacity using the FSSExp payload. In particular, the balloons generate data that is downloaded to a ground station using federations with the other balloons. This campaign allows to anticipate how the FSSExp payload will work in the FSSCat mission. Therefore, it becomes a proof-of-concept of satellite federations. The FSSExp payload has been adapted to be integrated in the balloon structure, but the subsystems remained as the ones installed in the spacecraft. This chapter presents the overall stratospheric balloon campaign performed outside Moscow, Russia, and discusses the metrics retrieved during its execution. The results demonstrate that using the proposed protocol the federations can be correctly established, and it is possible to increase the communications range using intermediate nodes. Despite the promising results, the FSSExp payload has been designed to demonstrate the concept of satellite federations with the proposed protocol suite. Its current design corresponds to a prototype that requires further development to achieve commercial performance.

The main contribution of the present research is 1) an explanation of the FSSExp payload concept, 2) details of the payload architecture and the subsystems that compose it, 3) the design of the software architecture implemented for the FSSExp payload, and 4) a proof of the benefits of using the proposed solution by evaluating the retrieved results of the stratospheric balloon campaign. The research presented in this chapter has been published in the International Geoscience and Remote Sensing Symposium (Camps et al., 2018; Ruiz-de-Azua et al., 2019c, 2020d), and an article has been submitted to the peer-reviewed IEEE Access journal (Ruiz-de-Azua et al., 2020c). Furthermore, the design process that has experienced the RF-ISL board from the ^{3}Cat-4 mission until the FSSCat context has been published in the URSI Atlantic Radio Science Meeting (Ruiz-de-Azua et al., 2018a), the International Geoscience and Remote Sensing Symposium (Ruiz-de-Azua et al., 2019a), and the Specialist Meeting on Reflectometry using GNSS and other Signals of Opportunity (GNSS+R) (Ruiz-de-Azua et al., 2019d). The details of these publications are accessible in Chapter 9 with the identifiers [VIII], [X], [XI], [XII], [XVI], and [XVIII].

The remaining of this chapter is structured as follows. First, Section 7.2 introduces the FSSCat mission to clarify the context over which the FSSExp payload has been developed. The concept of the experiment conducted with the stratospheric balloon campaign is presented in Section 7.3. The details of the FSSExp payload architecture are presented in Section 7.4, while its software implementation is described in Section 7.5. Section 7.6 elaborates the characteristics of the RF-ISL board, essential for the FSSExp payload. The stratospheric balloon campaign and the results obtained are discussed in Section 7.7. Finally, Section 7.8 concludes the chapter and proposes new research lines to enhance current FSSExp payload design.

7.2 FSSCat mission

The FSSCat mission was officially presented during the announcement of the Copernicus Masters awards at Tallin, Estonia on November 2017. The mission proposal was selected as the winner of the ESA Sentinel Small Satellite Challenge (S^3), and the overall winner of the Copernicus masters competition. The ESA S^3 Challenge goal was to stimulate ground-breaking satellite designs, testing and manufacturing solutions leading to small missions that could complement or provide added value to the current Sentinel satellite missions. FSSCat mission is formed by two 6U CubeSats in support of the Copernicus Land and Marine Environment services. Among the different services, the mission would provide relevant measurements for the Copernicus Land Monitoring Service (CLMS), and the Copernicus Marine Environment Monitoring Service (CMEMS). The lack of soil moisture data in these services, and the lack of an ESA SMOS follow-on mission (Kerr et al., 2010), makes the FSSCat a possible candidate to provide the needed measurements to complement the CLMS. Furthermore, current user demands are requesting measurements of the ice status on the poles, emphasizing on the coverage and thickness. The FSSCat mission combines different EO instruments to provide the desired observables.

The CubeSats in the mission are known as ^{3}Cat-5/A and ^{3}Cat-5/B, and orbit following a train formation; i.e. one after the other (Figure 7.1). With this tandem, each satellite can observe the same target area with its own EO instruments. Therefore, this formation promotes the use of instruments that provide different measurements. The ^{3}Cat-5/A spacecraft is equipped with UPC's Flexible Microwave Payload - version 2 (FMPL-2) (Munoz-Martin et al., 2019), a combined microwave radiometer and Global Navigation Satellite System Reflectometer (GNSS-R) instrument to monitor ice cover, ice thickness, and soil moisture. To complement the measurements performed with the FMPL-2, the ^{3}Cat-5/B spacecraft has the Cosine's Hyperscout-2, a hyperspectral Visible and Near-Infrared (VNIR) and Thermal Infrared (TIR) camera with artificial intelligence for cloud detection (Esposito et al., 2019).

The mission is also a technology enabler of satellite federations, because it integrates two technology demonstrators. An O-ISL device for CubeSats is equipped in each spacecraft to establish the first optical link between satellites with these small platforms. The demonstration of this technology would promote the development of high-capacity devices that are required in intermittent satellite networks (as suggested in Chapter 6). The additional technology demonstrator is the FSSExp payload which corresponds to the implementation of the federation protocol suite in a hardware system. This system is equipped in each spacecraft to establish different federations. Specifically, the execution of the FSSExp payloads would demonstrate the feasibility to deploy federations between two CubeSats to share downlink and storage opportunities (Figure 7.2). The execution of these technology demonstrators is expected to be conducted

[17] `https://nanosatlab.upc.edu`. Last access: July 24th, 2020.

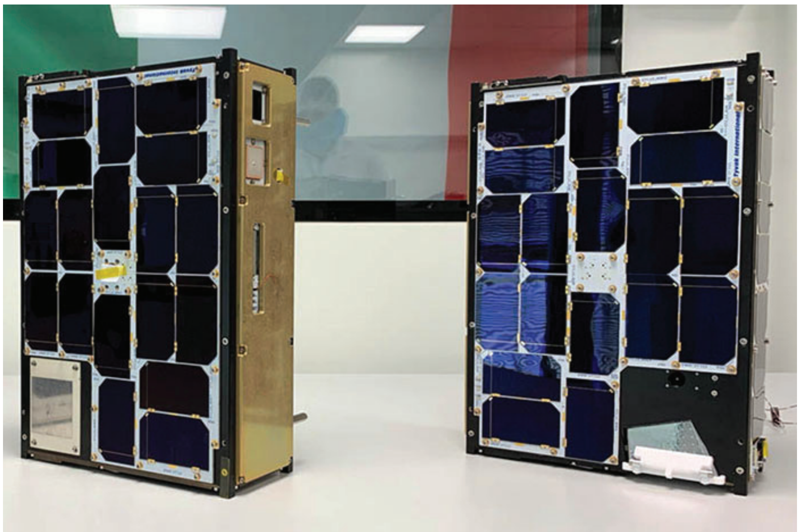


Figure 7.1: Artist view of the FSSCat mission concept with both CubeSats. (c) UPC NanoSat Lab[17].

at the et of the mission, three months later it launches. Nowadays, both satellites are correctly in-orbit deployed.

The development of the FSSExp payload has experienced different phases. The former phase has enabled the design of the system and its implementation. Furthermore, a verification plan has ensured the proper functionality of the payload in space environment conditions. As part of this verification plan, the payload has been evaluated in a stratospheric balloon campaign, where three balloons established federations by the execution of this payloads. The following sections present the details of this campaign, and the retieved results which demonstrate the correct execution of the system.

Figure 7.2: Modules that correspond the FSSExp payloads integrated in both spacecrafts.

7.3 Federated Satellite System Experiment

The design of the federation protocol suite prepares the way to develop a system which enables to deploy satellite federations. This is the purpose of the FSSExp payload which integrates the necessary hardware and software components in a single device. The execution of this payload enables to perform different experiments to evaluate the behavior and performance of the designed protocols. These experiments are thus the necessary proof-of-concept to demonstrate the feasibility of deploying federations between satellites. The FeDeCoP is designed to establish

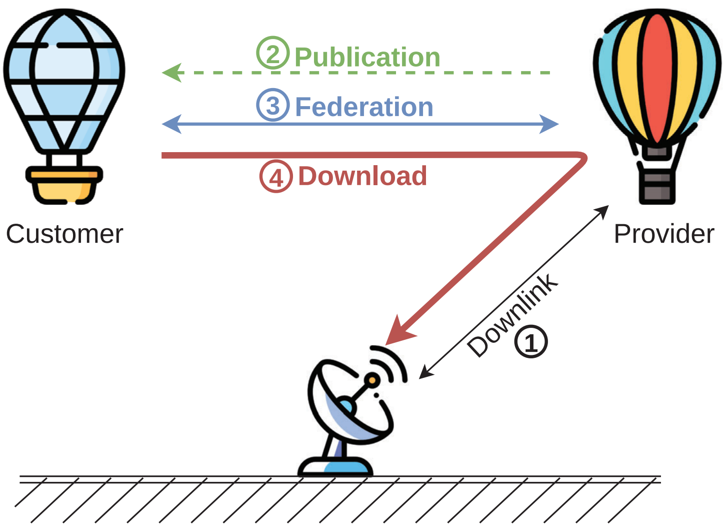


Figure 7.3: Representation of the establishment of a federation between balloons. The numbers correspond to the time sequence when each action is performed (from low values to greater ones).

federations that share heterogeneous resources as services, such as memory storage or processing capacity. However, in the FSSCat mission the different FSS experiments are focused on current EO needs. Therefore, federations in which the satellites share downlink opportunities and storage capacity are deployed during the mission.

As part of the verification plan of the FSSExp payload, a stratospheric balloon campaign was conducted in December 2019. In this campaign, three stratospheric balloons established different federations sharing downlink opportunities using each one a model of the FSSExp payload. This scenario was representative to the FSSCat one, although differences between satellites and balloons dynamics exist. To have a successful campaign, a portable ground station system was developed, with which downlink contacts can be established with the balloons to download their data. Using these downlink contacts, a stratospheric balloon decides to share itss download capacity with the others, becoming the service provider. Consequently, the other balloons can download data through federations with the service provider. Figure 7.3 presents a schematic view of the expected experiment performed during the campaign.

The experiment was conceived as follows. The ground station connects to one of the stratospheric balloons, e.g. balloon A. After the connection handshake, balloon A starts to download data until it transfers all the generated packets. At this moment, balloon A publishes the availability of the downlink contact using the OSAD protocol (Ruiz-de-Azua et al., 2020a). This publication is received by balloon B, which decides to establish a federation. Therefore, balloon B transmits a request of the service, triggering the negotiation phase of the federation. Once this phase is accepted by the balloon A, the roles of both balloons are defined: being balloon A as the service provider, and balloon B as the customer. At this moment, the balloons start the federated phase in which the customer sends its data to the service provider in order to be downloaded. After the correct download, the service provider notifies the customer of the correct data transfer. This federation remains active until the contact with the ground station ends (i.e. service ending) or until the customer does not need the service any more. At this moment, the closure phase finishes the connection, and each balloon goes back to their nominal operations (i.e. generating data). As a result, balloon B has additional downlink opportunities to transmit data to the ground.

Note that although the federations performed in the stratospheric balloon campaign and the FSSCat mission are focused on sharing downlink opportunities, the designed protocols are generalist. These protocols can be used to establish federations for other purposes, such as memory storage. For that reason, an additional federation based on sharing storage capacity has also been conducted during the campaign. In this federation, a balloon publishes the possibility to store data, which is used by the others. With these two types of federations, the versatility of the proposed protocol has been demonstrated. Future campaigns may focus on the demonstration of the adaptability of the protocol to other kind of services.

7.4 The FSSExp payload architecture

As previously presented, the stratospheric balloon campaign aimed at evaluating the performance of the proposed protocol, and the benefits of applying it in satellite missions. Although the original FSSExp payload was designed for the FSSCat mission, a tuning of the hardware was conducted for the stratospheric balloon campaign. Figure 7.4 presents these subsystems that compose four models of the FSSExp payload, developed specifically for the stratospheric balloon campaign. The subsystems that compose an FSSExp payload are (1) a microSD card as a storage system, (2) the Interface Board, (3) the On-Board Computer (OBC), (4) the Long Range (LoRa) payload, (5) the RF-ISL board. This section presents the details of this architecture used for the balloon campaign.

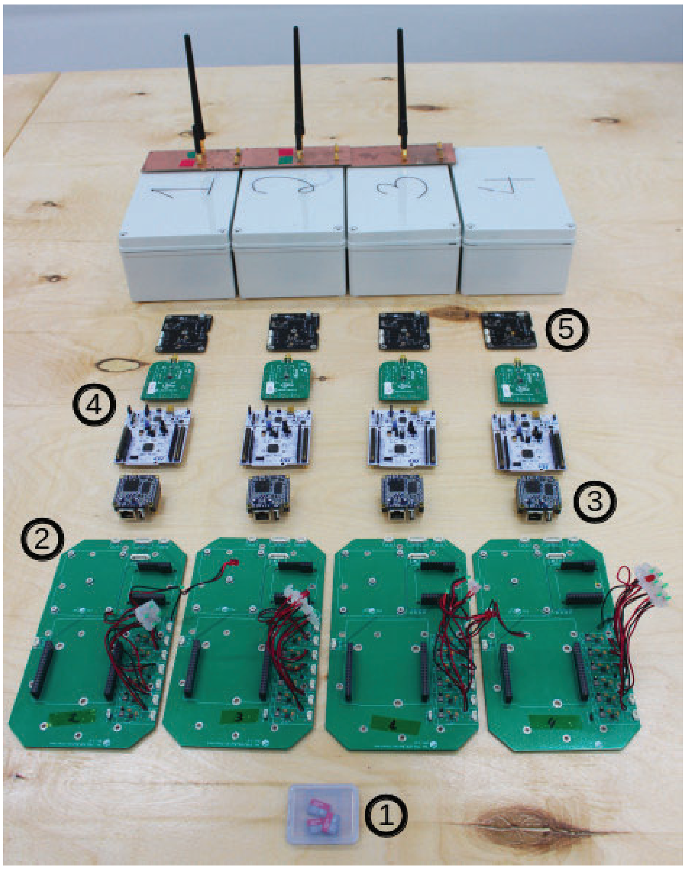


Figure 7.4: Subsystems of the FSSExp payload for the stratospheric balloon campaign

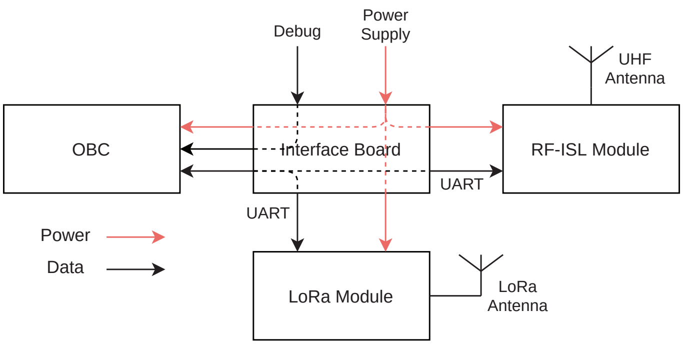


Figure 7.5: Block diagram of the subsystems and their interconnections that compose the FSSExp payload.

A star architecture has been used to interconnect the subsystems, in which a central module unifies all the data and power interfaces (Figure 7.5). This module is the Interface Board, and it distributes the incoming power supply (at 5 V) to the other subsystems. It incorporates a network of DC-to-DC converters which provides seven independent power lines at 3 V, 3.3 V, and 5 V. Additionally, this board unifies the data interfaces of all the subsystems in different PicoBlade connectors placed around the board. With this design, the integration of the subsystems on top of this module becomes simple and easy.

The payload OBC is the NanoPi NEO (FriendlyARM, 2015) model, which is a single-board computer with a Quad-core Advanced RISC Machine (ARM) Cortex-A7. This processor architecture facilitates the integration of a Linux-based Operating System (OS), ideal to speed up the software development and its replication. This OBC executes thus the software components necessary to perform the experiment, such as the federation protocol or the data retrieval. All this software, including the OS, is installed in a micro Secure Digital (SD) card of 32 GB of storage capacity, essential to store all the resulting metrics of the experiment. By default, the NanoPi NEO has different data interfaces, which follow standardized intra-board communications protocols, such as the Inter-Integrated Circuit (I2C). For the FSSExp payload, only the Universal Asynchronous Received-Transmitter (UART) ports have been used to communicate with the other subsystems. One of those subsystems is the RF-ISL board, which provides the communications means needed to exchange packets between the stratospheric balloons. In particular, a wireless communications link is established using a receiver-transceiver unit with transmission

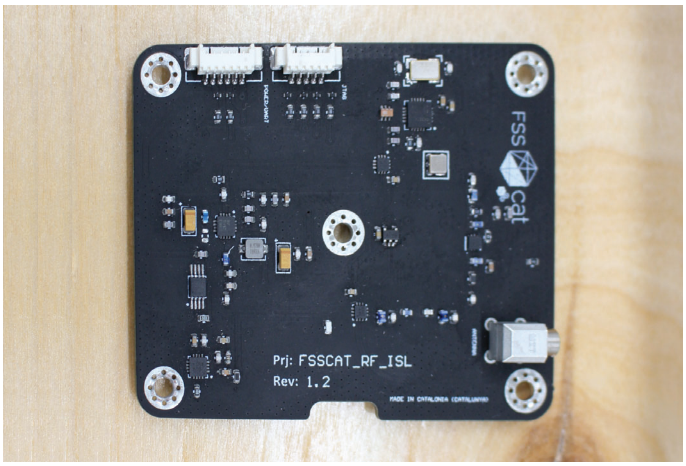


Figure 7.6: Top view of the RF-ISL board

and reception chains. Figure 7.6 presents the top view of the RF-ISL board, which detailed in Section 7.6. An additional subsystem is included in the payload to characterize the performance of the LoRa technology (Vangelista et al., 2015) in satellite platforms. In particular, a Commercial Off-The-Shelf (COTS) board with the LoRa SX-126X chip (Sensing and Division, 2015) is integrated with the Interface board, being properly supplied and connected to the OBC to store the different metrics of its execution. Despite the FSSExp payload hosts this additional subsystem, its execution and analysis are not part of the FSS experiment itself. Therefore, the details of this additional subsystem and the results of its performance are not presented in this chapter.

At the end, all the subsystems are integrated on the Interface board, and placed in their corresponding locations. Figure 7.7 shows a picture of the FSSExp payload with all the subsystems integrated in the corresponding structure. This structure of 23 cm length, 14 cm width, and 9.5 cm height provides also an interface panel with two D-subminature connectors of DB-9 type. One of them is used to provide the external power supply and to connect with the debug port of the OBC. This port allows accessing the Linux user space, and thus manage the file system among other actions. The secondary DB-9 connector incorporates the Join Test Action Group (JTAG) port to flash the code in the RF-ISL board. Once the payload is powered up, the software is executed without needing any additional telecommand. The FSSExp payload consumes around 1.4 W (i.e. 280 mA at 5 V) during its nominal execution.

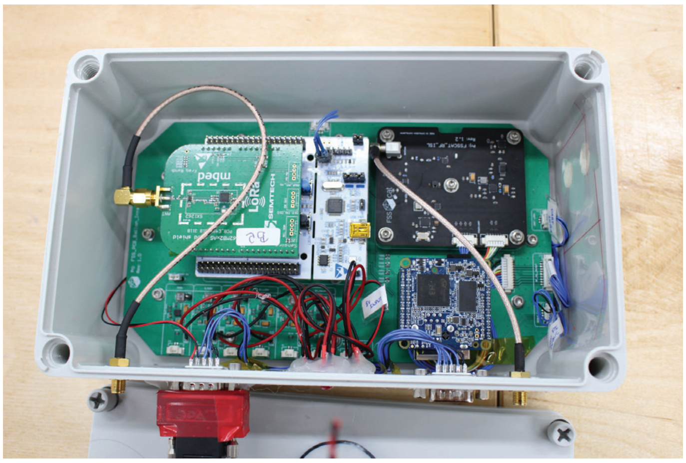


Figure 7.7: FSSExp payload with its subsystems once integrated

7.5 The FSSExp Software Architecture

To complement the previous explanation of the different subsystems that compose the FSSExp payload, the payload software architecture is detailed in this section. The implementation of the software has been focused on providing an effective, and replicable solution. For that reason, its implementation has been conducted over a Linux-based OS, and using the Java language (Gosling et al., 2000). This language is based on the Object-Oriented Programming (OOP) framework (Brogi et al., 1992), in which the concept of an "object" represents a software component with data or code. This framework facilitates the organization of the code, its production, and a possible extension. Additionally, this language is characterized by its portability to multiple hardware platforms. Therefore, its use helps to replicate the FSSExp software in other systems.

The FSSExp execution is composed of multiple software threads which conduct different functionalities of the experiment. The software architecture has been conceived around these

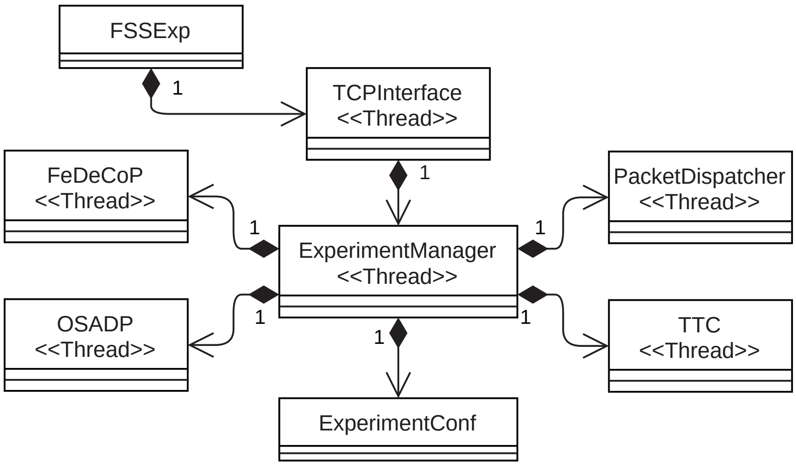


Figure 7.8: Threads of the FSSExp payload software

threads. The *ExperimentManager* thread controls the different status of the experiment. It is thus the responsible to trigger the execution of the additional threads after reading the corresponding configuration file. Due to its significance, a centralized approach has been followed in which the *ExperimentManager* is the central coordinator. Figure 7.8 presents the UML diagram of this centralized thread architecture. Although the *ExperimentManager* is the central thread, the execution of the FSSExp software starts with the *FSSExp* class. This one performs all the necessary verifications of the required components before starting the nominal execution. For instance, it verifies that the file system correctly initialized. It is thus conceived as a boot loader of the FSSExp software. This class also initializes the debug interface, which is used by an external computer to interact with the FSSExp payload. In particular, it triggers the *TCPInterface* thread which creates a TCP server to exchange commands and data. This interface is conceived to evaluate the status of the payload during the ground operations, such as the test phase. The payload itself is totally autonomous, being not necessary to send a command that triggers its execution.

The remaining threads are focused on specific functionalities of the experiment. The Telemetry, Tracking and Control (TTC) functions are conducted by the *TTC* thread. This thread is the responsible to interact with the ground station to download the data, and to receive telecommands. The execution of the proposed protocols is performed in the *OSADP* and *FeDeCoP* classes. All these three threads interact with the RF-ISL board to exchange packets with another node. An uncontrolled access to this board could provoke that two threads may end up attempting to transmit and receive at the same time. This situation would degenerate into a malfunction of the entire payload. A mechanism to manage this situation has been conceived.

7.5.1 Interface with the RF-ISL board

The *PacketDispatcher* thread is conceived to avoid a concurrent problem by multiple threads access to the RF-ISL board. Specifically, a single, and synchronized interface has been developed using the *PacketDispatcher* over which all the requests are directed. Figure 7.9 presents the UML diagram of this interaction between the protocol threads and the *PacketDispatcher*. This interaction is based on offering two dedicated buffers to collect all the requests from the threads that want to interact with the RF-ISL board. The *TxPacketBuffer* is used to notify the *PacketDispatcher* about the transmission of a packet. Let's make an example to clarify this interaction. If the *FeDeCoP* thread wants to transmit a packet, it stores the packet into the *TxPacketBuffer*.

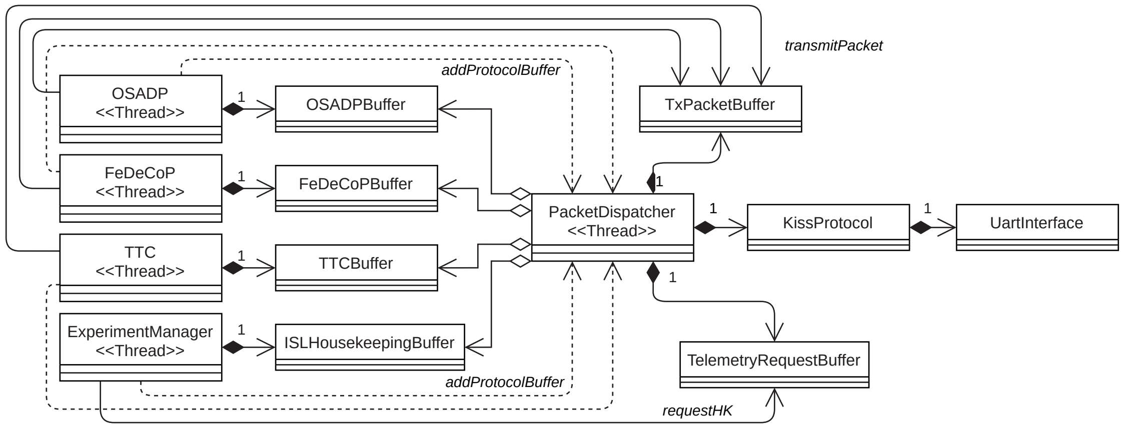


Figure 7.9: UML diagram of the interaction between the different threads and the interface to the RF-ISL board

Periodically, the *PacketDispatcher* evaluates the content of this buffer, searching for a possible packet to transmit. If it is the case, it encapsulates the packet following the intra-board communications protocol stack and sends it to the RF-ISL board. Additionally, the *PacketDispatcher* also checks if the RF-ISL board has received any packet from other nodes. If it is the case, the dispatcher evaluates to which thread the packet is intended, and place it in the corresponding output buffer. One output buffer is allocated per protocol thread, ensuring the independency between the threads. The *PacketDispatcher* dispatches thus the incoming packets from the RF-ISL board to the corresponding output buffer, like a postman. The transmission and reception processes are asynchronous, which does not block the thread after transmitting a packet in order to receive one.

The packets that are passed from the threads are properly encapsulated by the *PacketDispatcher*. This encapsulation has been designed specifically for the FSSExp case. It adds multiple fields as a header to manage the point-to-point communications, and generates a byte structure called the FSSExp packet. These packets are thus exchanged between the balloons through the ISLs. Figure 7.10 presents the structure of this FSSExp packet. The first six bits are used to identify the *source* node of the transmitted packet and its *destination* node. Only three bits have been used to identify a node because in the campaign was only composed of three balloons and a ground station (i.e. four nodes). The destination field helps the *PacketDispatcher* to evaluate if the received is intended for this node, and forward it to the threads. Otherwise, the packet is discarded. Additionally, another identifier has been considered to emulate a broadcast address that is accepted by all the nodes. This type of transmission is required by the OSADP. The *prot num* field represents the protocol number, which corresponds to the identifier of each thread. It is thus used by the dispatcher to select the output buffer in which the received packet is stored. The *timestamp* field is the transmission date that is used as a resulting metric of the experiment to evaluate the performance of the protocols. The *length* field corresponds to the size of the thread packet that is being encapsulated (byte unit). Note that this field enables to transmit up to 256 bytes of data, which in the stratospheric balloon case is in line with a data structure. Finally, a CRC of 2 bytes, known as CRC-16, is implemented as a *checksum* of the entire FSSExp packet.

Moreover, the intra-board communications protocol to interact with the RF-ISL board is implemented with an extension of the Keep It Simple Stupid (KISS) protocol (Wolfe, 2004). The

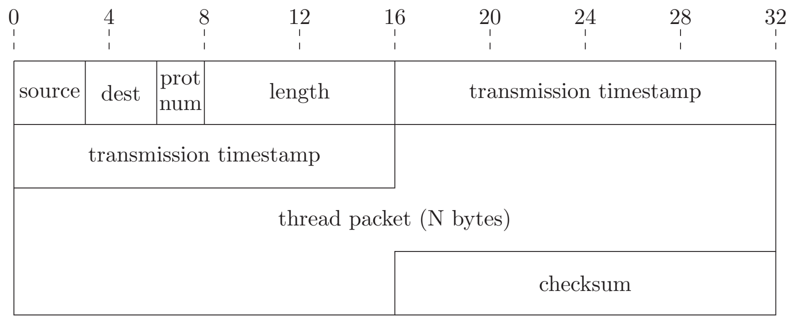


Figure 7.10: Structure of a FSSExp packet (bit unit)

original protocol included some special characters to identify the start and end of a frame. In this case, the protocol has been extended with an additional byte to identify some commands to interact with the RF-ISL board. Specifically, these commands corresponds to the actions that can be done with the board, such as transmit a packet, request if a packet has been received or request its data. The resulting frame is then forwarded to the RF-ISL board using a UART interface. These additional bytes added by the KISS protocol are then removed in the RF-ISL board, transmitting only the FSSExp packet. Figure 7.11 clarifies the different layers of encapsulation that a packet from a thread has.

