



**HAL**  
open science

# Centralized and Decentralized Routing Solutions for Present and Future Space Information Networks

Juan Andrés A Fraire, Elías L Gasparini

► **To cite this version:**

Juan Andrés A Fraire, Elías L Gasparini. Centralized and Decentralized Routing Solutions for Present and Future Space Information Networks. *IEEE Network*, 2021, 35 (4), pp.110-117. 10.1109/MNET.011.2100102 . hal-03494101

**HAL Id: hal-03494101**

**<https://hal.science/hal-03494101v1>**

Submitted on 18 Dec 2021

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

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

# Centralized and Decentralized Routing Solutions for Present and Future Space Information Networks

Juan A. Fraire, *Senior Member, IEEE* and Elías L. Gasparini

**Abstract**—Space Information Networks (SINs) are pushing the boundaries of networking into near-Earth orbit and beyond. Nevertheless, the harsh and costly space context forces most risk-prone missions to be designed with minimal chance of failure. The careful consideration of routing protocols used to connect the fleet is by no means the exception. Although decentralized SIN routing schemes received most of the attention of the research community, centralized approaches are captivating the industry, as these come with tighter control and do enable troubleshooting of valuable space assets from a mission control center on ground. In this paper, we settle the matter by qualitatively and quantitatively comparing both SIN paradigms. We propose two novel centralized routing schemes, and evaluate their resulting performance against Contact Graph Routing (CGR), the decentralized state-of-the-art. In this evaluation, computational effort, memory utilization, and energy consumption are taken as figures of merit. The outcome provides compelling evidence that centralized routing schemes can safely, successfully and efficiently connect SINs with up to 10k daily contacts, while CGR can be better entrusted with larger-scale SIN deployments.

**Index Terms**—Space Information Networks, Contact Graph Routing, Store-Carry-and-Forward, Delay-Tolerant Networking.

## I. INTRODUCTION

ADVANCES in key enabling technologies such as miniaturization, formation flying and inter-satellite communications, are making of Space Information Networks (SINs) an increasingly attractive means to retrieve and transport data to and from remote places of our planet. With more than 50,000 satellites to be launched in the next 10 years [1], the “new space” community is thriving with significant reductions in components’ weight, size, price, and acquisition lead time. As a result, new private investors and companies are joining the experienced public sector in deploying cross-linked constellations of small and nano-satellites [2] in Earth orbit, as well as ambitious networked deep space missions.

Despite the increasing popularity of SINs, no careful analysis is available regarding the adequacy of implementations or adaptations of protocols and routing schemes originally designed for terrestrial Internet. Notably, orbital dynamics, large distances, and limited power resources impose severe constraints with respect to delay and disruption on the transmission of data across SIN nodes. Although meshed and resource-rich mega-constellations are on the roadmap, early

deployments will need to render sparse and partitioned topologies with unstable end-to-end-paths. Furthermore, modest SIN missions devoted to asynchronous data collection and Internet of Things services are better fitted to communications via sporadic links. Thus, besides an Internet-like scheme of immediate forwarding (IF), sparse and partitioned SINs will require the calculation and exploitation of multi-hop store-carry-and-forward (SCF) routes (see Fig. 1).

The traditional Internet only implements IF, which can be viewed as a particular case of SCF with no storage needed, as partitions or delays are rarely an issue in Earth networks. The fact that the more general SCF routing approach is not part of the Internet paradigm is a challenge to direct and integral migration of terrestrial routing protocols to SINs. Another differentiating feature of SINs is that their topological evolution are induced by predictable spacecraft trajectories along orbits, which in turn can be pre-computed and thus exploited for routing purposes.

In this context, the state-of-the-art routing approach is Contact Graph Routing (CGR) [3], [4], developed and maintained by JPL, NASA. CGR exploits the topological information encoded in so-called *contact plans* to enable computation of routes on-board and on-demand, making CGR a decentralized routing scheme. Route computations occur at every SIN node including those in ground and space, which enables the in-time utilization of latest traffic queuing status. However, space nodes are typically equipped with resource-constrained on-board computers and are powered by limited amounts of energy, harvested from solar panels. Moreover, decentralized routing approaches constitute a challenge for close follow-ups of autonomous routing decisions on remote and disconnected nodes.

These issues are driving the more conservative players in space industry to explore centralized or software defined networking approaches [5], while at the same time, the more sophisticated SIN protocols and decentralized routing schemes are gradually validated in orbit. But the centralized paradigm has its drawbacks. Although computational power, and up to some extent also time, can be considered as being unbounded on ground premises, the lack of up-to-date information on the traffic status can provoke an over-computation of potentially needless routes. Furthermore, centralized approaches demand extra energy and bandwidth for transmitting and provisioning individual route tables to each of the node in the SIN fleet.

This work presents the first in-depth qualitative and quantitative comparative analysis of the decentralized state-of-the-art with novel centralized approaches to compute SIN routes. Inspired by CGR variants [6], a novel Centralized

J. A. Fraire is with Univ Lyon, Inria, INSA Lyon, CITI, F-69621 Villeurbanne, France, CONICET - Universidad Nacional de Córdoba, Córdoba, Argentina and Saarland University, Saarland Informatics Campus, Saarbrücken, Germany (e-mail: juan.fraire@unc.edu.ar).

