On the Role of Delay Tolerant Networks and Contact Graph Routing in Direct-to-Satellite IoT

Sebastián I Montoya, Diego Maldonado, Juan A Fraire, Sandra Céspedes

On the Role of Delay Tolerant Networks and Contact Graph Routing in Direct-to-Satellite IoT

Sebastián I. Montoya*, Diego Maldonado†, Juan A. Fraire†‡, Sandra Céspedes§
*Pontificia Universidad Católica de Chile, Santiago de Chile, Chile
†Inria, INSA Lyon, CITI, UR3720, 69621 Villeurbanne, France
‡CONICET - Universidad Nacional de Córdoba, Córdoba, Argentina
§Department of Computer Science & Software Eng., Concordia University, Montreal, Canada.

***Abstract*—This paper explores the integration of Delay-Tolerant Networking (DTN) and Contact Graph Routing (CGR) within Direct-to-Satellite Internet of Things (DtS-IoT) networks, utilizing the FLoRaSat discrete-event simulator based on Omnet++. By incorporating a DTN model and the CGR algorithm, the study evaluates the efficacy of these technologies in optimizing data routing and handling across emerging Low-Earth Orbit (LEO) satellite networks. The research delves into various satellite fleet configurations, including Star and Delta constellations, across different numbers of orbital planes and with the integration of opportunistic Inter-Satellite Links (ISLs). Results using the FLoRaSat simulator demonstrate that the DTN store-carry-and-forward approach over ISLs, enhanced by CGR, significantly reduces end-to-end delivery delays. Specifically, the implementation achieves an average end-to-end delivery delay as low as 10 minutes in 4-plane Star constellations with 24 satellites and immediate forwarding in 8-plane Delta constellations of equivalent size, underscoring the potential of DTN and CGR to improve the efficiency and reliability of emerging DtS-IoT.**

***Index Terms*—Delay-Tolerant Networking, Contact Graph Routing, Direct-to-Satellite Internet of Things**

I. Introduction

The Internet of Things (IoT) field has experienced a remarkable expansion, moving beyond Earth's boundaries and into outer space [1]. In addition to IoT access networks on Earth, there is a growing trend in making satellite-based IoT more accessible. This paradigm enables global connectivity and facilitates data transmission in remote and challenging geographies by bypassing traditional ground-based networks [2].

Current research and commercial efforts are exploring the use of space infrastructure in Low-Earth Orbit (LEO) to offer Internet-based services such as broadband access and IoT connectivity [3]. Traditional indirect connections to the satellite (i.e., from end devices to a ground station and from the ground station to the satellite) are now mixed with direct-to-satellite (DtS) connectivity. In the DtS-IoT architecture, end devices connect directly to the satellite using technologies like 5G-enabled satellite access or Low-Power Wide-Area solutions [4], including NB-IoT [5] and LoRa/LR-FHSS [6]. Moreover, bidirectional connections between satellites aid in overcoming the time-limited visibility faced by LEO satellites. Using inter-satellite links (ISL), recently validated in nano-satellite missions such as GOMX-4 [7], data can be relayed to and from the IoT end devices via other satellites, even those in different orbits.

In addition to its expansion on Earth, the IoT will play a crucial role in gathering data in outer space through near-Earth [8] and deep-space missions [9]. This includes collecting telemetry from sensors on spacecraft, scientific data from space research such as exoplanet detection, and measurements obtained by sensors deployed on Mars to gather humidity levels. Therefore, a mesh of interconnected space-based nodes capable of transporting packets to their destinations may also benefit the interconnection of deep-space and near-Earth DtS-IoT deployments with other nodes placed on Earth, the Moon, and other space assets.

When challenged by delays or disruptions, connectivity in space relies on delay- and disruption-tolerant (DTN) architectures rather than the traditional TCP/IP architecture. A mesh network of adaptable DTN space-based nodes can effectively route data while accommodating temporary disruptions caused by long distances and orbital dynamics experienced by spacecraft operating near and far from Earth. Schedule-Aware Bundle Routing (SABR), standardized by the Consultative Committee for Space Data Systems (CCSDS), is the reference routing strategy for DTN in space networks. In our simulation, we use Contact Graph Routing (CGR), which is the current reference implementation of SABR. CGR aims to address the issues of prolonged disconnections, fluctuating communication distances, and signal propagation delays encountered in space environments. It incorporates methods to handle routing within delayed and disrupted space networks by considering upcoming connectivity through a contact plan [10].

This paper investigates the applicability and benefits of DTN within DtS-IoT networks, specifically through deploying the CGR [10], [11] algorithm to route data via inter-satellite links. Firstly, we delve into the theoretical underpinnings of DtS-IoT and CGR, elucidating their functionalities, benefits, and challenges. Secondly, we venture into a practical exploration, implementing and evaluating a DTN model integrated with CGR within the FLoRaSat framework, an advanced simulation environment for evaluating DtS-IoT protocols and networks [12]. We evaluate the robustness of CGR in handling high-latency and intermittent communication links characteristic of DtS-IoT over a series of LEO constellation topologies. Results enhance the understanding of data flow optimization in ISL-based DtS-IoT systems and lay the groundwork for optimizing data handling strategies in future deployments.

The remainder of this document is organized as follows. Section II provides the background on DtS-IoT, DTN, and related toolings. Section III depicts the DTN and CGR integration into the FLoRaSat simulation model. Section IV discusses the results obtained for different Walker constellations. Finally, Section V concludes the paper.

II. Background

A. Direct-to-Satellite IoT

Direct-to-Satellite IoT (DtS-IoT) is an emerging paradigm that enables IoT devices to communicate directly with satellites, enhancing global connectivity, especially in remote, underserved, or geographically challenging regions where terrestrial networks are untenable or nonexistent [8]. This paradigm's primary benefit is the global extension of IoT applications such as cross-border logistics, maritime, agriculture, etc. It facilitates direct data transmission from sensors and devices to satellites, circumventing the need for conventional ground-based networks. As satellite constellations expand, DtS-IoT's prevalence is set to increase significantly [3].

1) Space-Terrestrial Integration via LPWANs: Proprietary DtS-IoT solutions have existed for many years, such as Iridium, Orbcomm, Globalstar, and Kinéis. However, recent advancements aim at integrating terrestrial Low-Power Wide-Area (LPWA) solutions [4] such as NarrowBand IoT (NB-IoT) [5] and Long Range/Frequency Hopping Spread Spectrum (LoRa/LR-FHSS) [6] into space-based operations [2], [1]. These technologies offer long-range communication capabilities with reduced device energy consumption, making them attractive candidates for future satellite IoT deployments. The integration of LPWAN technologies into DtS-IoT represents a pivotal shift. It harnesses the economies of scale from the established terrestrial market to reduce costs disruptively within the space sector. Consequently, the price of deploying an LPWAN device remains consistent, regardless of its application in terrestrial or space networks, eliminating financial barriers traditionally associated with space ventures. This paradigm shift has sparked a surge in innovation, with pioneering startups like Lacuna Space [13] and Sateliot [14] leading the charge, demonstrating the viability of LPWAN in space through successful in-orbit operations.

