Paving the Way for Link-State Routing in Satellite Networks with Fast and Seamless Ground Station Handovers

NICOLAS RYBOWSKI, ICTEAM, UCLouvain, Belgium
TOM ROUSSEAUX, ICTEAM, UCLouvain, Belgium
MAXIME PIRAUX, Aerospacelab, Belgium
CRISTEL PELSSER, ICTEAM, UCLouvain, Belgium
OLIVIER BONAVENTURE, ICTEAM, UCLouvain, Belgium and WEL Research Institute, Belgium

Satellite-based Internet access has mainly offered a bent-pipe model in which satellites relay packets between user terminals and ground stations strictly within reach. New constellations are now leveraging optical inter-satellite links to turn the constellation itself into a network by connecting satellites to their neighbours and providing Internet access through a set of ground stations reachable from the whole constellation. Operators deploying these networks aim to reuse existing IP technologies as much as possible. Satellite constellations are characterised by frequent connectivity changes between ground stations and satellites. Existing IP link-state routing protocols are known to trigger transient forwarding instabilities upon topological changes.

In this paper, we explore how link-state routing protocols like IS-IS and OSPF can operate in such constellations. We discuss the convergence of link-state routing protocols during connectivity changes between ground stations and satellites, and show that they can cause transient loops. Given the frequency of these changes, they would affect network availability. We propose an efficient method that leverages multicast trees to perform such topology changes in a seamless manner and prove the correctness of our approach.

CCS Concepts: • **Networks** → **Network protocol design**; **Routing protocols**; *Network simulations*; **Network dynamics**; **Network mobility**.

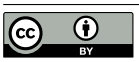
Additional Key Words and Phrases: OSPF, IS-IS, Network Stability, Satellite Constellation, Ground Station Handover

ACM Reference Format:
Nicolas Rybowski, Tom Rousseaux, Maxime Piraux, Cristel Pelsser, and Olivier Bonaventure. 2026. Paving the Way for Link-State Routing in Satellite Networks with Fast and Seamless Ground Station Handovers. *Proc. ACM Netw.* 4, CoNEXT4, Article 58 (December 2026), 26 pages. https://doi.org/10.1145/3856972

1 Introduction

While satellite Internet access has existed since the early 2000s, advances in the space industry over the last decade have turned this access network from an experimental technology into an infrastructure competing with terrestrial networks. Operators, therefore, entered a race for low latency. While communications satellites traditionally use high-altitude, often geostationary, orbits [41, 62], the propagation delay makes their use impractical given the importance of latency for Internet services [32]. Current and future constellations have significantly lower altitudes (Low Earth Orbit or LEO) and thus lower propagation delay. Covering the Earth from such low altitudes, however,

Authors' Contact Information: Nicolas Rybowski, nicolas.rybowski@uclouvain.be, ICTEAM, UCLouvain, Louvain-la-Neuve, Belgium; Tom Rousseaux, tom.rousseaux@uclouvain.be, ICTEAM, UCLouvain, Louvain-la-Neuve, Belgium; Maxime Piraux, maxime.piraux@aerospacelab.com, Aerospacelab, Louvain-la-Neuve, Belgium; Cristel Pelsser, cristel.pelsser@uclouvain.be, ICTEAM, UCLouvain, Louvain-la-Neuve, Belgium; Olivier Bonaventure, olivier.bonaventure@uclouvain.be, ICTEAM, UCLouvain, Louvain-la-Neuve, Belgium and WEL Research Institute, Wavre, Belgium.



This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2026 Copyright held by the owner/author(s).
ACM 2834-5509/2026/12-ART58
https://doi.org/10.1145/3856972

requires hundreds to thousands of satellites, each completing an orbit in around 90 minutes. As a result, the entire network is in constant motion w.r.t. the ground.

Early deployments of Internet access constellations, such as Eutelsat's OneWeb [4, 84] or SES's O3b [3], use a *bent-pipe* model, in which satellites relay traffic between user terminals and ground stations within their direct reach. In this type of architecture, demodulation may not even be required: the RF signal may be amplified and repeated to the ground. Such constellations are thus limited to serving as access networks. More recently, advances in free-space optical communication have enabled satellites to carry optical communication terminals to establish inter-satellite links (ISLs). This also changed the network topology of these constellations: the mesh formed by the ISLs can now function as a backbone network, as is the case for Starlink [53]. We argue that this is a fundamental change, bringing these networks into the scope of routing research. The community has not yet reached consensus on how routing is actually implemented within deployed constellations such as Starlink [39, 53, 56, 72]. Starlink applies a synchronised 15-second reconfiguration interval across the whole constellation [53], suggesting the use of a centralised control plane. In RFC 9717, an IETF participant posits that a reasonable architecture involves a link-state routing protocol synchronising the network topology state across satellites so that each can establish its forwarding policies [43]. During discussions with industrial stakeholders from the sector, we confirmed that multihop forwarding is expected to be used in future satellite constellations.

However, such a backbone differs from terrestrial ones in a crucial way: its topology changes constantly, by design. In terrestrial backbones, topological changes are dominated by unplanned failures and remain rare events. Measurements on the Sprint network report times between failures reaching up to 1.5 hours for 90% of them [38, 47]. In a constellation, satellites continuously enter and leave the sight cones of the ground stations connecting the constellation to the Internet, establishing *feeder links* that typically last only minutes [84]. A ground station therefore performs frequent *handovers*, i.e., disconnecting from a leaving satellite to reconnect to another one. We show in Section 3 that, depending on the constellation, a single ground station experiences from tens up to hundreds of such handovers per hour. Each one, consisting of two topological modifications, a link removal and a link addition, must be advertised to every satellite to adapt packet forwarding.

Link-state routing protocols distribute topological modifications by *flooding* [24, 54], which provides no ordering guarantee. Depending on the order in which routers update their forwarding tables, packets may transiently loop between two routers or be dropped, even for planned changes [25]. Known countermeasures order the updates of routers' forwarding tables (FIB) atop the flooding process [25, 66] or split a change into a sequence of loop-free metric updates [11, 12]. We demonstrate that these techniques are a poor fit for constellations as they rely on conservative timers, scale poorly with the topology size, and handle only one type of change per convergence, such that a single handover requires two distinct convergences. At hundreds of handovers per hour, a constellation operated this way would spend a significant share of its time converging, and every additional second of convergence increases its exposure to concurrent topological failures. To the best of our knowledge, the current literature on routing in satellite networks neither explores nor adapts link-state routing protocols to ensure seamless topological changes.

In this paper, we consider how link-state routing protocols should evolve to support such dynamic topologies, with a focus on next-generation satellite constellations. Our contributions are threefold. First, we analyse the limitations of state-of-the-art FIB ordering mechanisms when applied to ground station handovers, and derive the constraints that a FIB synchronisation mechanism should respect (Section 4). Second, we propose a synchronisation method that nests both topological modifications of a handover in a single convergence. Our method distributes nested updates along a multicast tree, which implicitly embeds a loop-free FIB update ordering without relying on timers. We formally prove that this method suppresses any transient forwarding event (Section 5). Third, we evaluate

our method by simulating more than ten thousand handovers on four representative constellations, showing that it converges two to three times faster than the state-of-the-art sequential ordering while reducing the number of convergences by up to half (Section 6). The artefacts of this paper are publicly available[1]. The solution we propose focuses only on seamlessly resynchronising the replicated databases that the link-state-based routing protocols constitute, i.e., advertising topological changes without transiently breaking the IP forwarding. It does not aim at optimising route computation in satellite constellations.

This paper is organised as follows. Section 2 provides background on the structure of satellite constellations. Section 3 motivates our work by justifying the use of IGPs in satellite constellations, then by quantifying the dynamicity of ground station handovers and explaining its impact on packet forwarding. Sections 4, 5 and 6 present our contributions as outlined above. Section 7 reviews the literature related to routing in satellite constellations, as well as techniques to avoid transient forwarding events upon topology reconfiguration with IGPs. Finally, Section 8 concludes this paper and discusses future research directions.

2 Background

Satellite constellations are composed of a set of orbital planes, each containing several satellites. Planes sharing the same altitude, inclination and satellite count, but differing in their angle to the equator plane, form a *shell*. In this paper, we consider single-shell constellations following the Walker pattern, which is the prevalent pattern in existing constellations [60]. There exist two variants of the Walker pattern, Delta (resp. Star), where the longitude of the ascending node of orbital planes, i.e., the intercept between the orbit and the equator plane, is distributed over 360 (resp. 180) degrees around the equator plane [76]. Each pattern enables different network topologies. We consider that satellites establish up to four ISLs with their immediate neighbours to form a +Grid [8] connectivity pattern. Figure 1 depicts such topologies for a simple constellation of 9 satellites distributed over 3 orbits. The Walker Star pattern is mainly composed of in-plane links, i.e., between satellites within the same orbit, each orbit forming a ring. Cross-plane links, i.e., between satellites of different orbits, may also exist, though those between the last and first orbit are less feasible as ascending satellites from one orbit face descending satellites from the other [60], leaving an open *seam*. The Walker Delta pattern does not suffer from this seam effect. It corresponds to the Star topology augmented with the dashed grey links in Figure 1, which connect the first and last orbits. The topology formed by applying the +Grid principle may then be represented as a torus. Considering that the length of an ISL bounds its one-way delay, in-plane ISLs have a stable latency since endpoints travel around the same orbit. For cross-plane ISLs, their latency oscillates between a minimum and a maximum as the two orbits cross and then move away from each other.

To enable their service, satellite constellations connect to terrestrial networks through *ground stations* distributed around the Earth. A satellite and a ground station then form what is called a *feeder link*. We posit that such links have asymmetric IGP metrics to prevent the ground station from forwarding traffic from the constellation back to it. We justify that hypothesis because feeder-link modems[2] are more constrained in terms of bandwidth than next-generation ISLs[3]. Ground stations co-located with a terrestrial point of presence act as *gateways* between the constellation and the Internet.

[1]The artefacts of the paper are available at https://doi.org/10.5281/zenodo.23243267.
[2]https://www.enensys.com/products/wideband-dvb-s2x-modem-ip
[3]https://www.satellitetoday.com/government-military/2025/06/05/mynaric-reports-condor-mk3-delivery-milestone-progress-on-mk3-1-terminal-for-sda

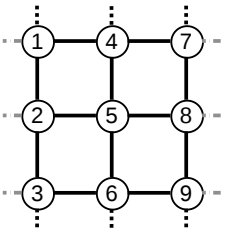
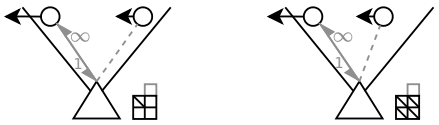


Fig. 1. Illustration of the Walker patterns for a simple constellation of 9 satellites distributed across 3 vertical orbits. A Walker Delta topology exhibits all the links present in the figure, while the dashed grey links connecting the left- and right-most orbits are absent in Walker Star.



(a) Simultaneous (b) At removal

Fig. 2. Types of handovers. The triangle (resp. circles) represents a ground station and its sight cone (resp. satellites). The arrows show the satellites' displacement direction. The available antennas on the ground station are shown by a grid of 4 slots, the 5^{th}, grey one being dedicated to handovers. The solid (resp. dashed) grey line illustrates an active (resp. candidate) connection. The IGP metrics are shown in grey.

As shown in Figure 2, each ground station has a sight cone determined by its minimum elevation angle (relative to the horizon), defining where feeder links are feasible. While User Terminals (UTs), such as Starlink antennas, usually support a single satellite connection [2], ground stations can support 10 or more feeder links [4]. We consider that satellites carry sufficient hardware to establish 4 simultaneous feeder links [4, 8], potentially to different ground stations. As satellites enter and leave their sight cone, a feeder link can be established and disconnected. We refer to these events as *connectivity events*. We define a *handover* as a ground station experiencing two opposite connectivity events, e.g., disconnecting from a satellite to reconnect with another satellite. Two kinds of handovers occur naturally. First, as shown in Figure 2a, a ground station performs a handover when a satellite enters the sight cone of a ground station while another leaves it. Second, three conditions are necessary: (*i*) a ground station is saturated, i.e., each radio has established a feeder link, (*ii*) a candidate satellite moving through the sight cone, and (*iii*) a satellite with an active feeder link leaves the sight cone. In such a situation, depicted in Figure 2b, a ground station may directly reconnect to a candidate satellite. Note that the 5 radio slots of Figure 2 are for illustration purposes only. In the remainder of this paper, we consider that each ground station leverages 10 radios (resp. 25 for Starlink), one of which is dedicated to performing handover in a make-before-break manner, which limits the capacity of ground stations to 9 (resp. 24) feeder links towards the constellation.