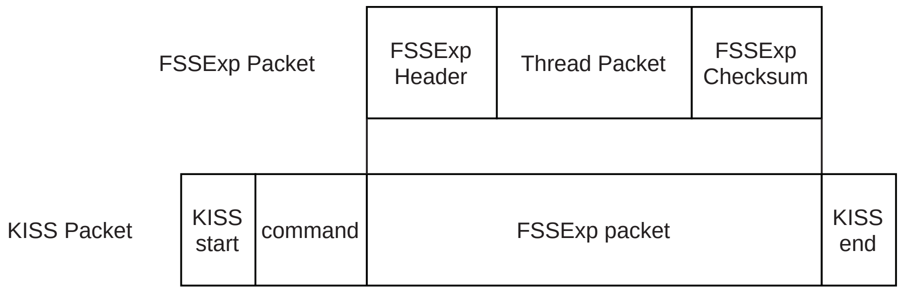


Figure 7.11: Encapsulation of the FSSExp packet to communicate with the RF-ISL board

7.5.2 Data generation and protocol execution

Returning to Figure 7.9, the *PacketDispatcher* provides another interface to retrieve the telemetry of the RF-ISL board. Following the same strategy as in the transmission case, the *TelemetryRequestBuffer* contains all the requests to retrieve the housekeeping of the RF-ISL board. Unlike in the transmission/reception case, this request is synchronous, and the *PacketDispatcher* directly processes all these requests. The data block is then placed in the corresponding output buffer, known as the *ISLHousekeepingBuffer*. This telemetry is retrieved by the *ExperimentManager* thread which collects all the status of all the FSSExp payload, and it is represented in the *HousekeepingItem* class. This *HousekeepingItem* is encapsulated in a *PayloadDataBlock* class which includes the identifier of the balloon, and the creation date in the format of a timestamp. The *ExperimentManager* generates a *PayloadDataBlock* instance every 20 seconds. These data blocks are downloaded to the ground using a direct connection or over a federation. The additional information in the *PayloadDataBlock* is used thus to evaluate on ground which balloon has generated the telemetry, and verify that if they are downloaded through a federation.

As these generated *PayloadDataBlock* packets are downloaded, the *ExperimentManager* must interact with the *TTC* and *FeDeCoP* threads. As in the previous cases, a dedicated synchronized buffer is used to manage this situation. Specifically, the *PayloadBuffer* contains all

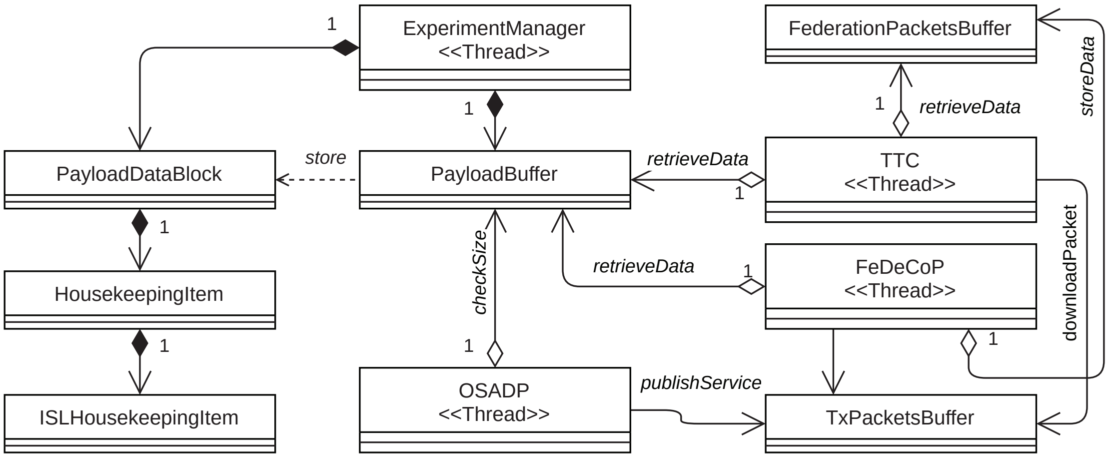


Figure 7.12: UML representation of the generated telemetry data and its transmission by the system threads

the telemetry packets that must to be download, which represents the memory usage of the balloon. The interaction of the threads with this buffer is conducted as follows. When the *TTC* thread establishes a downlink contact with the ground station, it downloads the packets that are stored in the buffer. Once the buffer is empty, and the downlink is still available, the *TTC* thread notifies the *OSADP* one that the resource can be shared. This one starts the protocol to publish the downlink service to the other balloons. If a request of the service is received, the *FeDeCoP* thread starts the execution of the federation as previously presented. In this federation, the balloon customer transmits packets to the supplier which are stored in the *PayloadBuffer*. On the other hand, the packets that are received by the *FeDeCoP* thread are stored in the *FederationPacketsBuffer*. This additional buffer aims at differentiating the packets that are internally generated with respect to the ones that are received with the federation. This distinction is used to prioritize the downlink of the internally generated packets rather than the federation ones. Therefore, the *TTC* thread always checks firstly the *PayloadBuffer*, and then the *FederationPacketsBuffer*. If a packet is detected, it automatically downloads it to the ground. Once the downlink is closed, the *TTC* thread notifies the *FeDeCoP* thread, which closes properly the federation. Note that the *TTC* thread downloads the incoming packets of the federation, and the new ones generated by the *ExperimentManager* thread during the execution of a federation. Figure 7.12 presents the UML diagram of the different classes that interact during the federation execution.

7.5.3 Outcome metrics and storage process

The performance of the FSSExp software must be characterized in order to evaluate the feasibility of the proposed solution and its behavior. For that reason, multiple metrics are stored during its execution. As previously presented, the FSSExp payload has a microSD card as a storage system with 32 GB of capacity. This system is populated with the journaling file system fourth extended filesystem, well-known as ext4 (Carrier, 2005). In this file system, multiple files are created to store the corresponding execution metrics. The strategy followed with the selection of the metrics has been to store as much information as possible, which ensures the proper characterization of the FSSExp payload after the stratospheric balloon campaign.

Following this premise, the *PayloadBuffer* and the *FederationPacketsBuffer* classes have been associated to a file. This association is based on the storage in the file of each item that is inserted in the buffer, i.e. a payload data block or a FSSExp packet. Therefore, all the FSS-

Exp packets and payload data blocks generated during the execution of the FSSExp are completely stored (including the corresponding headers) in the *FederationPacketFile* and the *GeneratedPacketFile* respectively. Additionally, to this automatic storage, other classes are conceived to specifically store important metrics. The *HousekeepingStorage* class has been used to store the housekeeping items generated by the *ExperimentManager*. With this information stored in the *HousekeepingFile*, the behavior of the RF-ISL board is accessible to be processed, such as its temperature. The *DownloadedPacketsStorage* class keeps the information of all the packets that have been downloaded by the *TTC* thread. Following the same strategy, the *ISLPacketsStorage* class stores all the transmitted and received packets by the *PacketDispatcher* thread. Note that the received packets are only stored if they are intended to the balloon that receives them. Otherwise, the packet is discarded, and thus not stored in the file. Alternatively, a sniffed mode has been developed in the *PacketDispatcher* thread. This mode allows to store all the packets that are received in the *PacketDispatcher*, without considering to which balloon they are destined. This mode is really useful to monitor the packets that are being transmitted in the communications channel. For that reason, this mode has been enabled during the stratospheric balloon campaign in all the nodes (including the ground station). Figure 7.13 presents a global UML view of this storage system.

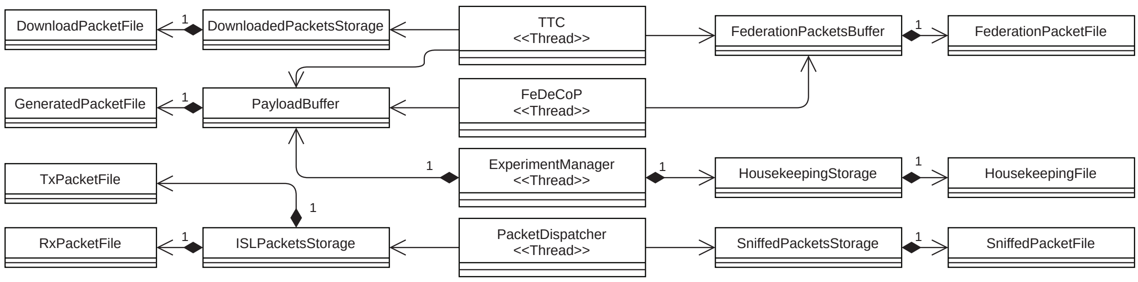


Figure 7.13: Storage system of the different metrics that characterizes the performance of the FSSExp execution

7.6 The RF-ISL Board

The RF-ISL board is a device that is used to establish the needed communications links between the stratospheric balloons. Those links are used to exchange the different messages generated by the proposed communications protocols. Therefore, the design of this board has been considered to offer a half-duplex interface. Figure 7.14 presents a block diagram of the different components that conform the board. Among the different components, some of them are related to the RF circuit which is designed to perform the low-level communications functions (e.g. modulation, signal conditioning, amplification, etc.) Two lines are defined to achieve the half-duplex interface—the transmission and the reception lines—in which the Low Noise Amplifier (LNA) and the Power Amplifier (PA) stand out as the essential components of each line. These components interact with the transceiver which processes the signal according to a custom configuration. A dedicated microcontroller interacts with this transceiver to perform other functionalities related to the medium access, and the interaction with external subsystems. This section presents the details of the RF circuit, and the software components executed in the microcontroller that all together conform the RF-ISL board.

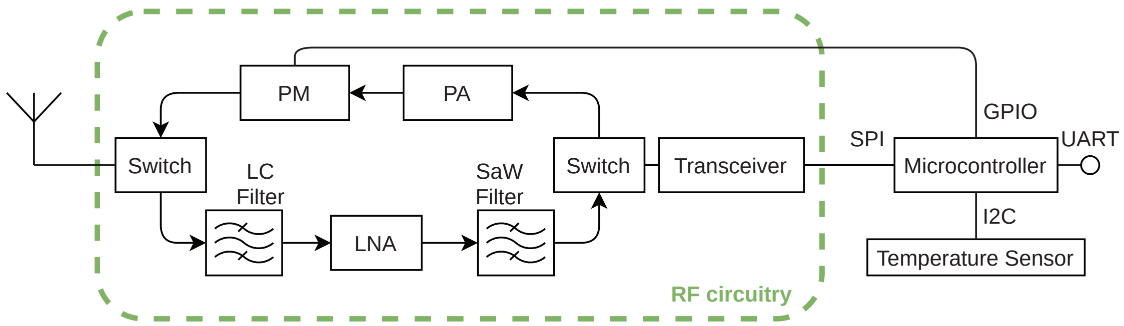


Figure 7.14: Block diagram of the RF-ISL board

7.6.1 Components of the RF circuit

The Texas Instrument CC1101 transceiver (Instruments, 2020) is the cornerstone of the RF circuit which transmits and receives messages through a RF interface. This sub-1GHz transceiver is designed to configure different aspects of the communications, such as type of modulation, central frequency, bandwidth, data rate, etc. The flexibility that offers these components helped to properly design the specifications of the device by means of different tests performed in the laboratory. At the end, the transceiver is configured to operate at 437.35 MHz—UHF amateur band—with 58 kHz of signal bandwidth. Moreover, this transceiver integrates a firmware that modulates the incoming packets using the Gaussian Minimum Shift Keying (GMSK) (Lui and Tsai, 2006). The messages provided by the microcontroller are encapsulated in a custom packet format designed by the manufacturer of the transceiver, but it can be configured at your discretion. This additional encapsulation provides mechanisms to synchronize at bit and at packet levels, and a FEC code. Further details about this encapsulation are presented in Section 7.6.2.

This transceiver has been complemented with a dedicated RF front-end circuit that improves its original characteristics. This circuit is composed of two parts or chains: the reception and transmission chains. The reception chain aims at conditioning the received analog signal to improve the resulting SNR. It is thus composed of wide band-pass filter designed with capacitors and inductances, a LNA, and a narrow band-pass filter. The former filter, known as LC filter, is designed following the π architecture, and it has been conceived to filter those signals out of a wide bandwidth centered in 433 MHz. The outcome filter has 115 MHz of effective bandwidth, which ensures a maximum insertion loss of 1 dB at the operational frequency; i.e. 437.35 MHz. Although the bandwidth of the filter is large, the fact that it provides low insertion loss favors to achieve a low noise figure. With this in mind, a LNA is included in the reception chain to provide 16.1 dB of gain with a noise figure of 0.85 dB. This component helps to mitigate the impact on the noise figure of the entire reception chain. Additionally, a Surface Acoustic Wave (SAW) filter is placed after the LNA to work with the narrow bandwidth of 6 MHz. This second filter has been included because the same UHF band is largely used to telecommand CubeSats. Therefore, to mitigate any possible interference, we have decided to reduce the bandwidth as maximum as possible. As the filter is applied after the LNA, its noise figure is also compensated. The combination of all these components provides a reception chain with 2.5 dB of noise figure, and 12.5 dB of gain. The transmission chain aims at amplifying the signal from the transceiver to overcome the path loss due to the propagation of the signal in the communications medium. For that reason, a PA is used to provide a maximum outcome signal of 31.2 dBm at 437.35 MHz. A Power Meter (PM) is included in the design to measure the transmission power of the RF-ISL board. This measurement is included as part of the board telemetry, and it is crucial to verify the communications performance.

As indicated previously, the RF-ISL board provides a half-duplex interface between the balloons. This communications mode is characterized by not being able to receive packets while the device is transmitting. Two causes have conditioned the half-duplex design in the RF-ISL board: (1) the CC1101 transceiver can only work in this mode, and—the more important—(2) the inclusion of a wireless full-duplex interface would increase the complexity of the design, and with it the risk to component failure. Two switches at the boundaries of the chains have been included in the design to select one of the two modes—transmission or reception. By means of a dedicated General-Purpose Input/Output (GPIO), the microcontroller can select the mode of the RF circuit before forwarding the message to the transceiver. These switches, however, have 0.3 dB of insertion loss which needs to be taken in consideration to ensure that the link budget is feasible. One of the switches is located just after the transceiver, while the other one is just before the antenna. This antenna is a monopole that ensures an omnidirectional radiation pattern. This pattern has been selected due to the uncertainty of the balloon trajectory, and the spin that they may experience. Instead of using a directive antenna that could enhance the communications range, an omnidirectional antenna has been preferred to dissociate the connectivity to the spin of the balloon. The selected antenna provides 1.2 dB of gain.

7.6.2 Software components of the RF-ISL board

The central component of the RF-ISL board is an ARM Cortex microcontroller, which executes the different software elements necessary to establish the ISL between the balloons (Figure 7.15). These elements are executed over the Free Real-Time Operating System (FreeRTOS) which is a real-time framework with different features to manage the threads interruptions. The design of the RF-ISL software is composed of multiple threads being each one responsible of a specific functionality. The `UartInterface` thread manages the interaction with the OBC of the FSS-Exp payload exchanging different packets following the KISS protocol. As previously presented, these packets contain commands to indicate which action should be performed in the RF-ISL board. Commands to transmit a packet, to check if a packet is received, or to retrieve the housekeeping of the board are implemented. After identifying the command, the `UartInterface` thread notifies the corresponding thread to process it.

The `ControlTask` thread processes the request to retrieve the housekeeping of the board by refreshing this information. The board housekeeping is composed of multiple parameters that represents the behavior of the board. Specifically, two temperature measurements are retrieved using an I2C connection with a temperature sensor, and a transmission power measurement is performed directly from an Analog-to-Digital Converter (ADC) pin. Furthermore, the main parameters of the transceiver are included as part of the board housekeeping, such as the Received Signal Strength Indication (RSSI). After refreshing the corresponding values, the `ControlTask` thread stores the housekeeping item in the `HousekeepingBuffer`, which is suddenly retrieved by the `UartInterface` thread.

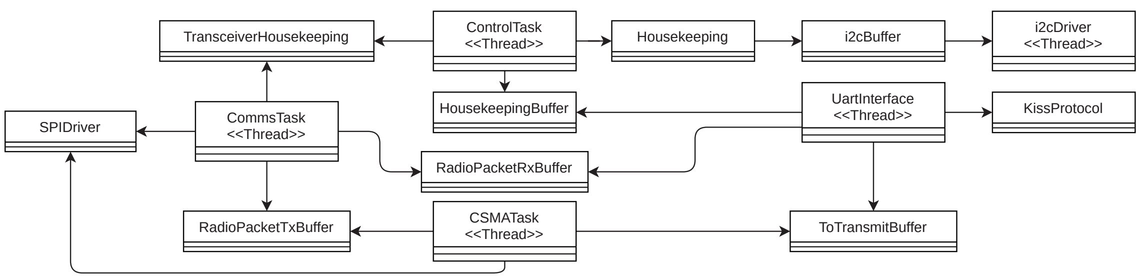


Figure 7.15: UML representation of the different software elements executed in the RF-ISL board

The CSMATask thread performs the algorithm to manage the access to the communications medium. In particular, it executes the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) medium access technique ('IEEE 802.11 Wireless Access Method and Physical Layer Specification') to mitigate the collision of transmitted packets. If the medium is not being used, this thread forwards the packet stored in the ToTransmitBuffer to the RadioPacketTxBuffer. All the packets in the RadioPacketTxBuffer are processed by the CommsTask thread, which accepts packets with a maximum length of 3285 bytes. After this, it performs a Reed-Solomon (RS) coding for erasure channels (Lacan et al., 2009) which enables to overcome the effect of error burst or losing an entire packet. This technique is implemented with the OpenFEC libraries, and it splits the incoming packet to a set of 223 byte fragments (i.e. K fragments). From these fragments, it generates another set which works as redundancy (i.e. R fragments). From the total transmitted fragments ($K + R$), it is just necessary to recover a set of them (K) to reconstruct the original and upper packet. This mechanism enables to overcome the loss of entire packets due to long fadings. Additionally, for each of those fragments, 32 parity bytes are generated with a coding rate of 255/223. This coding technique detects erroneous bits, and try to correct them. In the FSSExp case, however, the fragmentation is not performed, because all the packets are smaller than 223 bytes. Nevertheless, the 32 bytes of parity are computed as expected. After encoding the packet, the thread forwards it to the transceiver using a Serial Peripheral Interface (SPI). As previously presented, the C1101 transceiver has a firmware which enables to perform different functionalities. Apart of applying a specific modulation, this firmware performs a packet encapsulation which includes 4 bytes as a packet preamble, and 4 bytes more for packet synchronization. This encapsulation is removed when the packet is received in the remote transceiver. Figure 7.16 clarifies the different levels of encapsulations that a packet has in the RF-ISL board. Moreover, the CommsTask thread constantly checks if a packet has been received in the transceiver. If it the case, the thread decodes the packet applying the RS algorithm, and stores it in the RadioPacketRxBuffer. Therefore, the UartInterface thread checks this buffer while processing a command to evaluate if a packet is received. In the case that the buffer is not empty, the thread forwards the packet to the OBC of the FSSExp payload using again the KISS protocol.

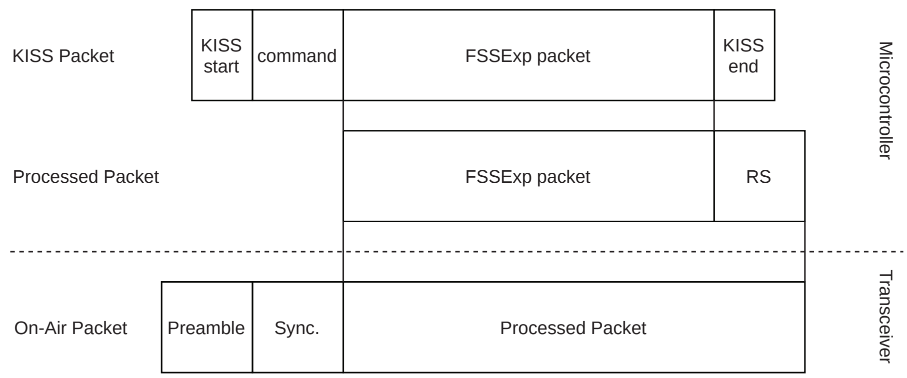


Figure 7.16: Encapsulation of messages that are received from the OBC until they are transmitted as packets by the RF-ISL board.

7.6.3 Link Budget definition

As part of a preliminary analysis before the stratospheric balloon campaign, a link budget between two RF-ISL boards is performed. Taking into consideration the characteristics of the RF-ISL board components, Table 7.1 presents the link budget between two RF-ISL boards communicating inside the stratospheric balloons. This link budget has been defined in the most demanding scenario, which is receiving at the minimum SNR required to ensure a Bit Error

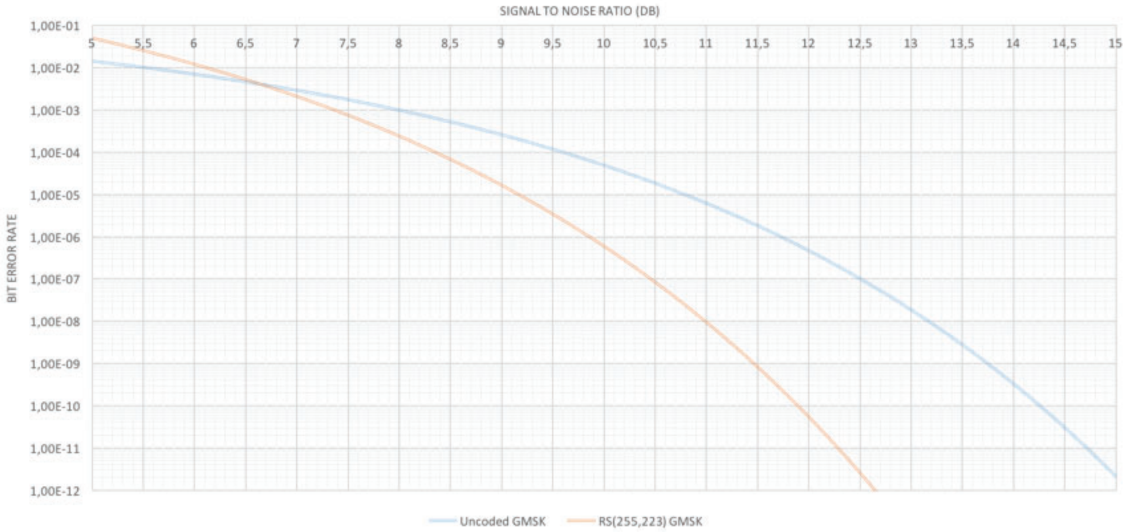


Figure 7.17: BER and SNR relationship for uncoded GMSK modulation, and with RS coding technique

Rate (BER) of 10^{-5}. Figure 7.17 presents the relationship between the BER and the SNR for this scenario. In this case, the minimum SNR required for an uncoded signal modulated with GMSK modulation is 10.8 dB (blue line in the figure). The use of the RS 255/223 coding provides 1.6 dB of gain, decreasing the required SNR to 9.2 dB (orange line).

Table 7.1: Link budget between two RF-ISL boards in balloons

Parameter	Close case	Unit
Frequency	437.35	MHz
Bandwidth	58	kHz
Transmission power	31.2	dBm
Antenna Gain	1.2	dB
EIRP	33.4	dBm
Modulation	GMSK	
Reed-Solomon Coding Rate	255/223	
Path Distance	400	km
Free Space Loss	137.3	dB
Received Antenna Gain	1.2	dB
Antenna Misalignment Loss	3	dB
Antenna Temperature	2900	K
Receiver System Temperature	228.1	K
Receiver System Gain	12.5	dB
Received Power at the antenna	-106.7	dBm
Noise Floor in transceiver	-116.0	dBm
Received SNR	9.3	dB
Required SNR for GMSK	10.8	dB
Required SNR with RS	9.2	dB
Link Margin	0.1	dB

Furthermore, the FSPL model (Friis, 1946) is used to compute the loss due to the signal propagation at 437.35 MHz. Complementing the propagation loss, a misalignment of the antennas is considered as part of the worst case scenario, generating 3 dB of loss. Moreover, the antenna temperature is determined using the P.372 recommendations of the ITU-R (Weinmann and Dostert, 2006). Considering that the launch pad of the balloons is located in a rural area close to a village, 2900 K of antenna temperature is considered. The link budget indicates that two stratospheric balloons can establish a communications link when the distance between them is less or equal to 400 km. These communications are established at 9.6 kbps transmission data rate. Considering the different bytes of header encapsulation and the size of the data packet of the FSSExp payload, the maximum throughput using the RF-ISL board is 5.2 kbps.

7.7 Stratospheric Balloon Campaign Results

As previously presented, the stratospheric balloon campaign aims at evaluating the design of the FSSExp payload, and in particular the FeDeCoP and the OSADP, in a realistic environment. This campaign becomes thus a proof-of-concept of the FSS and the IoSat paradigms. In order to achieve a scenario with variable connectivity, three stratospheric balloons are equipped with a FSSExp payload. The structural envelope with the different subsystems are integrated inside the structure of the balloon, while the monopole antenna is placed outside. Fadings due to balloon structure opacity are mitigated with this configuration. An additional FSSExp payload was implemented to work as part of the ground station.

The stratospheric balloons were launched in the Yaroslavl Oblast federal subject of Russia (north region of Moscow). Specifically, the coordinates in which the launch pad was located were 56.62ºN, 38.68ºE. The launch started at 12:12 on December 14^{th}, 2019. Figure 7.18 shows a picture of the launch pad just before starting to launch the stratospheric balloons. The launch was performed sequentially, keeping 5 minutes between each balloon rise. The wind speed on the ground was 18 km/h to the North, generating an initial distance between the balloons of 1.5 km. This distance enabled to evaluate the benefits of deploying federations to communicate with those balloons that are no longer in line-of-sight.

Figure 7.18: Launch pad with the three balloons ready to be released

(a) Balloon A without one external panel

(b) Balloon B already integrated

(c) Balloon C at the end of the campaign

Figure 7.19: Pictures of the three stratospheric balloons of the campaign

The balloons are identified in the following results according to its position in the launch sequence: Balloon A is the first one, Balloon B the second one, and Balloon C the last one. The three balloons present some structural differences that drive their trajectories. Therefore, the initial distance between the balloons was not kept during the entire flight. Figure 7.19 presents three pictures of the stratospheric balloons.

The ground station was installed in the same launch pad with a dedicated power supply system. In addition, the ground station was equipped with a computer that was used to transmit telecommands, and receive data from the balloons. This computer uses the FSSExp payload as a RF interface necessary to interact with the balloons. Two antennas were used for the ground station: a monopole antenna, and a coplanar Yagi antenna. The monopole was used in the former stages of the campaign, when the balloons are released and they are relatively close. The Yagi antenna was used as a backup solution if the communications of the balloons could not be established.

The campaign was successful, being able to deploy numerous federations, and to retrieve relevant metrics that represent the performance of the FSSExp payload. Additionally, the retrieved results are complemented with a simulation engine that is able to predict accurately the flight trajectory (Briatore et al., 2017). This section presents these results retrieved during the entire launch campaign. Section 7.7.1 evaluates the metrics related to the balloon flight, and the environment that the FSSExp payload was suffering. Then, Section 7.7.2 presents the first results retrieved with the interaction between a flying balloon, and the ground station. Additionally, it evaluates how the balloon flight dynamics can impact the communications performance. Finally, Sections 7.7.3, 7.7.4, 7.7.5 present the characteristics of four different federations established during the campaign.