2) Orbital Configuration: The DtS-IoT operates typically via Low-Earth Orbit (LEO) satellites, which offer lower latency and reduced transmission power requirements than their Geostationary Orbit (GEO) counterparts. Satellites are deployed following a specific configuration, as illustrated in Figure 1. In Walker Delta, the ascending nodes of the orbital planes are evenly spread out over 360 degrees, guaranteeing coverage of the most populated regions on Earth. Comparatively, Walker Star constellations distribute their ascending nodes 180 degrees, allowing for continuous coverage over more remote areas (e.g., closer to the poles).

3) Applications: DtS-IoT can lead to myriad applications, such as agricultural monitoring, environmental sensing, and maritime tracking, by ensuring that data from isolated devices can be reliably and efficiently transmitted to a centralized processing unit [8], [1]. This can thereby drive decisions and analytics in a global context.

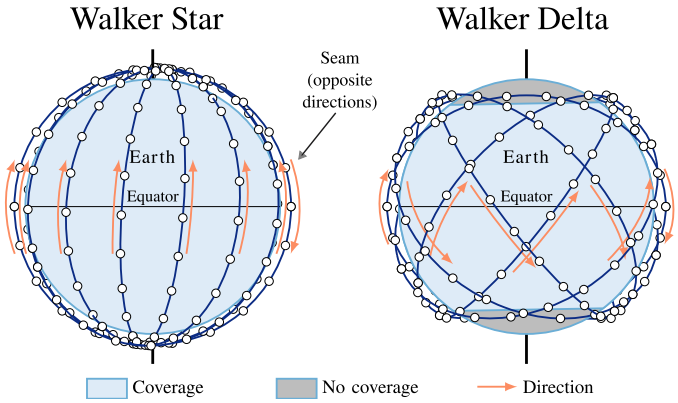

Figure 1: Walker Star and Delta Constellations.

4) Standardization: Significant effort has been made toward standardization, particularly in adapting existing technologies for the unique demands of satellite communication. The LoRaWAN Alliance has successfully adapted LoRa/LoRaWAN standards for the satellite context, ensuring that these LPWAN protocols can effectively utilize space-based operations. Concurrently, the 3GPP has made substantial advancements in standardizing Non-Terrestrial Networks (NTN) as part of their Release 16, 17, and subsequent iterations. These efforts aim to integrate satellite and aerial platforms into the global telecommunications infrastructure, providing a standardized framework that supports seamless, interoperable communication across terrestrial and non-terrestrial networks.

5) Challenges: Various challenges, such as energy-efficient communication, scalable network architectures, the trade-off between unlicensed and licensed bands, and reliable data transmission in constrained environments, are pivotal research areas that must be addressed to optimize and expand DtS-IoT deployments [8].

B. Delay Tolerant Networking

DTN is a specialized architecture tailored to overcome the unique challenges of space communications—such as vast interplanetary distances, signal disruptions, and stringent energy constraints [15]. Unlike conventional Internet protocols, DTN does not assume persistent end-to-end connectivity, which is often unfeasible in the complex and evolving topologies of large-scale space networks [16]. Originally developed to enable reliable data transfer across harsh space conditions, DTN reduces dependence on immediate node feedback. It achieves this through implementing the Bundle Protocol (BP), an overlay layer that enhances underlying protocols by introducing mechanisms like temporal data storage in transit [17]. This approach allows data packets to be retained until a viable transmission route emerges, adeptly handling the significant delays and frequent disruptions characteristic of interplanetary networks (IPNs) [18]. Furthermore, the store-carry-and-forward approach of DTN has also been considered in the con-

text of LEO with opportunistic ISLs, such as those described by DtS-IoT networks [19], [20]. These capabilities render DTN indispensable for sustaining dependable communication links in space exploration's vast and dynamic arena, where traditional terrestrial networking approaches are inadequate.

1) Contact Graph Routing: Traditional routing protocols (e.g., OSPF, OLSR, AODV) cannot accommodate store-carry-and-forward data handling. Thus, CGR is a fundamental routing strategy in predictable or scheduled DTN [10], particularly within the scope of space communications where conventional terrestrial networking protocols often fall short due to the unique challenges posed by the space environment [11]. CGR, developed by NASA's Jet Propulsion Laboratory (JPL), is primarily designed to efficiently determine viable routes for message delivery through a network, accounting for the high delay, intermittent connectivity, and other dynamic conditions inherent to space networks.

CGR constructs a contact graph, which represents the future scheduled contacts (communication opportunities) between nodes, annotated with their start times, end times, and the data rates available during the contacts [10]. When routing a message, CGR searches through this contact graph to find a series of time-ordered contacts that can transmit the message from the source to the destination while adhering to all the network constraints, such as contact times and data volume. The methodology optimizes for the earliest arrival time. Reducing the delivery time indirectly reduces the network's consumption of limited resources (like bandwidth and storage). This strategic optimization of resources and routing paths is vital in store-carry-and-forward data handling operations. The reader is referred to [10] for a detailed description of CGR algorithmics.

C. Simulator Tools

1) FLoRaSat: Simulator for DtS-IoT: FLoRaSat, standing for Framework for LoRa-based Satellite Networks, emerges as a crucial tool for simulating end-to-end satellite IoT networks, explicitly focusing on incorporating adaptations of LoRa and LoRaWAN for space-related applications [12]. FLoRaSat is publicly available under an open-source licence[1].

a) Platform and Libraries: Developed upon the OMNeT++ platform, a discrete-event simulator, FLoRaSat facilitates accurate and comprehensive simulations that seek to model and analyze the dynamics of space-terrestrial integrated IoT networks. The foundation of FLoRaSat lies in the integration of multiple pre-existing tools and frameworks—FLoRa, leosatellites, OS3, and INET—each contributing distinct features and capabilities to this new framework, thereby enabling a detailed simulation model that provides pertinent insights into the operational mechanics and challenges of deploying LoRa-based IoT networks in satellite environments.

b) Modules: The functionality and architecture of FLoRaSat are expansively detailed, covering diverse aspects from the Physical (PHY) layer to the Medium Access Control (MAC) layer, ensuring robust and adaptable simulations that mimic realistic space IoT networks. Furthermore, it extends its capabilities into simulating interactions with satellite gateways, featuring elements like orbital propagation, attitude control, and constellation creation to facilitate comprehensive simulation scenarios. The framework also demonstrates versatility in simulating ISLs dynamics, including topology control, dynamic link creation/destruction, and routing—essential for exploring connectivity and data transfer within satellite constellations. Meanwhile, features such as dynamic link creation/destruction and dynamic link latency updates are encapsulated in the Ground Segment.