E. L. Gasparini is with FAMAF - Universidad Nacional de Córdoba, Córdoba, Argentina.

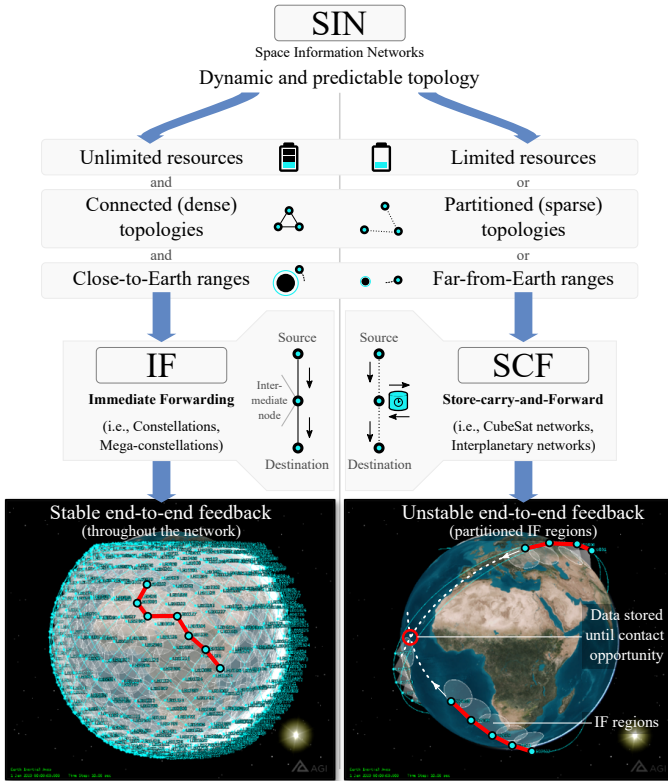


Fig. 1: SIN's Taxonomy: Immediate Forwarding (IF) and Store Carry and Forward (SCF).

First Ended (CFE) approach is described together with an exhaustive Breath-First Search (BFS) approach that ensures the discovery of all routes on ground. To the best of authors knowledge, CFE and BFS are the first centralized routing proposals applicable to a SIN routing model comprising both IF and SCF traffic flows. An in-depth performance evaluation of CGR, CFE and BFS is laid out with computational effort, memory and bandwidth utilization, and energy consumption as figures of merit.

The remainder of this paper is organized as follows. Section II provides the background of SINs, to then introduce the SIN routing model in III. Section IV presents the performance evaluation and comparison of centralized and decentralized schemes. Section V draws the conclusion and lists future research directions.

## II. BACKGROUND

Routing in Terrestrial Information Networks (TINs) has been studied for decades. TINs assume round trip times in the order of milliseconds, which enables a shared global view of the topology, and immediate feedback across all nodes in a path. As a result, intermediate nodes in paths work optimally by swiftly forwarding packets to the next hop, without the need of keeping a copy for prolonged time periods. Moreover, topological changes in TINs are infrequent, while data channels are bidirectional and mostly symmetric.

Routing in SINs enjoys none of the terrestrial assumptions. Topology is constantly changing as satellites orbit and planets rotate; while disruptions are numerous due to planet

occlusions or transponder duty cycling designed to fit energy constraints [7] (i.e., eclipses). In the case of deep space missions, unidirectional signal propagation delay in the order of minutes or hours also obstructs the immediacy assumption of classical terrestrial Internet protocols. These conditions generally force nodes to keep in-route packets in memory for potentially extended episodes (thus the *carry* in SCF) until links become available, or until the transmitted data has traversed the vast interplanetary space.

In the aim of extending Internet connectivity to remote areas, some SINs have been provisioned with extensive resources and densely connected topologies in order to comply with TIN stability assumptions. Thus, IF routing schemes are enabled by massive LEO deployments of tens of thousands of satellites such as SpaceX's Starlink mega-constellation in near-Earth space. Although expensive and resource-hungry, mega-constellations can leverage traditional mobile and Ad-Hoc networking solutions already matured in terrestrial environments. Issues and failures on remote nodes in the network can be reacted upon immediately thanks to instantaneous feedback from every element present in the system. SCF, on the other hand, is implemented by means of Delay/Disruption Tolerant Networking (DTN) architectures [8] and the Bundle Protocol [9], which are standardized by IETF and CCSDS. The Bundle Protocol operates as an overlay layer networking sporadically connected IF clusters over delayed and disrupted hops, as illustrated in Fig 1.

DTN protocols and routing schemes have received significant attention, especially from the space community. Several Bundle Protocol software stacks have been developed and flight tested, including NASA's Interplanetary Overlay Network (ION) [10], and  $\mu$ D3TN (former  $\mu$ PCN [11]). Near-Earth validations of DTN comprises the United Kingdom Disaster Monitoring Constellation (UK-DMC) in 2008, Japanese space agency (JAXA) geostationary satellites in 2012, the International Space Station (ISS) since 2018, and ESA's LEO flying laboratory, OPS-SAT in 2020. Deep space experiments such as NASA's DINET (Deep Impact Network) in 2008 also proved the utility of DTN over distances in the order of light minutes. Nevertheless, the maturity of DTN is still incomparable with Internet protocols in TINs, and will likely remain like this for the years to come.

This fact is not ignored by the thriving "new space" sector aiming at deploying SINs based on small- and nano-satellite constellations, or the most traditional established space industry developing SINs solutions for more challenging and expensive cislunar and interplanetary missions. These actors are already taking unprecedented risks by launching multiple complex spacecraft interconnected with sophisticated protocols and routing schemes at the core of the mission critical backbone. In this context, decentralized routing computation might be seen as an unnecessary risk, when compared with routes that are pre-computed on ground. Indeed, centralization facilitates debugging while keeping a better knowledge of the operational status relative to a SIN based on not yet widely tested protocols and routing algorithms. Furthermore, in partitioned networks, remote issues occurring in disconnected regions of the system might spend significant time

unnoticed, complicating a timely troubleshooting. Thus, it is highly likely that early SIN deployments will be rolled-out with a centralized concept of operations, as promoted among others by the software defined networking vision [5].

