Self-Organising LEO Small Satellite Constellation for 5G MTC and IoT Applications

Le Tian*, Nicolas Huot†, Olivier Chef‡, Jeroen Famaey§

*Information Engineering University, Zhengzhou, China
†Space Applications Services, Belgium
‡Celestia Antwerp, Belgium
§IDLab, University of Antwerp – imec, Belgium

***Abstract*—Small satellite constellations are an attractive alternative to provide ubiquitous connectivity to Internet of Things (IoT) and 5G machine-type communication (MTC) applications in areas where terrestrial network coverage is limited. In order to provide global coverage using small satellite networks flying in low earth orbit (LEO), a large constellation of at least hundreds of satellites is needed. As LEO satellites can only connect directly to the ground earth station (GES) sporadically, they need to forward data over inter-satellite links (ISLs) to reach the GES. In this paper, we propose a novel self-organising architectural design for small satellite constellations with ISLs, supporting automated commissioning and control of the satellite network. We specifically present an algorithm for autonomous ISL establishment and channel selection,that avoids interference among ISLs of the same satellite.**

***Index Terms*—Internet of Things, machine type communication, small satellite constellation, cubesats, self-organising networks**

I. Introduction

Small satellite constellations flying in low earth orbit (LEO) are envisioned as an attractive solution to provide ubiquitous connectivity to a wide variety of Internet of Things (IoT) and 5G machine-type communication (MTC) applications in areas where terrestrial network coverage is limited [1], [2]. LEO constellations, deployed between 500 and 2000 km above the Earth's surface, are especially attractive due to their low propagation delay and loss compared to the traditional geostationary earth orbit (GEO) satellites. However, in order to provide global coverage at LEO altitudes, a large constellation of hundreds of satellites is needed. Moreover, in order to support latency-sensitive applications, permanent connectivity between a ground earth station (GES) and all satellites in the constellation is required. As LEO satellites can only sporadically connect directly to a ground station (GS), inter-satellite communication links (ISLs) are needed to ensure such permanent connectivity. The advantage of using small satellites, is their relatively low production and launch cost. The lower cost to produce and launch small satellites, finally makes it feasible to create and deploy large constellations, consisting of tens or even hundreds of satellites, ensuring global coverage.

In such large constellations, manual commissioning and configuration of satellites is an arduous, error-prone and time consuming task. This problem is further amplified if the constellation changes due to failures, re-positioning of satellites, or the launch of new satellites. As such, there is a need for self-organising solutions that allow small satellite constellations to automatically organise and maintain themselves in a distributed manner. This includes the need for automated discovery and establishment of ISL and GES connections, as well as reliable end-to-end routing.

In this paper, we propose an architecture and protocol stack to design a self-organising small satellite network that is able to provide ubiquitous connectivity to terrestrial low-power IoT or MTC devices. Specifically, we present a fully distributed algorithm for fast autonomous ISL establishment and channel selection, that avoids interference among different ISL links. The designed protocols and algorithms were implemented in the event-based ns-3 network simulator. Finally, performance of the design is evaluated in ns-3, based on the characteristics of an ISL transceiver prototype.

The remainder of this paper is structured as follows. Section II gives a brief overview of related research on small satellite constellations. Section III introduces the proposed satellite design and network architecture. Subsequently, Section IV provides more details on the ISL establishment procedure. The evaluation results are presented in Section V and Section VI concludes the paper.

II. Related work

In this section, we give an overview of related research on the topic of small satellite constellations for IoT and 5G connectivity. Soret et al. [1] studied the applicability of LEO small satellite constellations to the 3 types of 5G use cases: enhanced mobile broadband (eMBB), ultra-reliable low-latency communications (URLLC) and MTC. They introduce the general satellite constellation architecture, provide a list of advantages of using satellites compared to terrestrial networks, and briefly survey the available technologies to implement the different required functionalities (e.g., connectivity, processing, networking). Both De Sanctis et al. [3] and Qu et al. [2] performed a similar study, but focused specifically on using LEO small satellite constellations for IoT connectivity. Among other things, they discuss constellation design and architecture, interference mitigation, network architecture, routing, and high-layer protocols. They conclude that LEO satellite constellations are a powerful supplement to terrestrial systems to provide ubiquitous IoT connectivity.

