International Journal of Satellite Communications and Networking

WILEY

ORIGINAL PAPER OPEN ACCESS

IDLB: An SDN-Based Load-Balancing Routing Protocol for Autonomous Satellite Constellation Networks

Manuel M. H. Roth[1] | Hartmut Brandt[1] | Hermann Bischl[1] | David Fernández Piñas[2] | Guray Acar[2]

[1]Institute of Communications and Navigation, German Aerospace Center (DLR), Cologne, Germany | [2]ESTEC, European Space Agency, Noordwijk, the Netherlands

Correspondence: Manuel M. H. Roth (manuel.roth@dlr.de)

Received: 18 April 2025 | **Revised:** 30 June 2025 | **Accepted:** 28 July 2025

Funding: This work has been supported by the Advanced Research in Telecommunications Systems Programme of the European Space Agency (ESA), activity code 3A.117, https://connectivity.esa.int/projects/ropro.

Keywords: distributed SDN | load-balancing | low Earth orbit | routing | satellite constellation networks

ABSTRACT

Routing in satellite constellation networks with intersatellite links has become an important aspect to enable broadband Internet access and to integrate into terrestrial networks. However, their dynamic characteristics and large physical size require specifically tailored solutions. To address these challenges, we propose and investigate a load-balanced routing protocol based on distributed software-defined networking. The approach relies on independent space-borne clusters with on-board controllers. Reduced signaling overhead is achieved by geographical intercluster routing algorithms. We evaluate the performance of the protocol in a custom-built system-level simulator, considering different architectures, design choices, and scenarios. Comprehensive comparisons with source-routed schemes and an upper benchmark demonstrate the viability of the solution. Notably, for the given scenario, the protocol can handle network loads of up to 15.0 Gbps before quality of service compliance falls below 95%. Compared with the 7.6 Gbps supported by source-routing, this represents an increase of 97.4%. This is achieved while maintaining an average routing convergence of 117.338 ms. The work provides valuable in-depth insights into the design of optimized routing protocols for satellite constellation networks.

1 | Introduction

In recent years, low Earth orbit (LEO) satellite constellation networks (SCNs) have emerged as an alternative to ground-based systems and to Geostationary Orbit (GEO)-based satellite broadband services [1]. Novel SCN designs typically rely on Intersatellite Links (ISLs) to create meshed networks, reducing the required number of ground stations and increasing service coverage [1, 2].

To support more potential users, routing traffic efficiently through such a network is critical. In order to maximize actual throughput, while complying with stringent Quality of Service (QoS) requirements, necessitates tailored routing protocols. By optimizing the flow of data through the network and minimizing congestion, broadband Internet access can be provided to many users. While optical ISLs are able to provide high data rates, bottlenecks may still occur in the ISL network. The asymmetric traffic patterns of the terminal distribution on ground can result in traffic spikes or geographic hot spots causing temporary link saturation. Therefore, to ensure QoS compliance, proactive congestion mitigation and adaptive load-balancing are necessary.

Due to the physical size of SCNs, centralized ground-based control entities are not expected to provide timely routing adjustments and cause significant signaling overhead. In contrast, autonomous satellite systems can solve delay-critical

problems in-space. By decentralizing network control, routing convergence time is reduced according to the sizes of the considered domains. So, using in-orbit routing protocols enables faster reactivity to handover or failure events. However, low-complexity solutions are necessary as on-board processing is limited.

A promising paradigm for managing SCNs is software defined networking (SDN), which offers flexibility and softwarization [3, 4]. Moreover, in the context of next-generation networking standards such as 6G, SDN is expected to be a key enabler [3, 5]. Using SDN, proactive load balancing is enabled using dynamic switching rule configuration based on the aggregated network state information at the controller.

Based on these insights, the independent distributed load-balanced routing (IDLB) protocol has been proposed [6]. By using space-borne sub-division of the network, a flat hierarchy within the constellation is established. The resulting clusters are controlled by an SDN control unit, a role which a satellite of the cluster takes over. For routing within this domain, the controller applies load-balancing strategies to maximize throughput. To this end, we propose a suitable low-complexity routing algorithm. For routing between clusters, we present an approach based on geographical information to reduce signaling overhead.