Previous efforts have sought to narrow down decentralized routing decision taking by means of pre-processing of contact plans on ground, also known as contact plan design [7]. Although this favors a tighter control of resulting routing tables, the final decision still rest on the decentralized CGR routine running on-board. On the other hand, static DTN routing, supported by CCSDS and IETF specifications, assumes the routing table is managed by mission operations personnel on Earth. However, in contrast with the more challenging decentralized approach, and despite its immediate utility to the space industry, the automated centralized route computation for SINS has been rather overlooked by the research community.

### III. SIN ROUTING MODEL

The routing model for SINS spans three distinct phases (see Fig. 2). The first phase is a) planning, where ground control premises predict future satellite positions with the help of orbital propagators. The node trajectories are enriched with communication models (i.e., antenna patterns, pointing, transmission power, receiver sensitivity, etc.) to derive the contact plan, which represents the upcoming connectivity of the network. In a second phase, b) routing, the contact plan serves as the basis for algorithms such as CGR to calculate routes. If route calculation takes place in-orbit, we are framed in a *decentralized* routing approach, which needs to assume the contact plan was timely provisioned to all orbiting nodes in advance. On the other hand, a *centralized* routing approach implies that route determination happens on ground, followed by their distribution to all nodes in the SIN. The main challenge of distributed CGR is the computational demand imposed on resource-constrained on-board computers, while the centralized approach crucially relies on timely distribution of route tables via long-range space links. In any case, the outcome is a route table that for each path indicates the next hop node, the best delivery time (BDT) expected for that route, and the time interval on which the route is valid, among other parameters. The information in the route table is finally consulted in the third phase, c) forwarding, by the satellites, in order to place the traffic in the corresponding outbound buffers for transmission towards the respective target nodes.

The contact plan is comprised of *contacts*. A contact  $C_{A,B,d}^{t_{start},t_{end},r}$  is a time interval  $(t_{start}; t_{end})$  during which it is expected that data can be transmitted by node  $A$  (the contact's sending node) at rate  $r$  such that it will be received by node  $B$  (the contact's receiving node) at distance  $d$  (thus, with a propagation delay of  $c/d$ , where  $c$  is the speed of light). The definition of contacts is flexible enough to model unidirectional links, asymmetric channels (different contacts in each direction), interplanetary communications, and time-evolving data volume utilization including prioritized traffic allocations. This model is at the core of CGR, which constructs contact graph structures to compute optimal routes, expressed as a finite sequence of contacts [4]. A route  $R_{src}^{dst}$  is valid

as long as a) the sending node ( $src$ ) for the first contact is the transmitting node, b) the receiving node ( $dst$ ) for the last contact is the destination node, c) the receiving node for contact  $i$  is the sending node for contact  $i + 1$ , and d) the time at which contact  $i + 1$  ends is no earlier than the time at which contact  $i$  begins [3]. It is straightforward to compute end-to-end delays from  $R_{src}^{dst}$ , which can be applicable to IF (when node delays along the route = 0) and SCF (with delays  $\geq 0$ ) traffic flows in SINS.

#### A. Contact Graph Routing

CGR is a decentralized routing approach. When using CGR, each orbiting SIN node explores the contact plan, constructs a contact graph, and then employs adaptations of Dijkstra's shortest path algorithm in order to compute the required routes for a specific outbound traffic and destination. A contact graph is a conceptual directed graph  $CG_{src}^{dst} = (V, E)$  where vertices  $V$  correspond to contacts  $C_{A,B,r}^{t_{start},t_{end},d}$  in the contact plan. Edges  $E$  in a contact graph can be seen as episodes of data retention (in SCF) at a node  $i$ , between the end of the earlier contact and the start of the subsequent contact (see Fig. 3-a)). This static graph representation facilitates the execution of network algorithms such as Dijkstra over time-evolving networks on simple graph structures [4]. In order to construct route tables out of subsequent Dijkstra calls, heuristics are applied in CGR [6], being an adaptation of the Yen's algorithm [12] the current approach is to add up to  $K$  paths to the table in the ION protocol stack. Indeed, CGR populates the route table for a given destination only when a local or in-transit bundle (Bundle Protocol data unit) needs to be forwarded. Thus, the fact that one contact graph exists for each source and destination pair nicely fits the on-demand routing triggered by the local node for a specific target node. But the consideration of CGR for centralized schemes requires of more appropriate data structures for an all-to-all route table pattern.

#### B. Centralized Heuristics

Decentralized CGR suffers the aforementioned issues of (i) demanding on-board processing capabilities which are not always available, and (ii) disposing sensible decision making to remote and potentially unreachable SIN nodes. Instead, a centralized CGR variant would overcome these issues at least during initial SIN deployments, until SCF protocols have proven to be reliable, and sufficient on-board (radiation tolerant) computation power has become the norm.