While the above works look more at the overall architecture, a lot of researchers have focused specifically on the problem of inter-satellite communications. Qi et al. [4] surveyed and compared various routing algorithms for LEO satellite constellations, focusing on routing based on virtual nodes and topologies. They classify them based on their single- or multi-path capabilities and offline or online topology creation. However, all the surveyed algorithms use static routing metrics and do not take into account link congestion dynamics. On-demand routing approaches, such as multi-service on-demand routing (MOR) [5], are able to overcome this issue. More recently, Radhakrishnan et al. [6] thoroughly surveyed existing literature on small satellite communications, from the physical up to the network layer.

The work presented in this paper is complementary to the approaches highlighted above. We propose a small satellite design based on an in-house developed prototype of an ISL transceiver, and consider its specific constraints and restrictions in building a distributed link establishment procedure and protocol stack for it. Our design is compatible with existing solutions for routing, and can thus be combined with them.

III. Satellite network architecture

In this section, we provide an overview of the satellite network architecture, first focusing on the overall constellation aspects and then zooming in on the design of a single satellite.

A. Satellite constellation

The envisioned small satellite constellation consists of $N \times M$ satellites, deployed in N orbital planes, each consisting of M satellites [4]. Each satellite maintains several different communication links. First, it is assumed that each satellite maintains an ISL to each of its two neighbours in the same orbital plane, referred to *intra-plane ISLs*. This assumes that each orbital plane contains enough satellites to maintain continuous line-of-sight among intra-plane neighbours (e.g., at an altitude of 600 km, at least 9 satellites are needed on each plane). Second, each satellite can sporadically connect to the nearest satellite in the two neighbouring orbital planes, referred to as *inter-plane ISLs*. Third, as satellites pass over a GES, they can temporarily establish a GES-LEO connection, allowing them to communicate with the ground (e.g., to transmit data received from IoT devices, or captured using on-board sensors, towards the application server). Finally, IoT devices may connect (either directly or via a gateway) to the satellites to transmit sensor readings or other data. The different types of connections and components involved in the envisioned architecture are summarized in Figure 1.

In the remainder of this paper, we focus on the commissioning and management of a single-plane constellation (i.e., assuming only intra-plane ISLs and the LEO-GES link) for clarity. The proposed solutions, however, can be trivially extended to support inter-plane ISLs as well.

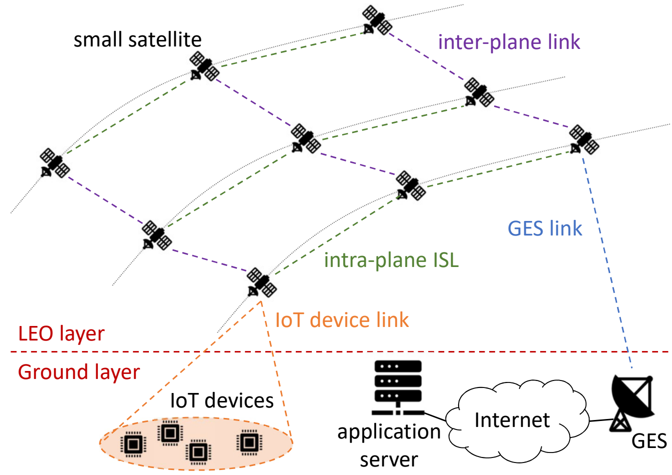

Fig. 1. Components involved in the satellite constellation architecture envisioned in this paper

B. Small satellite design

Figure 2 shows a schematic representation of the communication capabilities of a single satellite S_n, assuming only intra-plane ISLs. If considering also inter-plane ISLs, two additional ISL communication interfaces should be present. Each satellite is equipped with two ISL communication interfaces and one ground link interface. The ISL interfaces are used to communicate with the previous and next satellite in the orbital plane respectively, while the ground link interface is used to communicate with the GES and other devices on the ground (e.g., IoT gateways or sensors). The ground link interface can rely on off-the-shelf hardware, such as the Syrlinks EWC27 series[1], which comes in a form factor suitable for small satellites.

Each ISL interface is equipped with a custom-built K_a band radio.The transceiver uses a continuous modulation for ISL links. As such, the demodulator requires 80 ms to successfully synchronize with the carrier signal when establishing the link or changing to a different channel. Two unidirectional point-to-point links are set up between the satellite and each of its two orbital plane neighbours, one for transmission (TX) and one for reception (RX). This means the satellite has full-duplex ISL capabilities. As these point-to-point transmissions are directional, potential interference from other sources is negligible. Nevertheless, a satellite should not use the same frequency channel for transmitting and receiving at the same time. This would lead to receiver chain saturation on the modem, due to the high TX power used as well as the high sensitivity of the ISL receiver. To avoid this problem, we propose a self-organising ISL establishment and channel selection algorithm, described in Section IV.