2) DtnSim: Simulator for DTN: DtnSim is introduced as an event-driven simulation tool designed to evaluate DTN in various network scenarios, particularly in space-terrestrial networks that encounter disruptive and high-latency conditions [21]. DtnSim is publicly available online, providing a valuable tool in academic and practical contexts[2].

a) Platform: Like FLoRaSat, DtnSim is implemented in OMNeT++, a discrete event network simulator platform. It uses an event-driven framework to efficiently simulate scenarios at accelerated speeds, which is vital for analyzing space environments over extensive orbital periods.

b) Modules: DtnSim is flexible enough to evaluate various DTNs, including deep space systems and opportunistic LEO networks. It can transparently integrate existing flight software code as simulation modules due to its implementation in C/C++. Indeed, DtnSim interacts with the Interplanetary Overlay Network (ION) and its CGR library, which is the stack that JPL (NASA) developed. Also, DtnSim implements a variety of CGR flavors, including the so-called Rev17 (discussed in [22]), which we consider in this paper. However, DtnSim lacks the orbital dynamics and IoT-specific modules in FLoRaSat. The following section describes how CGR was ported from DtnSim and integrated into FLoRaSat.

III. Integration of CGR in DtS-IoT Data Delivery

To facilitate the transmission of packets sent by IoT nodes to distant destinations, we have incorporated the integration of CGR with ISL-capable LEO constellations in the FLoRaSat simulator. The implementation consists of two phases. The initial phase, explained in section III-A, relies on a contact graph generator. When provided with inputs of a specified number of satellites, IoT nodes/sinks, and distance constraints, the generator adeptly constructs a contact plan file conforming to the format required in the configuration step. Subsequently, as explained in section III-B, the second phase employs the generated contact plan file, or any other file adhering to the prescribed format, to execute simulations. The simulations are orchestrated using the specified contacts derived from the contact plan.

[1]The FLoRaSat Simulator is available to researchers and developers who want to experiment with it at https://gitlab.inria.fr/jfraire/florasat.

[2]The DtnSim Simulator is available to researchers and developers who want to experiment with it at https://gitlab.inria.fr/jfraire/dtnsim.

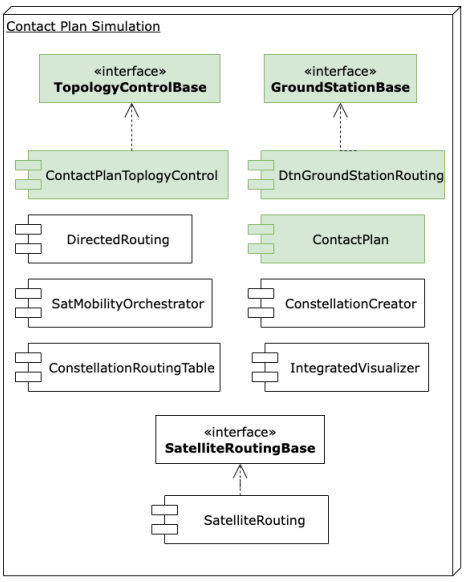

Figure 2: Contact Plan Simulation Architecture (Phase 1)

A. Phase 1) Contact Plan Generator

One of the primary goals of this implementation was to enable the FLoRaSat simulator to consume information from a contact plan file detailing connections between ground-based nodes, satellites, and inter-satellite links. To achieve the goal, we developed a contact plan generator that considers a fixed number of satellites, ground-based nodes, and distance constraints to generate both types of contacts: satellite-to-satellite and ground-to-satellite.

Figure 2 shows a component diagram with the high-level modules comprising the contact plan generator. Green modules correspond to the simulation components used by FLoRaSat to generate contact plans. Contacts are established through existing satellite mobility models. The design allows the simulator to be compatible with routing over DTN and non-DTN topologies (e.g., in the case of mega-constellations).

The satellite positions and parameters for the number of satellites and distance constraints can be manually inputted in the simulation's configuration file. Inter-satellite links are determined based on the distance between satellites. Therefore, if a satellite falls within a range of N kilometers from another satellite, it is considered a valid contact. The contact persists until the distance condition is no longer met. Each satellite is cross-referenced with adjacent satellites whenever the topology is updated, such as when satellites move and acquire new positions based on their mobility model. With this, the *TopologyControl* module determines whether a contact is starting, ongoing, or ending. Based on the contact information, a *ContactPlan* object is systematically updated each time a contact concludes. At the end of the simulation, the contact plan file contains the details of each contact. The initial configuration file (.ini)[3], contains six fundamental parameters for the contact plan generator phase: *a*) the number of ground-based nodes (i.e., IoT nodes and sinks), *b*) the number of satellites, *c*) the contact plan export file path, *d*) the general configuration for ground-based nodes, *e*) the maximum ISL distance, and *f*) the maximum ground-to-satellite range.

The general configuration of end terminals includes their positions in latitude and longitude coordinates, city names, IDs, type of node (e.g., IoT node with traffic generation or sink node), and local addresses. The main modules of the contact plan generator are detailed below.

ContactPlanTopologyControl. This module generates the contact plans after processing changes in the topology. In the *ContactPlanTopologyControl.cc* file, there are 3 essential methods. i) `updateTopology`. This method is called every time the topology is updated according to the update frequency parameter in the initial configuration file. ISL and ground-to-satellite are updated using this method as follows. i) `updateISL`. This method iterates over every satellite and retrieves four adjacencies: two adjacent satellites from the same plane, one adjacent satellite from the plane above, and one adjacent satellite from the plane below. Next, the method establishes the state of the contact, i.e., starting, ongoing, or ending. The state of the contact is determined by the distance between the pair of nodes under evaluation. ii) `updateGSL`. This method is analogous to `updateISL` except that it establishes the contact state between a ground-based node and a satellite. In the current implementation, the complexity of the method is $O(n \cdot m)$ where n is the number of ground-based nodes and m is the number of satellites.

ContactPlan. This module is called every time a new contact or range needs to be saved. At the end of the simulation, the module oversees finishing and recording any ongoing contacts. Afterward, the module exports all contact information to a file indicated in the initial configuration file.