A centralized routing algorithm computes on-ground all routes a SIN node would potentially need in a configurable time horizon in which orbital propagation remains accurate (typically in the order of days). As routes are consumed, route tables must be updated with recently computed data, similar to contact plan distribution in CGR. However, enumerating all possible paths between two nodes in a graph renders a non-polynomial output size, which means the route determination can easily become computationally intractable (not to mention the provisioning of large route table sizes). Since even terrestrial computational power might turn out insufficient, large

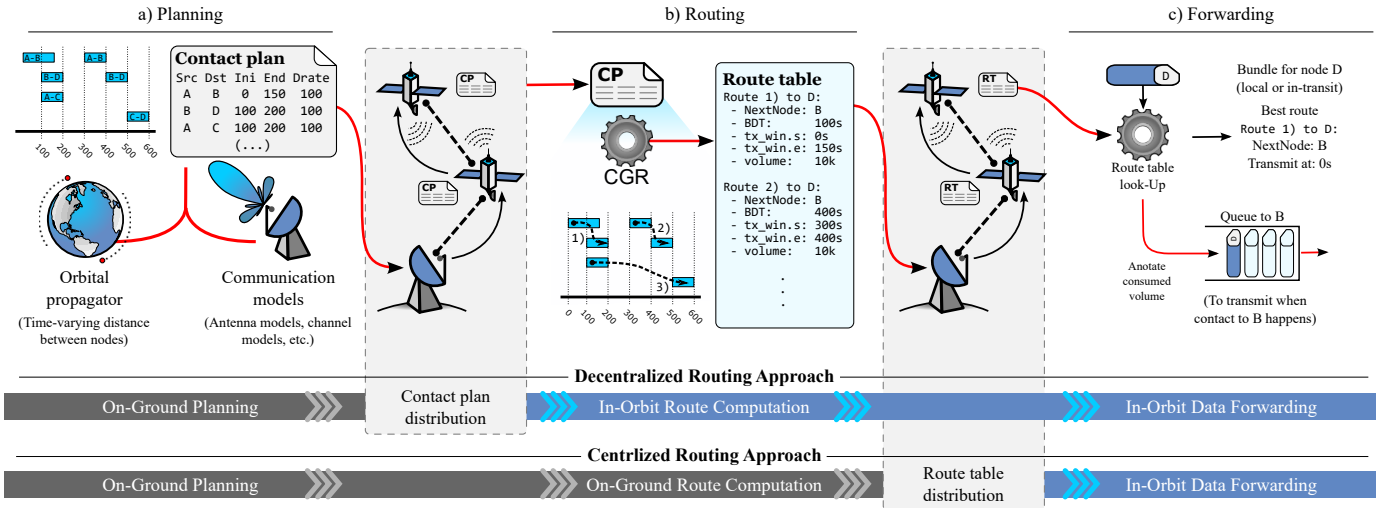


Fig. 2: Routing phases in SINs: a) planning, b) routing, and c) forwarding. The decentralized approach performs route computation in-orbit, while the centralized approach does it on-ground to later distribute the resulting routing tables.

contact graphs in centralized schemes also need heuristics to bound the number of discovered and provisioned routes. To this end, we propose two centralized algorithms inspired by CGR to produce routing tables on ground.

1) *Centralized First Ended (CFE)*: Motivated by the decentralized first ended approach in [6], this heuristic suppresses the contact with earliest ending time ( $\min(t_{end})$ ) from the latest route found by the Dijkstra algorithm. Subsequent calls using the same contact plan will thus find alternative and different paths. Instead of a contact graph, CFE operates over a multigraph data structure (where edges are contacts) for which simple Dijkstra adaptations exist [13]. While contact graphs are bounded by  $C$  vertices and  $C^2$  edges (assuming  $C$  contacts are listed in the contact plan), the multigraph is bounded by  $N$  vertices ( $N$  being the network nodes) and  $C$  edges (see Fig. 3-a). Furthermore, while decentralized CGR preemptively terminates Dijkstra once the destination node was found (routes are sought for specific traffic flows, with a specific destination node), CFE continues its execution to compute the best route to all other nodes in the network (i.e., the minimum spanning tree). The optimization metric remains in all cases the delivery time (relevant in SCF), being hop count the second optimization criteria (relevant in IF).

2) *Breath-First Search (BFS)*: A BFS approach is also possible over the multigraph, such that the best  $K$  paths are obtained for each destination, at each time slot  $t \in T$ , where  $T$  is an arbitrary contiguous time set. The BFS algorithm will backtrack in a BFS manner, pruning branches based on parameter  $K$ . For a given  $t$ , every contact with end time  $t_{end} < t$  will be ignored when constructing the graph. The rationale is that for some time  $t$  in the future, routes with expired contacts will no longer be valid, time by which alternative paths must be available in the route table. Then, BFS iterates over the resulting reduced graph, expanding one hop from the source node on every step (only for valid hops for a route), and storing the routes found up to  $K$  entries per destination. Note that in Fig. 3-b) two routes from A to C exist. The contact  $A \rightarrow C$  is analyzed in the first BFS expansion,

producing a new route. But after exploring the node B, a better route is found from A to C. If the  $K$ -parameter were to equal 1, the route entry from A to C would be replaced with the new route (keeping the most efficient route). Since the approach could revisit nodes already in the route path, loops shall be eliminated. Fortunately, this is easily achievable by a linear search over the resulting path. Nevertheless, BFS is a brute force approximation where the solution space is bounded by  $t$  and  $K$  in order to permit its practical use. Although less processing-efficient, BFS ensures no routes are missed. It can thus be used to cross-check other centralized approaches, while serving as baseline for comparison.

It is worth recalling that both CFE and BFS are centralized approaches, and by default compute routes for all source-destination pairs. As discussed in the evaluation section, a filtering that leaves only routes to nodes that are somewhat expected to be the destination by the actual traffic of the SIN can reduce the route table provisioning cost in terms of bandwidth and energy consumption.