7.7.1 Flight Characteristics and Environment

The stratospheric balloons are conceived with the necessary components to embed the payloads, and to ensure their correct operations during the entire flight. Furthermore, they are equipped with multiple sensors that help to characterize the flight conditions. Specifically, two sensors provide the internal, and the external temperatures of the balloon, while a barometer measures the pressure at different layers of the atmosphere. A Global Position System (GPS) receiver is also integrated to track the balloon position and altitude. However, this receiver follows the Coordinating Committee for Multilateral Export Controls (COCOM) regulations, which limits the correct operation of this device below 18 km of altitude. This limitation generated a gap of the retrieved data from the balloons which corresponds to this altitude boundary. Neverthe-

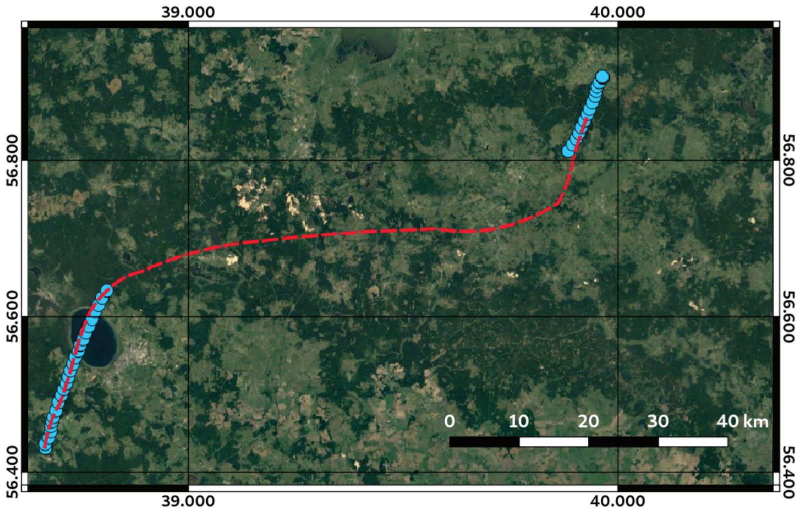
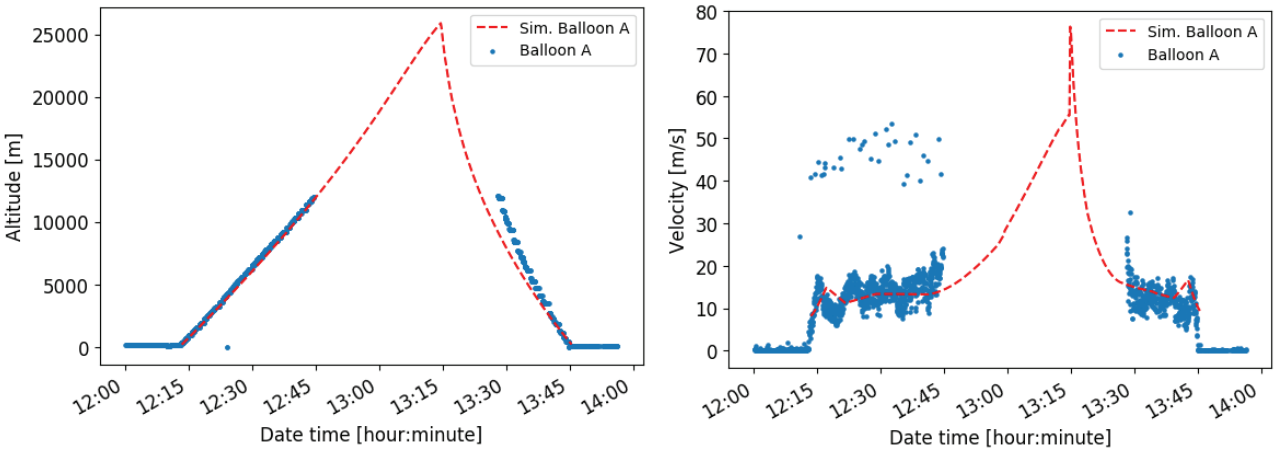


Figure 7.20: Balloon A ground track with the GPS measurements (blue dots), and the simulated trajectory (blue line).

less, the retrieved measurements have been complemented with the simulation of the balloon trajectory. As the following results demonstrate, the simulated trajectory is in good agreement with the measurements, stating that this trajectory is representative of the reality.

The results presented in this section are computed with the retrieved data of Balloon A platform. The data was collected after the recovery of the balloon, after it landed. The balloon was released at 12:12, and it landed at 13:45, completing a flight of 1 hour and 33 minutes. Figure 7.20 presents the ground track of the balloon represented over a map of the region. The flight started at the coordinates 56.62ºN, 38.68ºE, and it finished at the coordinates 57.10ºN, 39.96ºE. The blue dots presented in the figure corresponds to the GPS measurements retrieved during the flight, and the red dashed line the position of the simulated trajectory. The simulation matches the beginning and ending of the trajectory, serving as boundary conditions for a valid estimation of the remaining positions. Considering a straight line, the balloon landed 80 km (roughly) away from the launch pad.



(a) Altitude profile of the entire campaign.

(b) Velocity profile of the entire campaign.

Figure 7.21: Balloon A altitude (left) and velocity (right) with the measurements (blue dots), and the simulated trajectory (red dashed line).

The trajectory is determined by the different wind speed experienced in each altitude. The representation of the altitude profile provides another view of the balloon trajectory. Figure 7.21a plots this profile with the combination of GPS measurements (blue dots), and the simulated trajectory (red dashed line). As in the previous case, the simulated trajectory is coherent with the measurements. For that reason, the maximum altitude that Balloon A experienced can be estimated to 25906 meters. At this altitude, the helium balloon exploded, and Balloon A started to descend. Thi explosion happened at around 13:15, just after 57 minutes of its launch. The helium balloon explosion represents a singularity moment, after which the trajectory changes. This situation determines two phases of the trajectory: the ascending phase, and the descending one. The former phase is driven by the dynamics generated with the helium balloon, the mass structure, and the environment. The balloon rise follows at the beginning a slow and constant elevation. On the other hand, the descending phase is faster, requiring just 30 minutes to land. In this phase the trajectory is driven by the parachute, which is opened after the explosion.

These two phases can also be characterized with the velocity profile of the balloons, as shown in Figure 7.21b. This velocity is computed with the transformation of the GPS coordinates from the Geodetic Coordinate Reference System (CRS) (Mueller, 1985) to the Earth-Centered, Earth-Fixed (ECEF) CRS (Cai et al., 2011). In this case, the velocity profile shows the ascension phase starts with a speed of 12 m/s, which increases during the elevation. Again, when the helium balloon explodes, the velocity experiences a considerable growth up to 76.3 m/s. This growth is determined by the lapse of time that the balloon is in free fall and the parachute has not been opened yet. Once opened, the balloon velocity decreases considerably reaching again the 12 m/s. With this velocity, Balloon A performs a stable descent until it lands.

The ambient temperature fluctuates considerably during the flight, being able to reach negative values. The performance of the FSSExp payload, and in particular the RF-ISL board, can be impacted by these unusual temperatures. For that reason, Figure 7.22 presents the temperature profile during the entire flight. The temperature outside the balloon (red line) remains bellow de 0 ºC during the entire flight, reaching its (measured) minimum value of -35.2 ºC close to 13:30 (descending phase). Although outside of the balloon the temperature is extreme, the internal one remains stable between 15 and 20 ºC (green line). This stability is kept in all the locations inside the balloon, as the two temperature sensors in the FSSExp payload highlight (blue and orange lines). This temperature profile ensures that the components of the FSSExp payload work in its operational temperature range, being able to communicate with the other systems.

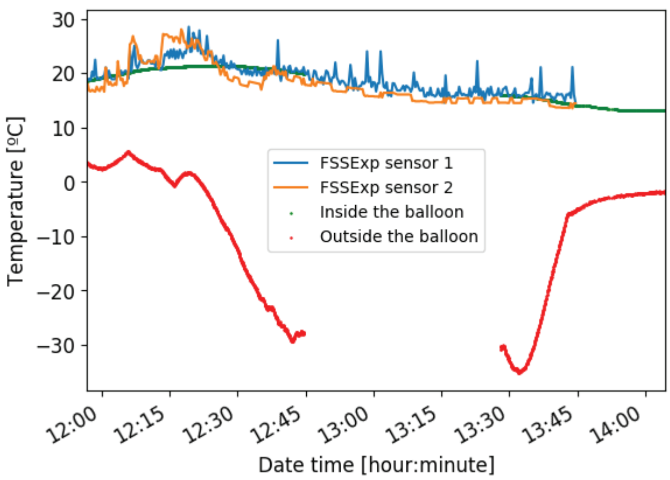


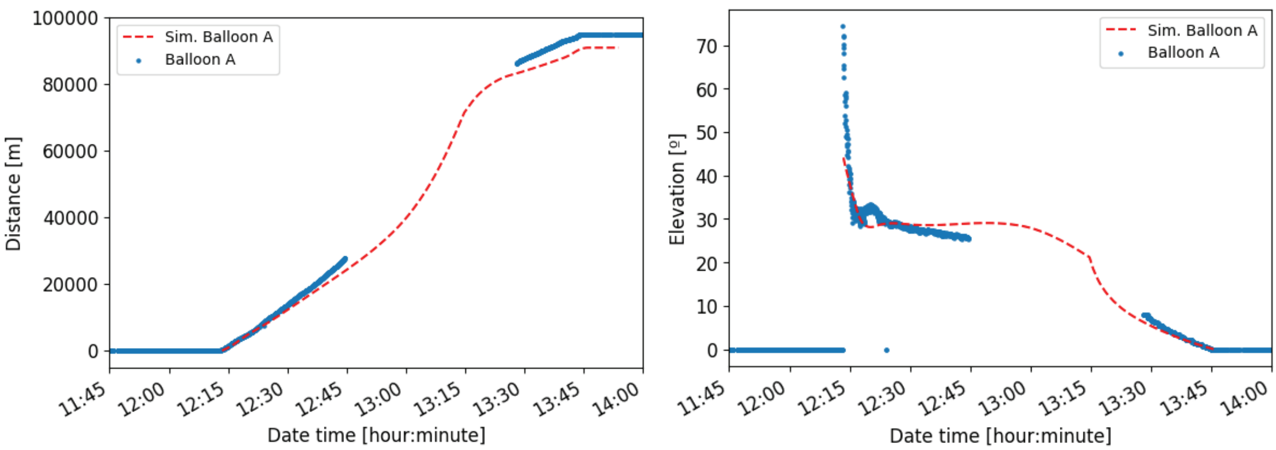
Figure 7.22: Temperature profiles during the balloon flight.

7.7.2 Communications Conditions

Since the release of the balloons, they start to separate of the ground station. The distance between the balloons, and the ground station is increased over time. Figure 7.23a presents this distance evolution between Balloon A and the ground station. This evolution follows a logarithmic curve, increasing until it reaches a maximum value, which corresponds to the moment that the balloon lands. This maximum distance is reached at 13:45, and it corresponds to 94986 meters. Note that this curve does not have a constant gradient, because the velocity profile does not remain constant. Considering the link budget presented in Table 7.1, the ground station should be able to communicate with the balloon during the entire flight. However, there are other fundamental aspects that impact this behavior.

As previously highlighted, the interaction between the nodes is feasible only if the SNR of the received signal is greater than the minimum required. This ratio is determined by the received signal power, and the noise floor in the receiver. In the laboratory, the noise floor measured with the RSSI from the FSSExp payload, and connecting a matched load as an antenna was -113.0 dBm (average). As presented in the link budget in Table 7.1, the expected noise floor due to the human activity is -104.7 dBm. However, the noise floor in the launch pad was -85.0 dBm (average), 19.7 dB greater than the expected one. Figure 7.24 presents the difference of these three noise floors. The blue line corresponds to the measured noise floor during the entire flight, the red line corresponds to the measure noise floor in the laboratory, and the green line corresponds to the estimated one in the launch pad. This increase generates an antenna temperature of 2.9×10^5 K, two order of magnitude greater than the one presented in the ITU-R recommendation. The existence of an interference signal is the cause of this increase. The link budget is thus impacted by reducing the maximum communications range to 55 km. Reviewing the distance profile in Figure 7.23a this implies that at certain moment between 13:00 and 13:15 the communications between the ground station and a balloon is no longer feasible.

Furthermore, there are another metrics that can impact the link budget. The elevation angle between the balloon and the ground station can also limit the communications. Figure 7.23b presents the elevation angle during the balloon flight. When the elevation angle is below 10 degrees, the communications becomes really impacted, being difficult to interact due to the ob-



(a) Distance profile of the entire campaign.

(b) Elevation profile of the entire campaign.

Figure 7.23: Balloon A distance (left) and elevation (right) with the measurements (blue dots), and the simulated trajectory (red dashed line).

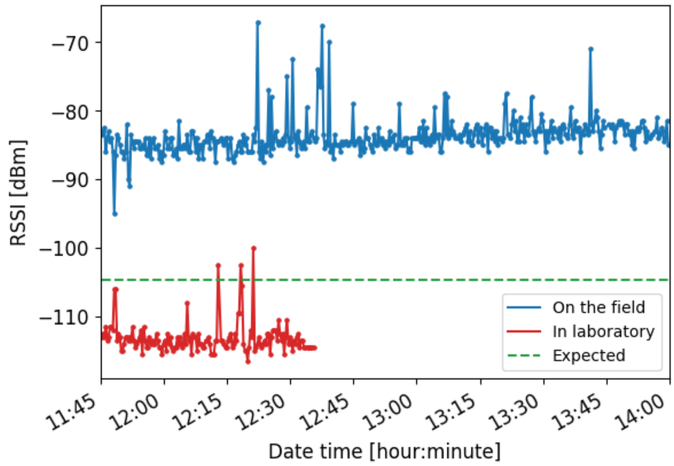


Figure 7.24: RSSI measurement of the noise floor on the launch pad (blue line), the expected one (green line), and in the laboratory (red line)

jects placed in the path. As the previous figure highlights, this condition is achieved after the 13:15. As a conclusion, the element that most impacts the link budget is the presence of an interference signal, which breaks the communications between the balloon and the ground station after 13:00 h.

The velocity of the balloon can also deteriorate the communications with the ground station. This velocity determines the shift that the frequency is experienced in reception due to the Doppler effect (Possel, 2017). If this shift is not properly compensated, the received signal can be placed outside the working band, and thus lost. Figure 7.25 presents this frequency shift according to the velocity profile of the balloon. The shift remains bellow the 120 Hz during all the flight. This shift can be compensated by the same transceiver of the RF-ISL board, which is able to manage frequency shifts up to a fourth of the bandwidth (i.e. 14.5 kHz). Therefore, the Doppler frequency shift does not impact the communications performance in this scenario.

At this point, the estimations and the measurements demonstrate that although the increase of the noise floor, the communications with the balloon and the ground station is feasible

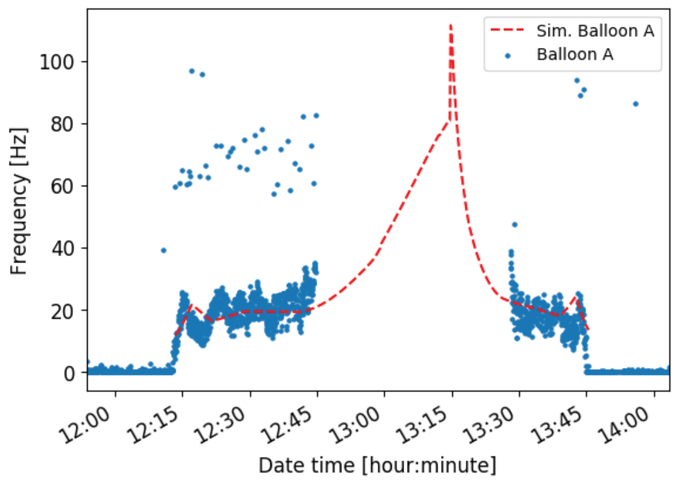


Figure 7.25: Doppler frequency shift with the measurements (blue dots), and the simulated trajectory (red dashed line)

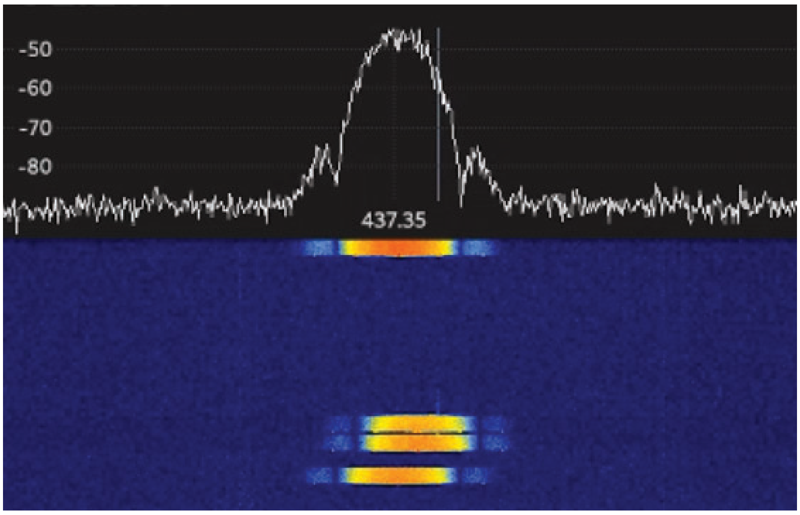
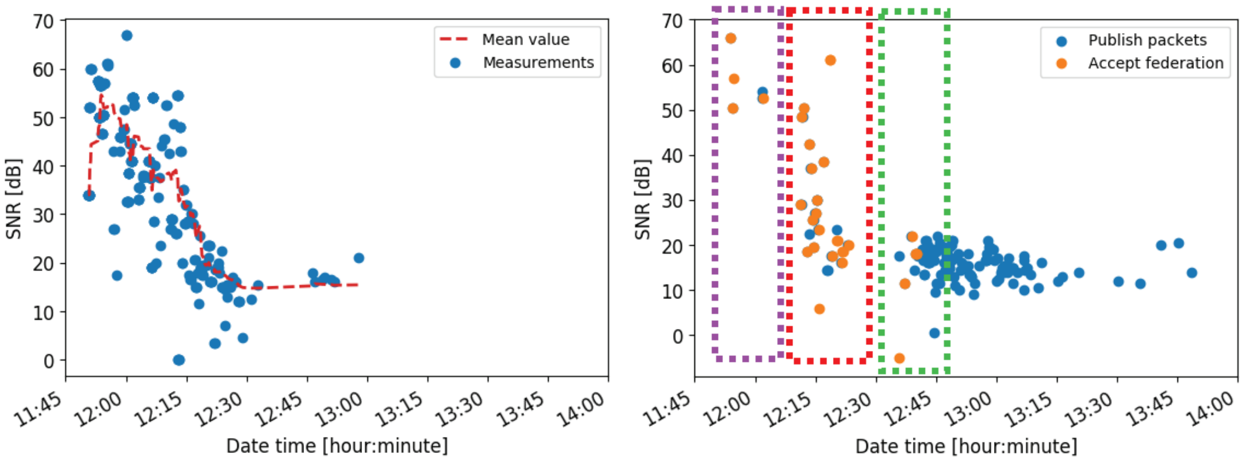


Figure 7.26: Signal sniffed from the ground station transmitted by two different stratospheric balloons during a federation

during almost all the ascending phase. In order to take some measurements of the received signals, a Software Defined Radio (SDR) module was installed near to the ground station. The goal of this module is to measure the frequency spectrum in order to sniff the signals that are being exchanged during the execution of the federations. Figure 7.26 presents a short fragment of the spectrum history retrieved with this device while the balloons were flying. At the top of the figure, the spectrum of the current received signal is plotted. The presented curve shows the shape of a GMSK modulated signal centered at 437.35 MHz. Additionally, at the bottom of the figure, a history of the spectrum is presented over time. A color scale is used in this history to highlight the power of the signal, being orange the great values, and blue the lowest ones. This figure presents the reception of four packets. Note that two of them are centered in the corresponding frequency, and other two are shifted in frequency. As previously indicated, this is not the consequence of the Doppler effect, because the transceiver can compensate it. Instead of this, the shift is mainly caused by the drift in the clock of the RF-ISL board. Specifically, the frequency shift is 13 kHz, which corresponds to 29.7 ppm. This shift does not impact the communications, because the transceiver can compensate up to 14.5 kHz.

The historic of all the sniffed packets in the ground station enables to evaluate the impact of the separation of the balloons in the communications performance. Figure 7.27a shows the evolution of the SNR corresponding to all the received signals from Balloon A. This value decreases coherently over time, because the distance increases. Before the release of the balloon, the SNR remains around 40 dB. After they take off, the SNR rapidly goes down until 16 dB (around 12:50). After this moment, the communications between the balloons and the ground station is lost. Note that this situation is coherent with the degradation of the link budget due to an interference signal, which determines that the communications should be lost after 13:00. Although the communications with the ground station is lost, the balloons kept interacting between them as Figure 7.27b highlights. This figure presents the SNR of the received packets in the three balloons that indicate the establishment of a federation: the `publish` (blue dots), and the `accept` packets (orange dots). Three different time periods are identified depending on the position of the balloons, and which nodes interacts in the federation. The former period (purple box) is delimited between 11:53 and 12:12 in which different federations are established having the balloons placed on the ground. Although the dynamics are not the flying ones, this situation enables to generate a reference performance. The following period (red box) is between 12:12 and 12:30,



(a) SNR of the sniffed packets in the ground station.

(b) SNR of publish (blue dots) and accept (orange dots) packets.

Figure 7.27: SNR of all the received packets in the ground station (left), and those related to the establishment of the federations. The three regions (purple, red, and green dashed lines) are identified according to the configuration of the balloons.

when the balloons are released sequentially. As it is presented in the next section, federations in which a balloon located on the ground can forward data through a flying balloon are established. Finally, the period between 12:30 and 12:45 (green box) shows federations established autonomously between the balloons to share storage capacity. After this moment, no longer federations were established. This situation happened because the balloons does not require more storage capacity, and they just started to publish the availability of the service. The following section presents the details of the federations established during each time period.

7.7.3 Federations on the ground

When the balloons are still on the ground, multiple federations were established to verify the correct operation of the payloads. Among all of them, Figure 7.28 presents the packet sequence of a federation. This sequence represents the different packets exchanged between the balloons and the ground station properly scheduled on time. Moreover, this representation identifies the types of the packets using different color and shape markers. For instance, all the blue markers correspond to packets exchanged in a downlink connection, while the red markers identify those exchanged in a federation. The rhombus (⋄) and the times (×) markers represents respectively the packets to establish and close a connection (i.e. a downlink or a federation). In the case of a federation, these are the `request`, the `accept`, and the `close` packets. The dot (∘) and the triangle (△) markers corresponds respectively to data and acknowledgement packets. Moreover, the packet that corresponds to a retransmission is represented with a "R" below the marker. Finally, the green markers corresponds to publications of the service. This kind of representation helps to understand the periods when a downlink or a federation is established with a dashed line of the corresponding color. It provides thus a global view of the scenario, and the different packets that are exchanged.

Returning to Figure 7.28, a federation between Balloon C and Balloon B is established in this lapse of time that goes from 12:01:00 to 12:02:30. At the very beginning, the ground station opens a downlink connection with Balloon C, which starts to download its own data during 50 seconds. At this moment, the balloon has downloaded all its data, and start to publish the

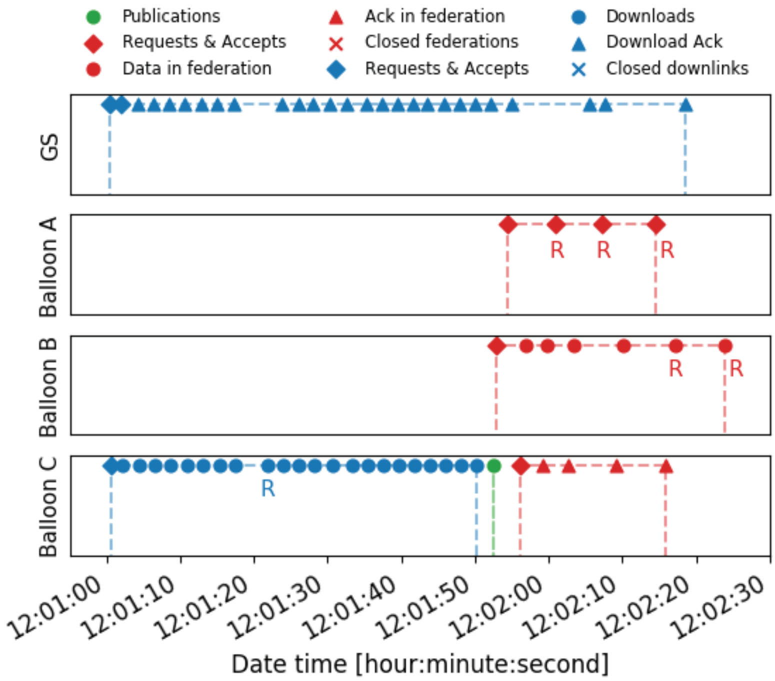


Figure 7.28: Packet sequence of federations established on ground

availability of the download service using the OSADP definition. Balloon B and Balloon A receives this publication, and decides to request the service to download its data. Balloon B reacts faster than Balloon A, being able to establish the federation with Balloon C. While Balloon A does not receive any `accept` packet, it retransmits the `request` packet in case that the original one could be lost. After three retransmissions, it stops. Meanwhile, Balloon B transmits data to Balloon C which downloads them to the ground station. As part of the test, Balloon C is turned off during the federation (at 12:02:16). After that, Balloon B performs different retransmissions of the last data packet because it has not received its acknowledgement. At 12:02:23, it decides that the connection is lost, and closes the federation internally. Figure 7.29 shows a graphical representation of this federation.

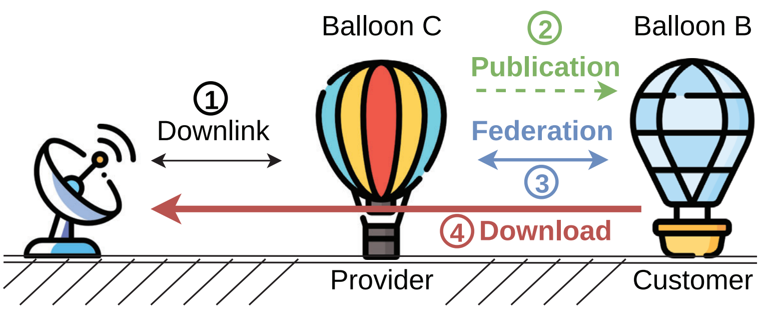


Figure 7.29: Representation of the federation established while balloons placed on ground. The numbers correspond to the time sequence when each action is performed (from low values to greater ones).

This federation enabled to download up to four federation packets from Balloon B. Figure 7.30 presents the memory usage of Balloon B and Balloon C (in data packet unit). Note that the buffer usage is part of the data retrieved from the balloons, and it is generated every 20 seconds. Therefore, the resolution in this representation is not the same than in the packet sequence. When Balloon C establishes the downlink connection, it reduces its buffer usage to zero pack-

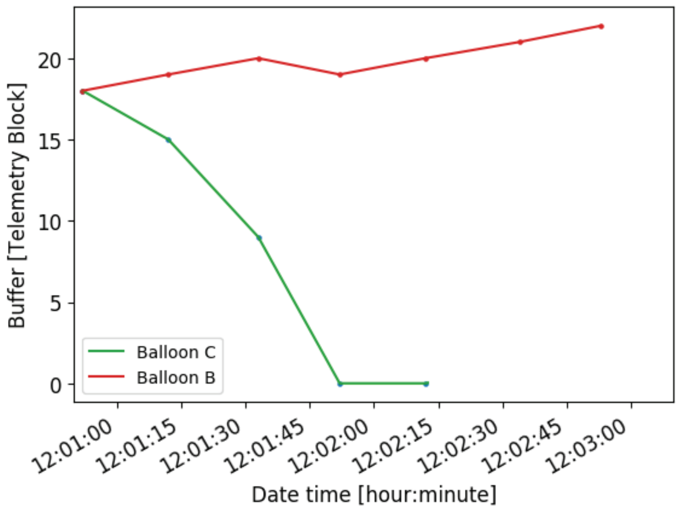


Figure 7.30: Buffer usage of Balloon C (green) and Balloon B (red) in data packet unit

ets by downloading them to the ground station. When the federation is established, Balloon B reduces its usage by using the federation. Once the federation is closed, Balloon B keeps producing new data packets, and thus increasing the usage of the buffer. This federation enables to download 4 telemetry packets during 33.3 seconds. The throughput achieved with the federation is 184.7 bps. At the same time, Balloon C downloads 22 data packets, achieving 756.9 bps of throughput. Note that the throughput values achieved are much smaller than the theoretical one (i.e. 2.6 kbps), presented in Section 7.6.3. The averaged Round Trip Time (RTT) of the different packets drives the communications throughput. This metric corresponds to the lapse of time that a data packet needs to be downloaded and acknowledged in order to transmit a new one.