B. Phase 2) Contact Graph Routing

This part of the work refers to implementing the CGR algorithm into the FLoRaSat simulator. This allows us to explore DTN routing in the context of Direct-to-Satellite IoT. Figure 3 illustrates the component diagram with the DTN's high-level functionality modules. Green modules correspond to the ones that provide FLoRaSat with routing capabilities to move bundles using CGR algorithms. The specifics of each module are detailed as follows.

DTNTopologyControl. This module oversees iterating over each satellite and ground-based node using a *ContactPlan* instance. Once the state of contact has been checked, the *DtnTopologyControl* creates the links between satellites and ground-based nodes, depending on the case and the contact information provided through the Contact Plan file. Once the topology is ready, nodes in the DtS-IoT network connected through a constellation of LEO satellites may transfer messages from a source IoT node to a destination sink in a store-carry-and-forward fashion.

[3]See sample file located at `simulations/dtnTutorial/contactplangenerator`.

Figure 3: DTN Simulation Architecture (Phase 2).

DTNGroundStationRouting. When the ground-based node is of type IoT node, it is tasked with the generation of bundles and transmission using the DTN logic in place. The *DtnTrafficGenerator* sub-module creates the necessary bundles at a time specified in the initial configuration file. The traffic generator sets traffic properties such as bundle number, start times, destination ID, and bundle size. Then, the *DtnPacketGenerator* sub-module receives a notification that triggers the generation of a *BundlePkt* instance, which is in turn sent to the core *DTN* sub-module. Depending on the routing algorithm in place, the DTN sub-module calculates the optimal route to reach the sink node at the destination. Once a bundle is forwarded to a satellite, the DTN module of said satellite handles the message and forwards it through ISL until it is delivered to the final destination.

DTNSatelliteRouting. This module handles and processes the messages received at the satellite. The *DTN* is the most critical sub-module since it calculates the optimal route, considering the contact plan and the bundles' custody model. The *DTN* interfaces with other *DTN* sub-modules to route bundles from the IoT node to the sink via intermediate satellites. The default routing model *Rev17* [22] was ported in the present work and can be set on the initial configuration file. The model incorporates advancements such as source routing extensions, adapting Dijkstra's algorithm, preventing routing loops, and multiple destination analysis.

IV. Evaluation

This section describes the simulation scenarios and analyzes the performance evaluation results of the operation of DTN and CGR in the context of DtS-IoT networks.

A. Scenarios

The scenarios are designed to reflect realistic DtS-IoT network configurations, including aspects of traffic patterns such as data sizes and transmission frequency, to analyze the effectiveness, efficiency, and robustness of the store-carry-and-forward approach in various operational settings. The simulation parameters are listed in Table I. The table shows that the study involves various realistic constellation configurations with satellites in orbit. Table II summarizes the 10 locations on Earth from where ground-based nodes exchange traffic through the satellite network. Every ground site behaves as an IoT node at certain times during the simulation. The IoT node generates nine sets of 100 bundles, each with one site from the remaining nine locations as the destination. The bundles are then transmitted toward the sink nodes, thus forming an all-to-all traffic pattern to sample multiple source-destination pairs worldwide. The constellation configurations and ground-based nodes locations are illustrated in Fig.4.

Table I: Simulation Parameters

Parameter	**Value**
Constellations	Walker Delta and Star (phasing = 0)
Altitude, Inclination	500 km, and 53 deg
Orbital Planes	4 and 8
Satellite Count	up to 64
GSL Data Rate	10 Mbit/s
ISL Data Rate	1 Mbit/s
ISL Config	2 across-track, 2 along-track
Max. ISL Range	5400 km
Max. GSL Range	2700 km
Min. Elevation	5 degrees
Bundles Sent	100 per sink node
Bundle Size	100 Bytes
Bundle Start Time	60 s
Bundle Interval	60 s
Bundle Generation	15 hours active + 9 hours iddle
CGR Variant	Rev17 with source routing

The interconnected constellation scenarios are compared with the same topology without ISL. In the case of configurations equipped with ISLs, the CGR algorithm facilitates data routing across multiple satellite hops. We generate contact plans based on four ISLs: two across-track (i.e., to neighboring orbital planes) and two along-track (i.e., within the same orbital plane). Conversely, in scenarios lacking ISLs, satellites transport data until it is forwarded directly to the destination. Even without ISLs, CGR is employed to identify the most optimal satellite, thus minimizing the data delivery time[4]. We leverage the source routing extension block of CGR so that only the sender node computes the route before embedding it into the transmitted bundle.

The metrics evaluated in these scenarios are averaged among all source-destination pairs in the all-to-all traffic pattern. The metrics include:

1) **End-to-End Delay:** Time elapsed from when a bundle

[4]Note that when no ISLs are present, CGR will still outperform a naive direct forwarding approach. With CGR, the sender device can compute which passing-by satellite will provide the fastest two-hop route to the destination (i.e., source-to-satellite and satellite-to-destination). A naive direct forwarding approach will send the bundle to the first passing-by satellite, which may not be the most expedited path to the destination. Therefore, CGR will consistently achieve equal or improved performance compared to direct forwarding methods.