### C. Failure Resilience

Space is a harsh environment. Radiation, temperature cycling, and limited power supply, among others might provoke unexpected failures that thus might undermine the proper establishment of a contact. When a next-hop contact fails, the node will need to fall back to the route table (pre-provisioned by a centralized node, or locally pre-computed) to obtain the next best route to the same destination, but with a different entry contact. An alternative route could start immediately (current time slot  $t$ ), later in the future, or not exist at all. In particular, centralized approaches with one path per time slot ( $K = 1$ ) can find alternative routes to the destination starting in subsequent slots, while distributed schemes can perform new local computations to reactively discover new paths (possibly starting at the current time slot). In addition, hybrid solution can be devised that harvest centralized route determination, but also include capacities to locally compute

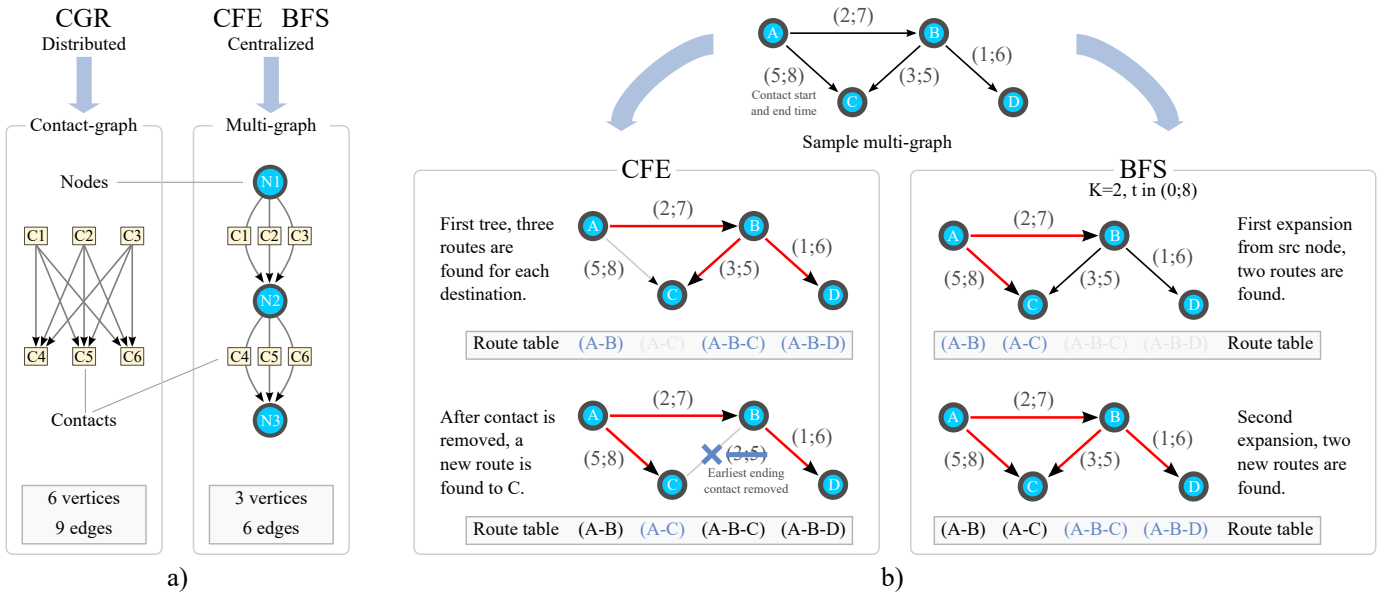


Fig. 3: a) Distributed and centralized graph models. b) Centralized CFE explores a multi-graph removing earliest ending contacts after each Dijkstra call, while BFS iterates over the graph storing up to  $K = 2$  routes for time  $t$ .

paths in case of failures. The greater subject of failure resilience in SIN routing is an appealing research path, yet to be explored.

#### IV. EVALUATION

In this section we evaluate and compare the decentralized and centralized approaches discussed above. We study CGR First-Ended [6] as the decentralized baseline, and contrast it with CFE and BFS (with  $K = 1$  and  $T$  slotted on a per-second basis). Note that CGR computes routes on-demand, and is thus sensitive to traffic patterns, while CFE and BFS proactively calculate routes for the worst-case traffic condition for all possible source and destination pairs.

In this study, telemetry traffics are considered with high, medium, and low generation rates, where a bundle destined to mission control on ground is generated every hour, two hours, and three hours respectively. The bundle size is assumed to be negligible with respect to the link capacities to avoid congestion and re-routing biases in the results. As CGR, CFE and BFS schemes decide to forward based on route tables with identical metrics (and optimal, assuming no losses/failures) - achieving the same packet delivery statistics- we focus on the effort required to obtain such tables, namely computation time, memory utilization, and energy consumption. Also, source routing is assumed, implying the end-to-end path is only computed and applied once in CGR, which favors a more efficient decentralized computation.

SINs scenarios are described by contact plans comprising the complete simulation period (i.e., no contact plan updates are performed within the simulation). The first scenario is a Walker constellation implementing the Ring Road Network in [14] of 12 LEO satellites (space nodes) collecting data from 100 points of interest (PoIs) and delivering them to 10 Internet gateways distributed on the Earth surface. The scenario comprises 9,700 contacts spanning a duration of 24

hours. The second scenario is the Lunar network in [15] composed of 8 satellites orbiting the Moon relaying data from 8 sensors on the far side of the Moon to 4 Lunar stations on its surface which in turn are connected to 3 NASA's DSN antennas on Earth (for a total of 20 space nodes). The scenario includes 10,000 contacts over two weeks (i.e., 714 contacts per day). Finally, we also consider a random network generator for 63 space nodes with varying topologies ranging from 5k up to 13k daily contacts (links assigned uniformly among node pairs). These random scenarios enable an extended sensitivity analysis of the figures of merit for larger SINs.

