Segment Routing based on Geographic Checkpoints

Hartig, Valentin; Bosk, Marcin; Mohan, Nitinder; Mendes, Paulo

DOI
10.1145/3697253.3697268

Publication date
2024

Document Version
Final published version

Published in
LEONET 2024 - Proceedings of the 2nd ACM Workshop on LEO Networking and Communication, Part of

Citation (APA)
Hartig, V., Bosk, M., Mohan, N., & Mendes, P. (2024). Segment Routing based on Geographic Checkpoints. In *LEONET 2024 - Proceedings of the 2nd ACM Workshop on LEO Networking and Communication, Part of: ACM MobiCom 2024* (pp. 25-30). (LEONET 2024 - Proceedings of the 2nd ACM Workshop on LEO Networking and Communication, Part of: ACM MobiCom 2024). Association for Computing Machinery (ACM). https://doi.org/10.1145/3697253.3697268

Segment Routing based on Geographic Checkpoints

Valentin Hartig
Technical University of Munich
Munich, Germany

Marcin Bosk
Technical University of Munich
Munich, Germany

Nitinder Mohan
Delft University of Technology
Delft, Netherlands

Paulo Mendes
Airbus
Munich, Germany

ABSTRACT

Low Earth Orbit (LEO) satellite constellations have highly dynamic network topologies, making conventional routing protocols inefficient. This paper presents Geographic Checkpoint Routing (GCR), a routing protocol that combines Geographic Routing and Segment Routing (SR) principles. Utilizing the structure of Walker Delta constellations, GCR eliminates the reliance on network topologies. It routes traffic through predefined geographic segments, offloads route computation to network edges, and allows traffic engineering through customizable policies without modifying satellite infrastructure. Simulations using the Starlink constellation show that GCR can match the performance of traditional source-based routing protocols without depending on network topologies.

CCS CONCEPTS

• **Networks** → **Routing protocols**; **Network protocol design**; *Naming and addressing*.

KEYWORDS

Time-Varying Networks, Geographic Routing, Segment Routing, Protocol Design, LEO Constellations

ACM Reference Format:
Valentin Hartig, Marcin Bosk, Nitinder Mohan, and Paulo Mendes. 2024. Segment Routing based on Geographic Checkpoints. In *2nd International Workshop on LEO Networking and Communication (LEO-NET 24), November 18–22, 2024, Washington D.C., DC, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3697253.3697268

LEO-NET 24, November 18–22, 2024, Washington D.C., DC, USA

ACM ISBN 979-8-4007-1280-7/24/11.
https://doi.org/10.1145/3697253.3697268

1 INTRODUCTION

LEO satellite constellations establish a global broadband Internet infrastructure with high bandwidth and low latency. A prominent example is Starlink [12]. However, LEO constellations present unique challenges for their integration with the terrestrial Internet due to their time-varying topologies induced by intermittently available links and satellites. Such variability poses challenges to IP-based location techniques [2], transport protocols, and congestion control, which have difficulties dealing with the variable, although predictable, latency of LEO constellations [1].

To support communication in the presence of long delays and/or intermittent connectivity, Delay-Tolerant Networking (DTN) data planes using Time-Varying Graphs (TVGs) are an alternative to Internet transport services. DTN, realized in the Bundle Protocol, introduces an overlay network, allowing nodes to store data until a connection is available [4].

DTN Link State Routing (LSR) protocols like Contact Graph Routing (CGR) rely on globally distributed contact plans, represented in TVG with vertices for contact opportunities. The complexity of TVG grows with the number of contacts, creating scalability challenges for route computation [13]. To mitigate this, CGR uses clustering to reduce computational effort while maintaining a link-state approach for intra- and inter-domain routing [11]. However, the impact of cluster size and number remains unclear, and dynamic clustering might add overhead, particularly with topological changes. Precomputed routes can handle predictable changes but not unexpected ones, like satellite tilting, affecting link availability.

Recent proposals, such as Stable Hierarchical Orbital Routing Technique (SHORT) [9], aim to mask the dynamics of LEO constellations with a structured network. The approach organizes the network into hierarchical routing domains and uses domain-specific routing via orbital-geodetic coordinates. The advantage of SHORT is its reliance on geographic coordinates, reducing dependency on topological changes of LEO constellations. The approach also highlights the importance of decentralized routing, minimization of updates, and computation overhead reduction in a LEO satellite system.