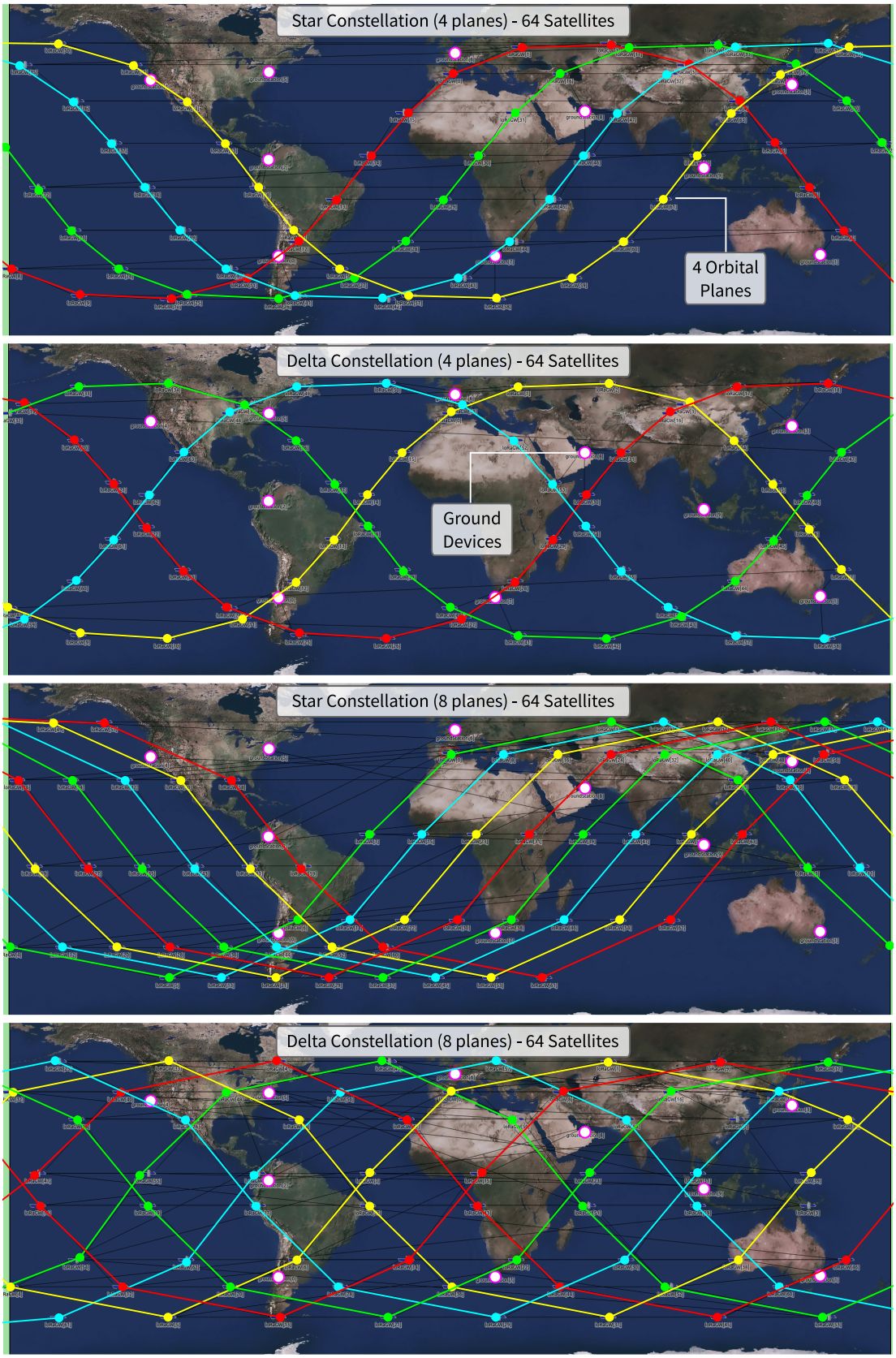

Figure 4: Star and Delta constellations with 4 and 8 planes. Screenshots from FLoRaSat were edited to highlight the orbits.

Table II: Ground Locations

Location	Latitude	Longitude	Altitude
Santiago	-33.45	-70.6667	520m
Sydney	-33.8688	151.2093	3m
Bogota	4.7110	-74.0721	2640m
Tokyo	35.6895	139.6917	40m
San Francisco	37.7749	-122.4194	16m
New York	40.7128	-74.0060	10m
Paris	48.8566	2.3522	35m
Cape Town	-33.9249	18.4241	12m
Dubai	25.276987	55.296249	0m
Singapore	1.3521	103.8198	15m

is sent from the IoT node until it is received at the sink node. We evaluate the distribution of delays across all successfully delivered bundles. The simulation is configured to ensure a delivery ratio of 100%, confirming that all bundles are accounted for in the analysis. Enough simulation time is given to ensure this. No packet loss is considered (e.g., a reliable underlying ISL data link layer is assumed).

2) **Hop Count:** This refers to the number of intermediary nodes (hops) a bundle passes through before reaching its destination. Each hop represents a transmission, reflecting the energy cost associated with the data transfer in the system. Our simulations guarantee that every transmitted bundle is successfully delivered, allowing for an accurate assessment of the system's energy efficiency.
3) **Routing Effort:** We monitor the number of times Dijkstra's algorithm is called during each simulation run to measure the computational effort required by the CGR. This metric helps evaluate and compare the processing load imposed by the routing strategy.

B. Analysis

In this section, we discuss the end-to-end delay and hop count results obtained from the two types of constellations, namely Walker Delta and Star, using 4 and 8 orbital planes and considering deployments with and without ISL.

4-Plane Scenarios

We first evaluate Star and Delta constellations deployed in 4 orbital planes, with and without ISLs. Results for each metric are discussed as follows.

End-to-End Delay: Figure 5 a) shows the simulation results for Delta and Star constellations, both with and without ISL. The results reveal insights into the network's performance concerning end-to-end delivery delay. Delays in the Delta constellation without ISL decrease as the satellite fleet size increases. This trend indicates improved coverage and an increased number of nodes, which enhance the availability of transmission opportunities. Consequently, the delivery delay is reduced due to the greater number of potential trajectories for message delivery, even when considering 2-hop routes. Adding ISLs to the Delta configuration substantially decreases delivery delay for fleet sizes of 32 or more satellites. Delta's smaller fleet sizes cannot meet the required ISL range. Thus,