In this work, the design of the IDLB protocol is described and analyzed in detail. As a lightweight solution specifically tailored to dynamic SCNs and their resource constraints, it is distinct from protocols used in terrestrial networks, for example, Border Gateway Protocol (BGP). Nevertheless, key concepts, such distribution and network hierarchies, and how they can be implemented effectively have been considered in the presented design. The main contributions of this work include:

- The design and implementation of a novel distributed SDN-based load-balancing routing protocol for SCNs. Dedicated intracluster and intercluster routing algorithms are formulated.
- The discussion of relevant design aspects of the proposed protocol. Namely, routing hierarchy, handover procedures, routing tables, geographic tiling, cluster arrangement, and multicast considerations.
- A quantitative evaluation of the protocol performance using system-level simulations with relevant benchmark approaches.
- An investigation of the impact of different system and protocol design aspects. These include comparisons between per-flow and per-packet routing, cluster sizes, intercluster routing approaches, and constellation sizes.

The work is structured as follows. In Section 2, the related works are summarized. An overview of the reference system is provided in Section 3. The design aspects of the proposed protocol are discussed in Section 4. Section 5 presents the considered intracluster and intercluster routing algorithms. In Section 6, the custom-built system-level simulator is introduced and described. Quantitative comparisons of the proposed approach and benchmark solutions are provided in Section 7. Finally, Section 8 concludes the paper.

2 | Related Works

In this work, we extend and improve upon the approach presented in a previous publication [6]. In general, for autonomous satellite systems several in-orbit routing approaches have been proposed, including QoS-aware and SDN-based schemes [7]. For QoS-based forwarding, decentralized as well as source-routed approaches have been investigated [2, 8]. A key limitation for these schemes is information retrieval. Optimizing paths on an end-to-end basis results in significant signaling overhead. With source-routing, each ingress node requires all relevant information [9]. However, the approach represents a relevant benchmark. It reflects the behavior of known terrestrial protocols, for example, OSPF. In general, comparisons with these protocols are difficult as they oftentimes face challenges in dynamic topologies [2]. In addition, due to the lack of dedicated backbone networks, to which all satellites are connected to, terrestrial protocols are not applicable without significant modifications making them difficult to include as benchmarks.

Besides centralized SDN schemes [10], there is research proposing distributed SDN [4] for SCNs [11, 12]. While these investigations focus on optimal controller placement and assume Shortest Path First (SPF) algorithms, we base our approach on a similar underlying architecture. Using a centralized controller on ground which enables nodes to make autonomous routing decisions has also been proposed [13]. To maximize throughput in SCNs, several approaches and algorithms have been considered. For instance, applying heuristics to solve the linear programming problem [14]. However, as this investigation proposes in-orbit decision-making, we focus less complex approaches.

For routing between clusters, that is, intercluster routing, IDLB makes use of geographical identifiers, similar to geographical routing methods [15, 16]. We apply the concept on a higher hierarchical level. In our scheme, the information is used for forwarding towards clusters, not individual nodes.

A similar distributed architecture based on Intermediate System to Intermediate System (IS-IS) has also been published [17]. In this context, Segment Routing Traffic Engineering (SR-TE) can also be applied to actively steer traffic, for example, to avoid known hot spots [18]. The proposed protocol adopts comparable link-state updates and handles interfaces between clusters in a similar fashion. But, the approach differs in its architecture and tailored design choices, for example, the applied load-balancing routing schemes. The hierarchical intercluster and intracluster split is also reminiscent of Interior Gateway Protocol (IGP) and BGP. However, BGP includes a multitude of features which are not required or can be encapsulated for routing within the satellite network. Thus, protocols considered for autonomous satellite constellations are typically designed as streamlined solutions.

The distributed SDN paradigm is also generally more flexible and adaptable given its softwarization of routing logic and fine-grained network control, enabling arbitrary flow allocations [13]. For satellite constellation networks evolving in size and alongside standards such as 6G, this flexibility is a key enabler [3, 5]. Standard IP routing protocols are more

difficult to override, and each device needs to be upgraded. The comparably lightweight IDLB protocol is tailored to SCNs and designed to be adjustable to diverse routing policies and extensions.