The quality of the channel can provoke the loss of packets, and thus their retransmission. This retransmission increases the effective RTT, because a Recovery Time Objective (RTO) is spend for each lost packet. Therefore, the worse channel quality, the more packet retransmissions, and thus greater the RTT is. The quality of the channel can be measured with the PDR metric, which corresponds to the ratio between the number of received packets and the transmitted ones. The RTO value becomes important when lower is the PDR. In this campaign, the RTO was set to 6 seconds. Moreover, the RTT is impacted by different features of the software implementation. The software has not been implemented with real-time thread, neither thread notifications. This makes that the threads periodically check if a packet is received, which can provoke that a received packet remains in the internal queue until a thread wakes up. In the FSSExp payload case, the *PacketDispatcher* thread has 50 ms of execution period, and the other threads 500 ms. This makes that the maximum time that a packet remains in the software until a reply is sent becomes 600 ms. However, the threads can be synchronized, and they can process the incoming packet sequentially, without spending time in the queue (it is negligible). Therefore, a maximum and minimum RTT, and thus the equivalent throughput, can be defined according to those two scenarios. Considering the stop and wait ARQ mechanism of the FeDeCoP and the downlink, Table 7.2 presents the maximum and minimum throughput with respect to the PDR that can be achieved in this campaign for each type of communications: direct download, or using a federation. In the case of the downlink, the minimum RTT is determined by the number of transmitted packets (n), its corresponding transmission time packet (t_{tx}), the number of retransmissions performed (r), and the number of packets correctly transmitted and received

($c = n - r$):

$$RTT_{min} = \frac{n \cdot t_{tx} + r \cdot RTO}{c} \tag{7.1}$$

Note that the FSSExp payload always transmit a packet of 255 bytes, by filling those that are smaller. Therefore, the transmission time is the same for all the packet types. After different measurements, the transmission time of a packet is 425 ms. In the same communications, the maximum RTT is determined by the number of transmitted packets (n), its corresponding transmission time packet (t_{tx}), the number of retransmissions performed (r), and the processing time (t_p) in each node per packet correctly transmitted (c):

$$RTT_{min} = \frac{n \cdot t_{tx} + r \cdot RTO + c \cdot t_p}{c} \tag{7.2}$$

where t_p is 600 ms according to the implementation of the FSSExp payload. The throughput (S) is the amount of bits (l) downloaded in the corresponding RTT:

$$S_{min} = \frac{l}{RTT_{max}} \; ; \; S_{max} = \frac{l}{RTT_{min}} \tag{7.3}$$

Returning to the throughput achieved in the scenario presented in Figure 7.28, both results can be compared with throughput reference presented in Table 7.2. In this scenario, the

Table 7.2: Theoretical throughputs (S)—maximum, minimum, and average—of downlink and federation transmissions with respect to the PDR

PDR	Max. RTT [ms]	Min. RTT [ms]	Max. S [bps]	Average S [bps]	Min. S [bps]
Downlink					
100.00 %	2050.00	850.00	1807.06	1278.17	749.27
66.67 %	8475.00	7275.00	211.13	196.19	181.24
50.00 %	14900.00	13700.00	112.12	107.61	103.09
40.00 %	21325.00	20125.00	76.32	74.18	72.03
33.33 %	27750.00	26550.00	57.85	56.6	55.35
28.57 %	34175.00	32975.00	46.58	45.77	44.95
25.00 %	40600.00	39400.00	38.98	38.41	37.83
Federation					
100.00 %	4100.00	1700.00	903.53	639.08	374.63
80.00 %	10525.00	8125.00	189.05	167.50	145.94
66.67 %	16950.00	14550.00	105.57	98.10	90.62
57.14 %	23375.00	20975.00	73.23	69.47	65.71
50.00 %	29800.00	27400.00	56.06	53.80	51.54
44.44 %	36225.00	33825.00	45.41	43.91	42.40
40.00 %	42650.00	40250.00	38.16	37.09	36.01
36.36 %	49075.00	46675.00	32.91	32.11	31.30
33.33 %	55500.00	53100.00	28.93	28.31	27.68
30.77 %	61925.00	59525.00	25.80	25.30	24.80
28.57 %	68350.00	65950.00	23.29	22.88	22.47
26.67 %	74775.00	72375.00	21.22	20.88	20.54
25.00 %	81200.00	78800.00	19.49	19.21	18.92

downlink communication has a 95.7 % of PDR, and the federation one has 80.0 %. The throughput achieved corresponds to 756.9 bps, and 184.7 bps respectively, which is equivalent to the 41.9 % and the 97.7 % of the maximum throughput wit this PDR (see Table 7.2). Although these throughput results are impacted by the fact that Balloon C was turned off while federated, the difference between each type of communications appears also in other federations. This difference is mainly determined by the different time needed to directly download data, and by using the federation. Figure 7.31 presents a boxplot[18] of the communication latency experienced in the packets directly downloaded, and the ones over a federation. The packets that are directly downloaded have a median latency of 1825 ms (roughly), while the federated ones have 6300 ms. This increase is mainly due to forwarding the packet over Balloon C, which also needs time to download it to the ground station. Therefore, always the packets that are downloaded over a federation suffer for a longer latency time.

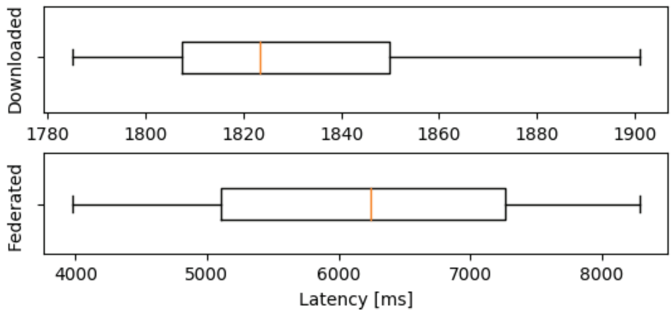


Figure 7.31: Latency of the downloaded packets (top) and the ones using a federation (bottom) in on-ground federations

These results demonstrated that the federations can be established with the OSADP, and FeDeCoP proposals. Additionally, these protocols can manage the disruption of the connection. The following section presents the performance of these protocols as performed in this one, but having the balloons flying in different configurations.

7.7.4 Federations while releasing the balloons

Other federations are established while the balloons started to lift from ground. Figure 7.32 presents the packet sequence of a set of them when Balloon A is raising. In this case, the customer of the federation is Balloon B which request up to eight federations in more than three minutes. Specifically, the ground station opens the downlink connection with Balloon A at 12:12:46, just after its release. The balloon starts to download its data, until it empties the buffer. Then, it publishes the service to the others. Balloon B requests the service at 12:12:54, which is suddenly accepted by Balloon A, starting the first federation. Balloon B downloads its data using the federation, as Balloon C performed when the federation was established on ground. This former federation remains opened until Balloon B empties also its buffer (at 12:13:20). At this moment, the federation is no longer needed, and Balloon B starts to close it by sending the corresponding `close` packet. After the correct three packet exchange, the federation is closed, and Balloon A starts to publish the service again (because the downlink is still opened). Note that Balloon A keeps downloading its own data while it is publishing the service.

At 12:13:32, Balloon B requests again the service of Balloon A, because it has generated an additional data packet. After downloading the packet with the federation, Balloon B starts again

[18] A boxplot represents a distribution function as a box delimited by the third quartile and the first quartile. Additionally, the median is represented as a line in this box.

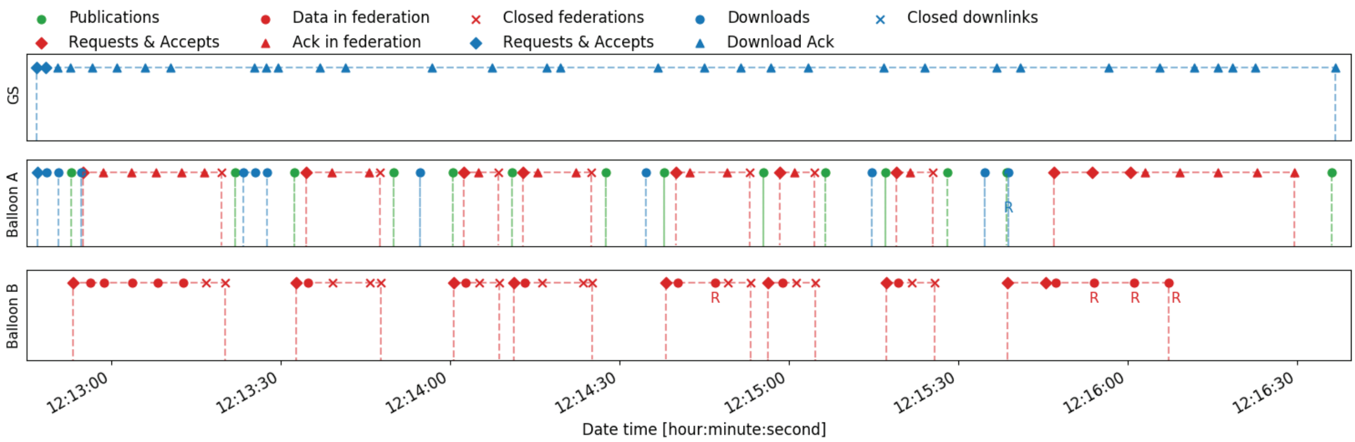


Figure 7.32: Sequence of packets exchanged in federations while the balloons start to be released

the process to close it. This situation is repeated during the following five federations. Note that during all these federations, Balloon B is placed on ground, while Balloon A is flying up. This means that Balloon B can download data to the ground station using the relay of Balloon A which is indeed over both nodes, as Figure 7.33 shows. From the first federation until the last one, Balloon A goes from 190 meters to 1517 meters with an elevation close to 90 º. Considering the position of the monopole antennas, this high elevation impacts the communications between the balloons, provoking some retransmissions. It is the case of the last federation, which indeed the connection is not totally established. In particular, Balloon B requests the service, that Balloon A accepts, and the first packet to be downloaded is retransmitted three times, because Balloon B does not receive the acknowledgement from Balloon A. Although Balloon A indeed transmits this acknowledgement, this situation ends by Balloon B closing the federation because it considers that the link is lost. This also demonstrates that some links between the balloons were asymmetric due to the RF environment.

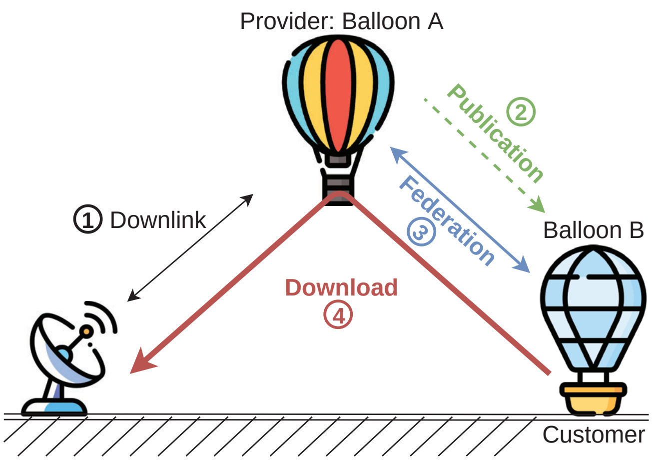


Figure 7.33: Representation of the federation established while balloons being released. The numbers correspond to the time sequence when each action is performed (from low values to greater ones).

The execution of these federations can also be interpret with the buffer usage of the balloons, as Figure 7.34 presents. When the downlink is established, Balloon A downloads all its data, and empties its buffer. Then, Balloon B can also empty its buffer thanks to the federation with Balloon A. After that, both buffers remain empty in the retrieving of the housekeeping. As previously indicated, this is due to the period value of generating data packets. Therefore, the

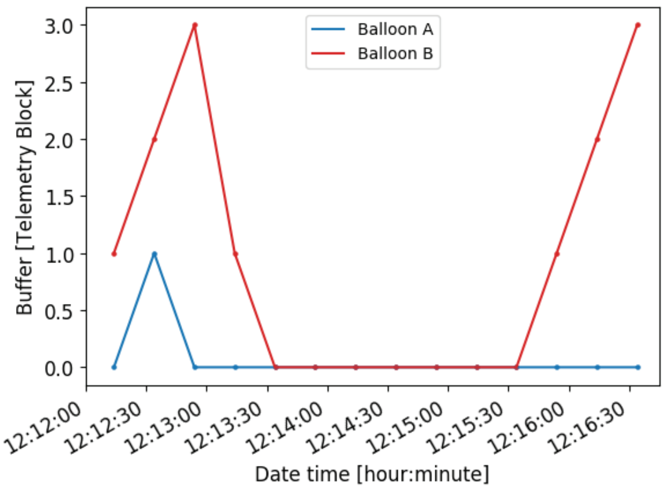


Figure 7.34: Buffer usage of Balloon A (blue) and Balloon B (red) in data packet unit while balloons are released

resolution of this information cannot detect the fluctuations of the buffer due to the exchange of the packets.

The global throughputs achieved in this scenario remains similar as in the ground case, being 790.6 bps for downlink communications and 167.4 bps for federation communications. This performance is achieved because the PDR also remains similar, with 100.0 % for downlink, and 84.4 % for federation. This throughput is equivalent to the 61.9 % and the 88.5 %—respectively—of the average throughput with this PDR values (see Table 7.2). The latencies experienced in this case, which are presented in Figure 7.35, are smoothly greater than in the ground case, due to these retransmissions. The median value of the latency in the downlink communications is 1875 ms, while the one for federation communications is 6550 ms. These values in line with the reference presented in Table 7.2.

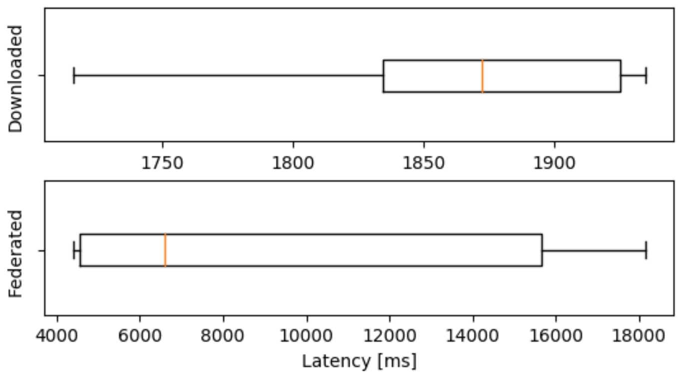


Figure 7.35: Latency of the downloaded packets (top) and using a federation (bottom) while balloons are released

After finishing this interaction, Balloon B was released, starting to lift. At this moment, the communications between Balloon A and the ground station experienced a degradation in the link quality, making it difficult to keep the connection opened. For that reason, the operator jumped to Balloon B in order to download data from Balloon A. Specifically, the ground station opened a downlink connection with Balloon B to promote the its federation with Balloon A, and thus its data. Figure 7.36 presents the packet sequence during this situation, demonstrating the establishment of multiple federations that enabled to download Balloon A data. The location of

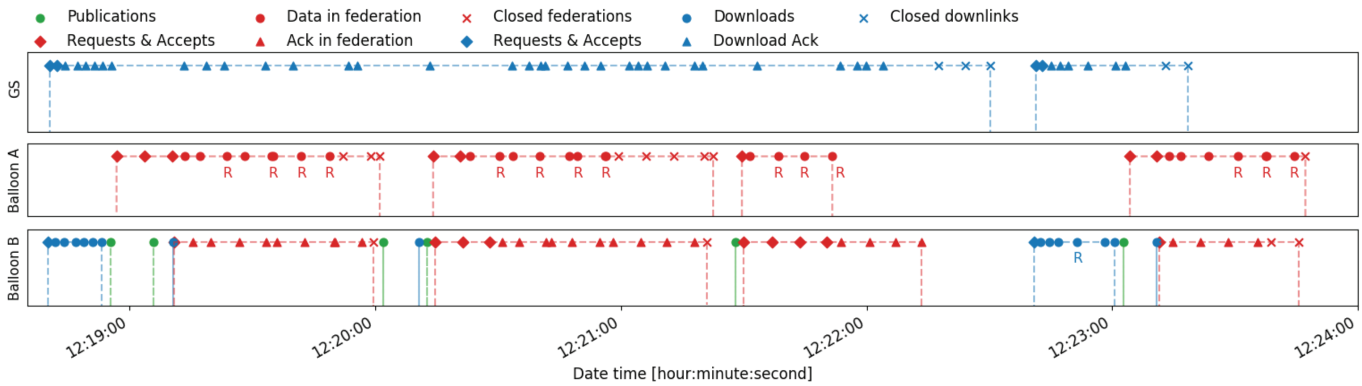


Figure 7.36: Sequence of packets exchanged in federations with a balloon as a data relay

the balloons and the ground station is represented in Figure 7.3. As in the previous cases, the ground station opens the downlink connection with Balloon B at 12:18:40, and this one starts to download data. In this case, the remaining data is less than the usual one, because it has already been downloaded using the previous federations with Balloon A. After finishing, it starts to publish the service, that Balloon A requests. Note that these requests are retransmitted by Balloon A, because Balloon B does not receive the packet. As in the previous cases, the presence of an asymmetric link complicates the communications.

The federation is established, and Balloon A starts to download its data. At this moment, the communications range from the ground station to Balloon A is increased thanks to Balloon B, which helps as a relay node. As in the previous case, multiple federations are established during more than 5 minutes. However, these ones are no closed at discretion of balloon criteria. Instead, the quality of the link is not as good as needed, and the asymmetry impacts enough to break the connection in the third federation. Therefore, Balloon A performs multiple retransmissions, although Balloon B receive all the packets. Additionally, the ground station loss the connection with Balloon B at the end, requiring to open again the downlink connection. Although those communications difficulties, the federations enables to download data from Balloon A, being able to properly empty its buffer. Figure 7.37 presents the buffer usage of both balloons during this scenario. Note how Balloon A empties its buffer using the different federations, multiple times.

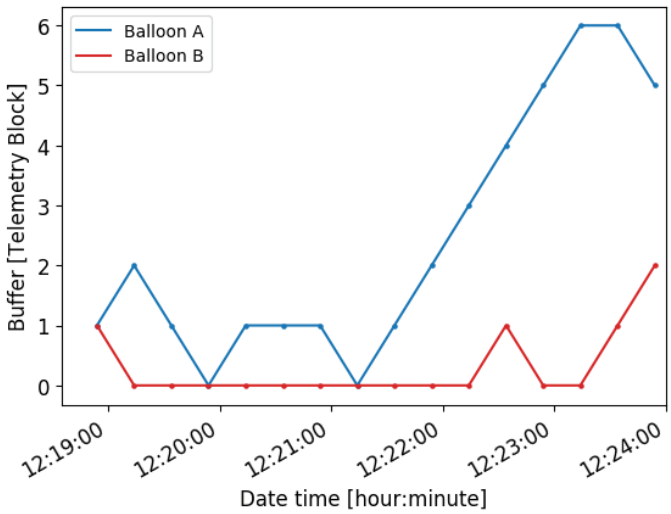


Figure 7.37: Buffer usage of Balloon A (blue) and Balloon B (red) in data packet unit with a balloon as a data relay

In this case, the achieved throughput changes between the federation and downlink communications with respect to the previous scenarios. In particular, the downlink remains during all the time with a PDR of 100.0 %, which enables to achieve a throughput of 621.2 bps that is the 48.6 % of the average throughput with this PDR. Meanwhile, the PDR of federation communications is 61.5 %, smaller than in the other cases. Therefore, its throughput also is reduced to 96.2 bps, which corresponds to the 98.1 % of the average throughput with this PDR. Note that the impact of the PDR over the throughput follows a logarithmic curve, being very sensible any fluctuation of the PDR when it remains close to the 100 %. Therefore, the throughput decreases more than 70 bps. This situation is also reflected in Figure 7.38, which presents the latency values. The latency on the downlink communications remains similar as in the previous cases, but the federation one has increased, which reduces the throughput.

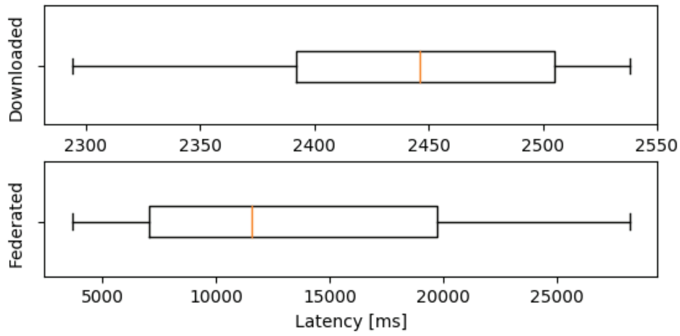


Figure 7.38: Latency of the downloaded packets (top) and using a federation (bottom) with a balloon as a data relay

As a summary, the proposed solution enables the deployment of federations which help to increase the communications range using intermediate nodes. However, the impact of the asymmetric link between the balloons has degraded the federations.

7.7.5 Federations to share storage capacity

After accomplish the federations to share downlink opportunities, federations to share storage capacity were target. For that reason, the ground station remained in silence, and it just sniffed any possible packet that could be transmitted between the balloons. At 12:35:26, Balloon A started to publish the availability of this service (Figure 7.39). This publication is received by Balloon B which request the service to store 38 data packets. The request is accepted by Balloon A, which provokes the establishment of the federation. Note that in this case the packet sequence shows that multiple federations are created due to the bad communications quality of the link. In par-

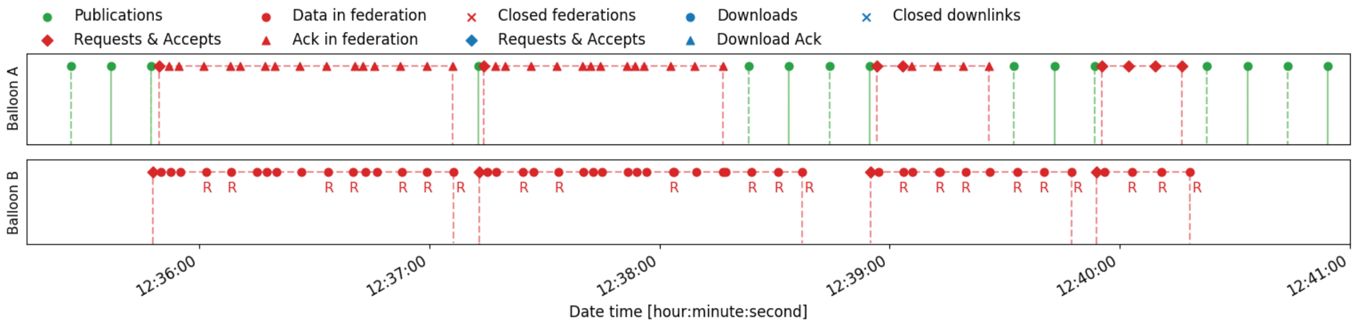


Figure 7.39: Sequence of packets exchanged in federations for storage

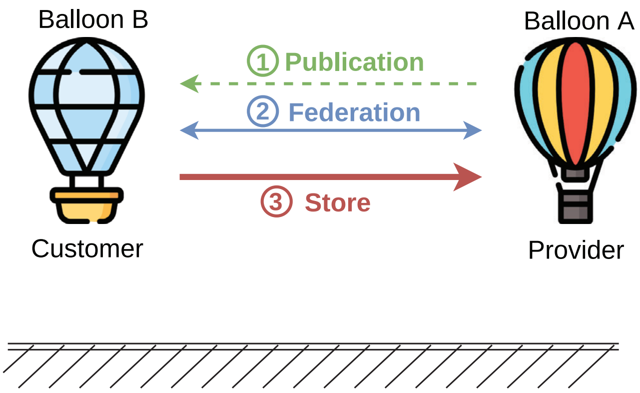


Figure 7.40: Representation of the federation established for storage. The numbers correspond to the time sequence when each action is performed (from low values to greater ones).

ticular, the federations are no closed at discretion of the balloon criteria, but the retransmission of the packets makes that the balloons considers that the link is broken.

As in the previous case, the asymmetric link between Balloon A and B is also present in this scenario. Balloon A receives all the packets that Balloon B is transmitting, including the retransmissions. However, Balloon B cannot receive all the acknowledgements from Balloon A, which provokes the retransmission of the original packet. Note that in this scenario Balloon A starts in 8372 meters of altitude, and finishes in 10344 meters. Meanwhile, Balloon B starts in 4699 meters of altitude, and finishes in 6970 meters. Figure 7.40 represents this scenario in which both balloons share storage capacity.

Returning to Figure 7.39, three satisfactory federations are established for more than 4 minutes and 30 seconds in total. The first federation remains opened during 1 minute and 13 seconds, the following one during 1 minute and 25 seconds, the next one during 52 seconds, and the final one during 27 seconds. This decrement is cause by the degradation of the link quality, which at the end provokes that the federations are broken suddenly and quickly. Although this poor communications, multiple data packets are exchanged between the balloons, following

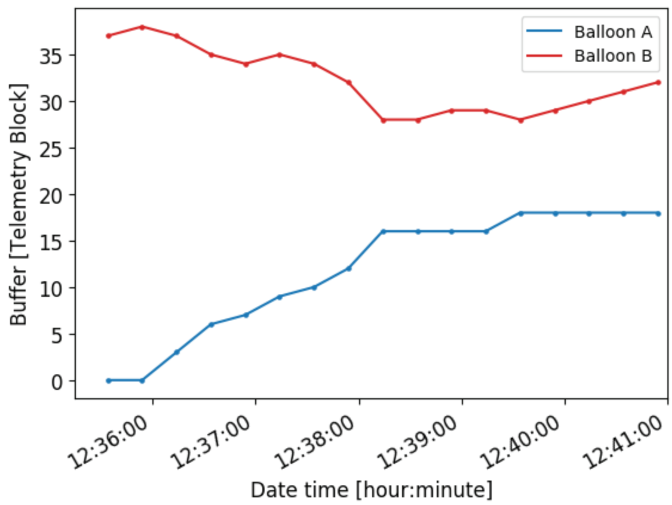


Figure 7.41: Buffer usage of Balloon A (blue) and Balloon B (red) during the federation for storage (data packet unit)

the demand of storing data. Figure 7.41 presents the buffer fluctuation in each balloon during this scenario. Balloon B starts to transfer data to Balloon A, which enable to decrease the buffer usage. Meanwhile, the buffer usage in Balloon A increases, because it is receiving the data from Balloon B. Note that there are lapses of times that this behavior is not achieved, and the buffer of Balloon A remains at the same usage. This is caused due to the break of the federation, and thus no longer communication is established during these lapses of time. Moreover, Balloon A does never reach the maximum capacity of the service that it has offered (i.e. 50 packets). For that reason, it remains publishing the service. At the end, when no longer federations are established, Balloon B keeps generating more data packets.

During the entire lapse of time, 19 data packets are exchanged between the balloons. Taking in consideration the duration of the different federations, the achieved throughput is 107.6 bps. The PDR in this scenario is greater than in the other ones, having that the 57.8 % of transmitted packets delivered correctly. Taking this link quality, the resulting throughput is equivalent to the 54.8 % of the average throughput with this PDR (see Table 7.2). Note that the current communications must be compared with the reference values as a downlink communication, because in this case there is no intermediate node that relays data. As in the other cases, this throughput is conditioned by the experienced latency in the communications. Figure 7.42 presents the statistics of this metric during the entire scenario. Note that the boundary values are greater than the previous cases, due to the number of retransmissions that are done. However, the median remains small, because at the end this kind of communications is similar to the one achieved in the direct downlink. Note however, that the latency increases due to the reduction of the PDR, and thus over time.

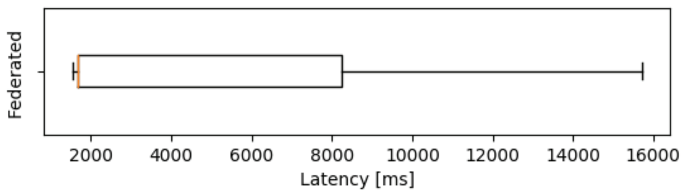


Figure 7.42: Latency of the packets stored using the federations

Although the difficulties achieved to establish federations due to the poor quality of the link, multiple federations to share storage capacity have been achieved. These federations have enabled Balloon B to store its data into Balloon A. This exchange reduced the usage of Balloon B buffer, while the buffer in Balloon A was increasing. These results demonstrate that the proposed protocols can manage different type of services following the same interaction. It is thus extensible to other and future services.

7.8 Summary

Chapter 6 has presented the federation protocol suite which allows a satellite to notify available services and establish federations. These protocols are part of the necessary technology development in order to the IoSat paradigm to be able to become a reality. The federation protocol suite is composed of two application-layer protocols: the OSADP and the FeDeCoP. The former allows a satellite to notify its available services to other satellites. These services correspond to the unallocated resources that a satellite has, and are not continuously available, existing opportunities to consume them only during lapses of time. With the OSADP, a satellite can use these opportunities by discovering those published services. Once the service is notified, a satellite can request it to the provider of the service. The FeDeCoP presents the different regulations to establish and maintain a federation between two satellites. The results retrieved from the different simulations motivated the development of a system that includes this protocol suite to deploy federations.

This chapter has presented the FSSExp payload which includes an implementation of the federation protocol suite, and the necessary hardware components to establish federations. The payload architecture has been reviewed, emphasizing on the different hardware subsystems that compose it, and the software design that implements all the necessary functions. The RF-ISL board is one of these subsystems that may be stood out because it provides the necessary means to establish a wireless communications interface between different FSSExp payloads. This payload becomes thus a system to verify the proposed protocol suite on a new environment, and taking in consideration the hardware limitations. The payload has been included in the FSSCat mission which aims at demonstrating the benefits of applying the FSS concept to EO missions. This mission promotes the execution of multiple payloads, which are distributed in two 6U CubeSats following a train formation. The FSSExp execution in this mission corresponds to a proof-of-concept of FSS, which would demonstrate the feasibility of deploying satellite federations.

As part of the verification plan of the payload, a stratospheric balloon campaign has been performed. Details of the stratospheric balloon campaign have been presented in this chapter, and a discussion of the different results has been conducted. This campaign was performed in the North region of Moscow in a rural area. The balloons flew during 1 hour and 45 minutes, and they established multiple federations. The presented results show the environment that the balloons have experienced, remaking that they reached a maximum altitude of 25,906 meters and a maximum velocity of 76.3 m/s. Additionally, the temperature inside the balloons remained stable between 15 ºC and 20 ºC, while the external temperature was close to -30 ºC. Unfortunately, the launch pad of the balloons was impacted by a RF interference, which increased the noise floor 19.7 dB. This limited the maximum communications range to 55 km. Although this limitation, the federations were established with different configurations: with balloons on the ground, raising, and flying. With the completion of these federations, it has been possible to extend the communications range with a balloon that was not in line-of-sight with the ground station. In particular, another balloon worked as a data relay, thanks to the federation, that helped to download data from this remote balloon. Another scenario demonstrated that the federation can also be established between a ground device, and a flying one, extending the paradigm to a seamless hybrid network of terrestrial and non-terrestrial nodes. Finally, a federation between the balloons to share storage capacity demonstrated that the proposed protocols can manage different kind of services using the same design.