Scenarios are generated using realistic orbital parameters obtained by the STK software from AGI, while the network traffic was simulated in the DtnSim tool. The CGR, BFS, and CFE routing algorithms have been implemented in the supporting Omnet++ framework<sup>1</sup>.

#### A. Computation Time

We analyze the computation time as a metric to assess the route calculation effort. To this end, we run all simulations in the same hardware, an Intel Core i7-4790, at 6.80 GFLOPS, single core and measure the time intervals for each of the CGR, BFS and CFE routines. To determine their duration if instead executed in on-board computers, we take as reference the Proton 200k OBC with a total computational power of 900 MFLOPS (a powerful flight computer which is only 7.5 times slower than our simulation hardware). We apply this factor to the recorded computation times.

As expected, computation time results in Fig. 4 show that CFE is more efficient than the brute force baseline in BFS.

<sup>1</sup>The detailed orbital parameters, communication models and constraints of Scenario 1 and 2 can be found in [14] and [15]. The random networks used for Scenario 3 are available as part of the software toolchain, which is open-source and is publicly available at <https://bitbucket.org/lcd-unc-ar/dtnsim/branch/centralized-model>.

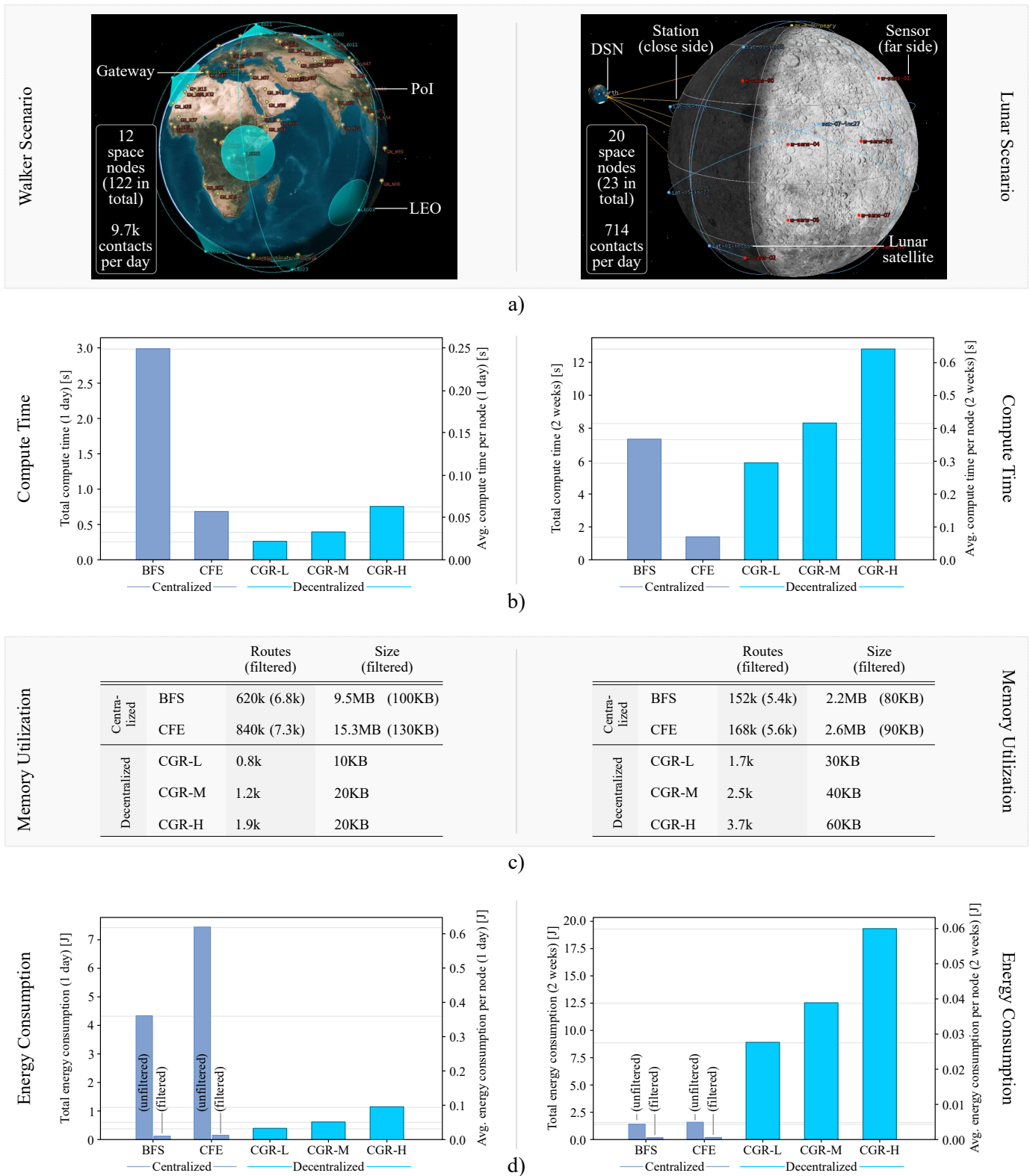


Fig. 4: a) Walker and Lunar scenarios, b) compute time, c) memory utilization, and d) energy consumption.