Hence, we argue that sustainable routing for LEO constellations must be able to reduce resource usage on satellites while being topologically agnostic. The solution should allow the operation of LEO networks based on service requirements, such as avoiding some geographical areas or some network nodes, e.g., for security inspections when roaming over country borders.

With this in mind, we propose a Geographic Checkpoint Routing (GCR) protocol that utilizes the grid-like structure of Walker constellations to provide topology-agnostic routing while minimizing satellite resources. This is done by leveraging SR and Geographic Routing. Routes are computed at the edges of the LEO constellation based on geographic segments, which allows traffic to flow without being impacted by the intermittent nature of links and satellites. With GCR satellites forward packets based solely on the information carried on the packet header, without any need to compute and store routes. GCR closely approximates shortest path routing in hop count and latency while maintaining low processing delays at satellites. Unlike topological source routing, GCR does not rely on information about network topologies.

2 BACKGROUND AND RELATED WORK

This section first introduces two technology blocks relevant to developing routing protocols for LEO constellations: Segment routing and Geographic routing. Afterward, we analyze current proposals for packet routing on LEO constellations.

2.1 Segment and Geographic Routing

SR is an architecture for IPv6 and MPLS networks that enhances traffic engineering, scalability, and network control [3]. It follows a source routing approach, where senders encode a route within the packet header as a series of instructions called Segments. The versatility of Segments differentiates SR from other source routing paradigms, such as Loose Source and Record Route (LSRR) and Strict Source and Record Route (SSRR). The primary components of SR include segments, policies, and domains, which enable network operators to define and manage routing paths and traffic flows.

Geographic routing, used primarily in Mobile Ad-Hoc Networks, offers significant advantages over topological routing by reducing the need to maintain network state information. Routing is based on the geographic positions of nodes, determined by, e.g., GPS, assuming that the destinations of packets can be geographically pinpointed. Many different geographic routing protocols have been proposed, including GPSR [6] and WEAVE [8]. The advantages of geographic routing include scalability with the network size, maintaining constant time complexity ($O(1)$) as the network expands, and the elimination of control traffic.

Greedy geographic routing is limited by its dependence on network density and challenged by voids, such as coverage gaps and concave regions that disrupt routing paths. They lead to a "local maximum" problem, where a node may be closer to the destination than any of its neighbors, but is obstructed by a void [6]. In such cases, greedy algorithms can't establish a sensible route to the destination.

Many algorithms assume a static and planar network graph on which they employ perimeter routing to overcome voids. The technique employs heuristics, e.g., the right-hand rule, to ensure that packets are forwarded along the perimeter of the void until a valid path is found. Kim et al. [7] showed that generating such planar graphs poses substantial implementation and computational challenges.

2.2 Routing in LEO Networks

Li et al. [9] introduce SHORT, a geographic divide-and-conquer approach to abstract and decouple LEO constellation dynamics from the routing algorithm. The authors use the invariance of geographic locations as hierarchical addresses for multi-layer satellite network organization. SHORT splits the satellite system into three domains: terrestrial encompassing ground stations and terminals, orbital shells, and inter-shell links. The orbital shells are addressed based on an orbital-geodetic coordinate aligning geographic coordinates with satellite orbits. These are then used to define hierarchical geographic areas towards which packets are routed through the constellation. Switches between orbital shells always utilize a ground station. The authors showcase the functionality of SHORT using the StarryNet framework, showing that their approach achieves hop counts and propagation delays close to optimal flat routing approaches.

Ma et al. propose a geographic routing algorithm that divides the Earth into areas based on latitude, assigning a unique geographic ID to each ground node and satellite [10]. The algorithm determines the path and satellite for transmissions based on the geographic location of the nodes. However, the authors do not address the challenges faced by geographic routing protocols.

Roth et al. [16] propose a geographic routing algorithm incorporating geographic information in the link layer. Satellites and terminals use MAC addresses which include geographic data, and periodically generate next-hop switching tables. Based on these, packets are routed to geographic areas represented as grid cells, corresponding to the destination terminals using area identifiers. Updates are broadcast locally to ensure all nodes within an area are aware of terminal locations and movements. Nodes then greedily forward packets to the next hop based on the geographic information in the MAC address and their respective routing tables.