The proposed protocol stack is shown in Figure 3. At the physical layer (PHY), the Digital Video Broadcasting Satellite Second Generation X (DVB-S2X) [7] standard is used. Due to

[1]https://www.syrlinks.com/en/space/nano-satellite/x-band-transmitter-ewc27

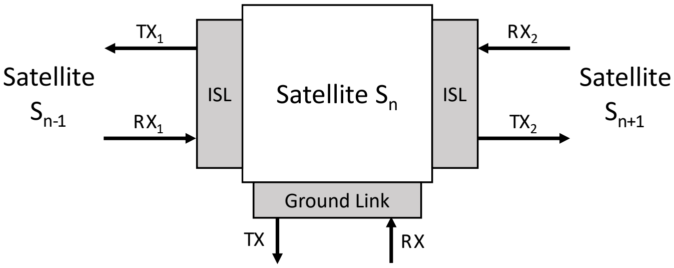

Fig. 2. Schematic representation of the wireless interfaces present on each satellite

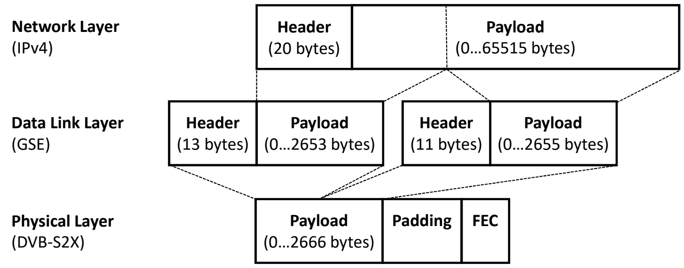

Fig. 3. Protocol stack used on the small satellites

design restrictions in the ISL radio (i.e., the use of continuous modulation), the PHY frame length is fixed to 2666 bytes, thus adding optional padding if the payload size is smaller. Moreover, a forward error correction (FEC) code is appended to each frame to minimize the frame error rate. As shown in Figure 3, multiple smaller data link layer packets can be aggregated into a single PHY frame, in order to maximize channel capacity utilisation. As the PHY frame size is fixed, there is no need for a synchronization header. As such, each PHY-layer frame is sent immediately after the previous one. If no data is available in the buffer, an empty (padding only) frame is sent instead.

At the data link layer, the DVB Generic Stream Encapsulation (GSE) protocol standard is used [8]. It supports packet fragmentation and error detection using a 32-bit cyclic redundancy check (CRC). To avoid packet fragmentation at the PHY, a maximum GSE packet size equal to the PHY frame size of 2666 bytes is enforced.

At the network layer, the standard Internet protocol version 4 (IPv4) is used. This enables interfacing with a variety of existing transport layer protocols, such as the User Datagram Protocol (UDP) or Transport Control Protocol (TCP). Moreover, it allows packets to or from the satellite constellation to be easily routed across the Internet, potentially using Network Address Translation (NAT) at the GES. As IPv4 packets can have a payload up to 65515 bytes, the GSE fragmentation feature is used to ensure that the maximum PHY payload size is not exceeded.

IV. ISL ESTABLISHMENT

The goal of ISL establishment protocol is to select a suitable channel for each ISL TX and RX interface. As stated above, a single satellite cannot use the same channel for TX and RX interfaces, as it can cause receiver chain saturation and self-interference. The presented approach works for satellites with any number of ISLs. As such, it can be easily extended to support inter-plane ISL links as well. From the well-known four-color map theorem, we can derive that at most 4 channels are required to ensure a solution exists for any arbitrary satellite constellation topology. Therefore, we assume 4 K_a band channels are available for use. For the intra-plane ISL case (cf., Figure 2), we need to ensure that $C_{TX1} \neq C_{RX1}$, $C_{TX2} \neq C_{RX1}$, $C_{TX1} \neq C_{RX2}$, and $C_{TX2} \neq C_{RX2}$. Here, $C_A \in \{1, 2, 3, 4\}$ represents the channel used for ISL transmitter or receiver $A \in \{TX_1, TX_2, RX_1, RX_2\}$.