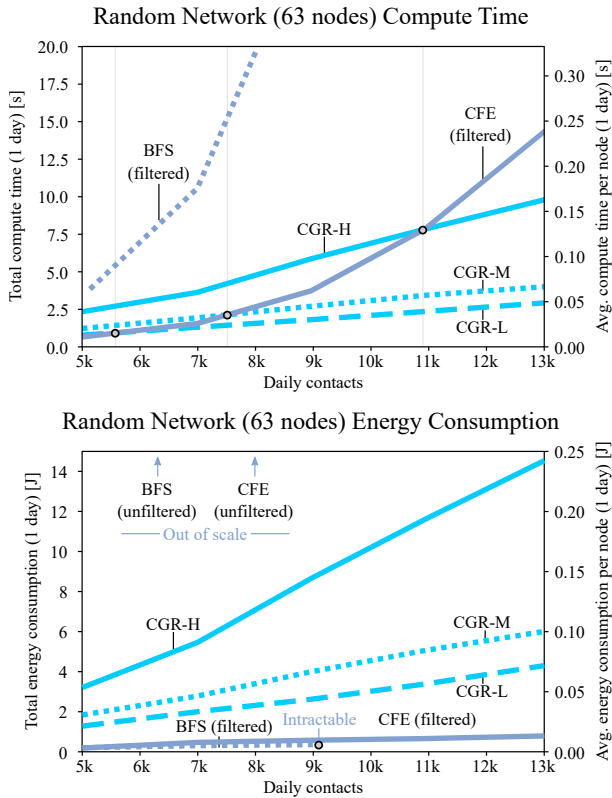


Fig. 5: Compute time (top) and energy consumption (bottom) results for random network scenarios.

The time requirements of centralized CFE can compete with decentralized CGR under high traffic rates in the Walker scenario, but lower traffic loads require less efforts if using the decentralized paradigm. In the less densely connected Lunar scenario, on the other hand, CFE outperforms CGR with respect to computation times in all cases. Even BFS is comparable with CGR under low and medium traffic loads. This trend is confirmed in the random network scenario outcomes, which is studied for a varying and higher density of contacts. Specifically, the plot in Fig. 5 clearly indicates that accumulated computation times in CFE are lower for contact plans with up to 5.5k, 7.5k, and 11k contacts per day for low, medium and high traffic rates respectively. In all cases, we notice that the computation times if averaged per node remain below the second of CPU dedication, an important evidence that slower on-board processors are already up to the challenge for decentralized CGR approaches.

It is important to recall that although useful for a quantitative comparison, the computation time for centralized approaches take places on ground (a qualitative difference). Also, we have disregarded optimization approaches such as parallelization or data-center class computation solutions likely available in most mission control centers.

### B. Memory Utilization

Since centralized SIN routing computes all routes for all possible source-destination pairs, it is relevant to monitor the memory (and bandwidth) utilization while provisioning the

SIN. As expected, the memory utilization table in Fig. 4 shows there is a 3 orders of magnitude difference in the number of routes computed on ground than those actually required by the decentralized CGR. This holds true for both Walker and Lunar topologies. However, the number of routes in the centralized approach can be reduced by up to 95% if filtering only those that include the mission control node as destination. Evidently, such filtering can only be applied when information of the forthcoming traffic is available.

It can be noted that centralized BFS and CFE find different number of routes. This is because they exploit different search approaches: CFE is based on Dijkstra over multi-graphs, while BFS uses brute force over contact graphs spanned by a choice of the parameters  $t$  and  $K$ . Nevertheless, we have cross-checked that discovered routes are metric-wise equivalent (delivery time and hop count) for every moment in the simulation for both CFE and BFS. This finding is empirical evidence that CFE is able to provide optimum network-wide connectivity with less number of routes.

An estimation of the route table occupation in memory is also presented in Fig. 4. In this case, we assume that each route is a stream of contact identifiers of 4 Bytes, plus an identifier for the corresponding contact plan file (distributed both in centralized and decentralized schemes), and 3% extra of coding overhead [3]. Route tables in the order of 100 KB are still manageable with typical RF uplinks. Nevertheless, unfiltered tables in the order of MB might present a challenge for massive mega-constellations, considering that a different table is to be independently uploaded to each node in the SIN.

### C. Energy Consumption

To study the energy consumption of the decentralized approach, we assume a power figure of 1.5W in the on-board processor at full activity, as reported in the Proton 200k computer data sheet. On the other hand, the centralized approach is penalized with the RF transmission cost of distributing the route table (contact plan distribution is neglected as is a common effort in both schemes). To this end, we assume a transmission consumption of 12W, which is a typical power consumption of modern X-Band transponders (including RF amplifiers), which can operate up to 150 Mbps. The route table sizes discussed above were used to compute the energy cost of provisioning the tables to each one of the SIN nodes.

Energy consumption results in Fig. 4 shows that significant power saving can be achieved in centralized schemes as long as an adequate route filtering is applied. Otherwise, for the larger Walker scenario, the route table size become prohibitively large, so that the cost of transmitting it via RF links notably exceeds the cost of computing it on-board. On the other hand, as expected for the decentralized CGR, higher traffic loads demand larger amount of processing time in orbit, which proportionally impacts the satellite's power budget. In any case, the outcomes of the Walker and Lunar scenarios correlates with the computation times reported in Section IV-A. Finally, the random scenario results in Fig. 5 confirms that the energy efficiency benefits of the centralized (and filtered) approach grows with the topology size. However, non-filtered



CFE is out of scale, which confirms the increasingly high energy cost of transmitting complete route tables across larger SIN topologies. BFS, on the other hand, becomes computationally intractable beyond 9k contacts. These outcomes reassure that centralized routing is practically applicable in small and early SIN operations, being the distributed scheme the most flexible concept for future large-scale SINS.