In all these federations the achieved throughput was equivalent to more than the 50 % of the average capacity considering the delivery ration of the packets. Despite this promising percentatge, the absolute value of the throughput may be enhanced in future investigations. These throughput values were mainly caused by the stop and wait ARQ technique applied in the FeDeCoP. This ARQ technique is ideal for half-duplex interfaces, but future research may evaluate how enhance this situation to increase the throughput. Additionally, the use of real-time threads, and the corresponding notifications may also boost the experienced throughput. Finally, the use of an ISL that separates the type of packets (i.e. control and data) may promote the optimization of the channel and the corresponding throughput growth, as proposed in Chapter 6. Nevertheless, the performance presented in this chapter corresponds to the first prototype version of the FSSExp payload, which opens the door to keep developing the system with future enhancements. Furthermore, the presented research has demonstrated that using the proposed protocols the FSS and IoSat approaches are feasible. The execution of the FSSExp payload in the FSSCat mission would also demonstrate the feasibility of these two paradigms. This may encourage the research on this topic, and to develop future spacecraft projects which integrates these protocols as part of its software core.

8

Conclusions

8.1 Introduction

Earth Observation satellite systems have become dependable resources for environmental and climate monitoring, weather forecast, governmental activities, modern agriculture, and countless other commercial and industrial applications. The incursion of 5G in the aerospace domain has motivated the conception of satellite systems as a promising platform to achieve desired applications that require global coverage (e.g. the mMTC use case). The numerous demands can be summarized in two main system requirements: (1) increase of data transfer capacity, and (2) decrease the end-to-end communications latency. As conveyed throughout this dissertation, some satellite systems are starting to adopt novel structural and functional paradigms grounded upon distribution and increased on-board capabilities in order to address these ever growing user and application requirements. Structurally dynamic and potentially large-scale, full-fledged DSS are being conceived as networks of sensing units that could become part of a self-governed, in-orbit infrastructure. Among the different distributed architectures, the FSS emerged as a candidate to exploit all the potential of DSS by means of establishing opportunistic collaborations among satellites to share unallocated or underutilized satellite resources. These collaborations—formally called federations—allows to conceive the space as a cloud system in which satellites can leverage from other resources to improve their current mission performance. Multiple aspects related to their design and operation are still open fields of study in the aerospace community, and some have been expressly tackled in this research. This final chapter discusses the specific contribution areas, recalling both the context at large and summarizing results, and concludes by identifying possible avenues for future research.

8.2 Research contribution

Studies to quantify the potential and the benefits of applying FSS in current satellite missions have demonstrated that this federated architecture can satisfy new demands, and materialize new commercial stakeholder perspectives (Golkar, 2013a; Lluch, 2017). Despite these expectations are applicable in numerous satellite activities, the potential of FSS is mainly remarkable in the EO market with new capabilities based on heterogeneous satellites, such as the fusion of data from different instruments. This future perspective encouraged multiple lines of research to enhance the integration of federations in satellite missions. This dissertation has presented the developments conducted so far, and mainly centered at different aspects of system definition: (1) a framework to quantify—in early stages of mission definition—the benefits of applying federations in return for increasing the complexity of satellites (Lluch and Golkar, 2019); (2) a mechanism to design satellite missions which includes federations taking in consideration the uncertainty of stakeholder intentions (Grogan et al., 2016b); (3) an auction-based mechanism to in-orbit determine the fair value of the available resources according to the satellite concurrence and demands (Pica and Golkar, 2017); and (4) the identification of the data security requirements in federations, and a preliminary approach to satisfy them (Maurich and Golkar, 2018). Despite the interesting technology maturity achieved, all these developments assumed that the necessary communication means to deploy federations were available, functional, and suitable for federation nature (e.g. ISL technologies, protocols to deploy and manage networks, protocols to negotiate and transact resources, and protocols to notify available services). However, this assumption needs still to be discussed in depth.

The review of the different satellite networks proposed in the past years conveyed the limitations of these proposals to tackle the sporadic, opportunistic, and heterogeneous nature of satellite federations. Some of the proposed satellite networks have been designed to deal with the satellite motion which may entail the fragmentation of the network and the isolation of satellite groups. By means of dedicated satellite architectures or centralized predictive approaches, these proposals could mitigate or coexist—respectively—with this network phenomenon. Nevertheless, they cannot tackle with the federation nature presented in the FSS concept, because their design was not conducted to satisfy this requirement. Specifically, heterogeneous orbits may coexist in the FSS being per se contradictory with a pre-defined satellite architecture. When this custom satellite system architecture is not included in the design process, the satellite mobility drives the capability to communicate. Satellite-to-satellite interfaces are characterized by being temporal links, which can be concatenated to generate per se a temporal route between two remote satellites. Due to this temporality, the routes and links may be destroyed over time, disrupting a satellite network into multiple connected fragments. Maurich and Golkar indicated in (Maurich and Golkar, 2018) the imperative necessity to address this situation for the sake of the established federations.

This dissertation has been founded on this necessity to **conceive a communications environment over which federations can properly been deployed**. The contributions crystallized in the previous chapters have tried to answer some questions that still were present in the research field of satellite federations: (1) What is the nature of end-to-end routes between remote satellites in sporadic and opportunistic satellite networks, and how can they be defined? (2) How can satellites be aware of the available resources offered by other satellites? and (3) what are the necessary mechanisms to deploy a federation in this opportunistic context? Therefore, the definition of this space paradigm and the exploration of tangible solutions to deploy satellite networks following the federation premises, naturally constituted the fruitful field of study of this doctoral research. The remaining sections present in details each of the contributions achieved during the completion of this doctoral research.

8.2.1 The IoSat paradigm: a network context for FSS

As previously stated, the development of FSS reached a stalemate that required the development of communications techniques that facilitate the establishment of satellite federations. This networked satellite system would be able to respect and adapt itself to the nature of satellite federations. These federations can be characterized by a set of unique features: (1) sporadic interactions that could not be anticipated; (2) opportunistic collaborations to obtain certain mission benefit from its establishment; (3) decision-making capacity in satellites to identify the potential interest on participating in a federation; and (4) composed of heterogeneous satellites with distinct resources, capacities, orbit trajectories, owners, and operators. These features—in conjunction with satellite motion—pose a considerable challenge in the definition of end-to-end communications routes composed of intermediate satellites. This task, conducted by the routing protocol, is a technology enabler to deploy federations in future satellite missions.

The discussion of current proposals to deploy satellite networks performed in Chapter 2 emphasized the scarcities of these proposals to tackle satellite federations. Snapshot networks are centered on the natural and predictive nature of satellite motion which conforms a sequence of topology pictures where inter-satellite connections remain stable during a lapse of time. Following a similar premise, DTN propose the construction of a contact plan that schedules inter-satellite networks over time to understand the network evolution, and leverage on determining time-forward routes. Additionally, the same type of network encourages the use of probabilistic techniques in case that prediction could not be affordable by leaking on reliability. The LEO satellite networks and MLSN promote the deployment of a dedicated satellite architecture to mitigate the motion impact. Despite the impressive performance achieved, the use of this custom architecture cannot deal with the heterogeneous nature of federations. Nevertheless, MLSN demonstrated the potential of combining different orbit altitudes to enhance the network performance. The mega-constellations encourage the deployment of hundreds or thousands of satellites to achieve a commercial performance. This approach may not be affordable due to other administrative challenges or the induced cost of this massive system. MANET technologies seem to be a promising approach thanks to its adaptive capacity to network events and node mobility. Nevertheless, the network disruption becomes its kryptonite by reducing the feasible instantaneous end-to-end routes. Last—but not least—, WSN demonstrated the benefits of including resource awareness mechanisms for embedded systems. Nevertheless, it has not been designed to efficiently manage node mobility.

None of the previous network proposals were conceived to deploy satellite federations, and all of them lack of certain feature that may be present in another one. This dissertation has tackled this issue by **the definition a new network environment adapted for satellite federations: The IoSat paradigm**. This paradigm promotes the development of a dynamic environment in which satellites decides to establish temporal networks—called ISN—to deploy federations when a benefit is awarded. This kind of satellite network is characterized by (1) having a dynamic topology due to satellite motion, (2) having nodes that follow the deterministic orbit trajectory, (3) being compounded of unreliable nodes that may decide to not participate in the network, (4) interconnecting nodes that are resource-constrained, (5) coexisting distinct technologies and operation strategies conforming a heterogeneous scenario, (6) being deployed sporadically and opportunistically, and (7) deploying distinct federations type which differ on traffic nature. Following the architecture of current Internet, the IoSat proposes the interaction of satellites from different systems to create a global network, instead of deploying a custom and massive infrastructures. This approach makes satellite systems more accessible than in the other case, which may be driven by administrative, cost, and high density issues. The development of a routing protocol that satisfies these features has to be properly conducted.

The definition of the IoSat paradigm and its features has been previously published in the peer-reviewed IEEE Access journal (Ruiz-de-Azua et al., 2018c), and in the 5th Federated and Fractionated Satellite Systems Workshop (Ruiz-de-Azua et al., 2017). Furthermore, the participation in the ONION project to define a technology roadmap for satellite networks has crystallized in the formal definition of this paradigm. This work has been published in the peer-reviewed IEEE Access and Acta Astronautica journals (Alarcón et al., 2018; Tonetti et al., 2020), the 70th International Astronautical Congress (Tonetti et al., 2019a), and the Living Planet Symposium (Tonetti et al., 2019b). The details of these publications are accessible in Chapter 9 with the identifiers [I], [II], [V], [VII], [XIV], and [XVII].

8.2.2 Identification of the required routing protocol capabilities

The design of a proper routing protocol requires a dedicated study to identify which capabilities should have it to be able to deploy ISN in the IoSat paradigm. The review of the routing protocols developed for each satellite network type—conducted in Chapter 2—has conveyed numerous features and capabilities that may be appropriate to deploy ISN in the IoSat paradigm. Among them, we have discussed which could be appropriate to ISN nature. The discussion concluded with **the identification of the appropriate features that an ideal routing protocol should include to define a route in the IoSat paradigm**: (1) a connectionless approach can better manage network mobility; (2) an adaptive protocol is able to learn the network topology changes, and reacts against unpredictable events (e.g. request of a federation); (3) predictive mechanisms need to be complemented with instantaneous route definition; (4) the routing table has to be computed in a distributed manner, instead of a centralized architecture; (5) a distance vector approach is suitable for uncertain size of the network, which could reach large dimensions; (5) the capability of managing multiple routes enhance the reaction time against path changes or congestion scenarios; (6) a hybrid approach to learn about the network topology that combines the discovery of the network from reactive protocols with the proactive maintenance seems to be the best strategy for ISN; and (7) the selection criteria must integrate different metrics—specifically, the satellite resource status—with the possibility to include route constraints. By means of this list, a custom routing protocol could be designed from scratch, or a comparison of current routing protocols developed for other satellite networks could be performed to evaluate if they could be appropriate for ISN.

8.2.3 Study of current routing protocols for the IoSat paradigm

Thanks to the identification of the required features of a routing protocol we have been able to evaluate if current routing protocols from other satellite networks could deploy ISN. The routing protocols developed for snapshot networks and predictive approaches for DTN are founded on a central architecture in which a capable infrastructure can compute the evolution of the entire network. This centralized approach does not match with the distributed feature for ISN, making these solutions not scalable and quite limited in flexibility. The protocols developed for LEO satellite networks and MLSN properly manage the congestion of the network, and provide low end-to-end latency. Nevertheless, they are customized to the proposed satellite architectures, which make them difficult to be applied in a heterogeneous scenario. Despite this limitation, these protocols take in consideration multiple routes simultaneously which is a desired feature for ISN. Replication-based DTN approaches cannot explicitly guarantee the delivery of the messages, which could not be acceptable for certain federations. Finally, MANET and WSN protocols are the ones that are—in mean—more suitable to deploy ISN. Specifically, all the hybrid solutions prevail against the other reactive and proactive protocols. Nevertheless, these hybrid approaches are zone-based, which could not be entirely integrated in satellite networks. All the protocols of the WSN have similar FoM, except the IRPL which has an 80 % as FoM. The capability to integrate different metrics in a single OF, which includes the status of the node resources, makes this protocol a relevant approach for ISN. However, the RPL family is not able to manage node mobility by default. Therefore, the protocol should be extended in order to be integrated in the IoSat paradigm. Despite this suitability, the maximum averaged FoM per satellite network does not exceed the 65 %, which indicates that there is—in mean—a 35 % of difference with respect to the ideal routing protocol. Furthermore, none of the reviewed protocols presented mechanisms to integrate the capability to decide the participation of a satellite in the construction of the network. **This result suggests that no single protocol can satisfy all the requirements of ISN, and they require to be extended with capabilities that are presented in other protocols**. Therefore, further investigations can be conducted to evaluate the performance of these protocols, and identify potential improvements to overcome these limitations.

8.2.4 The connectivity is fundamental for ISN

The possibility to apply proactive routing protocols from MANET into satellite networks has also been discussed and evaluated. With the previous analysis, routing protocols from MANET and WSN emerged as a promising capabilities that could be appropriate for ISN. The protocols developed for WSN has the ability to differentiate routes according to a group of metrics and constraints. These metrics may be the commonly used in routing protocols—i.e. number of hops, delay—, but they can be extended to other metrics related to the status of the nodes. This concern about the node resources is essential in a network composed of embedded systems, like sensors or satellites. Nevertheless, these protocols were not originally conceived for movable nodes, which complicates its direct application in satellite networks. On the other hand, the flexibility and adaptability of the protocols conceived for MANET are suitable capabilities to address the dynamic environment of ISN. The possibility to learn the topology over time would allow to react against link changes due to satellite motion. Chapter 4 has extended the study by simulating a scenario where satellites deploy a network using the OLSR protocol. This protocol has largely been used in other connected infrastructures, such as in networks composed of ground or flying vehicles.

The study has started by analyzing the dynamism of the network topology, suggesting that **the disruption of the network is a relevant phenomenon that influences the communications performance of satellite systems**. The routes that are composed of large number of intermediate satellites are more sensitives to changes than smaller routes, having short lifetime (roughly seconds). Typically, stable routes are less frequent that intermittent routes, which provokes that satellite networks can be considered as a dynamic environment. The utility of a route is an interesting metric to evaluate the potential interest of the network at system level, because it corresponds to the amount of data that can be transmitted in a feasible route with respect to the generated data. The results presented in Chapter 4 indicate that short-time routes have to be used for sporadic transmissions, while long-time routes for data streams. This is important to identify the data from which payload must be propagated at each route according to the temporal nature of the route. **The connectivity, and in particular the nature of the routes, are fundamental resources of a satellite system that must be properly allocated to optimize them**. The fragmentation of the network must be mitigated to improve the connectivity of the network. An option to ameliorate this connectivity could be the design of an appropriate satellite architecture that naturally mitigate this disruption—like LEO satellite networks—or the inclusion of a larger number of satellites—like mega-constellations. Another approach is the **integration of new and advanced communications capabilities into the satellites designs to improve their connectivity**: (1) farther communications ranges, (2) multiple devices to achieve an omnidirectional patter, and (3) predictive algorithms to anticipate feasible routes.

Therefore, the study demonstrated that the OLSR protocol cannot tackle with all the challenges that entail this type of satellite network. The development of new features for OLSR protocol that could anticipate this disruption (e.g. predictive algorithms or store-and-forward mechanisms) could enhance its performance in this satellite context. The research related to this study has been published in the peer-reviewed IEEE Access journal (Ruiz-de-Azua et al., 2018b). The details of this publication are accessible in Chapter 9 with the identifier [III].

8.2.5 Definition of a predictive algorithm for satellite contacts

The disruption of the network cannot be underestimated, because it can considerably degrade the communications performance. An option that has been emphasized during all the dissertation is the integration of capabilities presented in DTN solutions into MANET protocols. One of those capabilities is the prediction of network changes to anticipate future instantaneous routes or routes over time. These DTN solutions are founded on a central entity—the CPCE—which computes the entire contact plan of a satellite system. This plan corresponds to a sequence of scheduled inter-satellite contacts over time, which allows identifying feasible routes composed of an inter-satellite contact succession. These time-forward routes are thus defined in the time domain, instead of the common end-to-end routes which are defined in the space domain—being also called as instantaneous routes. As stated previously, this centralized approach is not suitable for ISN, because it is not easily scalable nor flexible. To overcome these limitations, new investigations aim at providing autonomy to satellite. Therefore, following this premise, Chapter 5 has presented a predictive protocol to estimate inter-satellite contacts in a distributed manner.

With this autonomous and de-centralized approach, a satellite can identify other neighbor satellites, and predict when a contact happens assuming an omnidirectional antenna pattern. The proposed solution uses a predictor which estimates the contacts leveraging from the determinism of the satellite trajectory. Therefore, it uses the satellite orbital elements to formulate simple model, without propagating the entire trajectory. To define the prediction algorithm, a contact between two satellites has been modeled as a satellite proximity or closeness. The proposed protocol has been evaluated in two different scenarios that could represent ISN: a hybrid constellation, and a massive satellite constellation. The results presented in Chapter 5 demonstrate that each satellite is able to construct the contact plan autonomously to identify feasible routes with remote satellites. However, it is highlighted that this final knowledge is achieved after a lapse of time in which the algorithm is discovering. The results also confirmed that this time is directly related to the ISL communications range, which determines the disruption of the network impact. With farther ranges, less time is required. However, an inflection point in which all the satellites have the needed knowledge appears in all the cases. Additionally, the results also highlight that the proposed algorithm is able to discover those remote nodes with which physically an interaction can be established. Therefore, depending on the disruption of the network, the satellite does not necessarily learn about the entire network, just about its community.

The achieved accuracy of these estimated contacts are more than 90 % when the satellite orbits does not differ considerably. This performance encourages to **conceive distributed predictive mechanisms for satellites that support or complement routing protocols to anticipate network disruption**. The prediction accuracy is degraded when the heterogeneity of the satellite system increases considerably. This situation may appear in the IoSat paradigm, which motivates the enhancement of this technology in future research. Due to its non-linear behavior, these predictors may be defined by means of AI that could enhance the final accuracy.

The design of this predictive algorithm has been published in the peer-reviewed IEEE Access journal (Ruiz-de-Azua et al., 2020b), and the International Astronautical Congress (Ruiz-de-Azua et al., 2019b). The details of these publications are accessible in Chapter 9 with the identifiers [IV], and [XIII].

8.2.6 The federation protocol suite: from the design to the implementation

The previous study has been centered on the design of routing protocols for ISN. Nevertheless, some other important questions remained still unanswered: How can satellites be aware of the available resources offered by other satellites? What is the necessary mechanisms to deploy a federation in this opportunistic context? Without the proper technology development to address these challenges the routing protocol could become useless. Therefore, Chapter 6 has presented **the design of the federation protocol suite which aims at providing the necessary means to notify available services and establish satellite federations**. These collaborations are founded on sharing underutilized satellite resources, which can be offered as services to other satellites. Due to the use of these resources, the offered services may not always be available. Due to the use of these resources, lapses of time in which the resource is available may exist. It becomes thus crucial to develop the fundamental protocols that allows to leverage these temporal opportunities.

Two protocols conform the proposed suite: the OSADP, and the FeDeCoP. The OSADP is responsible to publish the available services following a proper dissemination in the network which bounds the regions of the network where the publication is forwarded. A satellite that receives this publication retrieves information about the available service, and it can decide to request the establishment of the federation. This establishment is conducted with the FeDeCoP which proposes the execution of three phases: the negotiation, the consumption, and the closure phases. With these three phases, different types of federations can be established in a satellite system. The protocols have been evaluated in a heterogeneous satellite system which demonstrated its benefits by increasing the entire downlink capacity. Specifically, the execution of the protocol suite with a reduced number of satellites allows downloading 7% more than not using it. In this case, the potential benefits of applying this solution are limited by the network disruption associated to the small number of satellites deployed. **With further satellites, the system can duplicate the overall downloaded data**. This performance encourages to keep investigating on the development of new features that could be applied to these protocols in order to mitigate the impact of the network disruption.

The research associated to the federation protocol suite has been published in the peer-reviewed IEEE Access journal (Ruiz-de-Azua et al., 2020a). The details of this publication are accessible in Chapter 9 with the identifier [VI].

8.2.7 The necessity to conceive different communication planes

The results achieved in Chapter 6 with the execution of the federation protocol suite would not be possible if multiple communications plane would not be taken in consideration. The ideal ISL subsystem would provide large data rates, large communication ranges, and transmitting through an omnidirectional pattern. With these characteristics, it would be possible to have a system that provides fast transmission and high connectivity. Unfortunately, these characteristics cannot be achieved using a single subsystem—due to physical constraints. Nevertheless, the combination of multiple ones can tackle this challenge. Chapter 6 has proposed **the use of the control and data planes to separate what is related to the network signaling, and the data**

transmission. As in other scenarios, the combination of decoupled and independent planes allows to isolate network managing tasks and traffic, which optimizes the data transmission on the other plane. The control plane must provide a communications channel for those packets that manage the network, which do not require high transmission data rates. Nevertheless, it requires a subsystem that provides a large communications distance to increase the connectivity. Otherwise, the data plane provides a communications channel to transmit the generated data from the payloads, which requires a link with a high data rate. This ISL subsystem must be based on a directive antenna or laser to achieve large distances by specifically pointing to the neighbor satellite.

The combination of both planes provides a satellite-to-satellite communications interface with high connectivity, large range, and great data rate. Furthermore, this new device may include flexible mechanisms that allows to modify its functions according to each satellite. By means of SDR technology, this flexibility enables to interface different ISL devices that work with distinct characteristics. This is essential to be integrated in future satellite missions in order to enhance the system connectivity. As previously demonstrated, the conjunction of this device with the federation protocol suite may largely improve mission performance.

8.2.8 Development of the FSSExp payload

The results presented in Chapter 6 motivated the development of the FSSExp payload, which is a device that includes an implementation of the federation protocol suite, and the necessary hardware components to establish federations. Chapter 7 has presented the design of this payload by reviewing all its subsystems. Among these subsystems, the RF-ISL board stands out by allowing the creation of a wireless communications interface over which data can be exchanged. The development of this payload has been conducted over different phases. The design and implementation allowed the verification of the final system in the laboratory. This verification consisted in different executions in a controlled location, with a far distance configuration, and in the facilities that allows to emulate space conditions. Furthermore, a stratospheric balloon campaign has been performed as part of this verification plan. In this campaign, three stratospheric balloons established different federations sharing downlink opportunities and storage capacity using the FSSExp payload. Specifically, each balloon generated data that was downloaded to a ground station using federations with the other balloons.

The results of the stratospheric balloon campaign demonstrate that the FSSExp payload allows establishing federations. Numerous federations have been achieved in different configurations: (1) all the balloons were on ground, (2) between a balloon taking off and another on ground, (3) during the release of all the balloons, and (4) only between balloons. These different federations allowed to evaluate the performance of the protocols in a real scenario. The achieved throughput was equivalent to more than the 50 % of the average capacity considering the delivery ration of the packets. With the completion of these federations, it has been possible to extend the communications range with a balloon that was not in line-of-sight with the ground station. In particular, another balloon worked as a data relay, thanks to the federation, that helped to download data from this remote balloon. Another scenario demonstrated that the federation can also be established between a ground device, and a flying one, extending the paradigm to a hybrid network of terrestrial and non-terrestrial nodes. Finally, a federation between the balloons to share storage capacity demonstrated that the proposed protocols can manage different kind of services using the same design.

The presented design corresponds to the first prototype of the payload. Thanks to the balloon campaign, we have identified possible enhancements that could improve the performance its performance. Specifically, the throughput can be improved by the use of the suggested ISL device that separates the communications planes. Furthermore, the experience achieved in the balloon campaign demonstrated that communications interference may become a serious problem in future satellite missions. Therefore, the application of mitigation techniques could also improve the current performance.

The design and experimentation of the FSSExp payload has been published in the International Geoscience and Remote Sensing Symposium (Camps et al., 2018; Ruiz-de-Azua et al., 2019c, 2020d), and an article has been submitted to the peer-reviewed IEEE Access journal (Ruiz-de-Azua et al., 2020c). Furthermore, the design process that has experienced the RF-ISL board from the ^{3}Cat-4 mission until the FSSCat context has been published in the URSI Atlantic Radio Science Meeting (Ruiz-de-Azua et al., 2018a), the International Geoscience and Remote Sensing Symposium (Ruiz-de-Azua et al., 2019a), and the Specialist Meeting on Reflectometry using GNSS and other Signals of Opportunity (GNSS+R) (Ruiz-de-Azua et al., 2019d). The details of these publications are accessible in Chapter 9 with the identifiers [VIII], [X], [XI], [XII], [XVI], and [XVIII].

8.2.9 Development of an integrated simulation framework for networked DSS

The different studies conducted in this dissertation would not be possible without **the development of a software framework that enables to simulate networked DSS**. Chapter 3 presented the design of this framework which enables the simulation of autonomous DSS, and emulation of ISL communications with fidelity. Implemented on top of the NS-3 core, the simulator encompasses a programming environment through which custom satellite components can be modeled and controlled, as well as a user configuration area that facilitates the definition of multiple different DSS contexts. We have briefly presented how the architectural definition of a satellite is composed of three unique modules: operations, networking, and physical. Satellite behavior (e.g. modes of operations, execution of tasks) and flight software is emulated in the operations module, which leverages NS-3 components and is internally implemented as the application layer of a network node. The physical layer encompasses as many self-contained representations of subsystems, devices, or abstract models provided by the user. Finally, the networking module allows the user to define their own custom network stack, enabling the design, implementation, and verification of new ISN protocols. By means of a dedicated design, the software framework is able to emulate satellite-to-satellite and satellite-to-ground communications which are characterized by intermittent contacts. This framework encompasses thus the three main modules that could be critical technological enablers for future research on networked satellite systems. Despite its current design is functional, future research may still be focused on improving the framework with new functions and satellite models (e.g. pointing system, 5G use-cases traffic models, etc.)

Nowadays, the simulator is encompassed in an on-going project developed at UPC that was briefly introduced in (Ruiz-de-Azua et al., 2018d) and (Araguz et al., 2019). The details of these publications are accessible in Chapter 9 with the identifiers [IX], and [XV].

8.3 Future research

The work presented in this thesis has opened further questions that may hopefully foster new avenues of research. The design of ISN in the IoSat paradigm still entails multiple communications challenges. Throughout this dissertation, we have examined many satellite networks and their technologies to evaluate if they could be applicable in the IoSat paradigm. The discussion concluded that no single protocol can perfectly suit the nature of ISN, requiring the development of a dedicated protocol. This dissertation has introduced the challenges that this protocol have to tackle, and performed a preliminary study to identify the mechanisms to include in the OLSR protocol that would improve its performance. Nevertheless, the design of the routing protocol is still a task to be accomplished. We hope this study may motivate this design process in future research, generating a proposal that enables the deployment of ISN.

Among the different challenges that this kind of protocol must tackle, the disruption of the network is the most relevant. The results presented in this dissertation suggest the necessity to consider satellite connectivity as a valuable resource during the system design. This may encourage the development of new techniques and devices for this purpose. This enhancement must be performed to achieve the following goals: (1) the extension of farther communications range (roughly 3500 km), (2) the increase of the link capacity for the data transmission (roughly 100 Mbps), (3) the increase of the visibility of the satellites that could—closely—reach omni-directional patterns, and (4) the necessary flexibility to switch among different physical technologies. This enhancement on satellite connectivity would entail the global improvement of the satellite network, and thus mitigate the disruption of the network without the deployment of custom satellite architectures or massive satellite constellations.

The development of the predictive protocol presented in this dissertation demonstrated the feasibility to achieve a de-centralized approach to predict inter-satellite contacts. Despite the achieved results with the proposed solution, the presented approach has still some limitations that needs to overcome in order to achieve a competent performance in all the scenarios. We expect that these results motivate the development of new techniques that keep this distributed philosophy and can exceed the features of our approach. This prediction capability would support the development of any routing protocol by anticipating the disruption of the network, and determining the routes accordingly.

The research presented in this dissertation showed the development of the federation protocol suite which includes two protocols to establish federations. These protocols have been designed to be appropriate for federation nature. The results demonstrate that the use of these protocols can boost the performance of the satellite mission, but further enhancements can be applied. The disruption of the network has not been taken in consideration during the design of the protocols, which—as results suggest—drives the resulting performance. Therefore, the integration of store-and-forward techniques in these protocols could become a step forward on its application in future satellite missions that may be characterized by this network phenomenon. These investigations on enhance the protocol design may also be applicable in the FSSExp payload. The presented system corresponds to the first prototype of a device that helps to establish satellite federations. It has been conceived to be a proof-of-concept of the FSS and IoSat paradigms. Therefore, the improvement on its design with the new techniques applicable to the protocols, and hardware ameliorations could still encourage the integration of this module in future satellite missions.