3 GEOGRAPHIC CHECKPOINT ROUTING

Geographic Checkpoint Routing (GCR) is designed for routing in structured Time-Varying Networks, as is the case of LEO satellite networks. It combines the benefits of Geographic Routing with SR to improve scalability and efficiency. Exploiting the benefits of Geographic Routing is possible since GCR takes advantage of the inherently grid-like and planar nature of Walker constellations (e.g. Starlink). Combining Geographic Routing with SR allows complex route computation and network management functions to be offloaded from core nodes to the edges of the network. The distinction between *Core Nodes* and *Edge Nodes*, illustrated in Figure 1, defines the network structure required for the implementation of GCR. We refer to a group of edge and core nodes, e.g. one satellite network, as a *GCR Domain*. Nodes outside this special SR domain are collectively called *External Nodes*. In the case of a LEO constellation, satellites operate as core nodes, ground stations/terminals operate as edge nodes and Internet devices act as External Nodes.

Following the SR terminology, Edge Nodes serve as ingress and egress routers for the GCR Domain. Ingress nodes are responsible for appending a list of *Geographic Segments* to incoming packets, which are routed through Core Nodes towards egress nodes. To perform this task, ingress nodes rely on a *Geographic Lookup Service (GLS)* to determine the approximate geographic area of the egress node near the packet destination. After identifying this area, the ingress node consults a Checkpoint Policy Service (CPS) to gather the appropriate *Checkpoint Policy* for the packet, determining the most efficient geographic route. Core Nodes manage data forwarding and network connectivity within the domain.

3.1 Geographic Lookup Service

The GLS provides ingress nodes with the approximate geographical locations of egress nodes given the address of the required service. The GLS utilizes edge nodes to establish a decentralized DHT. A possible implementation features a Content Addressable Network (CAN) as the underlying DHT protocol. The CAN allows efficient access to the required data, as it is designed to be scalable in large-scale networks, and increases fault tolerance.

3.2 Geographic Segments

GCR extends the standard SR architecture with a *Geographic Segment* type, embedding geographic information in packet headers. A *Geographic Segment* is a list of Checkpoints, realizing a policy used for routing. Checkpoints define areas consisting of a geodetic coordinate and a radius q_j, defining the maximum allowable distance a satellite can deviate from the Checkpoint center. A Segment within the GCR Header must contain one or more Checkpoints, the last being the Destination Checkpoint, containing the egress node. Besides Destination Checkpoints, GCR defines Transit Checkpoints, through which distinct routing policies are executed, such as routing around congested areas.

Transit Checkpoints are used to forward packets along specific geographic paths, and are part of geographic segments included in the packet header. Their optimal radius can be calculated based on the constellation's Right Ascension of the Ascending Node (RAAN) difference $\Delta\Omega = \frac{2\pi}{P} \in [0, 2\pi]$ and phase difference $\Delta\phi = \frac{2\pi}{Q} \in [0, 2\pi]$, with P orbital planes and Q satellites. The number of Checkpoints contained in a Segment depends on the chosen policy.

Destination Checkpoints designate a packet's final destination area and enable a localized Ground to Space Link (GSL) search within it. For this, the Checkpoint radius is calculated based on a minimum elevation angle ϵ of the destination Edge Node and the radius of the coverage area R_i of a satellite. Based on these parameters, we ensure that a connection with the destination Edge Node is always possible when the Checkpoint is reached.

To calculate the optimal radius for Destination Checkpoints, we assume that the antenna steering angle of a satellite is only restricted by the minimum elevation angle ϵ of the Edge Nodes [17], as shown in Figure 1. If a satellite is within the cone defined by ϵ, the cone of its antenna angle also contains the Edge Node it can connect to. Then, the optimal radius is $q_j \in [0, \frac{R_i}{2}]$. If $q_j = 0$, then the Checkpoint is reached if it's at the border of the satellite's coverage area. If $q_j = \frac{R_i}{2}$, then the satellite is located directly at the border of the Checkpoint, and its antenna still covers the entire Checkpoint. Therefore the Edge Node can be located anywhere inside the radius of the Checkpoint and still connect to the satellite. This defines an upper bound for the distance of a satellite to the Checkpoint given by $d_{i,j} = R_i - q_j$. The ingress node selects a q_j based on the egress node's position accuracy (e.g. stale GLS data) and its mobility.