Stateless Multicast with BIER. Classical multicast protocols such as PIM-SM [23] explicitly build dedicated trees for each multicast group. Since the trees' state is distributed on each router, such solutions scale poorly with the size of the topology and the number of groups [18]. Bit Index Explicit Replication (BIER) [81] solves this scalability issue by embedding a condensed version of the tree, called a *bitstring*, directly in BIER packets that encapsulate the actual data-plane packets. Each bit in the bitstring, in the Least Significant Bit (LSB) order, represents a router from the topology. Setting a bit to one means that the router must decapsulate the BIER packet. The forwarding procedure of BIER relies on the BIER Forwarding Information Table (BIFT), which has the same role as the FIB for IP forwarding. It contains, for each BIER-enabled router in the topology, a forwarding bitmask (FBM) that represents the set of routers that are reachable through that router in terms of shortest path. A detailed example of the BIER forwarding process is provided in Appendix A.

While there is no per-tree state, routers still have to build the BIFT. Similarly to the RIB/FIB build for IP forwarding, BIER leverages IGPs to distribute connectivity information to build its BIER Routing Information Table (BIRT) [30, 61], the equivalent of the IP RIB. IGPs derive the FBM for each node based on the results of the SPT computation and the BIRT content, and synchronise the BIFT simultaneously with the FIB. Hence, computing the BIER state does not introduce much

overhead. Vendors support BIER forwarding in their commercial routers[4]. While BIER is not reliable by design, the literature proposes some solutions to enable reliability at the flow-level [16, 17] and resilience against link and node failures [9, 51]. Regarding the scalability of BIER, a standard bitstring may address up to 4096 nodes [80]. It is possible to further expand BIER scalability [52, 81]. However, since IGPs are known to scale poorly on very large topologies [14, 44], we argue that the scalability of BIER with its basic bitstring will not be an issue in this work.

3 Motivation

One of the original approaches to routing in satellite constellations, *DT-DVTR* by Werner *et al.* [77–79], leverages periodical snapshots of the topology on which the forwarding paths are pre-computed off-line. Such forwarding snapshots are uploaded to the satellites, then applied according to a planned schedule thanks to a synchronised clock throughout the constellation. Disjoint backup paths are also included to react in case of the failure of a primary path. Another original approach, proposed by Ekici *et al.* [19], suggests the use of static *logical locations* filled by the nearby satellites. A so-called *decision map*, containing minimum-hop and minimum-delay paths between logical locations, is computed off-line and then loaded on the satellites. They apply local forwarding decisions mapped to the pre-computed paths and their current relative position to the nearest logical location. These forwarding rules include congested link avoidance and rerouting around failed satellites.

We assume that the next-generation constellations will behave as backbone networks and require more flexibility than statically defined schedules and forwarding paths. Recent research confirms this intuition by proposing SDN-based path computation [22, 64]. However, updating SDN rules at scale while preserving forwarding policies is known to be challenging [74]. IGPs, on their side, are replicated databases focusing on reliably synchronising routing information such as IP prefixes, segment routing (SR) and multicast information, e.g., for BIER. They provide shortest paths within the physical topology and enable virtual topologies with multi-topology extensions or source-routed paths. IGPs also enable fast-reroute techniques (FRR) such as TI-LFA [6] to protect forwarding paths from link and node failures. Recent literature proposes OSPF extensions to leverage the predictability of satellite orbits [57] to compute optimized paths and reduce the number of convergences. Vissicchio *et al.* [75] even propose to leverage IGPs to reliably reprogram the forwarding tables based on data provided by SDN controllers.

While there is an increasing interest in the use of IGPs in satellite constellations, none of the prior works includes the feeder links provided by ground stations in their path computations, except for a study that proposes algorithms to select handovers to apply [83], but that does not analyse the impact of such handovers at the protocol level. Even if we introduce ground stations and feeder links into pre-computed schedules, the lack of dynamic information on the topology may trigger paths including failed ground stations or feeder links. We argue that leveraging IGPs to propagate the feeder link connectivity in the topology ensures correct egress node selection, while bringing the benefits mentioned above.

This section motivates our work by first quantifying how frequently ground station handovers occur in representative constellations, and then by explaining why advertising each of these topological changes with a *classical* routing protocol endangers packet forwarding.

[4]Juniper https://community.juniper.net/blogs/jeffrey-zhang/2024/04/26/bier-mvpn-in-ptx-express-5 and Nokia https://infocenter.nokia.com/public/7750SR225R1A/index.jsp?topic=%2Fcom.nokia.Multicast_Guide%2Fbier_forwarding-ai9erp9kqu.html document the configuration of BIER they support on their products.

3.1 Feeder links dynamicity

We consider four different satellite constellations in this work. IRIS2 is an upcoming high-LEO constellation with the lowest satellite count considered in this paper. It lies at an altitude of 1200 km and contains 264 satellites [34]. SCN-288 is a Walker Star constellation evaluated in prior work on constellation routing [64]. It contains 288 satellites and lies at an altitude of 780 km. *LEO* is a synthetic constellation designed as an intermediate Walker Delta constellation within this work.

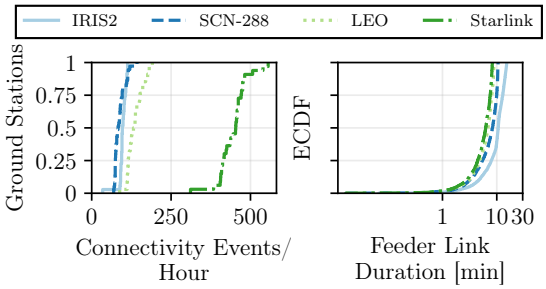


Fig. 3. Constellations have very dynamic connectivity through feeder links.

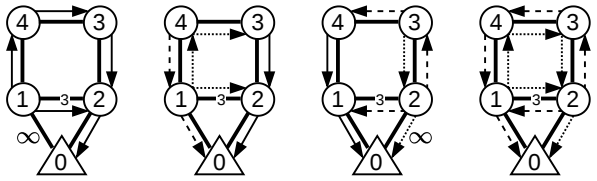
It is located at an altitude of 600 km and has 432 satellites. Finally, we consider one 1584-satellite Starlink shell [60, 70] at an altitude of 550 km. To characterise the dynamicity of handovers for these constellations, we leverage a public dataset [84] referencing 40 ground stations of the OneWeb constellation. We select the ones collocated with points-of-presence (PoPs) that provide the ground stations with a direct connection to the Internet, reducing the set to 35 ground stations. Figure 3 provides insights on the dynamicity of the feeder links. It depicts, for each constellation, the cumulated hourly connectivity event rate per ground station and the ECDF of feeder link duration. First, we see that every ground station is subject to numerous connectivity events each hour, especially for the large Starlink shell. Second, we observe that the connection durations are quite short; many last less than ten minutes, while the longest last for fifteen minutes.

3.2 Routing in satellite constellations

From a routing protocol perspective, a connectivity event is a topological modification that must be advertised to the whole network to adapt its packet forwarding. Until every satellite has received the modification and updated its forwarding state accordingly in a process called *convergence*, satellites forward packets based on inconsistent views of the topology. During this transient period, packets may be sent towards a feeder link that no longer exists and be dropped, or bounce between two satellites that each consider the other as their nexthop towards the ground station [10, 25]. Such transient loops delay and reorder packets, waste ISL capacity, and turn into packet drops once their TTL expires. Synchronising the forwarding tables in a network topology with a convergence is therefore a critical operation that takes time [11, 12, 15, 66]. The longer a convergence lasts, or the more frequent the convergences are, the more the system is at risk of an unplanned topological failure, e.g., a failing ISL, occurring while one is in progress. In turn, resolving these failures requires additional convergences, which further prolong the forwarding disruptions. Once these failures have been propagated, distributed routing takes them into account within future convergences by design, bringing more resilience to the routing.

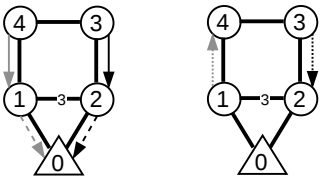
4 Link State Routing for Satellite Constellations

In terrestrial ISPs, topological changes are mainly due to failures and maintenance [50]. In satellite constellations, as shown in Figure 3, there is a large number of **planned** modifications (feeder links that come and go), which we aim to propagate with link-state routing while (*i*) ensuring forwarding stability, (*ii*) reducing exposure to failures through fewer and faster convergences, and (*iii*) scaling with topology size. In this section, we first present how flooding works in link-state-based IGPs (Section 4.1). Then, we illustrate in Section 4.2 how a single topological modification triggers



(a) Initial state (G_1) (b) Addition of $(0,1)$ (G_2) (c) Removal of $(0,2)$ (G_3) (d) Path changes summary

Fig. 4. A simple topology illustrating transient loops triggered by topological modifications. Link metrics are unitary unless specified otherwise. Solid arrows represent paths from each node to Node 0. Dashed (resp. dotted) arrows represent added (resp. removed) paths w.r.t. the previous step.



(a) $rSPDAG_{G_2}(0)$ (b) FIB Orderings

Fig. 5. Applying the oFIB framework. $rSPDAG_{2\to0}(2)$ (resp. $rSPDAG_{1\to0}(1)$) is depicted with solid black (resp. grey) arrows. Dotted arrows indicate the ordering relationship.

transient instabilities. Finally, in Section 4.3, we show existing methods to avoid transient events and their limitations.

4.1 Distributing Topological Modifications with Flooding

Link-state Interior Gateway Protocols (IGPs) such as OSPF [24, 54] and IS-IS [1] distribute each router's local topology view as Link State Advertisements (LSAs), building a replicated Link State DataBase (LSDB). Each router advertises its identifier, loopback address, and the administrative *metric* of each of its links. From its LSDB, a router typically runs Dijkstra's shortest path algorithm to compute shortest paths towards each router and stores the resulting nexthop(s) in its Forwarding Information Base (FIB). Modern routers use Equal-Cost Multi-Path (ECMP) [36, 37], and can thus store multiple nexthops for a given destination. Any topological modification triggers resynchronisation of every LSDB, a process called *convergence*.

Constraint 1. *Every node in the topology **must** eventually receive any topological modification(s), even if its FIB remains unchanged.*

LSDB inconsistencies, e.g., keeping an old LSA that contains a link which does not exist anymore, cause routers to compute non-existent paths, leading to packet loss. IGPs therefore use the flooding method to keep LSDBs synchronised, i.e., upon receiving an LSA, a router forwards it to all neighbours except the sender. This method ensures that every router delivers the new LSA. The unordered nature of flooding may trigger transient forwarding instabilities [25], but eventually converges to a loop-free state.

4.2 Routing Loops within a Toy Example

Figure 4 illustrates a simple topology that experiences loops due to the addition of link $(0,1)$ followed by the removal of link $(0,2)$. It may represent an abstract constellation of four satellites and one ground station (Node 0). The topological modifications consist of the handover of the ground station from satellite 2 to satellite 1. We consider the satellites directly connected to a ground station as *border satellites* of the constellation. Link metrics are unitary and symmetric unless specified otherwise. The metric of 3 between satellites 1 and 2 is chosen solely to illustrate the loops. Feeder links are the only asymmetric ones. Their inbound metrics are large enough to prevent traffic from bouncing back to the constellation through them. Figure 4a shows that, initially, each node selects Node 2 as the sole border satellite to reach the ground station.

Adding or removing a feeder link. After the addition of link $(0, 1)$, several nodes update their FIB in reaction to the topological modification, as illustrated by Figure 4b. Node 1 replaces its nexthops, Nodes 2 and 4, with a cost of four with Node 0 (unitary cost). Node 4 also replaces Node 3 with Node 1 as the nexthop to reach the ground station, and decrements the cost from three to two. If Node 4 updates its FIB before Node 1, a transient loop appears between them. It lasts until Node 1 updates its FIB. In a larger topology, any node using Node 4 in one of its paths to reach the ground station is also impacted by the loop.

Similarly, in Figure 4c, nodes update their FIB in reaction to the removal of link $(0, 2)$. Node 2 removes Node 0 as its nexthop to reach Node 0, and adds Nodes 1 and 3 as its new nexthops with a cost of four. Node 3 replaces Node 2, its old nexthop with a cost of two, with Node 4, its new nexthop with a cost of three. If Node 2 updates its FIB before Node 3, a transient loop is triggered between both nodes until Node 3 updates its own FIB.