As satellites can use a Global Navigation Satellite System (GNSS) to determine their location, they initiate the channel selection procedure as soon as they take up the right position in their assigned orbital plane. When the connection is lost (e.g., due to a failure in a neighbouring satellite), they restart the channel selection process in order to re-establish the ISL. Such ISL connection loss can be detected when a pre-configured number of subsequent acknowledgments are not received.

The proposed ISL establishment protocol uses three message types to exchange information between neighbouring satellites. The *ChannelRequest* message contains the channel that is used when sending the message (i.e., 0, 1, 2 or 3), a unique ID of the TX interface over which the message was sent (e.g., its pre-configured IP address), and the list of channels currently in use by the satellite's other connected interfaces. The *ChannelResponse* message contains the same channel number as the *ChannelRequest* to which it is a response, a unique ID of the RX interface on which the corresponding request was received, and a unique ID of the TX interface that sent the request. Finally, the *ChannelAcknowledgment* contains the same information as the response, and serves as a final verification in the three way handshake.

The channel selection algorithm itself follows a different procedure for the TX and RX interfaces of the ISL. They are described separately in the remainder of this section.

A. TX channel selection

The channel selection process is performed independently for each ISL TX interface on the satellite, and starts with a random back-off. The *BackOff* timer is initialized to a random value in the interval $[CC_{\min}, CC_{\max}]$, which is configurable. A back-off is used to avoid all interfaces trying to initialize at exactly the same time, which can lead to deadlock situations. Figure 4 shows the details of the process.

Once the back-off timer expires, the interface is put into **connected** state and it selects a legal channel C_{TX} to use. A legal TX channel is any channel not currently in use by any of the satellite's RX interfaces (either in **listening** or **connected** mode), as the satellite cannot use a channel for TX that it is already using for RX. At a specific interval, defined by the configurable *PacketIntervalCh* parameter, a ChannelRequest message is sent on the TX interface using channel C_{TX}. This

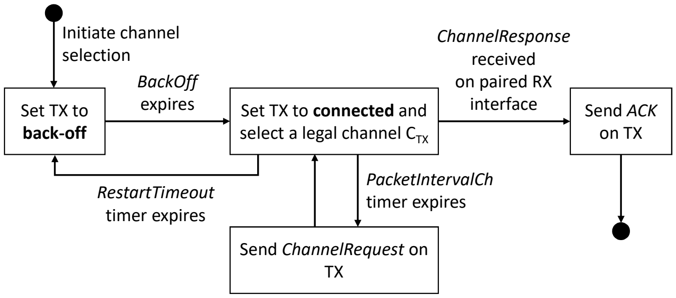

Fig. 4. Flow chart depicting the channel selection process for TX interfaces

ChannelRequest message contains the channel that all other currently **connected** interfaces of the satellite are using.

If the request message is received successfully on the corresponding RX interface of the neighbouring satellite, eventually a corresponding ChannelResponse will be received on the paired RX interface, which is being established using the RX channel selection procedure explained below. In the satellite schematic shown in Figure 2, the paired interface of TX_1 is RX_1 and of TX_2 is RX_2. When the ChannelResponse message is received by the paired RX interface, the paired RX interface is also successfully connected and a ChannelAcknowledgment message is sent to acknowledge reception of the response. If no response is received by the paired RX interface before the configurable *RestartTimeout* expires, then the TX interface returns to the back-off state and starts a new random *BackOff* timer. The *RestartTimeout* is needed, as in rare cases the distributed channel selection process may result in a deadlock, where no legal configuration exists. The timeout thus ensures that in such a case the process restarts and will eventually converge.

B. RX channel selection

The channel selection process for RX interfaces is shown in Figure 5. Every RX interface that is not connected will always be in the **listening** state. In this state, the satellite cycles through the available channels (i.e., any channel not currently used by a **connected** TX interface of the satellite). The channel is changed at a configurable interval *CycleIntervalCh*. While listening, the RX interface may receive either a ChannelRequest or ChannelResponse message. A response is received if the satellite had previously initiated channel selection on the paired TX interface. A request may be received if the satellite has not yet successfully established a channel on the paired TX interface. If a ChannelResponse is received, the RX state is changed to **connected** and the current channel is set as the RX interface's selected channel. As explained above, the paired TX interface will also send a ChannelAcknowlegment.