To define the Checkpoint radius q_j, we must calculate the coverage area radius R_i of the antenna of the satellite, depicted in Figure 1 (right). Let's assume that the satellite is located exactly at angle ϵ above the true horizon of Checkpoint j. Then R_i equals the great circle distance between the coordinates of the Checkpoint and the nadir point of the satellite. Since the latter is unknown, R_i can be calculated via the central angle β between the nadir point of the satellite and the Edge Node, using the great circle distance, i.e., $R_i = R_e \cdot \beta$, with R_e being the earth's radius. β can be found by looking at the triangle defined by the Core Node, Edge Node, and earth's center point, as shown in Equation 1, with h_i being the altitude of the satellite's orbit.

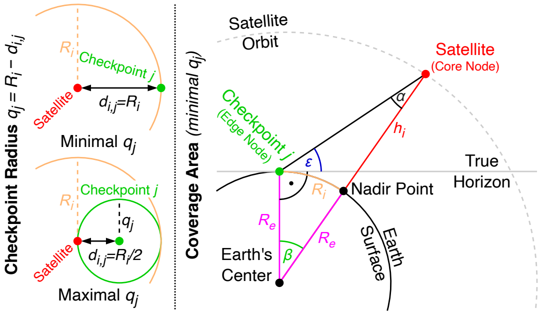

Figure 1: Checkpoint radius and satellite coverage.

$$\beta = 90^\circ - sin^{-1}\left(\frac{R_e}{R_e + h_i} \cdot sin\left(\epsilon + 90^\circ\right)\right) - \epsilon \tag{1}$$

3.3 Checkpoint Policy Service

The CPS generates Checkpoint Policies that are instantiated as geographic segments to guide packets through the Core Network. We implement two policies:

Shortest Geographic Path (SGP) Policy places Transit Checkpoints along the shortest great circle line between the ingress node and Destination Checkpoint. This path is calculated using Spherical Linear Interpolation (Slerp) to account for the Earth's curvature, and defines the number of Transit Checkpoints. The shortest geographical path doesn't always equate to the lowest hop count or latency [18].

Greedy Policy uses a single Destination Checkpoint for greedy geographic routing through the GCR Domain. It selects as the next hop the Core Node that minimizes the geographic distance to the destination. This decentralizing path calculation is considered as a fallback when other policies cannot be enforced, or Edge Node resources are limited.

The CPS can support other routing policies enabling traffic engineering and detailed path optimization. These policies can be based on various criteria, such as minimum hop count, latency, or congestion, and can include constraints like avoiding certain areas or optimizing for economic efficiency.

3.4 Packet Forwarding Mechanism

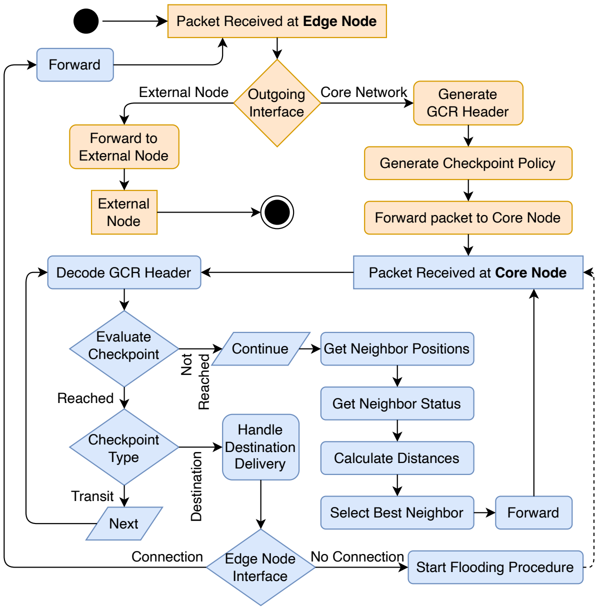

Figure 2: GCR packet forwarding decision tree.

GCRs forwarding mechanism can be split into two parts, one handling the packets at Edge Nodes, the other at Core Nodes, as indicated in Figure 2. Upon receiving a packet, Edge Nodes decide whether to route the packet via the GCR Domain, attaching corresponding geographic segments, or to another External Node. This decision is made autonomously by consulting the GLS and CPS or is influenced by preconfigured global SR policies. In the latter case, a service segment is attached to the incoming packet, signaling that the traffic should be sent over the core network.