Constraint 2. *Nodes should update their FIB in an order that avoids transient forwarding events.*

Such transient loops occur naturally due to the unordered nature of flooding, even for planned modifications. A transient loop may reorder and delay packets. If it lasts long enough, it may saturate the link and cause congestion that also affects unrelated traffic [50]. Even worse, when the packets' TTL expire, routers drop them, effectively turning the transient loop into a transient black-hole. Therefore, such transient forwarding instabilities must be addressed in topologies as dynamic as satellite constellations. Section 5.1 presents our ordering that prevents transient loops.

4.3 Avoiding transient loops

Some solutions to avoid or minimise transient loops were proposed [11, 12, 25], but they all rely on the flooding process. The specificities of the problem allow an ordering that better scales with the topology. Indeed, feeder links are asymmetric, preventing traffic from bouncing back to the constellation through ground stations. This enables a new kind of ordering that allows merging link removal and addition updates of handovers in a single convergence.

Delaying FIB updates to minimise loops. The biggest advantage of link-state IGPs, and their main source of transient instabilities, is their ability to react to topological modifications. The drawback is that a topological modification may require multiple messages to be entirely described. For example, when a router fails, the whole topological modification is fully described only when a router has received the LSA of each of the failed router's neighbours. Routers with high connectivity, i.e., a large number of neighbours, are another example of this phenomenon. Advertising many links may not fit in a single MTU. Hence, IGPs perform fragmentation and slice the LSA into multiple ones. Modifying a single link metric is sufficient to trigger the flooding of many LSAs in that context. Computing the shortest path first (SPF) for each LSA received by a router, potentially multiple times in a short span of time, increases the number of FIB updates, hence the potential number of transient events.

Vendors, then the IETF with RFC8405 [15], introduced a back-off delay to aggregate as many LSAs as possible in a single SPF computation. When a router receives a first update, it waits for an initial SPF delay. At timer expiration, if the router did not receive additional updates, it computes the SPF and applies the related FIB updates. If additional routing updates arrive before the timer expiration, the delay is increased. RFC8405 suggests 50 ms as the initial delay value. This method purposely increases the exposure of the system to additional topological modifications, such as failures, to decrease the number of FIB updates susceptible to trigger transient forwarding events. Authors of RFC8405 acknowledge that this method does not protect the system from all possible micro-loops [15], breaking Constraint 2. While that approach may decrease the number of loops, it also increases the duration of the convergence and the duration of existing loops.

The oFIB method. François and Bonaventure [25] propose the Ordered FIB Updates (oFIB) framework, a method to avoid transient events while advertising one or more updates of the same kind, i.e., link addition or removal, in a single convergence. The method relies on the Reverse Shortest Path DAG (*rSPDAG*) rooted at a neighbouring node of the updated link. It represents the shortest path from every node to the root. Figure 5a depicts $rSPDAG_{G_2}(0)$, the *rSPDAG* rooted at Node 0 in the graph G_2, shown in Figure 4b.

Upon asymmetric metric modification of link $X \rightarrow Y$, $rSPDAG_{X \rightarrow Y}(X)$ is defined as the *sub-rSPDAG* rooted at X in the *rSPDAG* rooted at Y, i.e. all shortest paths that use X to reach Y. Figure 5a shows $rSPDAG_{1 \rightarrow 0}(1)$ in grey solid arrows and $rSPDAG_{2 \rightarrow 0}(2)$ in black solid arrows, both computed on G_2. François and Bonaventure [25] prove that every link susceptible to present a transient loop belongs to that *rSPDAG* such that, according to the type of metric change (increment or decrement), ordering FIB updates as per the forward or backward traversal of the *rSPDAG* suppresses all transient loops. In the case of symmetric metrics, the loop has a size of two, i.e., only the neighbours of the loops are implied. In the case of asymmetric metrics, the loop may be longer than two hops. Mérindol [49] demonstrates that, if ECMP is enabled in the network, such loops can be summarized as a succession of loops of size two. This implies that ordering the FIB updates between the two neighbours of a loop is sufficient to suppress it. In Figure 5b, grey arrows show the ordering that solves the loop triggered by the metric decrement on link $(0, 1)$, defined by the top-to-bottom traversal of $rSPDAG_{1 \rightarrow 0}(1)$. Indeed, by updating the FIB of Node 1 before Node 4, we avoid the transient loop. Respectively, black arrows show the ordering that solves the loop triggered by the metric increment on link $(0, 2)$, defined by the bottom-to-top traversal of $rSPDAG_{2 \rightarrow 0}(2)$. Indeed, by updating the FIB of Node 3 before Node 2, we avoid the transient loop. oFIB being designed atop flooding, the SPF delay mentioned above impacts convergence duration.

Computing metric modification sequences. François *et al.* [27] introduce a new method to avoid transient loops that does not require modifications to the control-plane. It relies on manual unitary link metric increments or decrements. They show that such small changes are loop-free and sufficient to progressively reach the desired metric. Clad *et al.* [11] propose splitting a metric change that might trigger transient events into a sequence of smaller metric changes that do not, each requiring its own convergence. They focus on computing minimal sequences. It is also applied to router changes [12]. We view such techniques as FIB-update ordering mechanisms because they insert intermediate FIB updates to ensure loop-free forwarding.

System exposure to concurrent topological failures. FIB ordering mechanisms cannot, by design, handle a concurrent convergence during a planned change. Without concurrency control, a second convergence starts as soon as a topological failure occurs during a first ordered convergence. Their FIB updates may interleave, potentially causing transient events even though the first convergence was ordered. The oFIB framework suggests mitigating this issue with an "Abandoning All Hope" (AAH) mechanism [66] that falls back to conventional unordered convergence.

Constraint 3. *The time spent in convergence should be the shortest possible to avoid concurrent convergences.*

Because oFIB is designed atop the flooding process [25, 66], which is slow and does not scale well [14, 44, 59], it does not respect Constraint 3. Since Clad *et al.*'s solution increases the number of convergences, it also breaks Constraint 3.

Constraint 4. *The synchronisation mechanism should be the most efficient possible in terms of message exchange.*

While the oFIB framework suggests the AAH mechanism [66], it does not propose an actual architecture because detecting and classifying concurrent failures is difficult. First, the full description of a failure may require multiple LSAs as described above. Receiving a single LSA may not be sufficient to correctly identify the failure. Second, to avoid overloading slower nodes in the topology, IGPs implement pacing strategies by typically rate-limiting the generation of LSAs. The typical value is 33 ms between two LSAs for IS-IS [14]. Third, propagating these LSAs through the topology takes time because of the link latencies and processing delay on each node. These elements, with the replication strategy of the flooding algorithm, explain (*i*) why IGPs converge slowly, (*ii*) why they scale poorly, and (*iii*) why concurrency control between convergences is not practicable. They also imply that an improved FIB synchronisation mechanism should not send as many messages as the flooding process to scale in large topologies such as satellite constellations.

5 Seamless Ground Station Handovers along Multicast Trees

Based on the constraints defined in Section 4, we propose an improved FIB synchronisation method. It allows batching both events of a handover in a single convergence, and it implicitly embeds the FIB ordering in a multicast tree. We model the handover events as metric decrements from (resp. increments to) large metric values that will attract traffic to (resp. divert from) the new (resp. old) feeder link [25]. Unplanned events remain handled by classical mechanisms and are considered out of the scope of this paper. Indeed, we assume that classical fast-reroute (FRR) solutions such as TI-LFA [6] are deployed. Such mechanisms ensure correct forwarding in case of a single link failure, which is enough to mask that kind of failure during an ongoing convergence. Then, the failure can be repaired in a dedicated loop-free FIB synchronisation. FRR techniques can handle multiple concurrent link failures, but this is not guaranteed depending on their location in the network. More complex failure scenarios are left as future work, yet are briefly introduced in Section 8. We restrict the scope to topological connectivity, i.e., advertising routed identifiers and loopback addresses. Advertising external prefixes is left for future work, but briefly discussed in Section 8. Since both connectivity events defining the handover are distributed at once, our propagation method allows us to get rid of SPF delay and speeds up the convergence. We also formally prove that our method allows loop-free convergence for such handovers. Due to space limitations, the proof is in Appendix B. One should note that such a handover should happen within a given timeframe. Indeed, the advertisement for the addition of the new feeder link should happen **after** its physical connection. Similarly, the advertisement for the removal of the old feeder link should happen **before** its physical connection is removed. Otherwise, satellites may try to forward traffic through a non-existing link, because it does not exist yet, or because it does not exist anymore. As introduced in Section 2, we reserve one radio to perform such handover in a make-before-break manner.

5.1 Nesting FIB Updates in a Single Ordered Convergence

Ground stations do not forward traffic from the constellation back to the constellation, i.e. they do not act as a nexthop to reach a satellite from another satellite. So only the paths towards the ground station will change after the topological modifications triggered by the handover. Because of this and for the sake of readability, in the remainder of this section, we will refer to nexthops to the ground station only as "nexthops".

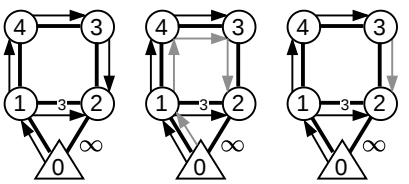
We take advantage of that property to perform a single, ordered and loop-free convergence per handover. Our ordering is defined by the *(r)SPDAG* computed on the **post-convergence topology**, and rooted in the ground station. Figure 6a illustrates this ordering for the handover from Node 2 to Node 1. As Node 2 will lose its feeder link while Node 1 has got its, the *(r)SPDAG* is shown in Figure 6a, i.e., on the topology where only Node 1 has a feeder link (i.e., the post-convergence one).

In practice, we first update the set of satellites that will have the ground station as a direct nexthop after the handover (Node 1 in our example). They have loop-free paths to the ground station as they are directly connected to it. Then, we propagate the update to the set of satellites whose nexthop is in the upstream set (Nodes 2 and 4). They have loop-free paths to the ground station because their nexthops have loop-free paths. We continue this propagation recursively, each path being prepended hop by hop. By construction, each satellite that updates its FIB has loop-free paths, as it has for nexthop a satellite that already has loop-free paths to the ground station. A specific feature remains. When a satellite has multiple nexthops (Node 2), it has to wait for all its nexthops (Nodes 1 and 3) to update their FIB before updating its own FIB. So it will respect the property that each nexthop already has loop-free paths to the ground station. A formal proof of this seamless propagation can be found in Appendix B. It relies on two facts. First, as ground stations do not forward traffic from the constellation back to the constellation, only paths **towards** the ground stations will change. This means that we only have to avoid loops towards the ground station. Second, updating nodes of a path hop-by-hop, from its destination to its source, is loop-free. If we combine these properties, we get that updating satellites along the paths from the ground station to any node is loop-free.

5.2 Distributing Ordered FIB Updates along Multicast Trees

The shortest-path multicast tree rooted in the ground station is computed on the **post-convergence topology**. We distribute the routing update containing both topological changes defining the handover along this tree. However, it lacks a few links to match the *(r)SPDAG*, which requires adding some explicit synchronisation for the missing links. Hence, each node receives the LSAs describing the complete handover from (one of) its nexthop(s), as well as explicit synchronisation from its other nexthops. This respects Constraint 2. It is interesting to note that, since the oFIB ordering is also defined by an *(r)SPDAG* [25], it could also use our propagation method with the shortest-path multicast tree. However, multicast-enabled oFIB would remain less effective than our propagation. Indeed, it would use two convergences (one for the link addition and one for the link removal) through two different multicast trees. Moreover, the metric increment update would have to travel down the entire tree before updating while ascending back, as routers the furthest away have to update first [25]. This would lead to three traversals of the network. We name that multicast-oFIB approach the *Sequential (S)* method.

Figure 6 depicts how the handover from Node 2 to Node 1 as border satellite can be distributed along multicast trees. Since there exist two equal-cost paths between Node 1 and 2, multiple shortest-path multicast trees are possible. Figure 6b shows one grey tree and one black tree. In ECMP-enabled networks, both links should be covered by our ordering. We chose to let BIER select the multicast tree while adding an explicit synchronisation for links from the DAG that the multicast tree does not cover. Figure 6c depicts the DAG with the grey arrow representing the explicitly-synchronised link. We see that this ordering avoids all the potential loops presented in Figure 4d. In practice, when a node receives the LSAs of a handover, it updates its LSDB and incrementally computes its SPT.