## V. CONCLUSIONS AND OUTLOOK

Decentralized routing solutions for SINS based on Contact Graph Routing (CGR) enjoys accurate in-flight information of the current traffic, which enables a flexible on-demand computation of the exact routes required for data forwarding. However, we discussed that in-orbit autonomous processing might be considered an unnecessary burden in early SIN deployments, where a tighter control might seem preferable by the most conservative industries. Therefore, we embarked in a qualitative and quantitative comparison of centralized and decentralized routing approaches.

We initially introduced BFS and CFE, the first two practical centralized routing approaches for SINS able to cover both Immediate Forwarding (IF) and Store-Carry-and-Forward (SCF) traffic flows. The performance of decentralized CGR and centralized BFS and CFE was evaluated on realistic and random SINS of increasing fleet and contact topology sizes.

Results showed that the centralized CFE scheme is computationally more efficient when computing routes for SINS of moderate sizes, up to 5-10k daily contacts. Route table entries, however, might grow considerably in size. Thus, unless an adequate route filter can be applied based on the expected traffic knowledge, the practical provisioning of routes constitutes a challenge both in terms of bandwidth and of energy consumption of the space nodes. In any case, for large-scale SINS characterized by more than 10k daily contacts, decentralized schemes emerge as the most flexible and autonomous option.

Future efforts in this line include the assessment of other filtering criteria that could extend the applicability and scalability [4] of centralized routing schemes in larger SINS. Also, broader evaluations of traffic patterns with bulk data generation that thereby could provoke link and buffer congestion, are currently being looked at. Finally, the resilience of hybrid centralized/distributed SCF and IF-specific algorithms with respect to various failure models is an appealing topic on our research roadmap.

## ACKNOWLEDGMENTS

This research has received support from the ERC Advanced Grant 695614 (POWVER) and Proof-of-Concept Grant 966770 (LEOpowver), as well as by the DFG Grant 389792660, as part of TRR 248 (<https://perspicuous-computing.science>). Authors would like to acknowledge Scott C. Burleigh from the Jet Propulsion Laboratory, California Institute of Technology and Holger Hermanns from Saarland University for their careful revision of this article and the valuable exchanges on the subject.

## REFERENCES

- [1] J. R. Kopacz, R. Herschitz, and J. Roney, "Small satellites an overview and assessment," *Acta Astronautica*, vol. 170, pp. 93 – 105, 2020. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0094576520300540>
- [2] J. Alvarez and B. Walls, "Constellations, clusters, and communication technology: Expanding small satellite access to space," in *2016 IEEE Aerospace Conference*, 2016, pp. 1–11.
- [3] Consultative Committee for Space Data Systems (CCSDS), "Schedule-aware bundle routing (SABR) (blue book, recommended standard CCSDS 734.3-B-1," <https://public.ccsds.org/Pubs/734x3b1.pdf>, July 2019.
- [4] 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. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1084804520303489>
- [5] Y. Bi, G. Han, S. Xu, X. Wang, C. Lin, Z. Yu, and P. Sun, "Software defined space-terrestrial integrated networks: architecture, challenges, and solutions," *IEEE Network*, vol. 33, no. 1, pp. 22–28, 2019.
- [6] 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.
- [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] 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>
- [9] K. Scott and S. Burleigh, "Bundle protocol specification," Internet Requests for Comments, RFC Editor, RFC 5050, November 2007. [Online]. Available: <http://www.rfc-editor.org/rfc/rfc5050.txt>
- [10] S. Burleigh, "Interplanetary overlay network: An implementation of the dtm bundle protocol," 2007.
- [11] M. Feldmann and F. Walter, "μPCN A bundle protocol implementation for microcontrollers," in *2015 International Conference on Wireless Communications and Signal Processing (WCSP)*. IEEE, 2015, pp. 1–5.
- [12] J. Y. Yen, "Finding the  $K$  shortest loopless paths in a network," *Management Science*, vol. 17, no. 11, pp. 712–716, 1971.
- [13] S. Biswas, B. Alam, and M. Doja, "Generalization of dijkstra's algorithm for extraction of shortest paths in directed multigraphs," *Journal of Computer Science*, vol. 9, pp. 377–382, 01 2013.
- [14] J. A. Fraire, M. Feldmann, and S. C. Burleigh, "Benefits and challenges of cross-linked ring road satellite networks: A case study," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–7.
- [15] M. Feldmann, J. A. Fraire, and F. Walter, "Tracking lunar ring road communication," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–7.

**Juan A. Fraire** Juan is researcher at Inria, Lyon, France and CONICET, professor at Universidad Nacional de Córdoba and guest professor at Saarland University in Germany. He researches on space networking and supporting informatics techniques. Juan is the founder of the STINT workshop and has co-authored more than 55 papers as well as the "Delay-Tolerant Satellite Network" book.

**Elías L. Gasparini** obtained his computer scientist degree in 2020 at FAMAF, from the Córdoba National University in Argentina. Elías research interest revolves around algorithms for space networks, and is one of the lead developers of the DTNSim simulator. Currently, he performs as microelectronics for optical network design staff at Inphi in Córdoba, Argentina.