Upon arrival at a Core Node, the packet handler retrieves the current geographic segment. It starts a *Checkpoint Evaluation Procedure* verifying if the packet has reached the Checkpoint indicated in the segment. For this, it calculates the great circle distance from the current satellite to the Checkpoint and compares this distance to the Checkpoint radius.

If the packet has arrived at the Checkpoint, the packet handler checks whether the Checkpoint is of type destination. If so, it starts the *Destination Delivery* procedure by checking for an interface to the Destination Edge Node. If found, the packet is forwarded to the Edge Node. Otherwise, the node initiates restricted *Checkpoint Flooding* to find a nearby satellite within the checkpoint that has a link to the destination Edge Node. The Core Node packet handler subsequently completes the current Checkpoint and evaluates the next one. If the Checkpoint was not reached, the *Core Network Forwarding Procedure* finds the optimal next hop by locating the neighbor with the shortest distance to the checkpoint. For this, the packet handler obtains the current positions of all neighboring nodes, excluding the sender of the current packet to avoid loops, and evaluates these for availability and congestion, prioritizing less congested nodes. It then calculates the great circle distance from each neighbor to the Checkpoint and selects the closest available neighbor with the best status.

4 EXPERIMENTAL EVALUATION

We implement GCR in OMNeT++ [14] based FLoRaSat simulator [5] to evaluate its performance using various checkpoint policies and constellation configurations. GSLs are managed by FLoRaSat's Topology Controller, ensuring Edge Nodes always connect to all Core Nodes (satellites) within their minimum elevation angle.

Simulations use the second orbital shell of Starlink, consisting of 72 orbital planes, each with 22 satellites, at an inclination of 53.2° and an altitude of 540 km. The mobility orchestrator updates satellite positions every 100 milliseconds, while the topology controller adjusts satellite links at the same frequency. The satellite constellation configuration repeats every 5764 seconds, corresponding to its orbital period. Thus, the simulation time was set to this duration to cover all possible movement patterns. The epoch for the constellation was set to the first day of 2020.

We evaluated the impact of topology changes on two GCR routing configurations: GCR with a Greedy Checkpoint Policy (*Greedy*) and GCR with a SGP Policy (*Trace*), using Topological Shortest Path Source Routing (*Topologic*) as a benchmark. Each scenario involves a northwest-to-southeast route between Germany and Australia using descending orbits, largely considered as a worst-case in LEO routing. Other scenarios, like those using ascending orbits, show similar performance and are omitted for brevity.

The effectiveness of the three routing methods is evaluated against the number of traversed hops. Figure 3 displays a moving average for all successfully delivered packets with a window size of 10,000 packets. No packets were lost due to routing loops or errors, ensuring the reliability of this metric. The paths taken by the three different routing methods are visualized in Figure 4 and Table 1 shows the corresponding average packet delays over different routes. In an optimal scenario using Topological Source Routing, a packet traverses an average of 12 to 13 hops to reach its destination, including the ground station. However, the source path must be recalculated with each topology change, which is not reflected in the RTT measurements of our simulation. This makes the method highly inefficient in real-world scenarios. GCR Greedy Checkpoint Policy typically results in a higher average of 14 to 15 hops due to the policy's tendency to overshoot the Checkpoint and require backtracking to the final destination. On the other hand, GCR SGP Policy achieves a lower average of 13 to 14 hops compared to the Greedy Checkpoint Policy by enforcing the shortest geographic path between Transit Checkpoints. The policy achieves a similar average delay to the best-case Topologic scenario without requiring a topological understanding at each ingress node and outperforms the Greedy method.

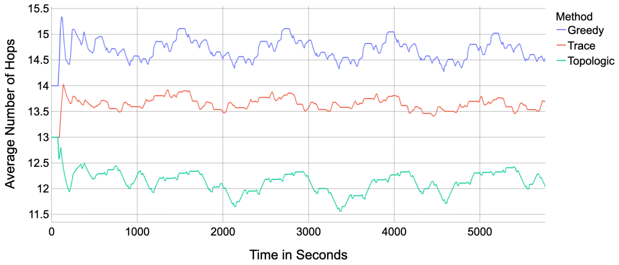

Figure 3: Moving average of hops over simulation time for various calculated Germany-Australia routes.