One of the outcomes retrieved from the stratospheric balloon campaign for the verification of the FSSExp payload is the dramatic impact of RF interferences in communications performance. This undesired phenomenon may degrade the expected performance of satellite communications until reaching the loss of the link. Future research may be centered on developing mitigation techniques in communications, which have already been applied in other fields, such as in EO satellite missions (Forte Véliz, 2014; Querol, 2018; Tarongí Bauzá, 2013). Furthermore, the aggregation of numerous devices in the same channel may also provoke a massive competition to access the medium that its effects could entail the same results as RF interferences. Therefore, it is crucial that future investigations tackle this challenge.

Among the different challenges that this dissertation has intended to contribute, other questions about FSS remain unanswered. One of the most relevant is the data security level that was briefly tackled in (Maurich and Golkar, 2018). New security mechanisms (e.g. blockchain) that follow the nature of satellite federations need to be developed in future research in order to be possible the integration of these satellite interactions in future missions.

Finally, the integration of the proposed protocols into a scenario that includes autonomous operations would lead to evaluate the proposed protocols in other scenarios and applications. The dissertation of Araguz (2019) suggested the necessity to extend his research with accurate modeling of satellite communications in DSS. The merge of both research may provide new insights of satellite systems which could motivate the development of new applications and services. Furthermore, the developed protocols have not been applied for broadband telecommunications services. A study to evaluate the performance of these protocols in this context may enable to evaluate their benefits and limitations. However, the software framework presented in this dissertation to simulate networked DSS needs to be extended to accomplish this goal. With this development, we hope that this framework could help in future research of similar topic.

8.4 Final remarks and open discussion

The evolution of space systems is currently and constantly seeking the incorporation of a myriad ground-based technologies that promise to improve the way in which satellites can offer services to users on the Earth. Some of the most audacious DSS propositions, if not all, are facing a change of paradigm that is partly motivated by the Internet of Things, the extraordinary miniaturization of computing platforms, cloud computing and virtual services, and the recent advancements in machine learning and artificial intelligence. Many of the research efforts seem to indicate that we should expect the development of very complex satellite systems in the years to come. Among the different proposals, the FSS with the IoSat paradigm propose an interconnected space that require the collaboration of different satellites owned and operated by distinct space entities.

Aside from the technical difficulties and obstacles to solve, this vision may encourage the improvement of current missions by extending the capabilities of satellites with new resources that are not physically located inside them. This could-based conception entails a major change on the traditional view of satellite systems, which could not be directly accepted by the community. Nevertheless, like in many other areas in ground technology (autonomous vehicles, UAVs, smart cities), many are the advancements that are constantly pushing to make this paradigm change a reality. The future of Earth observation systems and broadband telecommunications is very likely to benefit from the design of these collaborations, and the proposer communication interface to establish them. These systems may be able to observe, learn, react, and predict how to balance their actions according to the communications evolution in satellite networks. These new capabilities could perceive the space as a heterogeneous environment in which satellites autonomously interact for an explicit or implicit amelioration in the final service. This novel paradigm will excel in uncountable qualitative properties far beyond revisit times and latency, offering innumerable humanitarian, environmental, agricultural, industrial, and commercial opportunities. Only the time and the research effort will decide if this sophisticated paradigm shift will become a reality.

9

List of publications

Relevant peer-reviewed journal publications associated to this dissertation:

[I] J. A. Ruiz-de-Azua, A. Calveras, and A. Camps. “Internet of satellites (IoSat): Analysis of network models and routing protocol requirements.” In: *IEEE Access*. Volume 6, pp. 20390–20411. June 2018. DOI: {10.1109/ACCESS.2018.2823983}

[II] E. Alarcón, A. Alvaro, C. Araguz, G. Barrot, E. Bou-Balust, A. Camps, S. Cornara, J. Cote, A. Gutiérrez Peña, E. Lancheros, O. Lesne, D. Llaveria, I. Lluch, J. Malés, A. Mangin, H, Matevosyan, A. Monge, J. Narkiewicz, S. Ourevitch, H. Park, S. Perotti, U. Pica, A. Poghosyan, P. Rodriguez, J. A. Ruiz-de-Azua, P. Sicard, M. Sochacki, S. Tonetti, S. Topczewski. “Design and Optimization of a Polar Satellite Mission to Complement the Copernicus System.” In: *IEEE Access*. Volume 6, pp. 34777–34789. June 2018. DOI: {10.1109/ACCESS.2018.2844257}

[III] J. A. Ruiz-de-Azua, A. Camps, and A. Calveras. “Benefits of using mobile ad-hoc network protocols in federated satellite systems for polar satellite missions.” In: *IEEE Access*. Volume 6, pp. 56356–56367. September 2018. DOI: {10.1109/ACCESS.2018.2871516}

[IV] J. A. Ruiz-de-Azua, V. Ramírez, H. Park, A. Calveras, and A. Camps. “Assessment of Satellite Contacts using Predictive Algorithms for Autonomous Satellite Networks.” In: *IEEE Access*. Volume 8, pp. 100732–100748. May 2020. DOI: {10.1109/ACCESS.2020.2998049}

[V] S. Tonetti, S. Cornara, G. Vicario de Miguel, S. Pierotti, J. Cote, C. Araguz, E. Alarcón, A. Camps, D. Llaveria, E. Lancheros, J. A. Ruiz-de-Azua, E. Bou-Balust, P. Rodriguez, M. Sochacki, J. Narkiewicz, A. Golkar, I. Lluch i Cruz, H. Matevosyan. ”Mission and system architecture for an operational network of earth observation satellite nodes.” In: *Acta Astronautica*. Volume 176, pp. 398-412. June 2020. DOI: {10.1016/j.actaastro.2020.06.039}

[VI] J. A. Ruiz-de-Azua, A. Calveras, and A. Camps. “A Novel Dissemination Protocol to Deploy Opportunistic Services in Federated Satellite Systems.” In: *IEEE Access*. Volume 8, pp. 142348-142365. August 2020. DOI: {10.1109/ACCESS.2020.3013655}

Relevant conference publications associated to this dissertation:

[VII] J. A. Ruiz-de-Azua, A. Calveras, and A. Camps. “Internet of satellites (IoSat): An interconnected space paradigm.” In: *5th International Federated and Fractionated Satellite Systems Workshop*. Toulouse, France. November 2017.

[VIII] J. A. Ruiz-de-Azua, J. F. Muñoz-Martín, A. Solanellas, P. Vía, R. Castellà, L. Fernández, C. Díez, X. Matabosch, Q. Abella, A. Aguilella, N. Miguélez, A. Cortiella, O. Liébana, A. Navarro, and A. Camps. “3Cat-4: a 1U Cubesat Demonstration GNSS-R Mission“. In: 2^{nd} *URSI Atlantic Radio Science Meeting*. Gran Canaria, Spain. May 2018.

[IX] J. A. Ruiz-de-Azua, C. Araguz, A. Calveras, E. Alarcón, A. Camps. “Towards an integral model-based simulator for Earth observation satellite networks.” In: *IEEE International Geoscience and Remote Sensing Symposium*. pp. 7403–7406. Valencia, Spain. July 2018. DOI: {10.1109/IGARSS.2018.8517811}

[X] A. Camps, A. Golkar, A. Gutierrez, J. A. Ruiz-de-Azua, J. F. Munoz-Martin, L. Fernandez, C. Diez, A. Aguilella, S. Briatore, R. Akhtyamov, and N. Garzaniti. “FSSCAT, the 2017 Copernicus Masters'“ESA Sentinel Small Satellite Challenge” Winner: A Federated Polar and Soil Moisture Tandem Mission Based on 6U Cubesats.” In: *IEEE International Geoscience and Remote Sensing Symposium*. pp. 8285–8287. Valencia, Spain. July 2018. DOI: {10.1109/IGARSS.2018.8518405}

[XI] J. A. Ruiz-de-Azua, J. F. Muñoz, L. Fernández, M. Badia, D. Llavería, C. Diez, A. Aguilella, A. Pérez, O. Milian, M. Sobrino, A. Navarro, H. Lleó, M. Sureda, M. Soria, A. Calveras, and A. Camps. “3Cat-4 Mission: A 1-Unit Cubesat for Earth Observation with a L-band Radiometer and a GNSS-Reflectometer using Software Defined Radio.” In: *IEEE International Geoscience and Remote Sensing Symposium*. pp. 8867–8870. Yokohama, Japan. July 2019. DOI: {10.1109/IGARSS.2019.8898317}

[XII] J. A. Ruiz-de-Azua, L. Fernandez, J. F. Muñoz, M. Badia, R. Castella, C. Diez, A. Aguilella, S. Briatore, N. Garzaniti, A. Calveras, A. Golkar, and A. Camps. “Proof-of-concept of a Federated Satellite System between two 6-Unit cubesats for Distributed Earth Observation Satellite Systems.” In: *IEEE International Geoscience and Remote Sensing Symposium*. pp. 8871–8871. Yokohama, Japan. July 2019. DOI: {10.1109/IGARSS.2019.8900099}

[XIII] J. A. Ruiz-de-Azua, V. Ramírez, H. Park, A. Calveras, and A. Camps. “Predictive Algorithms to Assess Inter-Satellite Links Availability in Autonomous Satellite Networks.” In: 70^{th} *International Astronautical Congress*. Washington, USA. October 2019.

[XIV] S. Tonetti, S. Cornara, G. Vicario, S. Pierotti, J. Cote, C. Araguz, E. Alarcón, A. Camps, D. Llaveria, E. Lancheros, J. A. Ruiz-de-Azua, E. Bou-Balust, P. Rodriguez, M. Sochacki, J. Narkiewicz, A. Golkar, I. Lluch i Cruz, H. Matevosyan. “Mission and System Architec-

ture for an Operational Network of Earth Observation Satellite Nodes." In: *70th International Astronautical Congress*. Washington, USA. October 2019.

[XV] C. Araguz, J. A. Ruiz-de-Azua, A. Calveras, A. Camps, E. Alarcón. "Simulating distributed small satellite networks: A model-based tool tailored to decentralized resource-constrained systems." In: *70th International Astronautical Congress*. Washington, USA. October 2019.

[XVI] J. A. Ruiz-de-Azua, J.F. Munoz-Martin, L. Fernandez, M. Badia, D. Llaveria, C. Diez, A. Aguilella, A. Perez, O. Milian, M. Sobrino, A. Navarro, H. Lleó, O. Balcells, M. Sureda, M. Soria, and A. Camps. "The Flexible Microwave Payload - 1: a combined GNSS-Reflectometer and L-band Radiometer for the 3 Cat-4 Cubesat Mission." In: *IEEE GNSS+R*. Benevento, Italy. May 2019.

[XVII] S. Tonetti, S. Cornara, G. Vicario, S. Pierotti, J. Cote, C. Araguz, E. Alarcón, A. Camps, D. Llaveria, E. Lancheros, J. A. Ruiz-de-Azua, E. Bou-Balust, et al. "Mission and System Performance Analyses for an Operational Network of Earth Observation Satellite Nodes." In: *Living Planet Symposium*. Milan, Italy. May 2019.

[XVIII] J. A. Ruiz-de-Azua, L. Fernández, M. Badia, A. Martón, N. Garzaniti, A. Calveras, A. Golkar, A. Camps. "Demonstration of the Federated Satellite Systems concept for Future Earth Observation Satellite Missions." In: *IEEE International Geoscience and Remote Sensing Symposium*. [Accepted]. September 2020.

Supplementary peer-reviewed journal publications during the doctorate:

[XIX] J.F. Munoz-Martin, L. Fernandez, J. A. Ruiz-de-Azua, and A. Camps. "The Flexible Microwave Payload-2: A SDR-Based GNSS-Reflectometer and L-Band Radiometer for CubeSats." In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. Volume 13, pp. 1298-1311. Mars 2020. DOI: {10.1109/JSTARS.2020.2977959}

Supplementary conference publications during the doctorate:

[XX] J. F. Munoz-Martin, L. Fernandez, J. A. Ruiz-de-Azua, and A. Camps. "The Flexible Microwave Payload-2: Architecture and testing of a combined GNSS-R and L-band radiometer with RFI mitigation payload for CubeSat-based Earth Observation Missions." In: *IEEE International Geoscience and Remote Sensing Symposium*. pp. 5185–5188. Yokohama, Japan. July 2019. DOI: {10.1109/IGARSS.2019.8898402}

[XXI] J. F. Munoz-Martin, L. Fernandez, J. A. Ruiz-de-Azua, and A. Camps. "The Flexible Microwave Payload-2: Design, Implementation, and Optimization of a GNSS-R and Radiometry Processor for CubeSat-Based Earth Observation Missions." In: *IEEE International Geoscience and Remote Sensing Symposium*. pp. 5248–5251. Yokohama, Japan. July 2019. DOI: {10.1109/IGARSS.2019.8899300}

[XXII] J.F. Munoz-Martin, L. Fernandez, J. A. Ruiz-de-Azua, and A. Camps. ”The Flexible Microwave Payload -2: A SDR-based GNSS-R instrument for CubeSats.” In: *IEEE GNSS+R*. Benevento, Italy. May 2019.

[XXIII] L. Fernandez, J. A. Ruiz-de-Azua, and A. Camps. ”Deployable L-Band Helix Antenna for a GNSS-R Receiver onboard a 1U CubeSat.” In: *IEEE GNSS+R*. Benevento, Italy. May 2019.

[XXIV] L. Fernandez, M. Sobrino, A. Aguilella, A. Solanellas, M. Badia, J. F. Munoz-Martin, J. A. Ruiz-de-Azua, M. Sureda, and A. Camps. “Deployment mechanism for a L-band Helix Antenna in 1-Unit CubeSat.” In: *70^{th} International Astronautical Congress*. Washington, USA. October 2019.

[XXV] L. Fernández, J. A. Ruiz-de-Azua, A. Calveras, and A. Camps. “Evaluation of LoRa Modulation for data retrieval of ocean monitoring sensors with LEO satellites.” In: *IEEE International Geoscience and Remote Sensing Symposium*. [Accepted]. September 2020.

Bibliography

3GPP (2019a). *Release 15 Description*. Tech. rep. 3GPP TR 21.915 V15.0.0. 3rd Generation Partnership Project.

— (2019b). *Study on New Radio (NR) to support non-terrestrial networks*. Tech. rep. 3GPP TR 38.811 V15.1.0. 3rd Generation Partnership Project.

— (2019c). *Study on using Satellite Access in 5G*. Tech. rep. 3GPP TR 38.822 V16.0.0. 3rd Generation Partnership Project.

Akhtyamov, Rustam, Ignasi Lluch, Hripsime Matevosyan, Dominik Knoll, Udrivolf Pica, Marco Lisi and Alessandro Golkar (2016). 'An implementation of Software Defined Radios for federated aerospace networks: Informing satellite implementations using an inter-balloon communications experiment'. In: *Acta Astronautica* 123, pp. 470–478. DOI: 10.1016/j.actaastro.2015.12.043.

Akyildiz, Ian, Eylem Ekici and Michael Bender (2002). 'MLSR: a novel routing algorithm for multilayered satellite IP networks'. In: *IEEE/ACM Transactions on networking* 10.3, pp. 411–424. DOI: 10.1109/TNET.2002.1012371.

Alarcón, Eduard, Angel Alvaro Sanchez, Carles Araguz, Gilbert Barrot, Elisenda Bou-Balust, Adriano Camps, Stefania Cornara, Judith Cote, Antonio Gutiérrez, Estefany Lancheros, Olivia Lesne, David Llavería, Ignasi Lluch, Joao Malés, Antoine Mangin, Hripsime Matevosyan, Angel Monge, Janusz Narkiewicz, Stephane Ourevitch, Hyuk Park, Stephane Pierotti, Udrivolf Pica, Armen Poghosyan, Pedro odríguez, Joan Adrià Ruiz-de-Azua, Pierre Sicard, Mateusz Sochacki, Stefania Tonetti and Sebastian Topczewski (2018). 'Design and Optimization of a Polar Satellite Mission to Complement the Copernicus System'. In: *IEEE Access* 6, pp. 34777–34789. DOI: 10.1109/ACCESS.2018.2844257.

Alfano, Salvatore and Daniel Oltrogge (2015). 'Volumetric Encounter Analysis Enhancements'. In: *AAS Paper*, pp. 15–581.

— (2018a). 'Full Characterization of Satellite Conjunction Walk-ins'. In: *2018 Space Flight Mechanics Meeting*, p. 2232.

— (2018b). 'Volumetric assessment of satellite encounter rates'. In: *Acta Astronautica* 152, pp. 891–907. DOI: 10.1016/j.actaastro.2018.09.030.

Alfriend, Kyle and Hanspeter Schaub (2000). 'Dynamics and control of spacecraft formations-Challenges and some solutions'. In: *The Richard H. Battin Astrodynamics Symposium, College Station, TX*, pp. 205–223.

Aps, Gomspace (2018). *NanoCom SR2000*. [Online: April 21st, 2020]. URL: https://gomspace.com/UserFiles/Subsystems/datasheet/gs-ds-nanocom-sr2000-20.pdf.

Araguz, Carles (2019). 'In pursuit of autonomous distributed satellite systems'. PhD thesis. Barcelona, Spain: Technical University of Catalonia – UPC BarcelonaTech, Department of Electronics Engineering.

Araguz, Carles, Elisenda Bou-Balust and Eduard Alarcón (2018). 'Applying autonomy to distributed satellite systems: Trends, challenges, and future prospects'. In: *Systems Engineering* 21.5, pp. 401–416. DOI: 10.1002/sys.21428.

Araguz, Carles, Joan Adria Ruiz-de-Azua, Anna Calveras, Adriano Camps and Eduard Alarcon (2019). 'Simulating Distributed Small Satellite Networks: A model-based Tool Tailored to decentralized Resource-constrained systems'. In: *70th International Astronautical Congress (IAC)*. IAF.

Araniti, Giuseppe, Nikolaos Bezirgiannidis, Edward Birrane, Igor Bisio, Scott Burleigh, Carlo Caini, Marius Feldmann, Mario Marchese, John Segui and Kiyohisa Suzuki (2015). 'Contact graph routing in DTN space networks: overview, enhancements and performance'. In: *IEEE Communications Magazine* 53.3, pp. 38–46. DOI: 10.1109/MCOM.2015.7060480.

Atzori, Luigi, Antonio Iera and Giacomo Morabito (2010). 'The internet of things: A survey'. In: *Computer networks* 54.15, pp. 2787–2805. DOI: 10.1016/j.comnet.2010.05.010.

Awange, Joseph and John Kiema (2019). 'Disaster Monitoring and Management'. In: *Environmental Geoinformatics*. Springer, pp. 533–578.

Balasubramanian, Aruna, Brian Levine and Arun Venkataramani (2007). 'DTN routing as a resource allocation problem'. In: *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 373–384.

Baranyai, Lawrence, Enrique Cuevas, Schott Davidow, Christopher Demaree and Paul DiCaprio (2005). 'End-to-end network modeling and simulation of integrated terrestrial, airborne and space environments'. In: *IEEE Aerospace Conference*, pp. 1348–1353.

Barbarić, Dominik, Josip Vuković and Dubravko Babic (2018). 'Link budget analysis for a proposed Cubesat Earth observation mission'. In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, pp. 0133–0138.

Barcelo, Marc, Alejandro Correa, Jose Lopez Vicario, Antoni Morell and Xavier Vilajosana (2016). 'Addressing Mobility in RPL with Position Assisted Metrics'. In: *IEEE Sensors Journal* 16.7, pp. 2151–2161. DOI: 10.1109/JSEN.2015.2500916.

Barrios, Yubal, Antonio J Sánchez, Lucana Santos and Roberto Sarmiento (2020). 'SHyLoC 2.0: A Versatile Hardware Solution for On-Board Data and Hyperspectral Image Compression on Future Space Missions'. In: *Ieee Access* 8, pp. 54269–54287.

Barritt, Brian, Kul Bhasin, Wesley Eddy and Seth Matthews (2010). 'Unified approach to modeling & simulation of space communication networks and systems'. In: *2010 IEEE International Systems Conference*. IEEE, pp. 133–136.

Bate, Roger, Donald Mueller, Jerry White and William Saylor (2019). *Fundamentals of astrodynamics*. Dover publications.

Benz, Ursula, Klaus Strodl and Alberto Moreira (1995). 'A comparison of several algorithms for SAR raw data compression'. In: *IEEE Transactions on geoscience and remote sensing* 33.5, pp. 1266–1276.

Bhasin, Kul and Jeffrey L Hayden (2001). 'Space Internet architectures and technologies for NASA enterprises'. In: *Proceedings of IEEE Aerospace Conference*. Vol. 2, pp. 2–931. DOI: 10.1109/AERO.2001.931275.

Bista, Bhed Bahadur (2016). 'Improving Energy Consumption of Epidemic Routing in Delay Tolerant Networks'. In: *IEEE International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp. 278–283. DOI: 10.1109/IMIS.2016.74.

Bolton, Douglas K and Mark A Friedl (2013). 'Forecasting crop yield using remotely sensed vegetation indices and crop phenology metrics'. In: *Agricultural and Forest Meteorology* 173, pp. 74–84.

Bonin, Grant, Niels Roth, Scott Armitage, Josh Newman, Ben Risi and Robert E Zee (2015). 'CanX–4 and CanX–5 Precision Formation Flight: Mission Accomplished!' In:

Book, Blue (2019). 'Schedule-aware bundle routing'. In: *Consultative Committee for Space Data Systems (CCSDS)*.

Boukerche, Azzedine, Begumhan Turgut, Nevin Aydin, Mohammad Ahmad, Ladislau Bölöni and Damla Turgut (2011). 'Routing protocols in ad hoc networks: A survey'. In: *Computer networks* 55.13, pp. 3032–3080. DOI: 10.1016/j.comnet.2011.05.010.

Boyle, Alan (2020). *Amazon asks FCC to give swift approval to Project Kuiper satellite network despite SpaceX opposition*. https://www.geekwire.com/2020/amazon-asks-fcc-give-swift-approval-project-kuiper-satellite-network-despite-spacex-opposition/. [Online; accessed 17th July 2020].

Braun, Vitali, André Horstmann, Benedikt Reihs, Stijn Lemmens, Klaus Merz and Holger Krag (2019). 'Exploiting orbital data and observation campaigns to improve space debris models'. In: *The Journal of the Astronautical Sciences* 66.2, pp. 192–209.

Briatore, S, R Akhtyamov and A Golkar (2017). 'Design and flight test results of high speed optical bidirectional link between stratospheric platforms for aerospace applications'. In: *Laser Communication and Propagation through the Atmosphere and Oceans VI*. Vol. 10408. International Society for Optics and Photonics, p. 1040804.

Brogi, Antonio, Evelina Lamma and Paola Mello (1992). 'Objects in a logic programming framework'. In: *Logic Programming*. Springer, pp. 102–113.

Brown, Owen and Paul Eremenko (2006). *Fractionated space architectures: A vision for responsive space*. Tech. rep. DEFENSE ADVANCED RESEARCH PROJECTS AGENCY ARLINGTON VA.

Burgess, John, Brian Gallagher, David D Jensen, Brian Neil Levine et al. (2006). 'MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks.' In: *Infocom*. Vol. 6. Barcelona, Spain, pp. 1–11.

Burleigh, S, Carlo Caini, Jako Jo Messina and Michele Rodolfi (2016). 'Toward a unified routing framework for delay-tolerant networking'. In: *2016 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*. IEEE, pp. 82–86.

Burleigh, Scott (2007). 'Interplanetary overlay network: An implementation of the dtn bundle protocol'. In:

— (2010). *Contact Graph Routing*. Internet-Draft.

Cai, Guowei, Ben M Chen and Tong Heng Lee (2011). 'Coordinate systems and transformations'. In: *Unmanned rotorcraft systems*. Springer, pp. 23–34.

Caleb, Henry (2017). *Q&A | Telesat's Erwin Hudson opens up about LEO mega-constellation plans*. https://spacenews.com/qa-telesats-erwin-hudson-opens-up-about-leo-mega-constellation-plans/. [Online; accessed 17th July 2020].

— (2019a). *Lacuna Space aims to ride IoT wave with a 32-cubesat constellation*. https://spacenews.com/lacuna-space-aims-to-ride-iot-wave-with-a-32-cubesat-constellation/. [Online; accessed 22-April-2020].

— (2019b). *Megaconstellation ventures cautious about deployment milestones*. https://spacenews.com/megaconstellation-ventures-cautious-about-deployment-milestones/. [Online; accessed 17th July 2020].

— (2019c). 'What Airbus learned from building satellites with OneWeb'. In: *Spacenews*. [Online: July 1st, 2020].

— (2020a). *After bankruptcy, OneWeb's supply chain looking for new ways to keep busy*. https://spacenews.com/after-bankruptcy-onewebs-supply-chain-looking-for-new-ways-to-keep-busy/. [Online; accessed 17th July 2020].

— (2020b). *Bankruptcy court frees payment to OneWeb Satellites to restart satellite manufacturing*. https://spacenews.com/bankruptcy-court-frees-payment-to-oneweb-satellites-to-restart-satellite-manufacturing/. [Online; accessed 17th July 2020].

Caleb, Henry (2020c). *OneWeb files for Chapter 11 bankruptcy*. https://spacenews.com/oneweb-files-for-chapter-11-bankruptcy/. [Online; accessed 17th July 2020].

Cameron, Bruce G, Edward F Crawley, Geilson Loureiro and Eric S Rebentisch (2008). 'Value flow mapping: Using networks to inform stakeholder analysis'. In: *Acta Astronautica* 62.4-5, pp. 324–333.

Camps, Adriano, Alessandro Golkar, Antonio Gutierrez, Joan Adrià Ruiz-de-Azua, Joan Francesc Munoz-Martin, Lara Fernandez, Carlos Diez, Andrea Aguilella, Simone Briatore, Rustam Akhtyamov and Nicola Garzaniti (2018). 'Fsscat, the 2017 Copernicus Masters' "Esa Sentinel Small Satellite Challenge" Winner: A Federated Polar and Soil Moisture Tandem Mission Based on 6U Cubesats'. In: *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 8285–8287. DOI: 10.1109/IGARSS.2018.8518405.

Canada, Telesat (2018). *Telesat Ka-band NGSO constellation FCC filing SAT-PDR-20161115-00108*.

Carrel, Andrew R and Phil L Palmer (2005). 'An evolutionary algorithm for near-optimal autonomous resource management'. In: *8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*.

Carrier, Brian (2005). *File system forensic analysis*. Addison-Wesley Professional.

Chien, Steve, Mark Johnston, Jeremy Frank, Mark Giuliano, Alicia Kavelaars, Christoph Lenzen and Nicola Policella (2012). 'A generalized timeline representation, services, and interface for automating space mission operations'. In: *SpaceOps 2012*, p. 1275459.

Clausen, Thomas and Philippe Jacquet (2003). *Optimized link state routing protocol (OLSR)*. RFC 3626. RFC Editor.

Clohessy, WH and RS Wiltshire (1960). 'Terminal guidance system for satellite rendezvous'. In: *Journal of the Aerospace Sciences* 27.9, pp. 653–658.

Das, Priyanka, Prosenjit Chowdhury, Bikash Poudel and Tanmay De (2016). 'Fibonary Spray and Wait Routing in Delay Tolerant Networks'. In: *International Journal of Electrical and Computer Engineering* 6.6, p. 3205.

Demmer, Michael and Kevin Fall (2007). 'DTLSR: Delay tolerant routing for developing regions'. In: *Proceedings of the 2007 workshop on Networked systems for developing regions*. ACM, p. 5. DOI: 10.1145/1326571.1326579.

Diana, Rémi, Emmanuel Lochin, Laurent Franck, Cédric Baudoin, Emmanuel Dubois and Patrick Gelard (2017). 'DTN routing for quasi-deterministic networks with application to LEO constellations'. In: *International Journal of Satellite Communications and Networking* 35.2, pp. 91–108.

Dijkstra, Edsger W (1959). 'A note on two problems in connexion with graphs'. In: *Numerische mathematik* 1.1, pp. 269–271.

Directory, eoPortal. *TerraSAR-X Mission*. [Online: May 12th, 2020]. URL: https://earth.esa.int/web/eoportal/satellite-missions/t/terrasar-x.

Doroshkin, Alexander A, Alexander M Zadorozhny, Oleg N Kus, Vitaliy Yu Prokopyev and Yuri M Prokopyev (2019). 'Experimental Study of LoRa Modulation Immunity to Doppler Effect in CubeSat Radio Communications'. In: *IEEE Access* 7, pp. 75721–75731.

Ehsanfar, Abbas and Paul T Grogan (2020a). 'Auction-Based Algorithms for Routing and Task Scheduling in Federated Networks'. In: *Journal of Network and Systems Management* 28.2, pp. 271–297.