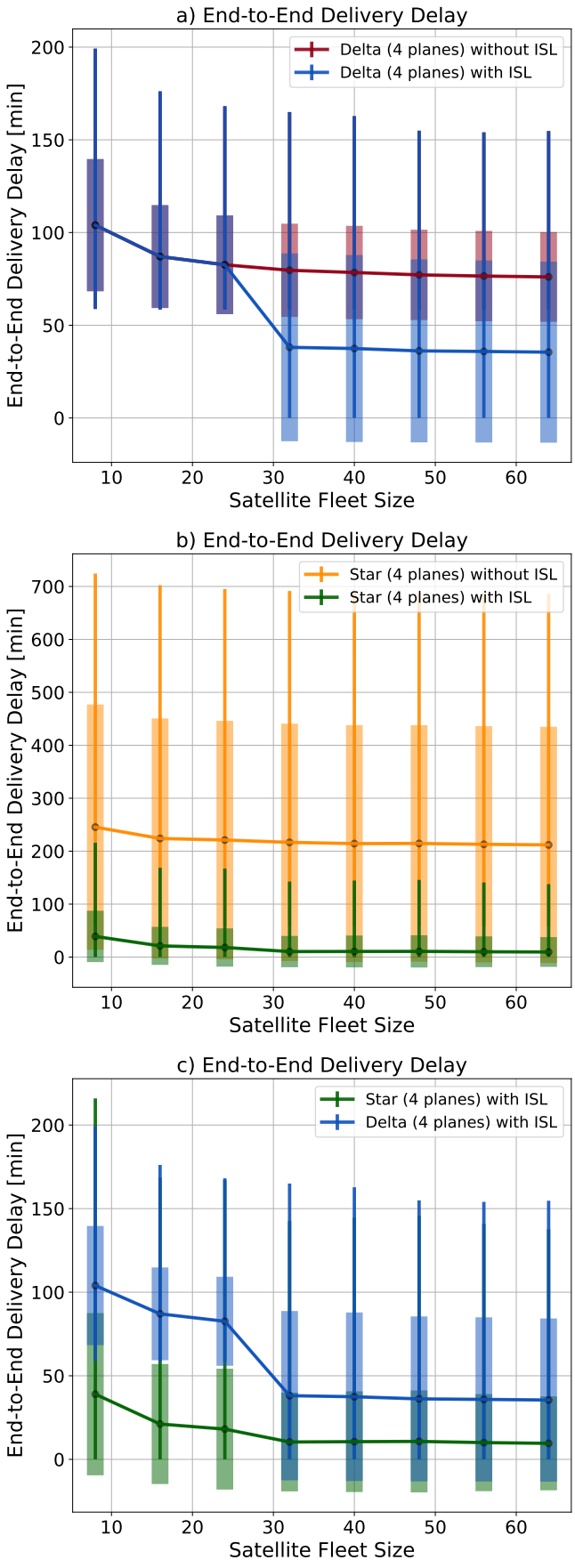

Figure 5: End-to-end delay for the Delta and Star Constellations using 4 orbital planes with and without ISL.

they behave as if no ISLs were present. The improved results for the 8-plane scenarios prove this is a consequence of the 4-plane parameter. For fleets with more than 32 satellites in Delta configuration, ISLs provide alternative routing paths and facilitate quicker data relays via ISLs. This is particularly advantageous in reducing the end-to-end delivery time. The result is a significant acceleration of data delivery and confirms the efficacy of opportunistic ISLs in enhancing communication paths. Note that the min/max spread is significant in the plots due to the different locations of the ground devices.

A similar pattern emerges in the Star constellation, where ISLs boost a delay reduction, as illustrated in Figure 5 b). Unlike Delta, the Star constellation can establish ISLs even with reduced fleet sizes. This is thanks to the proximity in the polar region, a phenomenon not present in inclined Delta constellations. The Star configuration with ISL shows lower delivery delays and reduced delay variability across fleet sizes, indicating a more consistent and dependable routing performance.

Figure 5 c) compares both configurations when equipped with ISL. The Star constellation frequently outperforms the Delta constellation, which is especially noticeable within smaller fleet sizes. This performance gap suggests the Star configuration may offer more efficient routing capabilities under specific fleet compositions. As mentioned, the Star constellation benefits from frequent over-the-pole connectivity, a topological advantage not present in the Delta configuration. As satellite fleet sizes expand, the Star constellation maintains a more stable delay profile, implying that its routing efficiency is better than Delta and less sensitive to changes in the number of satellites. Such stability is advantageous for scalability, allowing for reduced cross-linked constellations with improved relative performance.

Hop Count: The hop count metric is a crucial indicator of the satellite network's transmission effort and energy utilization. It accounts for the number of intermediary transmissions a bundle undergoes from origin to destination. For configurations without ISLs, the hop count remains constant at two: one transmission from the IoT device to the satellite and another from the satellite to the sink node. However, the hop count varies for ISL-enabled configurations and directly influences the energy consumed for data transmission.

As depicted in Figure 6, the Star constellation with ISL, while offering the best delivery times, incurs a higher transmission cost than the Delta constellation. This translates to increased energy expenditure on the satellites to relay the data across multiple hops. Notably, in Star configurations with 64 satellites, the hop count can escalate to as many as 18, whereas, in the Delta constellation, the largest observed hop count was 14. This observation highlights a trade-off in the Star constellation between performance and cost. While Star outperforms the Delta constellation in terms of delivery time for the same number of satellites, it demands more transmission effort, which impacts the satellite fleet's energy resources.

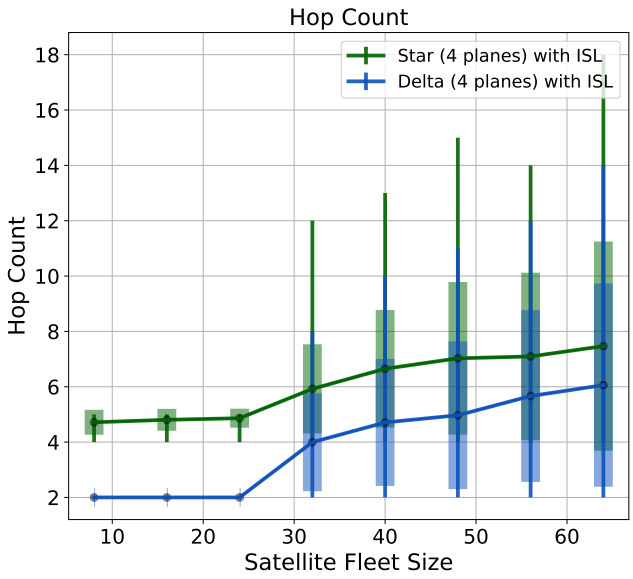

Figure 6: Hop count for the Delta and Star Constellations using 4 orbital planes with ISL. Configurations without ISL use 2 hops in all cases.

8-Plane Scenarios

In this configuration, we considered the same number of satellites deployed in a more granular 8-plane configuration for both Star and Delta, with the same traffic demand.

End-to-End Delay: The performance of the Delta constellation's end-to-end delivery delay critically depends on the number of orbital planes in operation. As evidenced by Figure 7 a), the augmentation from a 4-plane to an 8-plane constellation presents a stark improvement in delivery times, particularly noticeable in fleets with fewer than 30 satellites. This improvement indicates that the better-distributed 8-plane configuration does allow profit from the ISLs for smaller constellations. While the 8-plane configuration delivers lower delivery times for smaller satellite fleets, this architectural choice has significant economic implications. Deploying additional orbital planes typically incurs higher costs due to the need for more launch vehicles and the logistical complexity of coordinating multiple launches. Group-launching satellites are often organized plane-by-plane; thus, doubling the number of planes from 4 to 8 can substantially increase the project's overall expense.

The performance of the Star constellation, illustrated in Figure 7 b), shows consistency for 4 and 8 plane configurations. Indeed, unlike Delta, ISLs in Star already provide connectivity for smaller fleet sizes. However, when using 8 orbital planes, Delta outperforms Star in terms of end-to-end delivery delay.