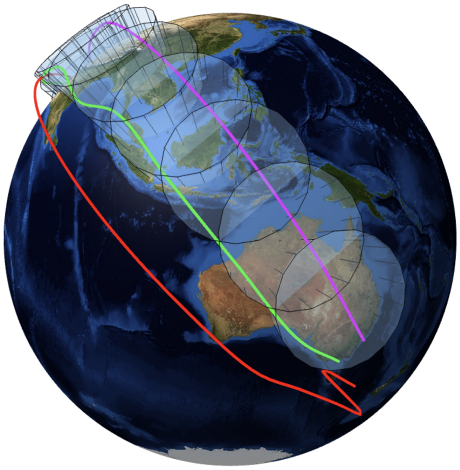

Figure 4: Route calculated by *Topologic* (purple), *Greedy* (red) and *Trace* (green) from Germany to Australia. [1]

Routing Policy	Propagation Delay (ms)	Processing Delay (ms)	RTT (ms)
Topologic	61.616	12.85	148.932
Greedy	78.195	14.23	184.85
Trace	62.77	12.60	150.74

Table 1: Average delays for tested routing methods.

5 DISCUSSION

Unlike Link State routing protocols, GCR is independent of topology changes, minimizing overhead. Extending SR with geographic segments allows GCR to discover alternative routes during link failures or congestion. The performance of GCR is influenced by the Checkpoint Policy, constellation density, and configuration, improving with the network scale. This enables adaptable and efficient traffic engineering without major changes to the routing algorithm.

[1] The texture of the globe was created from public domain material: https://commons.wikimedia.org/wiki/File:BlueMarble_monthlies_SMIL.svg.

The performance of GCR shows that it closely approximates shortest path routing in hop count and latency, especially with the SGP Policy. GCR maintains low processing delays at satellite nodes, while slightly increasing route distance, with an average Round Trip Time (RTT) of 150.74 milliseconds, compared to 283 milliseconds over fiber from Germany to Australia [15]. GCR effectiveness can be hindered by poorly placed Checkpoints, particularly in polar regions, highlighting the need for optimized policies that account for this.

6 CONCLUSION AND FUTURE WORK

Routing protocols for satellite constellations face challenges in complexity and scalability, which we address with GCR, a protocol that integrates SR with Geographic Routing. GCR scales efficiently with network size while increasing management flexibility and reducing effort in satellites. Simulation benchmarks with a Walker Delta constellation (Starlink) show GCR can match topological route performance, though it depends on Checkpoint Policy configuration and satellite constellation parameters. Future research will focus on optimizing Checkpoint Policies and implementing efficient GLS and CPS designs for real-time optimization.

ACKNOWLEDGMENTS

This work was partially supported by the German Federal Ministry of Education and Research joint project 6G-life (16KISK002) and National Growth Fund through the Dutch 6G flagship project “Future Network Services”.

REFERENCES

[1] Debopam Bhattacherjee, Waqar Aqeel, Ilker Nadi Bozkurt, Anthony Aguirre, Balakrishnan Chandrasekaran, P. Brighten Godfrey, Gregory Laughlin, Bruce Maggs, and Ankit Singla. 2018. Gearing up for the 21st century space race. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks* (HotNets ’18). Association for Computing Machinery, Redmond, WA, USA, 113–119. ISBN: 9781450361200. DOI: 10.1145/3286062.3286079.

[2] Tasneem Darwish, Gunes Karabulut Kurt, Halim Yanikomeroglu, Guillaume Lamontagne, and Michel Bellemare. 2022. Location management in internet protocol-based future leo satellite networks: a review. *IEEE Open Journal of the Communications Society*, 3, 1035–1062. DOI: 10.1109/ojcoms.2022.3185097.

[3] Clarence Filsfils, Stefano Previdi, Les Ginsberg, Bruno Decraene, Stephane Litkowski, and Rob Shakir. 2018. Segment Routing Architecture. RFC 8402. (July 2018). DOI: 10.17487/RFC8402.

[4] Juan A. Fraire, Olivier De Jonckère, and Scott C. Burleigh. 2021. Routing in the space internet: a contact graph routing tutorial. *Journal of Network and Computer Applications*, 174, 102884. DOI: 10.1016/j.jnca.2020.102884.