— (2020b). 'Mechanism design for exchanging resources in federated networks'. In: *Journal of Network and Systems Management* 28.1, pp. 108–132.

Ekici, Eylem, Ian F Akyildiz and Michael D Bender (2000). 'Datagram routing algorithm for LEO satellite networks'. In: *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Vol. 2. IEEE, pp. 500–508. DOI: 10.1109/INFCOM.2000.832223.

— (2001). 'A distributed routing algorithm for datagram traffic in LEO satelitte networks'. In: *IEEE/ACM Transactions on Networking (TON)* 9.2, pp. 137–147. DOI: 10.1109/90.917071.

Ekici, Eylem, Ian Akyildiz and Michael D. Bender (2002). 'A multicast routing algorithm for LEO satellite IP networks'. In: *IEEE/ACM Transactions On Networking* 10.2, pp. 183–192. DOI: 10.1109/90.993300.

Esposito, Marco, SS Conticello, Massimiliano Pastena and Bernardo Domínguez (2019). 'In-orbit demonstration of artificial intelligence applied to hyperspectral and thermal sensing from space'. In: *CubeSats and SmallSats for Remote Sensing III*. Vol. 11131. International Society for Optics and Photonics, p. 111310C.

European Union, Council of the (2014). *Regulation (EU) No 377/2014 of the European Parliament and of the Council of 3 April 2014 establishing the Copernicus Programme and repealing Regulation (EU) No 911/2010*.

Exploration, Holdings Space (2018). *LLC: SpaceX Ka-band NGSO constellation FCC filing SAT-LOA-20161115-00118*.

Faber, Nicolas, Yosuke Nakamura, Richard Alena, David Mauro, Chad R Frost, Gokul Bhat and Janise McNair (2014). 'Heterogeneous Spacecraft Networks: General concept and case study of a cost-effective, multi-institutional Earth observation platform'. In: *2014 IEEE Aerospace Conference*. IEEE, pp. 1–16.

Fernandez, Lara, Joan Adria Ruiz-de-Azua, Anna Calveras and Adriano Camps (2020). 'Evaluation of LoRa for data retrieveal of ocean monitoring sensors with LEO satellites'. In: *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. [Accepted]. IEEE.

Fischer, Daniel, David Basin, Knut Eckstein and Thomas Engel (2013). 'Predictable mobile routing for spacecraft networks'. In: *IEEE Transactions on Mobile Computing* 12.6, pp. 1174–1187. DOI: 10.1109/TMC.2012.93.

Forte Véliz, Giuseppe Francesco (2014). 'Contributions to Radio Frequency Interference detection and mitigation in Earth observation'. PhD thesis.

Foust, Jeff (2019). *OneWeb Satellites inaugurates Florida factory*. https://spacenews.com/oneweb-satellites-inaugurates-florida-factory/. [Online; accessed 17th July 2020].

— (2020). *Megaconstellation ventures cautious about deployment milestones*. https://spacenews.com/starlink-vs-the-astronomers/. [Online; accessed 17th July 2020].

Fraire, Juan A and Jorge M Finochietto (2015). 'Design challenges in contact plans for disruption-tolerant satellite networks'. In: *IEEE Communications Magazine* 53.5, pp. 163–169.

FriendlyARM (2015). *NanoPi NEO*. http://nanopi.io/nanopi-neo.html. [Online; accessed 20-June-2020].

Friis, Harald T (1946). 'A note on a simple transmission formula'. In: *Proceedings of the IRE* 34.5, pp. 254–256.

Gabbatiss, Josh (2018). 'First ship crosses Arctic in winter without an icebreaker as global warming causes ice sheets to melt'. In: *Independent journal*.

Gaffney, Leah M, Neal D Hulkower and Leslie Klein (1996). 'Non-GEO mobile satellite systems: a risk assessment'. In: *Space communications* 14, pp. 123–129.

Gates, Dominic (2015). *Elon Musk touts launch of 'SpaceX Seattle'*. https://www.seattletimes.com/business/elon-musk-touts-launch-of-lsquospacex-seattlersquo/. [Online; accessed 17th July 2020].

Generation Partnership Project, 3rd (2020). *3GPP: The Mobile Broadband Standard*. https://www.3gpp.org/. [Online; accessed 22-April-2020].

Geyer, Michael, Falk Mrowka and Christoph Lenzen (2010). 'TerraSAR-X/TanDEM-X mission planning-handling satellites in close formation'. In: *SpaceOps 2010 Conference Delivering on the Dream Hosted by NASA Marshall Space Flight Center and Organized by AIAA*, p. 1989.

Giraldo Rodriguez, Carlos, Laurent Franck, Cédric Baudoin and André-Luc Beylot (2010). 'Mobile ad hoc network assisted by satellites'. In: *28th AIAA International Communications Satellite Systems Conference (ICSSC-2010)*, p. 8716.

Gnawali, Omprakash (2012). *The Minimum Rank with Hysteresis Objective Function (MRHOF)*. RFC 6719. RFC Editor.

Golkar, Alessandro (2013a). 'Architecting Federated Satellite Systems for Successful Commercial Implementation'. In: *AIAA SPACE 2013 Conference and Exposition*, p. 5417.

— (2013b). 'Federated satellite systems: an innovation in space systems design'. In: *9th IAA Symposium on Small Satellites for Earth Observation. International Academy of Astronautics, Berlin, Germany*. Vol. 194.

— (2013c). 'Federated Satellite Systems: Leverages and Threats Identification by Stakeholders Value Network Modeling'. In: *31st AIAA International Communications Satellite Systems Conferences*, p. 5630.

Golkar, Alessandro and Ignasi Lluch (2015). 'The Federated Satellite Systems paradigm: Concept and business case evaluation'. In: *Acta Astronautica* 111, pp. 230–248. DOI: 10.1016/j.actaastro.2015.02.009.

Gonizzi, Pietro, Riccardo Monica and Gianluigi Ferrari (2013). 'Design and evaluation of a delay-efficient RPL routing metric'. In: *9th IEEE Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1573–1577. DOI: 10.1109/IWCMC.2013.6583790.

Gosling, James, Bill Joy, Guy Steele and Gilad Bracha (2000). *The Java language specification*. Addison-Wesley Professional.

Gounder, Vidyashankar V., Ravi Prakash and Hosame Abu-Amara (1999). 'Routing in LEO-based satellite networks'. In: *IEEE Wireless Communications and Systems: Emerging Technologies Symposium*, pp. 22–1. DOI: 10.1109/ETWCS.1999.897340.

Gregory, Mark, Frank F Heine, Hartmut Kämpfner, Robert Lange, Michael Lutzer and Rolf Meyer (2012). 'Commercial optical inter-satellite communication at high data rates'. In: *Optical Engineering* 51.3, p. 031202.

Grogan, Paul T, Alessandro Golkar, Seiko Shirasaka and Olivier L De Weck (2014a). 'Multi-stakeholder interactive simulation for federated satellite systems'. In: *2014 IEEE Aerospace Conference*. IEEE, pp. 1–15.

— (2014b). 'Multi-stakeholder interactive simulation for federated satellite systems'. In: *Aerospace Conference, 2014 IEEE*. IEEE, pp. 1–15. DOI: 10.1109/AERO.2014.6836253.

Grogan, Paul T., Koki Ho, Alessandro Golkar and Olivier de Weck (2016a). 'Bounding the value of collaboration in federated systems'. In: *Proceedings of Annual IEEE Systems Conference*, pp. 1–7. DOI: 10.1109/SYSCON.2016.7490657.

Grogan, Paul T, Koki Ho, Alessandro Golkar and Olivier L De Weck (2016b). 'Multi-actor value modeling for federated systems'. In: *IEEE Systems Journal* 12.2, pp. 1193–1202.

Grogan, Paul Thomas (2014). 'Interoperable simulation gaming for strategic infrastructure systems design'. PhD thesis. Massachusetts Institute of Technology.

Guangyu, Pei, Gerla Mario and Chen Tsu-Wei (2000). 'Fisheye state routing in mobile ad hoc networks'. In: *Proceedings of ICC*. s 1, pp. 71–78. DOI: 10.1109/ICC.2000.853066.

Guerra, André GC, Frederico Francisco, Jaime Villate, Fernando Aguado Agelet, Orfeu Bertolami and Kanna Rajan (2016). 'On small satellites for oceanography: A survey'. In: *Acta Astronautica* 127, pp. 404–423.

Haas, Zygmunt, Marc Pearlman and Prince Samar (2002). *The zone routing protocol (ZRP) for ad hoc networks*. Internet-Draft. Mobile Ad-Hoc Network (MANET) Working Group, IETF.

Haerri, Jerome, Fethi Filali and Christian Bonnet (2006). 'Performance comparison of AODV and OLSR in VANETs urban environments under realistic mobility patterns'. In: *Proceedings of the 5th IFIP mediterranean ad-hoc networking workshop*. i, pp. 14–17.

Handley, Mark (2018). 'Delay is not an option: Low latency routing in space'. In: *Proceedings of the 17th ACM Workshop on Hot Topics in Networks*, pp. 85–91.

Harris, Mark (2019). *Why satellite mega-constellations are a threat to the future of space*. https://www.technologyreview.com/2019/03/29/136268/why-satellite-mega-constellations-are-a-massive-threat-to-safety-in-space/. [Online; accessed 17th July 2020].

Hastings, Daniel E and Carole Joppin (2006). 'On-orbit upgrade and repair: The hubble space telescope example'. In: *Journal of spacecraft and rockets* 43.3, pp. 614–625.

He, Feng, Qin Liu, Tao Lv, Chunyan Liu, Hejiao Huang and Xiaohua Jia (2016). 'Delay-bounded and minimal transmission broadcast in LEO satellite networks'. In: *2016 IEEE International Conference on Communications (ICC)*, pp. 1–7. DOI: 10.1109/ICC.2016.7510734.

He, Jia-fu, Jiang Yong, Bian Dong-ming and Guang-xia Li (2008). 'Routing strategy research based on ISL states and topology snapshot in LEO satellite constellation'. In: *11th IEEE International Conference on Communication Technology (ICCT)*, pp. 13–16. DOI: 10.1109/ICCT.2008.4716089.

Heidt, Hank, Jordi Puig-Suari, Augustus Moore, Shinichi Nakasuka and Robert Twiggs (2000). 'CubeSat: A new generation of picosatellite for education and industry low-cost space experimentation'. In: *14th Annual/USU Conference on Small Satellites*. Utah State University. Digital Commons.

Heine, Frank, Gerd Mühlnikel, Herwig Zech, Sabine Philipp-May and Rolf Meyer (2014). 'The European Data Relay System, high speed laser based data links'. In: *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, pp. 284–286. DOI: 10.1109/ASMS-SPSC.2014.6934556.

Heinzelman, Wendi Rabiner, Anantha Chandrakasan and Hari Balakrishnan (2000). 'Energy-efficient communication protocol for wireless microsensor networks'. In: *33rd IEEE International Conference on System sciences*, pp. 1–10.

Henderson, Thomas Ross (1999). 'Networking over next-generation satellite systems'. PhD thesis.

Highfield, Vaughn (2018). *Elon Musk claims SpaceX's Starlink internet satellite service will be IP-less*. https://www.alphr.com/space/1008632/Elon-Musk-SpaceX-Starlink-internet. [Online; accessed 17th July 2020].

Hill, George William (1878). 'Researches in the Lunar Theory'. In: *American journal of Mathematics* 1.1, pp. 5–26.

Huang, Bormin (2011). *Satellite data compression*. Springer Science & Business Media.

Ilcev, Stojce Dimov (2011). 'Orbcomm space segment for mobile satellite system (MSS)'. In: *2011 10th International Conference on Telecommunication in Modern Satellite Cable and Broadcasting Services (TELSIKS)*. Vol. 2. USA, pp. 689–692. DOI: 10.1109/TELSKS.2011.6143205.

Inagaki, Tokuya and Susumu Ishihara (2009). 'HGAF: A power saving scheme for Wireless Sensor Networks'. In: *Information and Media Technologies* 4.4, pp. 1086–1097. DOI: 10.2197/ipsjjip.17.255.

Instruments, Texas (2020). *CC1101 Low Power Sub-1 GHz RF Transceiver*. https://www.ti.com/lit/ds/symlink/cc1101.pdf. [Online; accessed 20-June-2020].

Jabbar, Waheb A, Mahamod Ismail, Rosdiadee Nordin and Suki Arif (2016). 'Power-efficient routing schemes for MANETs: a survey and open issues'. In: *Wireless Networks*, pp. 1–36. DOI: 10.1007/s11276-016-1263-6.

Jahnke, Stephan Siegfried, Claudia Philpot, Volker Maiwald and Oliver Romberg (2017). 'Recent Developments on DLRs Post-ISS Concept'. In: *International Astronautical Congress (IAC)*.

Jilla, Cyrrus D. (May 2002). 'A Multiobjective, Multidisciplinary Design Optimization Methodology for the Conceptual Design of Distributed Satellite Systems'. PhD thesis. Cambridge, Massachusetts: Massachusetts Institute of Technology, Department of Aeronautics and Astronautics.

Kashitani, Tatsuki (2002). 'Development and application of an analysis methodology for satellite broadband network architectures'. In: *20th AIAA International Communication Satellite Systems Conference and Exhibit*, p. 2019.

Kawamoto, Yuichi, Hiroki Nishiyama, Nei Kato and Naoto Kadowaki (2013). 'A traffic distribution technique to minimize packet delivery delay in multilayered satellite networks'. In: *IEEE Transactions on Vehicular Technology* 62.7, pp. 3315–3324. DOI: 10.1109/TVT.2013.2256812.

Kelso, Thomas Sean (1985). *CelesTrak*. https://celestrak.com. [Online: April 22nd, 2020].

Kerr, Yann H, Philippe Waldteufel, Jean-Pierre Wigneron, Steven Delwart, François Cabot, Jacqueline Boutin, Maria-José Escorihuela, Jordi Font, Nicolas Reul, Claire Gruhier et al. (2010). 'The SMOS mission: New tool for monitoring key elements ofthe global water cycle'. In: *Proceedings of the IEEE* 98.5, pp. 666–687.

Kim, Hyung-Sin, Jeongyeup Paek and Saewoong Bahk (2015). 'QU-RPL: Queue utilization based RPL for load balancing in large scale industrial applications'. In: *12th IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pp. 265–273. DOI: 10.1109/SAHCN.2015.7338325.

Kodheli, Oltjon, Eva Lagunas, Nicola Maturo, Shree Krishna Sharma, Bhavani Shankar, JF Montoya, JC Duncan, Danilo Spano, Symeon Chatzinotas, Steven Kisseleff et al. (2020). 'Satellite Communications in the New Space Era: A Survey and Future Challenges'. In: *arXiv preprint arXiv:2002.08811*.

Korçak, Ömer, Fatih Alagöz and Abbas Jamalipour (2007). 'Priority-based adaptive routing in NGEO satellite networks'. In: *International journal of communication systems* 20.3, pp. 313–333. DOI: 10.1002/dac.823.

Kramer, Herbert J and Arthur P Cracknell (2008). 'An overview of small satellites in remote sensing'. In: *International journal of remote Sensing* 29.15, pp. 4285–4337.

Kurosu, Takashi, Masaharu Fujita and Kazuo Chiba (1995). 'Monitoring of rice crop growth from space using the ERS-1 C-band SAR'. In: *IEEE Transactions on Geoscience and Remote Sensing* 33.4, pp. 1092–1096.

Lacan, Jérôme, Vincent Roca, Jani Peltotalo and Sami Peltotalo (2009). 'Reed-Solomon forward error correction (FEC) schemes'. In: *IETF RFC 5510*.

Lancheros, Estefany, Adriano Camps, Hyuk Park, Pierre Sicard, Antoine Mangin, Hripsime Matevosyan and Ignasi Lluch (2018). 'Gaps Analysis and Requirements Specification for the Evolution of Copernicus System for Polar Regions Monitoring: Addressing the Challenges in the Horizon 2020–2030'. In: *Remote Sensing (MDPI)* 10.7. ISSN: 2072-4292. DOI: 10.3390/rs10071098.

LeBrun, Jason, Chen-Nee Chuah, Dipak Ghosal Ghosal and Michael Zhang (2005). 'Knowledge-based opportunistic forwarding in vehicular wireless ad hoc networks'. In: *61st IEEE Vehicular technology conference*. Vol. 4, pp. 2289–2293. DOI: 10.1109/VETECS.2005.1543743.

L'Ecuyer, Tristan S and Jonathan H Jiang (2010). 'Touring the atmosphere aboard the A-Train'. In: *Phys. Today* 63.7, pp. 36–41.

Lee, Jae-Wook, Jun-Woo Lee, Tae-Wan Kim and Dae-Ung Kim (2000). 'Satellite over satellite (SOS) network: A novel architecture for satellite network'. In: *IEEE INFOCOM*. Vol. 1, pp. 315–321. DOI: 10.1109/INFCOM.2000.832201.

Leguay, Jérémie, Timur Friedman and Vania Conan (2005). 'DTN routing in a mobility pattern space'. In: *ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 276–283. DOI: 10.1145/1080139.1080146.

Léon, Laura, Per Koch and Roger Walker (2018). 'GOMX-4-The Twin European Mission for IOD Purposes'. In: 32^{nd} *Annual AIAA/USU Conference on Small Satellites*.

Leonov, Alexey V and George A Litvinov (2018). 'Applying AODV and OLSR routing protocols to air-to-air scenario in flying ad hoc networks formed by mini-UAVs'. In: *2018 Systems of Signals Generating and Processing in the Field of on Board Communications*. IEEE, pp. 1–10.

Lindgren, A, E Davies, S Grasic and A Doria (2012). *Probabilistic routing protocol for intermittently connected networks*. RFC 6693. RFC Editor.

Lindgren, Anders, Avri Doria and Olov Schelén (2003). 'Probabilistic routing in intermittently connected networks'. In: *ACM SIGMOBILE mobile computing and communications review* 7.3, pp. 19–20.

Liolis, Konstantinos, Alexander Geurtz, Ray Sperber, Detlef Schulz, Simon Watts, Georgia Poziopoulou, Barry Evans, Ning Wang, Oriol Vidal, Boris Tiomela Jou et al. (2019). 'Use cases and scenarios of 5G integrated satellite-terrestrial networks for enhanced mobile broadband: The SaT5G approach'. In: *International Journal of Satellite Communications and Networking* 37.2, pp. 91–112.

Liu, Cong and Jie Wu (2008). 'Routing in a cyclic mobispace'. In: *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*. ACM, pp. 351–360. DOI: 10.1145/1374618.1374665.

Liu, Runzi, Min Sheng, King-Shan Lui, Xijun Wang, Di Zhou and Yu Wang (2015). 'Capacity analysis of two-layered LEO/MEO satellite networks'. In: *81st IEEE Vehicular Technology Conference*, pp. 1–5. DOI: 10.1109/VTCSpring.2015.7145726.

Llatser, Ignacio, Guillaume Jornod, Andreas Festag, David Mansolino, Iñaki Navarro and Alcherio Martinoli (2017). 'Simulation of cooperative automated driving by bidirectional coupling of vehicle and network simulators'. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 1881–1886.

Lluch, Ignasi (2017). 'A framework for architecting federations of systems'. PhD thesis. Moscow: Skolkovo Institute of Science and Technology.

Lluch, Ignasi and Alessandro Golkar (2014a). 'Resource balancing analysis of federated satellite systems'. In: *AIAA SPACE 2014 Conference and Exposition*, p. 4270.

— (2014b). 'Satellite-to-satellite coverage optimization approach for opportunistic inter-satellite links'. In: *Aerospace Conference, 2014 IEEE*. IEEE, pp. 1–13. DOI: 10.1109/AERO.2014.6836307.

— (2015a). 'Design Implications for Missions Participating in Federated Satellite Systems'. In: *Journal of Spacecraft and Rockets* 52.5. DOI: 10.2514/1.A33172.

— (2015b). 'The value proposition of novel distributed space systems: a literature survey on Ilities'. In: *3rd Federated Satellite Systems Workshop*, pp. 1–8.

— (2019). 'Architecting federations of systems: A framework for capturing synergy'. In: *Systems Engineering* 22.4, pp. 295–312.

Lluch, Ignasi, Paul T Grogan, Udrivolf Pica and Alessandro Golkar (2015). 'Simulating a proactive ad-hoc network protocol for Federated Satellite Systems'. In: *Proceedings of IEEE Aerospace Conference*, pp. 1–16. DOI: 10.1109/AERO.2015.7118984.

Lu, Xiaofeng and Pan Hui (2010). 'An energy-efficient n-epidemic routing protocol for delay tolerant networks'. In: *IEEE 5th International Conference on Networking, Architecture and Storage*, pp. 341–347.

Lu, Yong, Fuchun Sun and Youjian Zhao (2013). 'Virtual topology for LEO satellite networks based on earth-fixed footprint mode'. In: *IEEE communications letters* 17.2, pp. 357–360. DOI: 10.1109/LCOMM.2013.011113.122635.

Lui, Gee L and Kuang Tsai (2006). *Gaussian minimum shift keying (GMSK) precoding communication method*. US Patent 7,072,414.

Lunden, I (2014). 'Google Confirms Purchase of Satellite Startup Skybox Imaging from 500 M'. In: *TechCrunch, AOL, Inc*.

Macker, Joseph (1999). *Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations*. RFC 2501. RFC Editor.

Macmillan, S and Nils Olsen (2013). 'Observatory data and the Swarm mission'. In: *Earth, Planets and Space* 65.11, p. 15.

Maier, Mark W (1998). 'Architecting principles for systems-of-systems'. In: *Systems Engineering: The Journal of the International Council on Systems Engineering* 1.4, pp. 267–284.

Marina, Mahesh K and Samir R Das (2006). 'Ad hoc on-demand multipath distance vector routing'. In: *Wireless communications and mobile computing* 6.7, pp. 969–988. DOI: 10.1002/wcm.432.

Marshall, William and Chris Boshuizen (2013). 'Planet labs' remote sensing satellite system'. In:

Matevosyan, Hripsime, Ignasi Lluch, Armen Poghosyan and Alessandro Golkar (2017). 'A Value-Chain Analysis for the Copernicus Earth Observation Infrastructure Evolution: A Knowledge-base of Users, Needs, Services, and Products'. In: *IEEE Geoscience and Remote Sensing Magazine* 5.3, pp. 19–35.

Maurich, Olga von and Alessandro Golkar (2018). 'Data authentication, integrity and confidentiality mechanisms for federated satellite systems'. In: *Acta Astronautica* 149, pp. 61–76.

Merts, Andre-Jan and Arno Barnard (2016). 'Simulating MANETS: A study using satellites with AODV and AntHocNet'. In: *Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference*. IEEE, pp. 1–5.

Mignone, Vittoria, Maria Angeles Vazquez-Castro and Thomas Stockhammer (2011). 'The future of satellite TV: The wide range of applications of the DVB-S2 standard and perspectives'. In: *Proceedings of the IEEE* 99.11, pp. 1905–1921.

Moeneclaey, Marc, Herwig Bruneel, Ignace Bruyland and Doo-Young Chung (1986). 'Throughput optimization for a generalized stop-and-wait ARQ scheme'. In: *IEEE transactions on communications* 34.2, pp. 205–207.

Moreira, Adriano JC, Rui T Valadas and AM de Oliveira Duarte. 'IEEE 802.11 Wireless Access Method and Physical Layer Specification'. In: ().

Mueller, Ivan I (1985). 'Reference coordinate systems and frames: Concepts and realization'. In: *Journal of Geodesy* 59.2, pp. 181–188.

Munoz-Martin, Joan Francesc, Lara Fernandez, Joan Adrià Ruiz-de-Azua and Adriano Camps (2019). 'The Flexible Microwave Payload-2: Architecture and testing of a combined GNSS-R and L-band radiometer with RFI mitigation payload for CubeSat-based Earth Observation Missions'. In: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, pp. 5185–5188.

Munoz-Martin, Joan Francesc, Lara Fernandez Capon, Joan Adria Ruiz-de-Azua and Adriano Camps (2020). 'The Flexible Microwave Payload-2: A SDR-Based GNSS-Reflectometer and L-Band Radiometer for CubeSats'. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13, pp. 1298–1311.

Neumann, Axel, Corinna Aichele, Marek Lindner and Simon Wunderlich (2008). 'Better approach to mobile ad-hoc networking (BATMAN)'. In: *IETF draft*, pp. 1–24.

Neumann, Axel, Ester López and Leandro Navarro (2012). 'An evaluation of bmx6 for community wireless networks'. In: *2012 IEEE 8th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, pp. 651–658.

Niehoefer, Brian, Sebastian Šubik and Christian Wietfeld (2013). 'The CNI Open Source Satellite Simulator based on OMNeT++'. In: *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*, pp. 314–321.

Nsnam (2019). *ESA Summer of Code in Space 2019*. https://www.nsnam.org/wiki/SOCIS2019.

O'Callaghan, Jonathan (2019). *The Risky Rush for Mega Constellations*. https://www.scientificamerican.com/article/the-risky-rush-for-mega-constellations/. [Online; accessed 17th July 2020].

ONION, Consortium (2016). *Stakeholders needs: user needs analysis and Earth observation infrastructure state of the art assessment*. Tech. rep. D2.2. Publicly available deliverable. Operational Network of Individual Observation Nodes.

Organization, World Meteorological (2011). *Observing Systems Capability Analysis and Review Tool*. https://www.wmo-sat.info/oscar/spacecapabilities. [Online: April 22nd, 2020].

Papapetrou, Evangelos and Fotini-Niovi Pavlidou (2008). 'Distributed load-aware routing in LEO satellite networks'. In: *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–5. DOI: 10.1109/GLOCOM.2008.ECP.565.

Papapetrou, Evangelos, Stylianos Karapantazis and F-N Pavlidou (2007). 'Distributed on-demand routing for LEO satellite systems'. In: *Computer Networks* 51.15, pp. 4356–4376. DOI: 10.1016/j.comnet.2007.05.008.

Perkins, Charles, Elizabeth Belding-Royer and Samir Das (2003). *Ad hoc on-demand distance vector (AODV) routing*. RFC 3561. RFC Editor.

Perkins, Charles E and Pravin Bhagwat (1994). 'Highly Dynamic Destination Sequence Distance Vector (DSDV) for Moblie Computers'. In: *Proceedings of the SIGCOMM 1994 Conference on Communications Architectures, Protocols and Applications*, pp. 234–244.

Pica, Udrivolf and Alessandro Golkar (2017). 'Sealed-Bid Reverse Auction Pricing Mechanisms for Federated Satellite Systems'. In: *Systems Engineering* 20.5, pp. 432–446.

Pinter Jr, Paul J, Jerry L Hatfield, James S Schepers, Edward M Barnes, M Susan Moran, Craig ST Daughtry and Dan R Upchurch (2003). 'Remote sensing for crop management'. In: *Photogrammetric Engineering & Remote Sensing* 69.6, pp. 647–664.

Portillo, Inigo del, Bruce G Cameron and Edward F Crawley (2019). 'A Technical Comparison of Three Low Earth Orbit Satellite Constellation Systems to Provide Global Broadband'. In: *Acta Astronautica* 159, pp. 123–135.

Possel, Markus (2017). 'Waves, motion and frequency: the Doppler effect'. In: *Einstein Online* 5.

Pratt, Stephen R, Richard A Raines, Carl E Fossa and Michael A Temple (1999). 'An operational and performance overview of the IRIDIUM low earth orbit satellite system'. In: *IEEE Communications Surveys* 2.2, pp. 2–10. DOI: 10.1109/COMST.1999.5340513.

Puig-Suari, Jordi, Clark Turner and William Ahlgren (2001). 'Development of the standard CubeSat deployer and a CubeSat class PicoSatellite'. In: *2001 IEEE aerospace conference proceedings (cat. No. 01TH8542)*. Vol. 1. IEEE, pp. 1–347.

Pultarova, Tereza (2015). 'Telecommunications-Space tycoons go head to head over mega satellite network [News Briefing]'. In: *Engineering & Technology* 10.2, pp. 20–20. DOI: 10.1049/et.2015.0221.

Puttonen, Jani, Budiarto Herman, Sami Rantanen, Frans Laakso and Janne Kurjenniemi (2015). 'Satellite Network Simulator 3'. In: *Workshop on Simulation for European Space Programmes (SESP)*. Vol. 24, p. 26.

Querol, Jorge (2018). 'Radio frequency interference detection and mitigation techniques for navigation and Earth observation'. PhD thesis.

Radhakrishnan, Radhika, William W. Edmonson, Fatemeh Afghah, Ramon Martinez Rodriguez-Osorio, Frank Pinto and Scott C. Burleigh (2016). 'Survey of Inter-satellite Communication for Small Satellite Systems: Physical Layer to Network Layer View'. In: *IEEE Communications Surveys & Tutorials* 18.4, pp. 2442–2473. DOI: 10.1109/COMST.2016.2564990.

Raman, Sanjay, Robert Weigel and Tim Lee (2016). 'The Internet of space (IoS): A future backbone for the Internet of Things?' In: *IEEE IoT Newsletter*.