If, instead, the RX interface received a ChannelRequest message, it will also set the RX state to **connected** on the current listening channel. Immediately, the paired TX interface will also be set to **connected** and a legal channel C_{TX} is selected for it. A legal channel is any channel not being used by any of its connected RX interfaces. Moreover, any

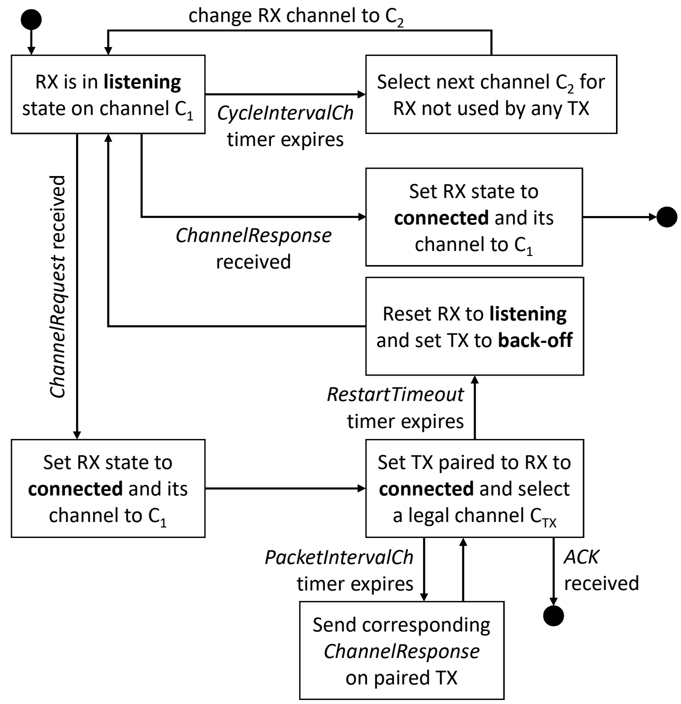

Fig. 5. Flow chart depicting the channel selection process for RX interfaces

channel used by a connected TX interface of the neighbour to which this TX interface links is also considered illegal (to avoid a deadlock). This information is obtained from the ChannelRequest message. At a specific interval, defined by the configurable *PacketIntervalCh* timer, a ChannelResponse (corresponding to the received ChannelRequest) is sent on the paired TX interface. If an ChannelAcknowledgment message is received on the paired RX interface, this pair is considered fully connected. If instead the configurable *RestartTimeout* expires, the RX interface's state is reset to **listening**, while the TX interface's state is reset to **back-off**. As before, this is to avoid deadlocks.

Note that there is a small chance the ChannelAcknowledgment is not received due to frame loss. In that case, the transmitting satellite will detect this problem once it starts transmitting data and it does not receive data acknowledgments. In that case, it will restart the TX channel selection process. As the frame loss probability is extremely low (i.e., $\sim 10^{-6}$ in practice), the chances of this occurring more than once in a row are negligible.

The influence on convergence speed of the channel selection process of the $CC_{\min}$, $CC_{\max}$, *PacketIntervalCh*, *CycleIntervalCh*, and *RestartTimeout* parameters is evaluated in Section V.

V. Results and Evaluation

This section presents and discusses simulation results obtained using the ns-3 network simulator. The aforementioned protocols and algorithms were implemented in ns-3, using the *point-to-point-channel* and *point-to-point-net-device* as a basis for the implementation of the ISLs and satellites. A satellite

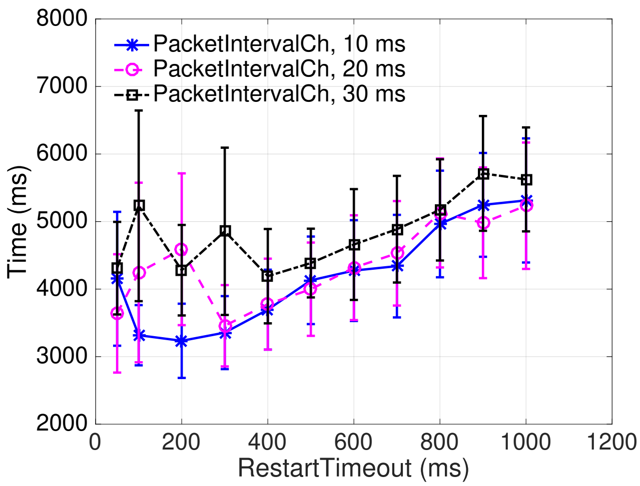

Fig. 6. Channel selection convergence time as a function of the *RestartTimeout* parameter and for different values of the *PacketIntervalCh* parameter

mobility model and satellite-to-GES connectivity model were developed based on the ns-2 satellite models.