(a) Our ordering (b) Two multicast trees (c) The ordering DAG

Fig. 6. Illustration of the multicast trees and *(r)SPDAG*. In (c), black arrows depict the multicast tree, and the grey arrow depicts the explicit synchronisation message.

If it has explicitly-synchronised nexthops, it waits for them to send explicit synchronisation. Then it updates its FIB and its BIFT, which allows it to identify to which nodes it has to transmit the

LSAs. Finally, it propagates the update to those nodes. In Figure 6c, it happens as follows. Node 1 first receives the LSAs and updates, then it distributes the LSAs to Nodes 2 and 4. Node 4 directly updates its FIB and BIFT, but Node 2 knows it has to wait for an explicit confirmation from Node 3. Indeed, both Nodes 1 and 3 are its nexthops to reach the root of the multicast tree, i.e., the ground station. When Node 3 receives the LSAs from Node 4, it updates its FIB and BIFT, then sends an explicit notification to Node 2, which can then update, avoiding the loops discussed in Section 4.2.

The case of explicit synchronisation. A node may determine its nexthops by looking up its FIB. For example, when Node 2 receives the update from Node 1, it determines that Node 3 is its nexthop. However, Node 3 cannot easily determine that it has to send explicit synchronisation to Node 2. Indeed, Node 3 is a leaf of the multicast tree shown with black arrows in Figure 6c, but it is a parent in the DAG shown with the additional grey arrow. However, its nearest common ancestor with Node 2 in the multicast tree, i.e. Node 1, can. Such computation also relies on direct lookups in the BIFT, which are lightweight operations. The idea is to detect the nodes that are reachable via a nexthop, but for which the BIER packets come from another path. In the current example, Node 1 directly replicates the routing update to Node 2, but it knows that it can also reach Node 2 via Node 3. Node 1 receives the routing update encapsulated in a BIER packet with the bitstring 11110, with the LSB identifying the ground station. Its FBM associated with Node 2 is 00100, and with Node 4 is 11100. Node 1 replicates the update to Node 2 with the bitstring $11110 \wedge 00100 = 00100$ and to Node 4 with the bitstring $11010 \wedge 11100 = 11000$. Now, Node 1 computes the *explicit synchronisation bitstring* for a given neighbour by applying a logical *and* operation with the original bitstring, the negation of the bitstring sent to the neighbour, and the neighbour's FBM. This allows identifying the nodes that are reachable via the neighbour but will not receive the update via that neighbour. For example, Node 1 computes for its neighbour 4, $11110 \wedge \neg 11000 \wedge 11100 = 11110 \wedge 00111 \wedge 11100 = 00100$, which indeed identifies Node 2. Node 1 embeds that explicit synchronisation bitstring along with the update it replicates to Node 4. As Node 4 does not have Node 2 as a direct neighbour, it forwards that explicit bitstring to Node 3. When Node 3 receives the explicit bitstring along with the routing update, it knows that Node 2 is a child in the DAG to which it has to send a confirmation.

Multicast is more efficient than flooding. We argue that our distribution method is more lightweight than flooding since (*i*) LSAs pass through each link at most once depending on the multicast tree, and (*ii*) explicit synchronisation messages are smaller than the routing update, only propagate locally, and only cross links of the *(r)SPDAG* that are not covered by the multicast tree. Hence, it respects Constraint 4. In addition to being less resource-intensive than the flooding process, our method does not rely on timers such as the *hold-down* or *MAX_FIB* timers defined in oFIB [25, 66]. The multicast tree root does not have to compute the FIB ordering either, since it is defined by a multicast tree encoded in the BIFTs.

We name our method *Nested (N)*, because it distributes and applies both handover updates in a single convergence, requiring only one traversal of the DAG. Besides requiring a single convergence, applying both topological modifications also ensures (*i*) that a handover triggers at most a single FIB update per node, since both topological modifications are delivered together, and (*ii*) that the IGP's SPF delay timer can be safely suppressed for the same reason. It is reasonable to assume that Jumbo frames are enabled, so aggregated topological modifications fit in a single MTU.

6 Evaluating Nested Ground Station Handovers Performances

In the previous section, we showed that our synchronisation method enables a ground station handover in a single convergence without triggering forwarding loops. Now, we simulate our method and compare it with the sequential approach on the constellations introduced in Section 3.1, along with the ground station set we leverage. Constellation parameters are detailed in Appendix C.

6.1 Methodology

Our evaluation framework consists of a toolchain of multiple stages. First, it computes all possible connectivity events over one orbital period at a fixed time step for a given constellation. As introduced in Section 2, we consider ground stations capable of establishing up to 9 feeder links, while satellites can establish up to 4 of them. The second stage defines the initial connections for each ground station and satellite, then computes a feasible (not necessarily optimal) complete sequence of events for the whole orbital period considering these hardware constraints.

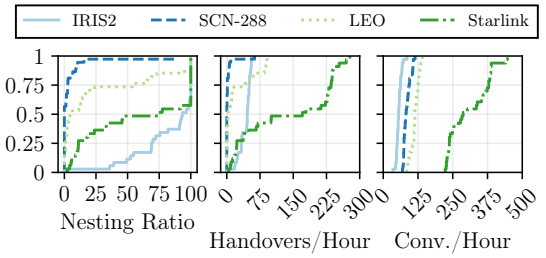


Fig. 7. Statistical view of the connectivity events, w.r.t. proportions of ground stations, selected by the second stage of the toolchain.

From this sequence, we identify the pairs of events that are eligible for a handover. Third, for each selected handover, we simulate the state of the constellation at the time of the handover to determine the latency of each link. Fourth, each handover is simulated with both the sequential and nested approaches using a custom discrete-time simulator implemented for this work. Finally, it produces the figures in this paper from the raw simulation output.

Ground Station Handovers Sample. Figure 7 depicts statistical views of the handovers we evaluate. First, it shows what we call the *nesting ratio*, which represents the proportion of single connectivity events that can be considered as a handover. A nesting ratio of 100 % means that all the connectivity events seen by a ground station can be nested, hence halving the total number of convergences during an orbital period. Due to space constraints, we discuss in detail the impact of this ratio on the selected handovers for each constellation in Appendix C.

Figure 7 also depicts the ECDF of the handover rate, measured in number of handovers per hour, w.r.t. each ground station for an orbital period. This rate is directly correlated to the nesting ratio since a ground station with a higher ratio will perform more handovers than one with a lower ratio. For the LEO constellation, 75 % of the ground stations perform at most a few tens of handovers per hour, while the remaining 25 % perform at most 80 handovers per hour. For the IRIS2 constellation, the handover rate is quite stable across the ground stations, with a maximum rate of around 70 handovers per hour. The handover rate is also quite stable across ground stations for SCN, with a few tens per hour. Only a very small number of ground stations perform more handovers, with a maximum of around 70 per hour. For the Starlink constellation, half of the ground stations have a high rate of at least around 170 handovers per hour up to slightly less than 300, while the other half perform at most 100 handovers per hour. Such a clear distinction between two classes of ground stations suggests that the parameters of the constellation favour the coverage of some stations. The orbital period of the four constellations being similar, the difference in rate mainly depends on the altitude and density of the constellations.

Each event, i.e., the single addition or removal of a connection, or a handover combining both kinds of connectivity events, requires one convergence to synchronise the topology. The last graph of Figure 7 shows the ECDF of the convergence rate, expressed in number of convergences per hour, for each constellation. LEO, SCN, and IRIS2 depict steady convergence rates across the ground stations, around 100 convergences per hour for LEO, 75 for SCN and 50 for IRIS2. The rate for IRIS2 is lower than for LEO, which is explained by the altitude difference. IRIS2 being higher and less dense, it triggers fewer connectivity events. For the Starlink constellation, the convergence rate is widely distributed between 200 and 400 convergences per hour for nearly every ground

station. This suggests that half of the ground stations that do not have a nesting ratio of 100 % are overwhelmed with numerous single connectivity events to advertise.

Convergence Simulator and Router Model. We simulate every handover for both FIB synchronisation methods with a custom discrete-time simulator. It simulates the propagation of routing updates and confirmation messages through the topology. Since routing updates fit in a single MTU, each message is delivered to a neighbouring node after a single one-way delay. We do not model the dynamics of the constellation, i.e., we simulate each handover in isolation. For each of them, we recompute the link latencies, from which we derive the link metrics. Also, we do not model the variation in latency between cross-plane links during a convergence in our simulator. We argue that such changes are negligible w.r.t. convergence durations.

We ensure in a post-processing phase that each node has correctly received the routing update(s) to guarantee Constraint 1. Each time a node applies the routing update and synchronises its FIB, the simulator recomputes the FIB of each node, then explores the paths between all node pairs to detect potential transient events. This sanity check, together with FIB content checks, ensures that the simulator correctly implements both the sequential and nested synchronisation methods, i.e., that Constraint 2 is respected. These checks are disabled for Starlink due to the computational overhead, given the size of this topology. However, no transient event has been detected for all the simulated handovers on the other topologies. Figure 8 shows the cumulative number of loops avoided with our FIB synchronisation method. These are statically computed by overlapping the pre- and post-convergence DAGs. Handovers on the three smallest topologies trigger at least a few tens of loops, with a maximum of around 80 loops. On the other hand, 75 % of the handovers on the Starlink constellation trigger at least 70 loops, and 25 % at least 500 loops, with a maximum of around 800 loops. We also model the internal timings of routers as follows. Updating a prefix in the FIB requires 100 µs, which is a conservative value [25]. We consider FIB updates as atomic operations that cannot be interrupted by external events.

Duplicating a routing update on an output link requires 10 µs, while originating a confirmation takes 5 µs. Indeed, the second operation is less resource-intensive and generates smaller messages. The SPT computation duration depends on the size of the topology, as well as its connectivity. We leverage the SPT model of Eramo *et al.* [21], who show that the SPT computation duration of a commercial Cisco router follows the law $\alpha \times n \times log(n)$, where n is the number of nodes and α is a constant experimentally defined to fit the router's SPT duration measurements. In another work, Eramo *et al.* [20] define an optimised SPT algorithm for large-scale networks and evaluate its performance. From these works, we model the SPT duration with $0.039 \times n \times log(n)$ in microseconds[5], where n is the number of nodes.

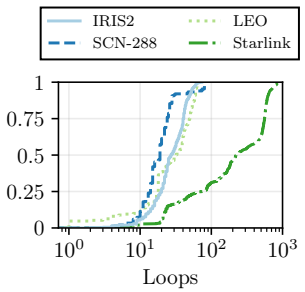


Fig. 8. ECDF of the number of loops avoided per handover.

6.2 Performances of the Synchronisation Methods

We measure (*i*) the convergence duration per handover, (*ii*) the cumulated convergence duration normalised by the orbital period and (*iii*) the FIB updates overhead. We also measure (*iv*) the number of messages exchanged normalised by the number of satellites and (*v*) the synchronisation overhead. We define the FIB update overhead as the deviation from the ratio of one update per satellite. Each node should update its FIB at most once to account for a given handover. Formally, the FIB update overhead is the number of FIB updates normalised with the number of satellites, minus one. We

[5]Eramo *et al.* [20] show that their algorithm computes the SPT in 0.35 ms in a topology of 893 nodes and 4154 links. By reversing the law, we obtain that $\alpha = 3.9 \times 10^{-5}$ ms, or 3.9×10^{-2} µs.

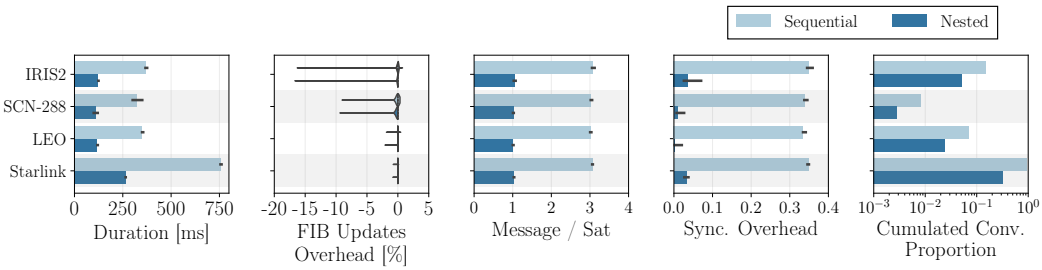


Fig. 9. Simulation results for both synchronisation methods. From left to right, (*i*) the median duration of the simulated convergences, (*ii*) the overhead of each method in terms of FIB update proportion, (*iii*) the median number of messages per satellite per convergence, (*iv*) the synchronisation overhead in terms of message proportion, and (*v*) the proportion of the cumulated convergences duration w.r.t. the orbital period. Bar plots with error bars represent the median value of the metric across the handover sample. The error bars indicate an interval around the median that contains 95 % of the samples.

define the synchronisation overhead as the proportion of confirmation messages normalised by the total number of messages. Figure 9 depicts statistical overviews of our simulations. The number of handovers evaluated for each constellation is given in Table 1.