Rao, Yuan and Ru-chuan Wang (2010). 'Agent-based load balancing routing for LEO satellite networks'. In: *Computer networks* 54.17, pp. 3187–3195. DOI: 10.1016/j.comnet.2010.06.019.

Ratasuk, Rapeepat, Nitin Mangalvedhe, Yanji Zhang, Michel Robert and Jussi-Pekka Koskinen (2016). 'Overview of narrowband IoT in LTE Rel-13'. In: *2016 IEEE conference on standards for communications and networking (CSCN)*. IEEE, pp. 1–7.

Raza, Usman, Parag Kulkarni and Mahesh Sooriyabandara (2017). 'Low Power Wide Area Networks: An overview'. In: *IEEE Communications Surveys & Tutorials* 19.2, pp. 855–873.

Richey, Randal (1986). *Determining the probability of close approach between two satellites*. Tech. rep. Air Force Institute of Technology Wright-Patterson Air Force Base.

Romano, Diego, Marco Lapegna, Valeria Mele and Giuliano Laccetti (2020). 'Designing a GPU-parallel algorithm for raw SAR data compression: A focus on parallel performance estimation'. In: *Future Generation Computer Systems*.

Ruf, Christopher S, Clara Chew, Timothy Lang, Mary G Morris, Kyle Nave, Aaron Ridley and Rajeswari Balasubramaniam (2018). 'A New Paradigm in Earth Environmental Monitoring with the CYGNSS Small Satellite Constellation'. In: *Scientific reports* 8.1, p. 8782.

Ruiz-de-Azua, Joan Adria, Anna Calveras and Adriano Camps (2020a). 'A Novel Dissemination Protocol to Deploy Opportunistic Services in Federated Satellite Systems'. In: *IEEE Access*. [Submitted].

Ruiz-de-Azua, Joan Adria, Victoria Ramirez, Hyuk Park, Anna Calveras and Adriano Camps (2020b). 'Assessment of Satellite Contacts using Predictive Algorithms for Autonomous Satellite Networks'. In: *IEEE Access*.

Ruiz-de-Azua, Joan Adria, Anna Calveras and Adriano Camps (2020c). 'Towards Federated Satellite Systems and Internet of Satellites paradigms: A Proof-of-Concept with Balloons'. In: *IEEE Access*. [Submitted].

Ruiz-de-Azua, Joan Adrià, Anna Calveras and Adriano Camps (2017). 'Internet of Satellites (IoSat): An interconnected space segment paradigm'. In: *5th Federated and Fractionated Satellite Systems Workshop Proceedings*, pp. 1–8.

Ruiz-de-Azua, Joan Adrià, Joan Francesc Munoz-Martin, Arnau Solanellas, Pol Vía, Ricard Castellà, Lara Fernández, Carlos Diez, Xavier Matabosch, Quirze Abella, Andrea Aguilella, Noemí Miguélez, Alex Cortiella, Oscar Liebana, Angel Navarro and Adriano Camps (2018a). '3Cat-4 Mission: a 1-Unit CubeSat Demonstration GNSS-R Mission'. In: *2^{nd} URSI Atlantic Radio Science Meeting*.

Ruiz-de-Azua, Joan Adrià, Adriano Camps and Anna Calveras (2018b). 'Benefits of Using Mobile Ad-Hoc Network Protocols in Federated Satellite Systems for Polar Satellite Missions'. In: *IEEE Access* 6, pp. 56356–56367. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2871516.

Ruiz-de-Azua, Joan Adrià, Anna Calveras and Adriano Camps (2018c). 'Internet of Satellites (IoSat): Analysis of Network Models and Routing Protocol Requirements'. In: *IEEE Access*, pp. 20390–20411. DOI: 10.1109/ACCESS.2018.2823983.

Ruiz-de-Azua, Joan Adrià, Carles Araguz, Anna Calveras, Eduard Alarcón and Adriano Camps (2018d). 'Towards an Integral Model-based Simulator for Autonomous Earth Observation Satellite Networks'. In: *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, pp. 7403–7406.

Ruiz-de-Azua, Joan Adrià, Joan Francesc Munoz-Martin, Lara Fernández, Marc Badia, David Llavería, Carlos Diez, Andrea Aguilella, Adrián Pérez, Oscar Milian, Marco Sobrino, Angel Navarro, Hector Lleó, Miquel Sureda, Manel Soria, Anna Calveras and Adriano Camps (2019a). '3Cat-4 Mission: A 1-Unit CubeSat for Earth Observation with a L-band Radiometer and a GNSS-Reflectometer Using Software Defined Radio'. In: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pp. 8867–8870. DOI: 10.1109/IGARSS.2019.8898317.

Ruiz-de-Azua, Joan Adrià, Victoria Ramírez, Hyuk Park, Anna Calveras and Adriano Camps (2019b). 'Predictive Algorithms to Assess Inter-Satellite Links Availability in Autonomous Satellite Networks'. In: *International Astronautical Congress (IAC) 2019*. IAF.

Ruiz-de-Azua, Joan Adrià, Lara Fernandez, Joan Francesc Muñoz, Marc Badia, Ricard Castella, Carlos Diez, Andrea Aguilella, Simone Briatore, Nicola Garzaniti, Anna Calveras, Alessandro Golkar and Adriano Camps (2019c). 'Proof-of-Concept of a Federated Satellite System Between Two 6-Unit CubeSats for Distributed Earth Observation Satellite Systems'. In: *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, pp. 8871–8874.

Ruiz-de-Azua, Joan Adrià, Joan Francesc Munoz-Martin, Lara Fernández, Marc Badia, David Llavería, Carlos Diez, Andrea Aguilella, Adrián Pérez, Oscar Milian, Marco Sobrino, Angel Navarro, Hector Lleó, Oscar Balcells, Miquel Sureda, Manel Soria, Anna Calveras and Adriano Camps (2019d). 'The Flexible Microwave Payload-1: : a combined GNSS-Reflectometer and L-band Radiometer for the 3 Cat-4 Cubesat Mission'. In: *GNSS+R Specialist Meeting on Reflectometry using GNSS and other Signals of Opportunity*. IEEE.

Ruiz-de-Azua, Joan Adrià, Lara Fernández, Marc Badia, Albert Martón, Nicola Garzaniti, Anna Calveras, Alessandro Golkar and Adriano Camps (2020d). 'Demonstration of the Federated Satellite Systems concept for Future Earth Observation Satellite Missions'. In: *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. [Accepted]. IEEE.

Russell, Kendal (2018). *Telesat Celebrates Launch of Its First LEO Satellite*. https://www.satellitetoday.com/business/2018/01/12/telesat-celebrates-launch-first-leo-satellite/. Accessed: 16th July 2020.

Safdar, Veena, Faisal Bashir, Zara Hamid, Hammad Afzal and Jae Young Pyun (2012). 'A hybrid routing protocol for wireless sensor networks with mobile sinks'. In: *7th IEEE International Symposium on Wireless and Pervasive Computing*, pp. 1–5. DOI: 10.1109/ISWPC.2012.6263665.

Samar, Prince, Marc R Pearlman and Zygmunt J Haas (2004). 'Independent zone routing: an adaptive hybrid routing framework for ad hoc wireless networks'. In: *IEEE/ACM Transactions on Networking (TON)* 12.4, pp. 595–608. DOI: 10.1109/TNET.2004.833153.

Sanchez Net, Marc, Iñigo del Portillo, Bruce Cameron and Edward F Crawley (2016). 'Architecting Information Security Services for Federated Satellite Systems'. In: *Journal of Aerospace Information Systems*, pp. 1–12. DOI: 10.2514/1.I010425.

Sayin, Alp, Mikhail Cherniakov and Michail Antoniou (2019). 'Passive radar using Starlink transmissions: A theoretical study'. In: *2019 20th International Radar Symposium (IRS)*. IEEE, pp. 1–7.

Schaub, Hanspeter (2004). 'Relative orbit geometry through classical orbit element differences'. In: *Journal of Guidance, Control, and Dynamics* 27.5, pp. 839–848.

Sector, ITU Radiocommunication (2017a). *Guidelines for evaluation of radio interface technologies for IMT-2020*. Tech. rep. ITU-R M.2412. International Telecommunication Union (ITU).

— (2017b). *Minimum requirements related to technical performance for IMT-2020 radio interface(s)*. Tech. rep. ITU-R M.2410. International Telecommunication Union (ITU).

Selding, Peter B. (2014). *WorldVu, a Satellite Startup Aiming To Provide Global Internet Connectivity, Continues To Grow Absent Clear Google Relationship*. `https://spacenews.com/41755worldvu-a-satellite-startup-aiming-to-provide-global-internet/`. [Online; accessed 17th July 2020].

Selding, Peter B. de (2015). 'Virgin Qualcomm Invest in OneWeb Satellite Internet Venture'. In: *Spacenews*.

Selva, Daniel (2012). 'Rule-based system architecting of Earth observation satellite systems'. PhD thesis. Massachusetts Institute of Technology.

Selva, Daniel, Alessandro Golkar, Olga Korobova, Ignasi Lluch, Paul Collopy and Olivier L de Weck (2017). 'Distributed Earth Satellite Systems: What Is Needed to Move Forward?' In: *Journal of Aerospace Information Systems* 14.8, pp. 412–438. DOI: `10.2514/1.I010497`.

Senouci, Mustapha Reda, Abdelhamid Mellouk, Hadj Senouci and Amar Aissani (2012). 'Performance evaluation of network lifetime spatial-temporal distribution for WSN routing protocols'. In: *Journal of Network and Computer Applications* 35.4, pp. 1317–1328. DOI: `10.1016/j.jnca.2012.01.016`.

Sensing, Semtech Corporation Wireless and Timing Products Division (2015). *AN1200.22: LoRa^{TM} Modulation Basics*. Tech. rep. Technical Report.

Shankar, Bhawani (1996). 'Mobile satellites gather speed'. In: *Microwave Journal* 39.10, pp. 138–143.

Shaw, Graeme B, David W Miller and Daniel E Hastings (1999). 'The generalized information network analysis methodology for distributed satellite systems'. PhD thesis. Citeseer.

Singh, Kuldeep and Anil Kumar Verma (2014). 'Applying OLSR routing in FANETs'. In: *2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies*. IEEE, pp. 1212–1215.

Song, Guanghua, Mengyuan Chao, Bowei Yang and Yao Zheng (2014). 'TLR: A traffic-light-based intelligent routing strategy for NGEO satellite IP networks'. In: *IEEE Transactions on Wireless Communications* 13.6, pp. 3380–3393. DOI: `10.1109/TWC.2014.041014.130040`.

Song, Pubin, Jing Wu, Hao Jiang, Jianguo Zhou, Zijing Cheng, Zhaojun Wang and Tianyi Cao (2015). 'Snapshot integration routing for high-resolution satellite sensor networks based on delay-tolerent network'. In: *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM)*, pp. 2400–2406. DOI: `10.1109/CIT.2015.7378753`.

Space, Bryce and Technology (2019). *2019 State of the Satellite Industry Report*.

Spyropoulos, Thrasyvoulos, Konstantinos Psounis and Raghavendra Cauligi (2005). 'Spray and wait: an efficient routing scheme for intermittently connected mobile networks'. In: *ACM SIGCOMM workshop on Delay-tolerant networking*, pp. 252–259.

Su, William, Johanna B. Malaer, Oshua Cho and Kevin Suh (2019). 'Using Mobility Prediction to Enhance Network Routing in LEO Crosslink Network'. In: *International Astronautical Congress (IAC) 2019*. IAF.

Sun, R, D Maessen, J Guo and E Gill (2010). 'Enabling inter-satellite communication and ranging for small satellites'. In: *9th Symposium on Small Satellites Systems and Services, Funchal, Portugal*. Vol. 31.

Sweeting, Martin (2018). 'Modern small satellites-changing the economics of space'. In: *Proceedings of the IEEE* 106.3, pp. 343–361.

Systems, Fossa (2020). *FossaSat-1*. https://fossa.systems/. [Online; accessed 12-May-2020].

Taleb, Tarik, Abbas Jamalipour, Nei Kato and Yoshiaki Nemoto (2005). 'IP traffic load distribution in NGEO broadband satellite networks'. In: *Lecture notes in computer science* 3733, p. 113.

Taleb, Tarik, Daisuke Mashimo, Abbas Jamalipour, Nei Kato and Yoshiaki Nemoto (2009). 'Explicit load balancing technique for NGEO satellite IP networks with on-board processing capabilities'. In: *IEEE/ACM Transactions on Networking* 17.1, pp. 281–293. DOI: 10.1109/TNET.2008.918084.

Tang, Zhu, Zhenqian Feng, Wei Han, Wanrong Yu, Baokang Zhao and Chunqing Wu (2015). 'Improving the snapshot routing performance through reassigning the inter-satellite links'. In: *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 97–98. DOI: 10.1109/INFCOMW.2015.7179361.

Tarongí Bauzá, José Miguel (2013). 'Radio frequency interference in microwave radiometry: statistical analysis and study of techniques for detection and mitigation'. PhD thesis.

Teles, J, MV Samii and CE Doll (1995). 'Overview of TDRSS'. In: *Advances in Space Research* 16.12, pp. 67–76. DOI: 10.1016/0273-1177(95)98783-K.

The European Space Agency (2018). *Curbing space debris in the era of mega-constellations*. https://www.esa.int/Enabling_Support/Preparing_for_the_Future/Discovery_and_Preparation/Curbing_space_debris_in_the_era_of_mega-constellations. [Online: July 17th, 2020].

Thompson, Amy (2020). 'SpaceX launches 60 Starlink satellites and lands rocket in dazzling nighttime liftoff'. In: *Space.com*. [Online: July 1st, 2020].

Thubert, Pascal (2012). *Objective Function zero (OF0) for the Routing Protocol for Low-power and lossy networks (RPL)*. RFC 6552. RFC Editor.

Tiainen, Arttu (2017). *Inter-Satellite Link Antennas: Review and The Near Future*.

Tonetti, Stefania, Stefania Cornara, G Vicario de Miguel, Stéphane Pierotti, Judith Cote, Carles Araguz, Eduard Alarcón, Adriano Camps, David Llaveria, Estefany Lancheros, Jona Adrià Ruiz-de Azua, Elisenda Bou-Balust, Pedro Rodríguez, Mateusz Sochacki, Janusz Narkiewicz, Alessandro Golkar, Ignasi Lluch i Cruz and Hripsime Matevosyan (2019a). 'Mission and system architecture for an operational network of earth observation satellite nodes'. In: *70th International Astronautical Congress (IAC)*. IAF.

Tonetti, Stefania, Stefania Cornara, G Vicario de Miguel, Stéphane Pierotti, Judith Cote, Carles Araguz, Eduard Alarcón, Adriano Camps, David Llaveria, Estefany Lancheros, Jona Adrià Ruiz-de Azua, Elisenda Bou-Balust, Pedro Rodríguez, Mateusz Sochacki and Janusz Narkiewicz (2019b). 'Mission and System Performance Analyses for an Operational Network of Earth Observation Satellite Nodes'. In: *Living Planet Symposium*. ESA.

Tonetti, Stefania, Stefania Cornara, G Vicario de Miguel, Stéphane Pierotti, Judith Cote, Carles Araguz, Eduard Alarcón, Adriano Camps, David Llaveria, Estefany Lancheros, Jona Adrià Ruiz-de Azua, Elisenda Bou-Balust, Pedro Rodríguez, Mateusz Sochacki, Janusz Narkiewicz, Alessandro Golkar, Ignasi Lluch i Cruz and Hripsime Matevosyan (2020). 'Mission and system architecture for an operational network of earth observation satellite nodes'. In: *Acta Astronautica*. DOI: 10.1016/j.actaastro.2020.06.039.

Trujillo, Alejandro E, Matthew T Moraguez, Andrew Owens, Samuel I Wald and Olivier De Weck (2017). 'Feasibility Analysis of Commercial In-Space Manufacturing Applications'. In: *AIAA SPACE and Astronautics Forum and Exposition*, p. 5360.

Tyvak Nano-Satellite Systems, Inc. (2019). *Platforms*. [Online: April 21st, 2020]. URL: https://www.tyvak.com/mp-files/tyvak_platforms.pdf.

(USGS), United States Geological Survey. *Landsat Missions - Landsat 8*. [Online: May 12th, 2020]. URL: https://www.usgs.gov/land-resources/nli/landsat/landsat-8?qt-science_support_page_related_con=0#qt-science_support_page_related_con.

Vahdat, Amin, David Becker et al. (2000). *Epidemic routing for partially connected ad hoc networks*.

Vangelista, Lorenzo, Andrea Zanella and Michele Zorzi (2015). 'Long-range IoT technologies: The dawn of LoRa™'. In: *Future access enablers of ubiquitous and intelligent infrastructures*. Springer, pp. 51–58.

Vasseur, JP, Mijeom Kim, Kris Pister, Nicolas Dejean and Dominique Barthel (2012). *Routing metrics used for path calculation in low-power and lossy networks*. RFC 6551. RFC Editor.

Vidal, Oriol, Greet Verelst, Jérôme Lacan, Eric Alberty, José Radzik and Michel Bousquet (2012). 'Next generation high throughput satellite system'. In: *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*. IEEE, pp. 1–7.

Vinton, Cerf, Scott Burleigh, Adrian Hooke, Leigh Torgerson, Robert Durst, Keith Scott, Kevin Fall and Howard Weiss (2007). *Delay-tolerant networking architecture*. RFC 4838. RFC Editor.

Voigt, Stefan, Fabio Giulio-Tonolo, Josh Lyons, Jan Kučera, Brenda Jones, Tobias Schneiderhan, Gabriel Platzeck, Kazuya Kaku, Manzul Kumar Hazarika, Lorant Czaran et al. (2016). 'Global trends in satellite-based emergency mapping'. In: *Science* 353.6296, pp. 247–252.

Walter, Felix and Marius Feldmann (2018). 'Contact Capacity Prediction in Ring Road Networks'. In: *2018 6th IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*. IEEE, pp. 77–82.

Wang, Lan and Stephan Olariu (2004). 'A two-zone hybrid routing protocol for mobile ad hoc networks'. In: *IEEE transactions on Parallel and distributed systems* 15.12, pp. 1105–1116. DOI: 10.1109/TPDS.2004.73.

Wang, Yu, Min Sheng, King-Shan Lui, Xijun Wang, Runzi Liu, Yan Zhang and Di Zhou (2015). 'Tailored Load-Aware Routing for Load Balance in Multilayered Satellite Networks'. In: *82nd IEEE Vehicular Technology Conference*, pp. 1–5. DOI: 10.1109/VTCFall.2015.7391166.

Wang, Zhenfei, Liying Zhang, Zhiyun Zheng and Junfeng Wang (2017). 'Energy balancing RPL protocol with multipath for wireless sensor networks'. In: *Peer-to-Peer Networking and Applications*, pp. 1–16. DOI: 10.1007/s12083-017-0585-1.

Weinmann, Frank and Klaus Dostert (2006). 'Verification of background noise in the short wave frequency range according to recommendation ITU-R P. 372'. In: *AEU-International Journal of Electronics and Communications* 60.3, pp. 208–216.

Werner, Markus (1997). 'A dynamic routing concept for ATM-based satellite personal communication networks'. In: *IEEE Journal on Selected Areas in Communications* 15.8, pp. 1636–1648. DOI: 10.1109/49.634801.

Werner, Markus, Cecilia Delucchi, H-J Vogel, Gerard Maral and J-J De Ridder (1997). 'ATM-based routing in LEO/MEO satellite networks with intersatellite links'. In: *IEEE Journal on Selected areas in Communications* 15.1, pp. 69–82. DOI: 10.1109/49.553679.

Wiedeman, Robert A and Andrew J Viterbi (1993). 'The Globalstar mobile satellite system for worldwide personal communications'. In: *Proceedings 3rd International Mobile Satellite Conference, IMSC'93*. Pasadena, CA, USA.

Williams, Ross et al. (1993). 'A painless guide to CRC error detection algorithms'. In: *Internet publication, August*.

Winter, Tim (2012). *RPL: IPv6 Routing Protocol for Low-power and lossy networks*. RFC 6550. RFC Editor.

Wolfe, Hanke (2004). 'KISS—Keep It Simple Stupid'. In: *Computer Fraud & Security* 2004.5, p. 11.

Wong, Yen, George Bussey, Jacob Burke and Serhat Altunc (2017). *SmallSat Constellation Intersatellite Link System Simulator*. Tech. rep.

Wood, Lloyd, Wesley M Eddy, Will Ivancic, Jim McKim and Chris Jackson (2007). 'Saratoga: a Delay-Tolerant Networking convergence layer with efficient link utilization'. In: *2007 International Workshop on Satellite and Space Communications*. IEEE, pp. 168–172.

Wu, Zhaofeng, Guyu Hu, Fenglin Jin, Yu Song, Yinjin Fu and Guiqiang Ni (2017). 'A novel routing design in the IP-based GEO/LEO hybrid satellite networks'. In: *International Journal of Satellite Communications and Networking* 35.3, pp. 179–199. DOI: 10.1002/sat.1174.

Yazıcı, Volkan, Ulas C Kozat and M Oguz Sunay (2014). 'A new control plane for 5G network architecture with a case study on unified handoff, mobility, and routing management'. In: *IEEE Communications Magazine* 52.11, pp. 76–85.

Yi, Xianqing, Zhili Sun, Fang Yao and Ye Miao (2013). 'Satellite constellation of MEO and IGSO network routing with dynamic grouping'. In: *International Journal of Satellite Communications and Networking* 31.6, pp. 277–302. DOI: 10.1002/sat.1032.

Younis, Ossama and Sonia Fahmy (2004). 'HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks'. In: *IEEE Transactions on mobile computing* 3.4, pp. 366–379.

Yu, Yan, Ramesh Govindan and Deborah Estrin (2001). *Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks*. Tech. rep. UCLA Computer Science Department.

List of abbreviations

AI	Artificial Intelligence.
ALBR	Agent-based Load Balancing Routing.
AMSR	Advanced Microwave Scanning Radiometer.
AODV	Ad-hoc On-demand Distance Vector.
AOMDV	Ad-hoc On-demand Multi-path Distance Vector.
ARM	Advanced RISC Machine.
ARP	Address Resolution Protocol.
ARQ	Automatic Repeat-Request.
ATM	Asynchronous Transfer Mode.
BATMAN	Better Approach To Mobile Ad-hoc Networking.
BER	Bit Error Rate.
BMX6	BatMan eXperimental version 6.
CBR	Constant Bit Rate.
CCSDS	Consultative Committee for Space Data Systems.
CGR	Contact Graph Routing.
CLMS	Copernicus Land Monitoring Service.
CNES	French National Centre for Space Studies.
COCOM	Coordinating Committee for Multilateral Export Controls.
COTS	Commercial-Off-The-Shelf.
CPCE	Contact Plan Computation Element.
CRC	Cyclic Redundancy Check.
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance.
CYGNSS	Cyclone Global Navigation Satellite System.
DAG	Directed Acyclic Graphs.
DLAR	Distributed Load-Aware Routing.
DRA	Datagram Routing Algorithm.
DSDV	Destination-Sequenced Distance-Vector.
DSS	Distributed Satellite System.
DTLSR	Delay Tolerant Link State Routing.
DT-DVTR	Discrete-Time Dynamic Virtual Topology Routing.
EAEpidemic	Energy Aware Epidemic.
ECEF	Earth-Centered, Earth-Fixed.
ECI	Earth-Centered Inertial.
EDRS	European Data Relay System.
EHEED	Extended Hybrid Energy Efficient Distributed Clustering.
EIRP	Equivalent Isotropical Radiated Power.
ELB	Explicit Load Balancing.
EO	Earth Observation.

ESA	European Space Agency.
ETX	Expected Transmission Count.
EU	European Union.
FEC	Forward Error Correction.
FeDeCoP	Federation Deployment Control Protocol.
FIFO	first in, first out.
FMPL-2	Flexible Microwave Payload - version 2.
FoM	Figure of Merit.
FreeRTOS	Free Real-Time Operating System.
FSPL	Free Space Path Loss.
FSR	Fish-eye State Routing.
FSS	Federated Satellite System.
FSSExp	Federated Satellite Systems Experiment.
GAF	Geographical Adaptive Fidelity.
GEAR	Geographic and Energy Aware Routing.
GMAT	General Mission Analysis Tool.
CMEMS	Copernicus Marine Environment Monitoring Service.
GMSK	Gaussian Minimum Shift Keying.
GNSS	Global Navigation Satellite Systems.
GNSS-R	GNSS Reflectrometry.
GPIO	General-Purpose Input/Output.
GPS	Global Position System.
GRI	Global Routing Information.
GUI	Graphical User Interface.
HEED	Hybrid Energy Efficient Distributed Clustering.
HC	Hop-Count.
HCAR	Hop-Constrained Adaptive Routing.
HL	Hop Limit.
HQRP	Hierarchical QoS Routing Protocol.
IAC	International Astronautical Congress.
IAF	International Astronautical Federation.
IETF	Internet Engineering Task Force.
IOL	Intra-Orbital Links.
ION	Interplanetary Overlay Network.
IoS	Internet of Space.
IoSat	Internet of Satellites.
IoT	Internet of Things.
IPv6	Internet Protocol version 6.
IRPL	Improved RPL.
IRQ	Inter-Quantile Range.
ISL	Inter-Satellite Link.
ISN	Inter-Satellite Network.
ISS	International Space Station.
IZRP	Independent Zone Routing Protocol.
I2C	Inter-Integrated Circuit.
I-VT	Instantaneous Virtual Topology.
JTAG	Join Test Action Group.
KISS	Keep It Simple Stupid.
KP-RPL	Kalman Positioning RPL.
LAOR	Location-Assisted On-demand Routing.
LCI	Life Cycle Index.

LEACH	Low-Energy Adaptive Clustering Hierarchy.
LEO	Low Earth Orbit.
LET	Link Expiration Time.
LNA	Low Noise Amplifier.
LoRa	Long Range.
LPLR	Local Packet Loss Ratio.
LPWAN	Low-Power Wide-Area Networks.
LRI	Local Routing Information.
MANET	Mobile Ad-hoc Networks.
MEO	Medium Earth Orbit.
MLSN	Multi-Layered Satellite Networks.
MLSR	Multi-Layered Satellite Routing.
MOVE	Motion Vector.
MPR	Multipoint Relay.
MRHOF	Minimum Rank with Hysteresis Objective Function.
MWR	Microwave Radiometer.
MWRI	Microwave Radiation Imager.
NASA	National Aeronautics and Space Administration.
NB-IoT	Narrow-Band Internet of Things.
NS-3	Network Simulator version 3.
OBC	On-Board Computer.
OCGR	Opportunistic Contact Graph Ruting.
OF	Objective Function.
OLSR	Optimized Link State Routing.
ONION	Operational Network of Individual Observation Nodes.
OSADP	Opportunistic Service Availability Dissemination Protocol.
OSCAR	Observing Systems Capability Analysis and Review.
O-ISL	Optical ISL.
PA	Power Amplifier.
PAR	Priority-based Adaptive Routing.
PDIRR	Packet Delivery In Route Ratio.
PDR	Packet Delivery Ratio.
PLSR	Predictable Link-State Routing.
PM	Power Meter.
PROPHET	Probabilistic Routing Protocol using Historic of Encounters and Transitivity.
QU-RPL	Queue Utilization based RPL.
RAPID	Resource Allocation Protocol for Intentional DTN.
RCM	Routing in Cyclic Mobility.
RET	Route Expiration Time.
RF-ISL	Radio Frequency Inter-Satellite Link.
RPL	Routing Protocol for Low Power and Lossy Networks.
RS	Reed-Solomon.
RSSI	Receive Signal Strength Indicator.
RTO	Recovery Time Objective.
RTT	Round Trip Time.
SABR	Schedule-Aware Bundle Routing.
SAR	Synthetic Aperture Radar.
SAW	Surface Acoustic Wave.
SD	Secure Digital.
SDN	Software Defined Networks.
SDR	Software Defined Radio.

SIR	Snapshot Integration Routing.
SIRAL	SAR Interferometer Altimeter.
SLT	Service Lifetime.
SNR	Signal-to-noise Ratio.
SOCIS	Summer of Code in Space.
SSO	Sun-Synchronous Orbit.
ST	Service Table.
STK	Satellite Tool Kit.
TC	Topology Control.
TCP	Transport Control Protocol.
TDRSS	Tracking Data Relay Satellite System.
TIR	Thermal Infrared.
TLAR	Tailored Load-Aware Routing.
TLR	Traffic-Light-based Routing.
TTC	Telemetry, Tracking and Control.
TZRP	Two-Zone Routing Protocol.
UART	Universal Asynchronous Received-Transmitter.
UDP	User Datagram Protocol.
UHF	Ultra High Frequency.
UML	Unified Modeling Language.
UPC	Universitat Politècnica de Catalunya (Technical Univ. of Catalonia)
VHF	Very High Frequency.
VNIR	Visible and Near-Infrared.
WGS	World Geodetic System.
WMO	World Meteorological Organisation.
WSN	Wireless Sensor Network.
ZRP	Zone Routing Protocol.
3GPP	3rd Generation Partnership Project.