We consider a single-plane satellite constellation consisting of 10 satellites, and a single GES located at Svalbard, Norway. Each satellite is equipped with 2 prototype K_a band transceivers to enable the ISLs, and a Syrlinks EWC27 for the ground link (supporting 100 Mbps). The satellites fly at 600 km altitude, resulting in an ISL length of 4750 km and a propagation delay of 14.4 ms. Lab experiments with the ISL transceiver showed that at this distance it can achieve 1.3 Mbps at a frame error rate of 10^{-6}. Its channel acquisition time is 80 ms, whenever it switches to a different channel.

Figure 6 shows the convergence time of the channel selection algorithm described in Section IV, as a function of the *RestartTimeout* and *PacketIntervalCh* parameters. The network is considered converged if a channel has been correctly negotiated for each ISL TX and RX interface of each satellite. The *CycleIntervalCh* parameter is always configured as 2 times the *PacketIntervalCh* parameter, to ensure that at least 1 request or response packet can arrive to the satellite before it switches to another RX channel. Moreover, the minimum and maximum back-off timers $CC_{\min}$ and $CC_{\max}$ are set to 0 and 1 second respectively. Each experiment was repeated 10 times, and the error bars represent the standard deviation of the mean across these 10 experiments.

The *RestartTimeout* determines how long the satellite will try to agree on a channel on each link before giving up, resetting to the starting state, backing off and starting again (to overcome deadlocks). As shown in Figure 6, if the timeout is too short, satellites may restart the channel selection process even if no deadlock occurred, resulting in a higher overall convergence time. If the restart timeout is too long, satellites may wait unnecessarily long after a deadlock has occurred, also leading to higher convergence times. As such, it is necessary to find the optimal point. A very short restart timeout (i.e., 100 ms or less) can indeed lead to increased convergence time. Restart timeouts above 300 ms also lead to significantly increasing convergence delay, as expected. The graph also shows that a lower *PacketIntervalCh* generally leads to better results, as the channel request and response packets are received faster and more frequently. From this graph, we can derive that *PacketIntervalCh* equal to 10 ms and RestartTimeout equal to 200 ms performs best. This results in a network-wide convergence time of 3.2 seconds. However, even suboptimal configurations lead to an average convergence time below 6 seconds, which is still within acceptable limits.

VI. Conclusion

In this paper we presented an architectural design, protocol stack and algorithms for a self-organising LEO small satellite constellation to provide ubiquitous connectivity to IoT and 5G MTC devices. Simulation results show that our proposed distributed ISL establishment algorithm is able to converge to a valid channel for each interface in as little as 3 seconds when optimally configured. In future work, we plan to evaluate our proposed solution in large-scale multi-plane LEO satellite constellations.

Acknowledgment

The research presented in this paper resulted from the Small Platform Inter-satellite Data Exchange Routes (SPIDER) project, which received funding from the European Space Agency (ESA).

References

[1] B. Soret, I. Leyva-Mayorga, M. Röper, D. Wübben, B. Matthiesen, A. Dekorsy, and P. Popovski, "LEO small-satellite constellations for 5G and beyond-5G communications," *arXiv:1912.08110 [cs.NI]*, 2019.

[2] Z. Qu, G. Zhang, H. Cao, and J. Xie, "LEO satellite constellation for internet of things," *IEEE Access*, vol. 5, pp. 18 391–18 401, 2017.

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

[4] X. Qi, J. Ma, D. Wu, L. Liu, and S. Hu, "A survey of routing techniques for satellite networks," *Journal of Communications and Information Networks*, vol. 1, pp. 66–88, 2016.

[5] S. Karapantazis, E. Papapetrou, and F. N. Pavlidou, "Multiservice on-demand routing in LEO satellite networks," *IEEE Transactions on Wireless Communications*, vol. 8, no. 1, pp. 107–112, 2009.

[6] R. Radhakrishnan, W. W. Edmonson, F. Afghah, R. M. Rodriguez-Osorio, F. Pinto, and S. C. Burleigh, "Survey of inter-satellite communication for small satellite systems: Physical layer to network layer view," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2442–2473, 2016.

[7] ETSI EN 302 307-2, "Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications (DVB-S2X)," 2014.

[8] ETSI TS 102 606, "Generic stream encapsulation (GSE) protocol," 2007.