Convergence duration. The first graph represents the median duration of the convergence triggered by a handover. For each bar plot, the error bar represents the interval containing 95 % of the samples. The median of our method is 124 ms (resp. 111 ms) compared to around 368 ms (resp. 325 ms) for the sequential approach on the IRIS2 (resp. SCN) constellation. The duration is slightly shorter for SCN, which is explained by a higher density of the constellation, 264 satellites for IRIS2 against 288 for SCN, and a lower altitude for SCN, 780 km against 1200 km for IRIS2. While SCN is denser than IRIS2, the lower altitude allows a smaller latency on ISLs. Also, SCN follows a Walker Star pattern, resulting in deeper multicast trees that cannot spread through the seam. On the other hand, LEO and Starlink lie almost at the same altitude; the main difference is that Starlink contains more than thrice the number of satellites than LEO, 1584 against 432. The ranges of link latencies for both constellations are similar and lie between 4 ms and 10 ms. The sequential method takes a median duration of 352 ms (resp. almost 759 ms), while our approach takes 119 ms (resp. around 265 ms) for the LEO (resp. Starlink) constellation. We observe that our approach is at least twice as fast as the sequential method, regardless of the topology. This confirms that our FIB synchronisation method is faster than the sequential ordering method. We have also evaluated both methods with a more pessimistic router model where the SPT computation duration is defined by $45 \times n$ in microseconds[6]. We found that our approach is still faster than the sequential one. For example, the sequential method takes around 6 s for Starlink against around 3 s for our approach. Both methods still reach sub-second convergence for the three other constellations. This suggests that the SPT computation duration is an important component in the propagation duration of our synchronisation method. One should also note that our model of the sequential method is very optimistic since additional synchronisations would be required in a real implementation to indicate the end of the first convergence and allow the start of the second convergence. Our measurements for that method correspond to a lower bound.

FIB overhead evaluation. The second graph depicts, with violin plots, the deviation from the expected FIB update count. A deviation of 0 indicates that every satellite has applied exactly a

[6]François *et al.* [26] experimentally defined that the full SPT computation from another commercial router takes 45 µs per node with an average connectivity of four links.

single FIB update in response to the handover. This is the expected value since the ground station is the sole destination for which the paths change. A positive deviation indicates the proportion of additional FIB updates across the topology. A negative deviation indicates that some nodes did not update their FIB, i.e., did not change their paths towards the ground station. This may happen in cases where the new path has exactly the same cost and goes through the same nexthop. Hence, such deviation is more related to the structure of the topology, the link metric, and the handover occurring rather than the FIB synchronisation method. First, we observe a negative deviation ranging from 1 to 16 % fewer FIB updates for the Walker Delta-based constellations. The distribution being especially narrow indicates that such cases are rare among the simulated handovers. Second, we note a higher maximum negative deviation around 10 % for the Walker Star constellation. Such a high value is also quite rare, but lower values around 0 are more common. This may be explained by the topological structure of the Star pattern. Regardless of the constellation pattern, the negative deviation for a given constellation is slightly more pronounced for our method w.r.t. the sequential approach since a FIB modification may require a single update against two with a sequential ordering. Third, we observe an overhead, i.e., a positive deviation, of about 1 % with the sequential approach for the IRIS2 constellation, while it is negligible for the other Delta-based constellations. Again, low deviations are more common for the Star-based constellation. Fourth, our method shows no overhead, which is expected since it is designed to trigger at most a single FIB update. The symmetrical nature of Walker patterns does not trigger many FIB updates overhead.

Synchronisation cost in terms of message count. The third graph of Figure 9 shows the number of routing messages that a satellite sends, normalised with the number of satellites in the constellation. As expected, our method requires, on median, that each satellite sends a single message against 3 for the sequential method: one to distribute and apply the link addition towards the ground station, one to distribute the link removal and the last to apply the link removal. The fourth graph depicts the overhead for both methods in terms of synchronisation messages. First, it shows that around 35 % (in median) of the exchanged messages for the sequential method are only dedicated to explicitly ordering the FIB updates between nodes when applying the link removal. This proportion does not vary much with the constellation since it corresponds to the rSPDAG rooted at the ground station. Second, it shows a smaller overhead for our method, with a maximum value around 4 % for the IRIS2 constellation. We also see that the proportion is more dependent on the constellation. This is explained since the confirmation messages are required to cover branches of the SPDAG that the multicast tree does not cover, both being rooted at the ground station. A low (resp. high) overhead indicates a low (resp. high) proportion of ECMP paths from the ground station towards satellites. Both graphs indicate that our synchronisation method is significantly more efficient than the sequential one in terms of explicit synchronisation messages needed.

Cumulated convergence duration w.r.t. the orbital period. The fifth graph of Figure 9 shows the cumulated convergence duration resulting from every simulated handover, normalised with the constellation's orbital period. For SCN, the cumulated convergence duration represents about 0.3 % of the orbital period (18 s), compared with more than 0.8 % (48 s) for the sequential approach. The proportion for SCN is low since only 150 handovers occur naturally during one orbital period. For LEO (resp. IRIS2), the cumulated duration represents around 2.4 % (resp. around 5 %) of the orbital period for our method, compared with more than 7 % (resp. 14.8 %) for the sequential one. This approximates to 2 min (resp. about 5 min) for our method on LEO (resp. IRIS2) and about 6 min (resp. 16 min) for the sequential method. For the Starlink constellation, the cumulated convergence duration for our method takes 32 % of the orbital period, against 91 % for the sequential approach. This corresponds respectively to 30 min and 87 min. These values show that our method is faster than a sequential approach and still leaves plenty of time to advertise the single connectivity events.

Handling simultaneous handovers. The predictable convergence of our synchronisation method and the regularity of constellation dynamics enable off-line handover scheduling. Such schedules spread simultaneous handovers over a short time window, preventing instability from concurrent gateway changes. In Section 5, we assume a handover is triggered just before the leaving satellite disconnects, so the schedule anticipates some handovers. For the handover sets considered here, the maximum anticipation is 6 s for Starlink, well below the median feeder-link duration. Due to space limitations, we can not detail further our method to compute such a schedule.

7 Related Works

The research community has been very active in studying the Starlink constellation [53, 56], considering end-to-end performance, its terrestrial infrastructure [31], user terminals [40], etc. To our knowledge, no work has looked at how routing is established.

Several works have studied the design of ISL topologies by exploring ISL link patterns beyond the +Grid [8], or investigating the impact of orbital parameters on latency, throughput, and coverage [5, 82]. These analyses focus on steady-state performance, while others [48] addressed handover continuity by proposing a topology where handovers occur between satellites sharing an ISL.

Some researchers proposed to introduce dynamicity in the constellation topology by replacing one of the four fixed ISLs with a dynamic link optimised for delay [46]. However, establishing OISLs can take several minutes [29], which is significant compared to their reported average dynamic link duration. Others posit that a near-second pointing, acquisition, and tracking (PAT) time could enable energy-efficient on-demand topologies [7].

Some researchers have looked at the stability of the computed routes. StableRoute [58] proposes route computation techniques to suppress unnecessary route changes, including those caused by arbitrary tie-breaking among equal-cost paths. Other researchers have studied the effect of route changes on transport connections and proposed alternative algorithms [69].

Distributed approaches for routing have been proposed based on Multi-Armed Bandit (MAB) algorithms. LEONS [22] designs a slice configuration policy using restless MABs to select network slice topologies spanning terrestrial, air, and satellite segments under partial observability, but the MAB reward requires per-segment throughput, delay and energy information whose signalling cost and timescale feasibility are not discussed. SkyLink [13] combines MABs with tile coding to perform independent hop-by-hop traffic distribution at each satellite using only local information. While it does not compute end-to-end paths, the feedback mechanism needed to attribute end-to-end outcomes to individual per-hop decisions is not discussed.

While source-routed approaches have been shown to remain attractive for constellations [65], most works propose new routing protocols. SHORT [45] organises multi-shell constellations into a three-tier routing hierarchy (terrestrial, orbital shells, orbits) and uses stable orbital-geodetic coordinates to decouple routing from topology dynamics. IDLB [64] proposes an SDN-based routing protocol that partitions the constellation into space-borne clusters, each managed by a controller satellite. While convergence times are reported for different cluster sizes, the protocol-level mechanisms for propagating topology changes within and between clusters are not fully specified. Geographic Check-point Routing (GCR) [33] proposes a protocol to establish Source Routing within constellations while being topologically agnostic by relying on their geography. Al-Hourani [2] models the duration of satellite connections and the handover strategies of user terminals. Such heuristics favour short-lived connections and frequent handovers, which is acceptable for a UT, but not for ground stations, whose handovers each trigger a constellation-wide synchronisation.

Hock *et al.* [35] find that the oFIB approach may temporarily overload some links while reconverging and suggest a heuristic defining specific link metrics to mitigate such overloads. Other approaches such as Fu *et al.* [28] and Shi *et al.* [67] tackle the same issue by computing more

specific FIB update ordering, but show that their approach can not suppress all link overloads. The previous methods focus on solving the microloop issue by modifying the control-plane behavior with FIB updates ordering and link metric optimisations. Other works such as FIFR by Nelakuditi *et al.* [55], Safeguard by Li *et al.* [42] or FCFR by Robertson *et al.* [63] address the issue by masking the transient instabilities with Fast ReRoute (FRR) techniques directly in the data-plane while the control-plane converges. None of them actually modify the convergence process.

8 Conclusion

Emerging satellite constellations are connected to the terrestrial networks using ground stations. We first characterize the dynamicity of the connectivity events between satellites and these ground stations. Then, we explore how link-state routing protocols could be used in these constellations. During the frequent handovers between the gateway and the constellation, these protocols cause transient forwarding instabilities. State-of-the-art Ordered FIB Updates (oFIB) loop avoidance mechanisms, tailored for IGPs, are not suitable for these satellite constellations.

We propose a new FIB synchronisation mechanism that efficiently distributes handover routing updates along multicast trees, and embeds an implicit FIB ordering directly within these trees. We formally prove our approach in Appendix B. We simulate it, and an optimized version of oFIB, and compare them over more than ten thousand handovers on four constellations. Our results show that our method is two to three times faster than a sequential method, while guaranteeing the same loop-free convergence property. While this paper provides a first overview of how to adapt the convergence of IGPs to satellite constellations, it opens new directions for future work.

Resilience against failures during a convergence. As discussed in Section 2, BIER can already be protected at the flow level and against topological failures. In Section 5, we assume that TI-LFA [6] is sufficient to mask a single-link failure before a dedicated loop-free convergence repairs it. These could have slots in the schedule introduced in Section 6. If the synchronisation method converges faster than the failure rate, it should not need to handle multiple concurrent link failures. However, no public data currently confirms this hypothesis, and node failures overlapping with a convergence require a dedicated study because they directly affect the advertised topology changes.

Advertising a default route or external prefixes. Vanbever *et al.* [73] show that external prefixes may still experience forwarding loops even when topological modifications within a routing domain are loop-free. We believe that our approach may still suppress such kinds of loops by rooting our ordering DAG in a central controller connected to every gateway. However, the impact of our ordering method on the forwarding of external prefixes still has to be evaluated.

Applying our ordering method on terrestrial networks. None of the hypotheses we posit in this paper specifically tie our approach to satellite constellations. Actually, it could also be leveraged to advertise border router migrations in terrestrial networks, which should be further studied.

Analysing the impacts of the Ground Stations' position and on the data-plane. We only leverage the OneWeb ground station dataset in our evaluations. More diverse datasets could help to evaluate the impact of the ground stations distribution on the convergence performance of our synchronisation method. Orthogonally, we mainly focused on analysing the IGPs convergence from the control-plane viewpoint. Further analysis could evaluate the impact of our synchronisation method on the data-plane, especially on the packet flows.

Acknowledgments

This work has been partially supported by the Walloon Region (Belgium) as part of the funding of the FRFS-WEL-T strategic axis. This research was partially supported by the European Union and Wallonia through the "Wallonia 2021-2027" program. We would also like to thank the anonymous reviewers for their valuable comments.