[5] Juan A. Fraire, Pablo Madoery, Mehdi Ait Mesbah, Oana Iova, and Fabrice Valois. 2022. Simulating lora-based direct-to-satellite iot networks with florasat. In *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 464–470. DOI: 10.1109/WoWMoM54355.2022.00072.

[6] Brad Karp and H. T. Kung. 2000. Gpsr: greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking* (MobiCom ’00). Association for Computing Machinery, Boston, Massachusetts, USA, 243–254. ISBN: 1581131976. DOI: 10.1145/345910.345953.

[7] Young-Jin Kim, Ramesh Govindan, Brad Karp, and Scott Shenker. 2005. Geographic routing made practical. In *2nd Symposium on Networked Systems Design & Implementation (NSDI 05)*. USENIX Association, Boston, MA, (May 2005). https://www.usenix.org/conference/nsdi-05/geographic-routing-made-practical.

[8] Michał Król, Eryk Schiller, Franck Rousseau, and Andrzej Duda. 2016. Weave: efficient geographical routing in large-scale networks. In *2016 International Conference on Embedded Wireless Systems and Networks (EWSN)*. Junction Publishing, Toronto, Canada, (Feb. 2016), 89–100. ISBN: 978-0-9949886-0-7. DOI: 10.5167/uzh-175020.

[9] Yuanjie Li et al. 2024. Stable hierarchical routing for operational leo networks. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking* (ACM MobiCom ’24). Association for Computing Machinery, Washington D.C., DC, USA, 296–311. ISBN: 9798400704895. DOI: 10.1145/3636534.3649362.

[10] Yanpeng Ma, Jinshu Su, Chunqing Wu, Jinshu Su, Xiaofeng Wang, Wanrong Yu, Baokang Zhao, and Xiaofeng Hu. 2012. A distribute and geographic information based routing algorithm for LEO satellite constellation networks. In *2012 Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*. IEEE, Palermo, Italy, (July 2012), 433–438. ISBN: 978-1-4673-1328-5. DOI: 10.1109/IMIS.2012.74.

[11] Pablo G. Madoery, Juan A. Fraire, Fernando D. Raverta, Jorge Finochietto, and Scott C. Burleigh. 2018. Managing routing scalability in space dtns. In *International Conference on Wireless for Space and Extreme Environments (WiSEE)*. IEEE, Huntsville, USA, (Dec. 2018).

[12] Nitinder Mohan, Andrew E. Ferguson, Hendrik Cech, Rohan Bose, Prakita Rayyan Renatin, Mahesh K. Marina, and Jörg Ott. 2024. A multifaceted look at starlink performance. In *Proceedings of the ACM Web Conference 2024* (WWW ’24). Association for Computing Machinery, Singapore, Singapore, 2723–2734. ISBN: 9798400701719. DOI: 10.1145/3589334.3645328.

[13] Michael Moy et al. 2023. Contact multigraph routing: overview and implementation. In *2023 IEEE Aerospace Conference*, 1–9. DOI: 10.1109/AERO55745.2023.10115729.

[14] 2024. OMNeT++ discrete event simulator. (July 30, 2024). Retrieved July 30, 2024 from https://omnetpp.org/.

[15] 2024. Ping time between munich and sydney. WonderNetwork. (May 11, 2024). Retrieved May 11, 2024 from https://wondernetwork.com/pings/Munich/Sydney.

[16] Manuel Roth, Hartmut Brandt, and Hermann Bischl. 2021. Implementation of a geographical routing scheme for low earth orbiting satellite constellations using intersatellite links. *International Journal of Satellite Communications and Networking*, 39, 1, 92–107. DOI: 10.1002/sat.1361.

[17] LLC Space Exploration Holdings. 2024. Application for fixed satellite service by space exploration holdings, LLC [SAT-MOD-20200417-00037]. Retrieved Mar. 12, 2024 from https://fcc.report/IBFS/SAT-MOD-20200417-00037.

[18] Gregory Stock, Juan A. Fraire, and Holger Hermanns. 2022. Distributed on-demand routing for leo mega-constellations: a starlink case study. In *2022 11th Advanced Satellite Multimedia Systems Conference and the 17th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, 1–8. DOI: 10.1109/ASMS/SPSC55670.2022.9914716.

Received ; revised ; accepted