Hop Count: Figure 8 a) illustrates the comparison of hop count for a Delta constellation with 4 and 8 orbital planes. The augmentation from a 4-plane to an 8-plane configuration causes ISL connectivity in the latter case to appear even for small fleet sizes. However, the increase in hops for constellations with 30 or more satellites does not result in

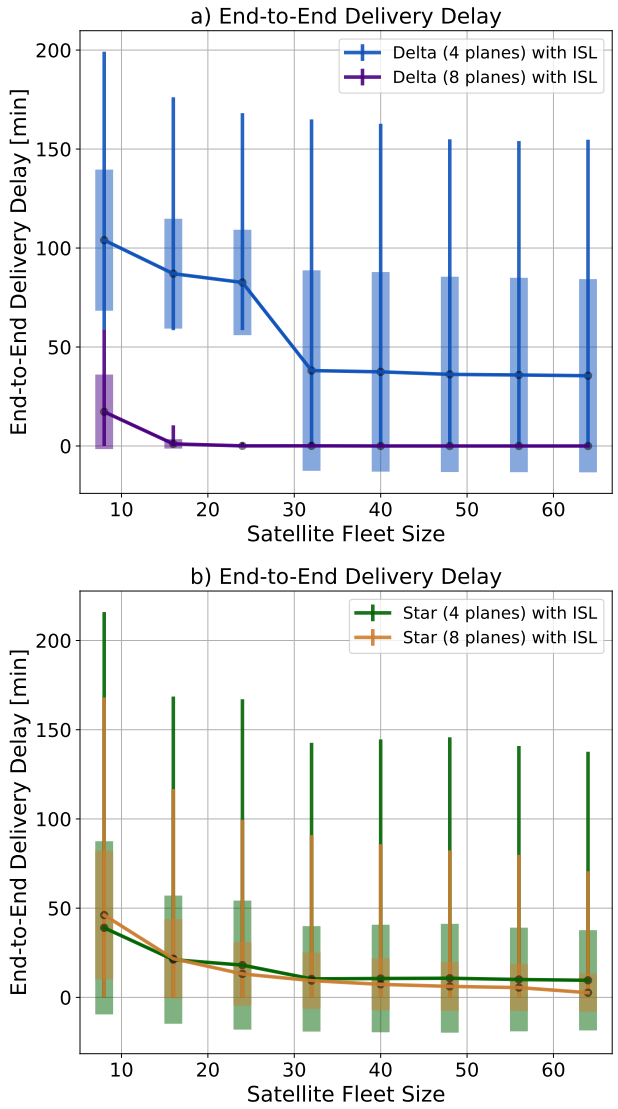

Figure 7: End-to-end delay for Delta and Star Constellations using 4 and 8 orbital planes with ISL.

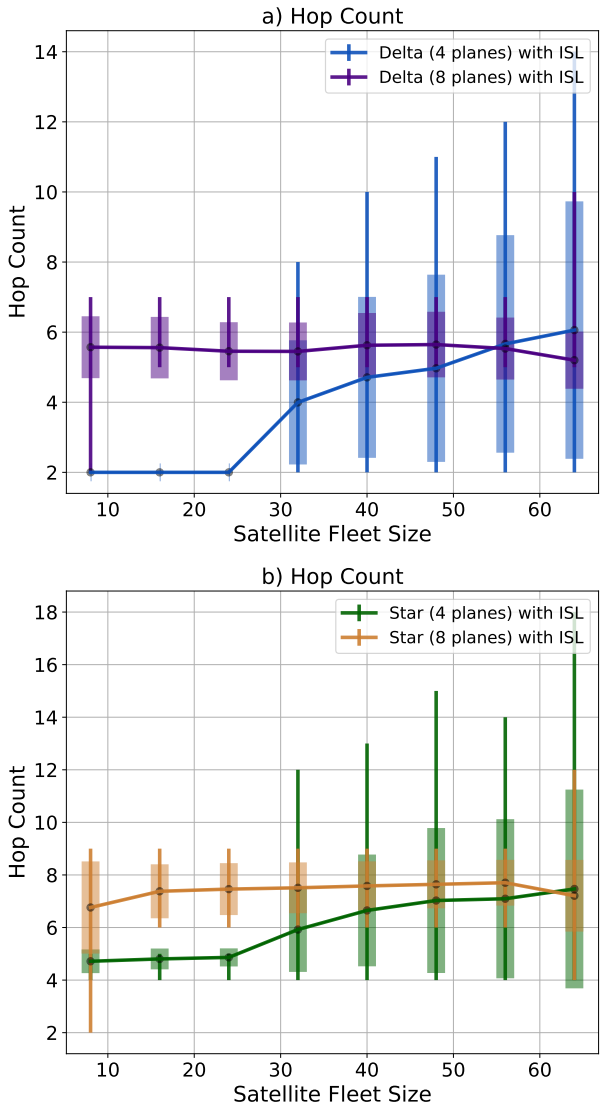

Figure 8: Hop Count comparison for the Delta and Star Constellations using 4 and 8 orbital planes with ISL.

significant energy consumption compared to the consumption spent in the 4-plane configuration. Furthermore, despite the higher number of hops remaining for the 8-plane configuration, end-to-end delay remains lower than in its counterpart with a 4-plane setting, as previously discussed. This could be seen as a contribution of CGR in finding paths that deliver packets faster despite having similar numbers of hops as alternative configurations. Note also that the peak hop count with 8 planes is reduced almost to half of the hops required for the 4-plane constellation in fleets with sizes larger than 30. The results for the Star constellation, depicted in Figure 8 b), show consistency with the end-to-end analysis. There is no meaningful difference in the average hop count for the two configurations, especially for larger fleet sizes; however, with 4 orbital planes, the peak number of hops is considerably higher than the 8-plane counterpart. As a result, there may be a significant increment in transmission costs for specific 4-plane Star configurations.

C. Routing Effort

Figure 9 plots the number of accumulated Dijkstra's calls on each simulation. Being the core subroutine of CGR, measuring the number of Dijkstra's calls directly indicates the overall routing effort performed by the routing module. This is especially important with the CGR under evaluation, considering that source routing was applied to find the best paths at the source node. In the case of DtS-IoT, the routing routine will run on an end device that, with a high probability, is resource-constrained. Results show that the effort scales linearly until 50 satellites. In any case, the total number of Dijkstra calls remains reasonable, considering they cover 24 hours for the evaluated system.

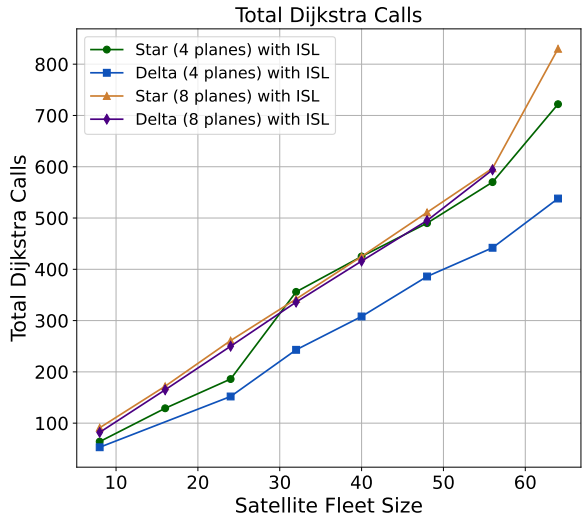

Figure 9: Dijkstra's calls recorded in 24 hours.