References

[1] 1990. Use of OSI IS-IS for routing in TCP/IP and dual environments. RFC 1195. doi:10.17487/RFC1195

[2] Akram Al-Hourani. 2021. Session duration between handovers in dense LEO satellite networks. *IEEE Wireless Communications Letters* 10, 12 (2021), 2810–2814.

[3] Richard J. Barnett. 2012. O3b – A different approach to Ka-band satellite system design and spectrum sharing. Presentation at the ITU Regional Seminar for RCC countries on Prospects for Use of the Ka-band by Satellite Communication Systems, Almaty, Kazakhstan. https://www.itu.int/dms_pub/itu-r/md/12/iturka.band/c/R12-ITURKA.BAND-C-0010!!PDF-E.pdf

[4] Richard J. Barnett. 2016. *OneWeb Non-Geostationary Satellite System.* Technical Report. https://fcc.report/IBFS/SAT-LOI-20160428-00041/1134939.pdf

[5] Suvam Basak, Amitangshu Pal, and Debopam Bhattacherjee. 2023. Exploring Low-Earth Orbit Network Design. In *Proceedings of the 1st ACM Workshop on LEO Networking and Communication* (Madrid, Spain) *(LEO-NET '23)*. Association for Computing Machinery, New York, NY, USA, 1–6. doi:10.1145/3614204.3616103

[6] A. Bashandy, S. Litkowski, C. Filsfils, P. Francois, B. Decraene, and D. Voyer. 2025. Topology Independent Fast Reroute Using Segment Routing. RFC 9855. doi:10.17487/RFC9855

[7] Dhiraj Bhattacharjee, Pablo G Madoery, Aizaz U Chaudhry, Halim Yanikomeroglu, Güneş Karabulut Kurt, Peng Hu, Khaled Ahmed, and Stéphane Martel. 2024. On-demand routing in LEO mega-constellations with dynamic laser inter-satellite links. *IEEE transactions on aerospace and electronic systems* 60, 5 (2024), 7089–7105.

[8] Debopam Bhattacherjee and Ankit Singla. 2019. Network topology design at 27,000 km/hour. In *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies* (Orlando, Florida) *(CoNEXT '19)*. Association for Computing Machinery, New York, NY, USA, 341–354. doi:10.1145/3359989.3365407

[9] Huaimo Chen, Mike McBride, Steffen Lindner, Michael Menth, and Toerless Eckert. 2026. *A Framework for Fast Reroute with Bit Index Explicit Replication (BIER-FRR).* Internet-Draft draft-ietf-bier-frr-14. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-ietf-bier-frr/14/ Work in Progress.

[10] Marco Chiesa, Andrzej Kamisiński, Jacek Rak, Gábor Ré tvári, and Stefan Schmid. 2021. A survey of fast-recovery mechanisms in packet-switched networks. *IEEE Communications Surveys & Tutorials* 23, 2 (2021), 1253–1301.

[11] François Clad, Pascal Mérindol, Jean-Jacques Pansiot, Pierre Francois, and Olivier Bonaventure. 2013. Graceful convergence in link-state IP networks: A lightweight algorithm ensuring minimal operational impact. *IEEE/ACM transactions on networking* 22, 1 (2013), 300–312.

[12] Francois Clad, Pascal Mérindol, Stefano Vissicchio, Jean-Jacques Pansiot, Pierre Francois, et al. 2013. Graceful router updates in link-state protocols. In *The 21st IEEE International Conference on Network Protocols (IEEE ICNP 2013)*.

[13] Wanja de Sombre, Arash Asadi, Debopam Bhattacherjee, Deepak Vasisht, and Andrea Ortiz. 2026. SkyLink: Scalable and Resilient Link Management in LEO Satellite Networks. *IEEE Transactions on Communications* 74 (2026), 1421–1434. doi:10.1109/TCOMM.2025.3636092

[14] Bruno Decraene, Les Ginsberg, Tony Li, Guillaume Solignac, Marek Karasek, Gunter Van de Velde, and Tony Przygienda. 2024. IS-IS Fast Flooding. RFC 9681. doi:10.17487/RFC9681

[15] Bruno Decraene, Stephane Litkowski, Hannes Gredler, Acee Lindem, Pierre Francois, and Chris Bowers. 2018. Shortest Path First (SPF) Back-Off Delay Algorithm for Link-State IGPs. RFC 8405. doi:10.17487/RFC8405

[16] Yoann Desmouceaux, Thomas Heide Clausen, Juan Antonio Cordero Fuertes, and W Mark Townsley. 2018. Reliable multicast with BIER. *Journal of Communications and Networks* 20, 2 (2018), 182–197.

[17] Yoann Desmouceaux, Juan Antonio Cordero Fuertes, and Thomas Heide Clausen. 2019. Reliable BIER with peer caching. *IEEE Transactions on Network and Service Management* 16, 4 (2019), 1826–1839.

[18] Christophe Diot, Brian Neil Levine, Bryan Lyles, Hassan Kassem, and Doug Balensiefen. 2000. Deployment issues for the IP multicast service and architecture. *IEEE network* 14, 1 (2000), 78–88.

[19] Eylem Ekici, Ian F Akyildiz, and Michael D Bender. 2001. A distributed routing algorithm for datagram traffic in LEO satellite networks. *IEEE/ACM Transactions on networking* 9, 2 (2001), 137–147.

[20] Vincenzo Eramo, Marco Listanti, and Antonio Cianfrani. 2008. Multi-path OSPF performance of a software router in a link failure scenario. In *2008 4th International Telecommunication Networking Workshop on QoS in Multiservice IP Networks*. IEEE, 197–203.

[21] Vincenzo Eramo, Marco Listanti, Antonio Cianfrani, et al. 2007. OSPF Performance and Optimization of Open Source Routing Software. *Int. J. Comput. Sci. Appl.* 4, 1 (2007), 53–68.

[22] Haitham H. Esmat, Beatriz Lorenzo, and Jianqing Liu. 2023. LEONS: Multi-Domain Network Slicing Configuration and Orchestration for Satellite-Terrestrial Edge Computing Networks. In *ICC 2023 - IEEE International Conference on Communications*. 6294–6300. doi:10.1109/ICC45041.2023.10278769

[23] Bill Fenner, Mark J. Handley, Hugh Holbrook, Isidor Kouvelas, Rishabh Parekh, Zhaohui (Jeffrey) Zhang, and Lianshu Zheng. 2016. Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised). RFC 7761. doi:10.17487/RFC7761

[24] Dennis Ferguson, Acee Lindem, and John Moy. 2008. OSPF for IPv6. RFC 5340. doi:10.17487/RFC5340

[25] Pierre Francois and Olivier Bonaventure. 2007. Avoiding transient loops during the convergence of link-state routing protocols. *IEEE/ACM Transactions On Networking* 15, 6 (2007), 1280–1292.

[26] Pierre Francois, Clarence Filsfils, John Evans, and Olivier Bonaventure. 2005. Achieving sub-second IGP convergence in large IP networks. *ACM SIGCOMM Computer Communication Review* 35, 3 (2005), 35–44.

[27] Pierre Francois, Mike Shand, and Olivier Bonaventure. 2007. Disruption free topology reconfiguration in OSPF networks. In *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*. IEEE, 89–97.

[28] Jing Fu, Peter Sjodin, and Gunnar Karlsson. 2008. Loop-free updates of forwarding tables. *IEEE Transactions on Network and Service Management* 5, 1 (2008), 22–35.

[29] Jason Gerard, Juan A Fraire, and Sandra Céspedes. 2025. Modeling Pointing, Acquisition, and Tracking Delays in Free-Space Optical Satellite Networks. In *2025 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*. IEEE, 1–6.

[30] Les Ginsberg, Tony Przygienda, Sam Aldrin, and Zhaohui (Jeffrey) Zhang. 2018. Bit Index Explicit Replication (BIER) Support via IS-IS. RFC 8401. doi:10.17487/RFC8401

[31] Sarah-Michelle Hammer, Vamsi Addanki, Max Franke, and Stefan Schmid. 2024. Starlink Performance through the Edge Router Lens. In *Proceedings of the 2nd International Workshop on LEO Networking and Communication* (Washington, DC, USA) *(LEO-NET '24)*. Association for Computing Machinery, New York, NY, USA, 67–72. doi:10.1145/3697253.3697273

[32] Mark Handley. 2018. Delay is Not an Option: Low Latency Routing in Space. In *Proceedings of the 17th ACM Workshop on Hot Topics in Networks (HotNets)*. 85–91. doi:10.1145/3286062.3286075

[33] Valentin Hartig, Marcin Bosk, Nitinder Mohan, and Paulo Mendes. 2024. Segment Routing based on Geographic Checkpoints. In *Proceedings of the 2nd International Workshop on LEO Networking and Communication* (Washington, DC, USA) *(LEO-NET '24)*. Association for Computing Machinery, New York, NY, USA, 25–30. doi:10.1145/3697253.3697268

[34] Caleb Henry. 2025. Europe Signs Contracts for IRIS^2 Constellation. *SpaceNews* (May 2025). https://spacenews.com/europe-signs-contracts-for-iris%C2%B2-constellation/

[35] David Hock, Matthias Hartmann, Tim Neubert, and Michael Menth. 2011. Loop-free convergence using ordered FIB updates: Analysis and routing optimization. In *2011 8th International Workshop on the Design of Reliable Communication Networks (DRCN)*. IEEE, 156–163.

[36] Christian Hopps. 2000. Analysis of an Equal-Cost Multi-Path Algorithm. RFC 2992. doi:10.17487/RFC2992

[37] Christian Hopps and Dave Thaler. 2000. Multipath Issues in Unicast and Multicast Next-Hop Selection. RFC 2991. doi:10.17487/RFC2991

[38] Gianluca Iannaccone, Chen-nee Chuah, Richard Mortier, Supratik Bhattacharyya, and Christophe Diot. 2002. Analysis of link failures in an IP backbone. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. 237–242.

[39] Liz Izhikevich, Manda Tran, Katherine Izhikevich, Gautam Akiwate, and Zakir Durumeric. 2024. Democratizing LEO Satellite Network Measurement. *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 8, 1 (2024), 13:1–13:26. doi:10.1145/3639039

[40] Bhaskar Kataria, Hammas Bin Tanveer, Rishab Nithyanand, and Rachee Singh. 2026. What Obstructed Skies Teach Us About Satellite Internet. In *1st New Ideas in Networked Systems (NINeS 2026)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 7–1.

[41] Oltjon Kodheli, Eva Lagunas, Nicola Maturo, Shree Krishna Sharma, Bhavani Shankar, Jesus Fabian Mendoza Montoya, Juan Carlos Merlano Duncan, Danilo Spano, Symeon Chatzinotas, Steven Kisseleff, Jorge Querol, Lei Lei, Thang X. Vu, and George Goussetis. 2021. Satellite Communications in the New Space Era: A Survey and Future Challenges. *IEEE Communications Surveys & Tutorials* 23, 1 (2021), 70–109. doi:10.1109/COMST.2020.3028247

[42] Ang Li, Xiaowei Yang, and David Wetherall. 2009. SafeGuard: safe forwarding during route changes. In *Proceedings of the 5th international conference on Emerging networking experiments and technologies*. 301–312.

[43] Tony Li. 2025. A Routing Architecture for Satellite Networks. RFC 9717. doi:10.17487/RFC9717

[44] Tony Li, Peter Psenak, Huaimo Chen, Luay Jalil, and Srinath Dontula. 2024. Dynamic Flooding on Dense Graphs. RFC 9667. doi:10.17487/RFC9667

[45] Yuanjie Li, Lixin Liu, Hewu Li, Wei Liu, Yimei Chen, Wei Zhao, Jianping Wu, Qian Wu, Jun Liu, and Zeqi Lai. 2024. Stable Hierarchical Routing for Operational LEO Networks. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking* (Washington D.C., DC, USA) *(ACM MobiCom '24)*. Association for Computing Machinery, New York, NY, USA, 296–311. doi:10.1145/3636534.3649362

[46] Yueyi Li, Junfeng Wu, Guohua Kang, Luyu Chen, Yuhuan Qiu, and Wenwen Zhou. 2024. A flexible topology control strategy for mega-constellations via inter-satellite links based on dynamic link optimization. *Aerospace* 11, 7 (2024), 510.

[47] Athina Markopoulou, Gianluca Iannaccone, Supratik Bhattacharyya, Chen-Nee Chuah, Yashar Ganjali, and Christophe Diot. 2008. Characterization of failures in an operational IP backbone network. *IEEE/ACM transactions on networking* 16, 4 (2008), 749–762.