V. Conclusions

In this paper, we present the integration and evaluation of Direct-to-Satellite IoT (DtS-IoT) deployments that use Delay-Tolerant Networking (DTN) with Contact Graph Routing (CGR) to connect IoT nodes to other places on Earth for the delivery of data packets. Through a blend of theoretical exploration and practical implementation, the research studied the role of CGR in enhancing the reliability and efficiency of data transmission within space-terrestrial integrated IoT networks. We provided details of the implementation of CGR within the FLoRaSat simulator, which showcases the feasibility of such integration and opens avenues for further research and development in optimizing routing strategies for space communications. We evaluated Star and Delta constellations, with 4- and 8-plane configurations, considering the added connectivity of inter-satellite links. Results showed that cross-linked Star constellations equipped with DTN and CGR routing could provide a 10-minute end-to-end delivery delay. On the other hand, Delta constellations deployed in an 8-plane configuration can provide minimal delivery delay for fleet sizes above 24 satellites when forwarding data between populated regions. Future work includes the trade-off study of ISLs vs more ground stations, and along-track ISL only.

Acknowledgement

This research has received support from the European Union's Horizon 2020 R&D program under the Marie Skłodowska-Curie grant agreement No 101008233 (MISSION project) and the French National Research Agency (ANR) under the STEREO project ANR-22-CE25-0014-01, the Mitacs Globalink Research Award No IT38693 and ANID Basal Project FB0008. We acknowledge the support of NSERC Canada, RGPIN-2024-05730.

References

[1] M. Centenaro, C. E. Costa, F. Granelli, C. Sacchi, and L. Vangelista, "A survey on technologies, standards and open challenges in satellite iot," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1693–1720, 2021.

[2] J. A. Fraire, S. Céspedes, and N. Accettura, "Direct-to-satellite iot-a survey of the state of the art and future research perspectives: Backhauling the iot through leo satellites," in *International Conference on Ad-Hoc Networks and Wireless*. Springer, 2019, pp. 241–258.

[3] I. del Portillo, B. G. Cameron, and E. F. Crawley, "A technical comparison of three low earth orbit satellite constellation systems to provide global broadband," *Acta Astronautica*, vol. 159, pp. 123–135, Jun. 2019.

[4] N. S. Chilamkurthy, O. J. Pandey, A. Ghosh, L. R. Cenkeramaddi, and H.-N. Dai, "Low-power wide-area networks: A broad overview of its different aspects," *IEEE Access*, vol. 10, pp. 81 926–81 959, 2022.

[5] M. A. Ullah, K. Mikhaylov, and H. Alves, "Analysis and simulation of lorawan lr-fhss for direct-to-satellite scenario," *IEEE Wireless Communications Letters*, vol. 11, pp. 548–552, 3 2022.

[6] P. K. Sharma, B. Yogesh, D. Gupta, and D. I. Kim, "Performance analysis of iot-based overlay satellite-terrestrial networks under interference," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, pp. 985–1001, 9 2021.

[7] G. Stock, J. A. Fraire, T. Mömke, H. Hermanns, F. Babayev, and E. Cruz, "Managing fleets of leo satellites: Nonlinear, optimal, efficient, scalable, usable, and robust," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 3762–3773, 2020.

[8] J. A. Fraire, O. Iova, and F. Valois, "Space-terrestrial integrated internet of things: Challenges and opportunities," *IEEE Communications Magazine*, vol. 60, no. 12, pp. 64–70, 2022.

[9] O. Ledesma, P. Lamo, J. A. Fraire, M. Ruiz, and M. A. Sánchez, "Architectural framework and feasibility of internet of things-driven mars exploration via satellite constellations," *Electronics*, vol. 13, no. 7, p. 1289, 2024.

[10] J. A. Fraire, O. De Jonckère, and S. C. Burleigh, "Routing in the space internet: A contact graph routing tutorial," *Journal of Network and Computer Applications*, vol. 174, p. 102884, 2021.

[11] G. Araniti, N. Bezirgiannidis, E. Birrane, I. Bisio, S. Burleigh, C. Caini, M. Feldmann, M. Marchese, J. Segui, and K. Suzuki, "Contact graph routing in dtn space networks: overview, enhancements and performance," *IEEE Comms Magazine*, vol. 53, no. 3, pp. 38–46, 2015.

[12] J. A. Fraire, P. Madoery, M. A. Mesbah, O. Iova, and F. Valois, "Simulating lora-based direct-to-satellite iot networks with florasat," in *2022 IEEE 23rd WoWMoM*. IEEE, 2022, pp. 464–470.

[13] L. Space. (2024) Lacuna Space - Connecting Sensors. [Online]. Available: https://lacuna.space/

[14] Sateliot. (2024) Sateliot Space - 5G IoT From Space. [Online]. Available: https://sateliot.space/#about

[15] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture," Internet Requests for Comments, RFC Editor, RFC 4838, April 2007. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4838.txt

[16] ——, "Delay-tolerant networking architecture," Tech. Rep., 2007.

[17] K. Scott, S. Burleigh, and E. Birrane, "Bundle protocol version 7," Internet Requests for Comments, RFC Editor, RFC 9171, January 2022. [Online]. Available: http://www.rfc-editor.org/rfc/rfc9171.txt

[18] A. Alhilal, T. Braud, and P. Hui, "The sky is not the limit anymore: Future architecture of the interplanetary internet," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, no. 8, pp. 22–32, 2019.

[19] C. Caini and R. Firrincieli, "Application of contact graph routing to leo satellite dtn communications," in *2012 IEEE International Conference on Communications (ICC)*, June 2012, pp. 3301–3305.

[20] ——, *DTN for LEO Satellite Communications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 186–198. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23825-3_18

[21] J. A. Fraire, P. Madoery, F. Raverta, J. M. Finochietto, and R. Velazco, "Dtnsim: Bridging the gap between simulation and implementation of space-terrestrial dtns," in *2017 6th IEEE SMC-IT*. IEEE, 2017, pp. 120–123.

[22] J. A. Fraire, P. G. Madoery, A. Charif, and J. M. Finochietto, "On route table computation strategies in delay-tolerant satellite networks," *Ad Hoc Networks*, vol. 80, pp. 31–40, 2018.