[48] Joseph Mclaughlin, Jee Choi, and Ramakrishnan Durairajan. 2023. ×Grid: A Location-oriented Topology Design for LEO Satellites. In *Proceedings of the 1st ACM Workshop on LEO Networking and Communication* (Madrid, Spain) *(LEO-NET '23)*. Association for Computing Machinery, New York, NY, USA, 37–42. doi:10.1145/3614204.3616110

[49] Pascal Mérindol. 2022. *Build and Measure Routing Systems*. HDR. Université de Strasbourg, Strasbourg, France.

[50] Pascal Merindol, Pierre David, Jean-Jacques Pansiot, Francois Clad, and Stefano Vissicchio. 2018. A fine-grained multi-source measurement platform correlating routing transitions with packet losses. *Computer Communications* 129 (2018), 166–183.

[51] Daniel Merling, Steffen Lindner, and Michael Menth. 2020. Comparison of fast-reroute mechanisms for BIER-based IP multicast. In *2020 Seventh International Conference on Software Defined Systems (SDS)*. IEEE, 51–58.

[52] Daniel Merling, Thomas Stüber, and Michael Menth. 2023. Efficiency of BIER multicast in large networks. *IEEE Transactions on Network and Service Management* 20, 4 (2023), 4013–4027.

[53] 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* (Singapore, Singapore) *(WWW '24)*. Association for Computing Machinery, New York, NY, USA, 2723–2734. doi:10.1145/3589334.3645328

[54] John Moy. 1998. OSPF Version 2. RFC 2328. doi:10.17487/RFC2328

[55] Srihari Nelakuditi, Sanghwan Lee, Yinzhe Yu, Zhi-Li Zhang, and Chen-Nee Chuah. 2007. Fast local rerouting for handling transient link failures. *IEEE/ACM Transactions on networking* 15, 2 (2007), 359–372.

[56] Jianping Pan, Jinwei Zhao, and Lin Cai. 2023. Measuring a Low-Earth-Orbit Satellite Network. In *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 1–6. doi:10.1109/PIMRC56721.2023.10294034

[57] Tian Pan, Tao Huang, Xingchen Li, Yujie Chen, Wenhao Xue, and Yunjie Liu. 2019. OPSPF: Orbit prediction shortest path first routing for resilient LEO satellite networks. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 1–6.

[58] Tian Pan, Guohao Ruan, Qiang Fu, Zhengjie Luo, Junkai Huang, Xingshuang Luo, and Tao Huang. 2025. StableRoute: When Dijkstra's Algorithm Meets Topology-Varying Satellite Networks. In *IEEE INFOCOM 2025 - IEEE Conference on Computer Communications*. 1–10. doi:10.1109/INFOCOM55648.2025.11044485

[59] Jeff Parker, Rajesh I. Balay, and Dave Katz. 2000. IS-IS Mesh Groups. RFC 2973. doi:10.17487/RFC2973

[60] Maxime Piraux and Juan A. Fraire. 2026. *A code to describe satellite constellations*. Internet-Draft draft-piraux-space-constellation-code-01. Internet Engineering Task Force. https://datatracker.ietf.org/doc/draft-piraux-space-constellation-code/01/ Work in Progress.

[61] Peter Psenak, Nagendra Kumar Nainar, IJsbrand Wijnands, Andrew Dolganow, Tony Przygienda, Zhaohui (Jeffrey) Zhang, and Sam Aldrin. 2018. OSPFv2 Extensions for Bit Index Explicit Replication (BIER). RFC 8444. doi:10.17487/RFC8444

[62] Aravindh Raman, Matteo Varvello, Hyunseok Chang, Nishanth Sastry, and Yasir Zaki. 2023. Dissecting the Performance of Satellite Network Operators. *Proceedings of the ACM on Networking* 1, CoNEXT3 (2023), 15:1–15:25. doi:10.1145/3629137

[63] Glenn Robertson, Nirupam Roy, Phani Krishna Penumarthi, Srihari Nelakuditi, and Jason M O'Kane. 2017. Loop-free convergence with unordered updates. *IEEE Transactions on Network and Service Management* 14, 2 (2017), 373–385.

[64] Manuel MH Roth, Hartmut Brandt, Hermann Bischl, David Fernández Piñ as, and Guray Acar. 2026. IDLB: An SDN-Based Load-Balancing Routing Protocol for Autonomous Satellite Constellation Networks. *International Journal of Satellite Communications and Networking* 44, 2 (2026), 109–127.

[65] Manuel M. H. Roth. 2023. Analyzing Source-Routed Approaches for Low Earth Orbit Satellite Constellation Networks. In *Proceedings of the 1st ACM Workshop on LEO Networking and Communication* (Madrid, Spain) *(LEO-NET '23)*. Association for Computing Machinery, New York, NY, USA, 43–48. doi:10.1145/3614204.3616109

[66] Mike Shand, Stewart Bryant, Stefano Previdi, Clarence Filsfils, Pierre Francois, and Olivier Bonaventure. 2013. Framework for Loop-Free Convergence Using the Ordered Forwarding Information Base (oFIB) Approach. RFC 6976. doi:10.17487/RFC6976

[67] Lei Shi, Jing Fu, and Xiaoming Fu. 2009. Loop-free forwarding table updates with minimal link overflow. In *2009 IEEE International Conference on Communications*. IEEE, 1–6.

[68] SpaceRISE Consortium. 2025. The EU Perspective on Authorisation of LEO Constellations. Presentation at ITU Space Connect. https://www.itu.int/space-connect/wp-content/uploads/sites/66/2025/02/Dominic_The-EU-Perspective-on-Authorisation-of-LEO-constellations.pdf

[69] Gregory Stock, Juan A Fraire, Santiago Henn, Holger Hermanns, and Andreas Schmidt. 2024. A stability-first approach to running tcp over starlink. In *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 1708–1713.

[70] 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

[71] Juan Arias Suarez, Louis Barbier, Marina Dehez-Clementi, and Oana Hotescu. 2026. SatGoNetEm: an Open Source Emulator of Satellite Constellations and Non-Terrestrial Networks. In *IFIP Networking 2026*. https://dl.ifip.org/db/conf/networking/networking2026/1571253461.pdf Table III uses 12 planes × 22 satellites for the IRIS ^{2} LEO shell, citing [**?**].

[72] Hammas Bin Tanveer, Mike Puchol, Rachee Singh, Antonio Bianchi, and Rishab Nithyanand. 2023. Making Sense of Constellations: Methodologies for Understanding Starlink's Scheduling Algorithms. In *Companion of the 19th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*. 37–43. doi:10.1145/3624354.3630586

[73] Laurent Vanbever, Stefano Vissicchio, Luca Cittadini, and Olivier Bonaventure. 2013. When the cure is worse than the disease: The impact of graceful IGP operations on BGP. In *2013 Proceedings IEEE INFOCOM*. IEEE, 2220–2228.

[74] Stefano Vissicchio and Luca Cittadini. 2016. FLIP the (flow) table: Fast lightweight policy-preserving SDN updates. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 1–9.

[75] Stefano Vissicchio, Olivier Tilmans, Laurent Vanbever, and Jennifer Rexford. 2015. Central control over distributed routing. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*. 43–56.

[76] John Gerard Walker. 1970. *Circular orbit patterns providing continuous whole earth coverage*. Technical Report.

[77] Markus Werner. 1997. A dynamic routing concept for ATM-based satellite personal communication networks. *IEEE journal on selected areas in communications* 15, 8 (1997), 1636–1648.

[78] Markus Werner, Cecilia Delucchi, H-J Vogel, Gerard Maral, and J-J De Ridder. 1997. ATM-based routing in LEO/MEO satellite networks with intersatellite links. *IEEE Journal on Selected areas in Communications* 15, 1 (1997), 69–82.

[79] Markus Werner, Jochen Frings, Frédéric Wauquiez, and Gérard Maral. 2001. Topological design, routing and capacity dimensioning for ISL networks in broadband LEO satellite systems. *International Journal of Satellite Communications* 19, 6 (2001), 499–527.

[80] IJ. Wijnands, E. Rosen, A. Dolganow, J. Tantsura, S. Aldrin, and I. Meilik. 2018. Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks. RFC 8296. doi:10.17487/RFC8296

[81] IJsbrand Wijnands, Eric C. Rosen, Andrew Dolganow, Tony Przygienda, and Sam Aldrin. 2017. Multicast Using Bit Index Explicit Replication (BIER). RFC 8279. doi:10.17487/RFC8279

[82] Wenyi Zhang, Zihan Xu, and Sangeetha Abdu Jyothi. 2024. An In-Depth Investigation of LEO Satellite Topology Design Parameters. In *Proceedings of the 2nd International Workshop on LEO Networking and Communication* (Washington, DC, USA) *(LEO-NET '24)*. Association for Computing Machinery, New York, NY, USA, 1–6. doi:10.1145/3697253.3697263

[83] Yaoying Zhang, Qian Wu, Zeqi Lai, and Hewu Li. 2022. Enabling low-latency-capable satellite-ground topology for emerging LEO satellite networks. In *IEEE INFOCOM 2022-IEEE Conference On Computer Communications*. IEEE, 1329–1338.

[84] Jinwei Zhao, Owen Perrin, Ali Ahangarpour, and Jianping Pan. 2025. Measuring the oneweb satellite network. In *2025 9th Network Traffic Measurement and Analysis Conference (TMA)*. IEEE, 1–10.

A Illustration of BIER Forwarding

In this section, we illustrate the BIER forwarding process based on Figure 1. The FBM associated with Node 2 at Node 1 is 110110110, meaning that Node 1 can reach Nodes 2, 3, 5, 6, 8 and 9 through Node 2. The FBM associated with Node 4 at Node 1 is 111111000. Suppose that Node 1 sends a multicast message to every router, i.e., it sets every bit in the bitstring to one. The forwarding procedure works as follows. Node 1 iterates over the bits set in the bitstring. The first LSB in the bitstring identifies Node 1 itself; it decapsulates the packet and forwards it to the multicast listeners attached to it, then it clears the bit. The second bit identifies Node 2. Node 1 extracts the FBM of Node 2 from its BIFT and applies a logical *and* operation between the bitstring and the FBM. The result, $111111110 \wedge 110110110 = 110110110$, determines the enabled nodes that will receive the multicast packet through Node 2. Node 1 duplicates the BIER packet, replaces the bitstring with the result of the *and* operation and forwards it to Node 2. Then, Node 1 clears the bit from the original bitstring with a logical *and* operation with the inverse of the FBM, leaving $111111110 \wedge \neg 110110110 = 111111110 \wedge 001001001 = 001001000$. The next bit set to one is Node 4. Node 1 applies the same operations as before, hence replicating to Node 4 the BIER packet with the new bitstring $001001000 \wedge 001001000 = 001001000$. After clearing the bits, the bitstring is empty, meaning that the forwarding procedure on Node 1 has finished. The other nodes apply the forwarding procedure in turn until the bitstring is empty, which stops the propagation of the BIER packet.

B Proof

We now prove that our method performs a ground station handover without triggering any routing loop by updating satellites in a specific order, defined by an *(r)SPDAG* rooted at the ground station (*GS*). This DAG is computed on the post-convergence topology, which should require explicit ordering computation at the root. However, the attentive reader will note that it is not required because the multicast tree representing the DAG is **dynamically built during the convergence**. Indeed, by definition of the ordering, a node first applies the topological modifications, i.e., updates its FIB and BIFT **before propagating** the LSAs. Hence, a node replicates the routing update on its post-convergence view of the topology. This thus does not require any explicit ordering computation besides the definition of the explicitly synchronised links in the ECMP case. The specificity of handovers that enables our ordering is that paths to reach border satellites do not move during the handover, as all modified links are connected to the ground station, which cannot be a hop. We consider ECMP, which means that a node α has a set of nexthops to another node ϵ, denoted $nh_{\alpha,G}(\epsilon)$, G being the network graph $G = (V, E)$. Similarly, there exists a set of equal-cost shortest paths from α to ϵ, denoted $Paths(G, \alpha, \epsilon)$. A routing loop is defined pairwise, i.e., two neighbors are their respective nexthops for a given destination. Formally, we define $Loop(\alpha, \beta) \equiv \beta \in nh_{_,\alpha}(GS) \wedge \alpha \in nh_{_,\beta}(GS)$. Topologies are omitted, as they are the topology that the node currently knows when the loop occurs. Loops with more than two nodes are composed of multiple two-node loops because we use ECMP and symmetric metrics[7] [49]. In the remainder of this demonstration, we use "loops" to mean two-node loops.

Lemma B.1. *A routing loop can only exist when multiple perceptions of the topology co-exist in the network.*

Proof. A routing loop can only exist during a convergence, while the topology change propagates. Otherwise, the routing loop would persist in a converged network. □

[7]Metrics of feeder links are not symmetric, but they cannot belong to a loop with a greater size, as ground stations do not forward traffic between satellites.

A *candidate* routing loop is a loop that may happen during the transition from topology G_0 to G_1.

$$L_{G_0,G_1}(\alpha,\beta) \equiv \beta \in nh_{\alpha,G_1}(GS) \wedge \alpha \in nh_{\beta,G_0}(GS) \tag{1}$$

Corollary 1. *If two nodes have the same LSDB, they cannot be the nexthops of one another.*

We say the loop is originated in α and terminated in β when β is a new nexthop for α in the new graph G_1.

Lemma B.2. *Every loop is originated within one and only one node:* $L_{G_x,G_y}(\alpha,\beta) \implies \neg L_{G_x,G_y}(\beta,\alpha)$.

Proof. By Corollary 1, if α has β as nexthop in G_y, β cannot have α as nexthop. Therefore, by definition of the origination, the loop cannot be originated in β. □

Let U be the set of nodes that are up-to-date, i.e., in a transition $G_0 \rightarrow G_1$, the set of nodes that already know the new LSDB (G_1).

Lemma B.3. *A routing loop originated in α and terminated in β occurs when α applies the routing update (i.e., uses β as a new nexthop) while β is not yet up-to-date (i.e. still uses α as nexthop).*

$$L_{G_0,G_1}(\alpha,\beta) \wedge Loop(\alpha,\beta) \implies \alpha \in U \wedge \beta \notin U$$

Proof. If α and β are both up-to-date (resp. not up-to-date), no routing loop can exist; otherwise, the routing loop would exist in the converged new (resp. old) topology. If β is up-to-date and α is not, as α will not use β as nexthop, there cannot be any loop between them. Hence, the sole case where a loop can exist is if α updates before β. □

Corollary 2. *To avoid this loop, α must apply the update once β is up-to-date.*

$$L_{G_0,G_1}(\alpha,\beta) \wedge (\alpha \in U \implies \beta \in U) \implies \neg Loop(\alpha,\beta) \tag{2}$$

Lemma B.4. *A routing loop originated in α and terminated in β can occur only when at least β is added to the set of nexthops of α.*

$$L_{G_0,G_1}(\alpha,\beta) \implies \beta \in nh_{\alpha,G_1}(GS) \wedge \beta \notin nh_{\alpha,G_0}(GS) \tag{3}$$

Proof.

$$L_{G_0,G_1}(\alpha,\beta) \implies \beta \in nh_{\alpha,G_1}(GS) \quad \text{by (1)} \tag{4}$$

$$L_{G_0,G_1}(\alpha,\beta) \implies \alpha \in nh_{\beta,G_0}(GS) \quad \text{by (1)}$$

$$\alpha \in nh_{\beta,G_0}(GS) \implies \beta \notin nh_{\alpha,G_0}(GS) \quad \text{by Corollary 1} \tag{5}$$

Combining (4) and (5) gives the equation to be proved. □

Theorem B.5. *During the transition from G_0 to G_1, it is sufficient to avoid any loop in the network that for every node α that adds a nexthop β, β is already up to date.*

$$(\forall \alpha,\beta \mid \beta \in nh_{\alpha,G_1}(GS) \wedge \beta \notin nh_{\alpha,G_0}(GS) : \alpha \in U \implies \beta \in U) \implies \nexists \epsilon,\delta \mid Loop(\epsilon,\delta)$$

Proof.

$$\neg L_{G_0,G_1}(\alpha,\beta) \implies \neg Loop(\alpha,\beta) \text{ If no loop can occur, none will occur} \tag{6}$$

$$L_{G_0,G_1}(\alpha,\beta) \wedge (\alpha \in U \implies \beta \in U) \implies \neg Loop(\alpha,\beta) \text{ By (2)} \tag{7}$$

$$(\beta \in nh_{\alpha,G_1}(GS) \wedge \beta \notin nh_{\alpha,G_0}(GS)) \wedge \neg L_{G_0,G_1}(\alpha,\beta) \implies \neg Loop(\alpha,\beta) \text{ By (6)} \tag{8}$$

$$\neg(\beta \in nh_{\alpha,G_1}(GS) \wedge \beta \notin nh_{\alpha,G_0}(GS)) \implies \neg L_{G_0,G_1}(\alpha,\beta) \implies \neg Loop(\alpha,\beta) \text{ By (3) and (6)} \tag{9}$$

Ordering candidate loops make them loopfree (7). Pairs of nodes from which one is a new nexthop of the other and that do not form a candidate loop are loopfree (8). Pairs of nodes from which none is a new nexthop of the other are loopfree (9). Combining (7) with (8), and (9) shows that ordering potential loops is necessary and sufficient to avoid any loops. Ordering all pairs of nodes with a

new nexthop is sufficient to avoid any loop, as it is a superset of all candidate loops, by (3). This proves the Theorem. □

We define the ordering set S, the set of directed links $l_{\alpha,\beta} \triangleq \alpha \rightarrow \beta$ that define an ordering relationship between nodes β and α. Formally, $S \triangleq \{l_{\alpha,\beta} \mid \beta \in nh_{\alpha,G_1}(GS) \wedge \beta \notin nh_{\alpha,G_0}(GS)\}$.

Theorem B.6. *All links that belong to a path towards GS in G_1, and that begin with a link in S, will respect the loopfree ordering. Formally,*
$\forall l_{\alpha,\beta} \in S, \forall P \in Paths(G_1, \alpha, GS) \mid l_{\alpha,\beta} \in P : \nexists l_{\epsilon,\delta} \mid l_{\epsilon,\delta} \in P \wedge l_{\delta,\epsilon} \in S$

Proof. By definition, all links $l_{\epsilon,\delta} \in S$ are the first link of a path to GS. As there cannot be loops in the converged network, if $l_{\epsilon,\delta}$ is in a path to GS, $l_{\delta,\epsilon}$ cannot be in one. □

We can build a Directed Acyclic Graph (DAG) called *rDAG* that merges all paths to GS beginning with a link in S. It is acyclic because of Theorem B.6. Formally, $rDAG \triangleq \{l_{\epsilon,\delta} \mid \forall l_{\alpha,\beta} \in S, \forall P \in Paths(G_1, \alpha, GS) : l_{\alpha,\beta} \in P \wedge l_{\epsilon,\delta} \in P\}$. By construction of *rDAG*, if we apply the update from GS to its leaf, we avoid routing loops as we respect the ordering of Theorem B.5.

Lemma B.7. *If a node α adds β as a new nexthop during a handover, this is due to β being used to reach GS via at least one border satellite O_a which it did not use beforehand.*

$$\beta \in nh_{\alpha,G_1}(GS) \wedge \beta \notin nh_{\alpha,G_0}(GS) \implies \exists P \in Paths(G_1, \alpha, GS) \mid l_{\alpha,\beta} \in P \wedge l_{O_a,GS} \in P \\ \wedge \nexists P \in Paths(G_0, \alpha, GS) \mid l_{O_a,GS} \in P$$

Proof. By contradiction, assume α adds β to its nexthop set while still using the same border satellites. Without loss of generality, let assume that β is used to reach GS *via* O_a. As O_a was already in the nexthop set, reaching O_a *via* β should be shorter or equal to the old paths. At least one link in the old or new path to O_a must have changed its cost. But only links with GS change during a handover. Then GS must be a hop to O_a from α, contradicting that O_a is a border satellite to GS. □

We illustrate this situation with the transition $G_2 \rightarrow G_3$ depicted in Figures 4b and 4c. We posit $G_0 = G_2$, $G_1 = G_3$, $O_a = Node1$, $\beta = Node4$ and $\alpha = Node3$.

Corollary 3. *Links in S are the ones used to reach the border satellites that will exist in G_1.*

$$\forall l_{\alpha,\beta} \in S : \exists O_a \mid l_{O_a,GS} \in G_1 \wedge \exists P \in Paths(G_1, \alpha, GS) \mid l_{\alpha,\beta} \in P \wedge l_{O_a,GS} \in P \tag{10}$$

Obviously, paths to reach border satellites do not move during the handover, as all modified links are connected to GS, which cannot be a hop.

$$\forall O \mid l_{O,GS} \in G_1, \forall P \in Paths(G_1, \alpha, O) : P \in Paths(G_0, \alpha, O)$$

Let *rspDAG* be the *rDAG* with all links reversed.

$$rspDAG = \{l_{\beta,\alpha} \mid l_{\alpha,\beta} \in rDAG\}$$

Let *spDAG* be the union of all shortest paths from GS to every node in the network.

$$spDAG = \{l_{\epsilon,\delta} \mid \exists \alpha, \exists P \in Paths(G_1, GS, \alpha) : l_{\epsilon,\delta} \in P\}$$

Theorem B.8. *In a network with symmetric links, rspDAG can be computed from GS with Dijkstra's shortest path algorithm on G_1 and is included in spDAG.*

Proof. As *rDAG* is composed of paths of next hops to GS, it is composed of shortest paths from every node in *rDAG* to GS. In a network using symmetric links, *rspDAG* represents the shortest paths from GS to every node in *rDAG*. Corollary 3 tells us that the shortest paths should be computed on G_1 because *rDAG* is composed of the shortest paths to GS only *via* border satellites

existing in G_1. By construction, those paths can be extended to cover all the nodes in the topology with shortest paths from *GS*, so *rspDAG* is included in *spDAG*. □

Theorem B.9. *spDAG corresponds to the shortest paths multicast tree from GS to every node with extra links when multiple paths to a node have the same cost.*

Proof. The optimal multicast tree from *GS* to every node is composed of shortest paths from *GS* to every node with arbitrary choice in case of ECMP. As *spDAG* contains all the shortest paths to every node, the tree is a set of *spDAG*. As their constructions are the same, links not in the tree are those that are used when multiple paths with the same cost exist. □

C Methodology Complements

Topology	LEO	IRIS2	Starlink	SCN
Code [60]	D:600:55:432/16/4	D:1200:55:264/12/1	D:550:53:1584/72/39	S:780:86.4:288/12/6
Feeder links/SAT	4	4	4	4
Feeder links/GS	9	9	24	9
Handovers	1162	2630	6926	150
Period [h]	1:36:41	1:49:25	1:35:38	1:40:26

Table 1. Constellation parameters, handover filter parameters, and handover count per constellation.

Table 1 shows (*i*) the constellation parameters with compact constellation codes defined by Piraux and Fraire [60], (*ii*) the count of radios per ground station and per satellite for each constellation, and (*iii*) the selected number of handovers per constellation and their orbital period. The constellation code for IRIS2 has been inferred from public data [34, 68, 71].

Nesting ratios of the constellations. In Figure 7, we show for each constellation the distribution of the nesting delay w.r.t. the ground stations. For the LEO constellation, 75 % of the ground stations show a nesting ratio of at most 25 %. This indicates that most of the ground stations are lightly loaded and mainly see isolated connectivity events. On the other hand, for the IRIS2 constellation, 75 % of the ground stations show a ratio of at least 70 %, and half of them show a ratio above 50 %. This suggests that half the gateways are always connected to the maximum number of satellites they support, while the other half is nearly saturated. Since IRIS2 lies at twice LEO's altitude, each satellite sees more ground stations, which may explain their saturation. For SCN, the nesting ratio is zero for half the ground stations, and a few tens of percent for the other half. This explains the low count of handovers that the ground stations can perform.

Defining the radio count per gateway for Starlink. Each constellation uses the filter parameter values defined in Section 2, except for Starlink, for which we increase the maximum number of satellite connections per ground station to 24. Since we use OneWeb ground stations as there is no reliable source for Starlink ground station positions, this station set may be undersized for Starlink's scale. By choosing a limit of 24 satellites per station for the Starlink constellation, the nesting factor curve lies between those of IRIS2 and LEO constellations, and about half of the ground stations are fully saturated with a nesting ratio of 100 %. The other half is mildly, saturated with a ratio topping at 50 %. With lower limits, the nesting factor reaches 100 % for nearly all gateways due to the constellation size being more than three times that of the LEO one, which could be unfair compared to other constellations. Higher limits further degrade the nesting factor, which is unfair to the Starlink constellation whose ground station network is probably denser than the OneWeb one. With the ground station connectivity limit chosen for Starlink, about half of the ground stations are fully saturated and show a nesting ratio of 100 %.

Received 5 June 2026; accepted 10 September 2026