3 | Reference System

3.1 | Space Segment

In this investigation, we consider two single-layer reference constellations. Nevertheless, the proposed protocol is generally suited to also support multilayer or multishell constellations. These reference systems are based on proposed designs for potential next-generation satellite systems. Hence, they do not represent currently deployed constellation networks. The numbers of satellites are supposed to represent current trends in constellation sizing. The first constellation consists of 288 satellites and is called "SCN-288." It represents the main focus of most investigations in this work. The second constellation consists of 1440 satellites, thus called "SCN-1440," and is only used to provide a comparison in terms of QoS compliance performance. Both reference constellations are summarized in Table 1. As quasi-polar Walker star constellations [20] with an inclination of 86.4°, they share similarities in terms of general design with the Iridium constellation [21, 22].

In SCN-288, the 288 satellites are arranged in 12 planes at an altitude of 780 km. There are 24 satellites in a plane, resulting in an angular distance of 15°. The angular phase offset between co-rotating planes is 7.5°. SCN-1440 corresponds to a larger version of SCN-288 at an altitude of 600 km. Here, 1440 satellites in 30 planes with 48 satellites per plane are proposed.

Circular orbits are assumed, individual satellites orbit the Earth in a constant distance relative to the center of mass of the planet. In real-world systems, the actual position of a satellite varies slightly due to atmospheric and gravitational influences. The effects of these small positional variations on the overall transmission characteristics is generally not significant. Consequently, only the theoretical positions of satellites are considered.

TABLE 1 | Summary of space segment of SCN-288 and SCN-1440.

Space segment characteristic	SCN-288	SCN-1440
Number of satellites	288	1440
Number of planes	12	30
Number of satellites per plane	24	48
Satellite altitude (km)	780	600
Orbital inclination (°)	86.4	86.4
Cross-seam planes spacing (°)	30	12
Co-rotating planes spacing (°)	15	6
Phase offset co-rotating planes (°)	7.5	3.75
Interplane ISL shut-down latitude (°)	80	80
Number of ISLs per satellite	4	4
Maximum ISL data rate (Mbps)	1000	1000
Output buffer per ISL (Mbit)	0.36–1.08	0.36–1.08

3.1.1 | Constellation Connectivity

Each satellite of the constellation possesses four ISLs: two intraplane ISLs and two interplane ISLs. The bandwidth and availability of all ISLs is considered equal in this work.

Based on this setup, a grid network emerges, which is interrupted by the seams as well as the interplane ISL shutdown latitudes. A shutdown of the interplane ISLs is required at the polar regions, because of the switch in relative position of the orbital planes. So, a neighboring satellite which was previously on one side will find itself on the other side of a given satellite, after the crossing of orbital planes in a polar region. This switch has to be accounted for when reactivating the interplane ISLs. The shutdown latitudes mainly depend on the pivoting speed of the antennas and the angular velocity between the two satellites.

We assume state-of-the-art optical laser terminals, which are able to pivot rapidly and can compensate elevated Doppler effects well. For instance, in a related project, Tesat's ConLCT laser communication terminal was assumed to pivot at 2.5° s_{-1} on both axes (azimuth and elevation) simultaneously and to handle Doppler shifts of up to 3 GHz [23]. Reorientation and target tracking of such a laser terminal are assumed feasible up to latitudes of ±80° at the defined orbit altitudes. Corresponding shutdown latitudes are considered in this work.

We assume an ISL data rate of 1 Gbps. Moreover, output buffers with a size of 0.36 Mbit are considered for each ISL, utilizing a first-in-first-out (FIFO) tail drop policy. With an assumed packet size of 1500 Byte, the usual Ethernet Maximum Transmission Unit (MTU), this means 30 packets can be buffered. Larger buffers did not significantly impact results, except for large clusters, where more than 30 signaling packets may be sent at once. In this case, a buffer size of 1.08 Mbit is used (also discussed in Section 7.4.2).

3.2 | Ground Segment

We consider a ground segment consisting of user terminals (UTs) and gateway stations (GWs). The distribution of these entities is shown in Figure 1. An overview of the considered ground segment characteristics is provided in Table 2.

We consider a nonhomogeneous UT distribution to reflect the requirements of real-world use cases. To this end, we sample locations from worldwide population density statistics [24] to approximate a plausible distribution. Metropolitan and densely populated areas are discounted in the sampling using a linear filtering function, which sets the expected density per square kilometer to a maximum of 100. For the sake of simplicity, we do

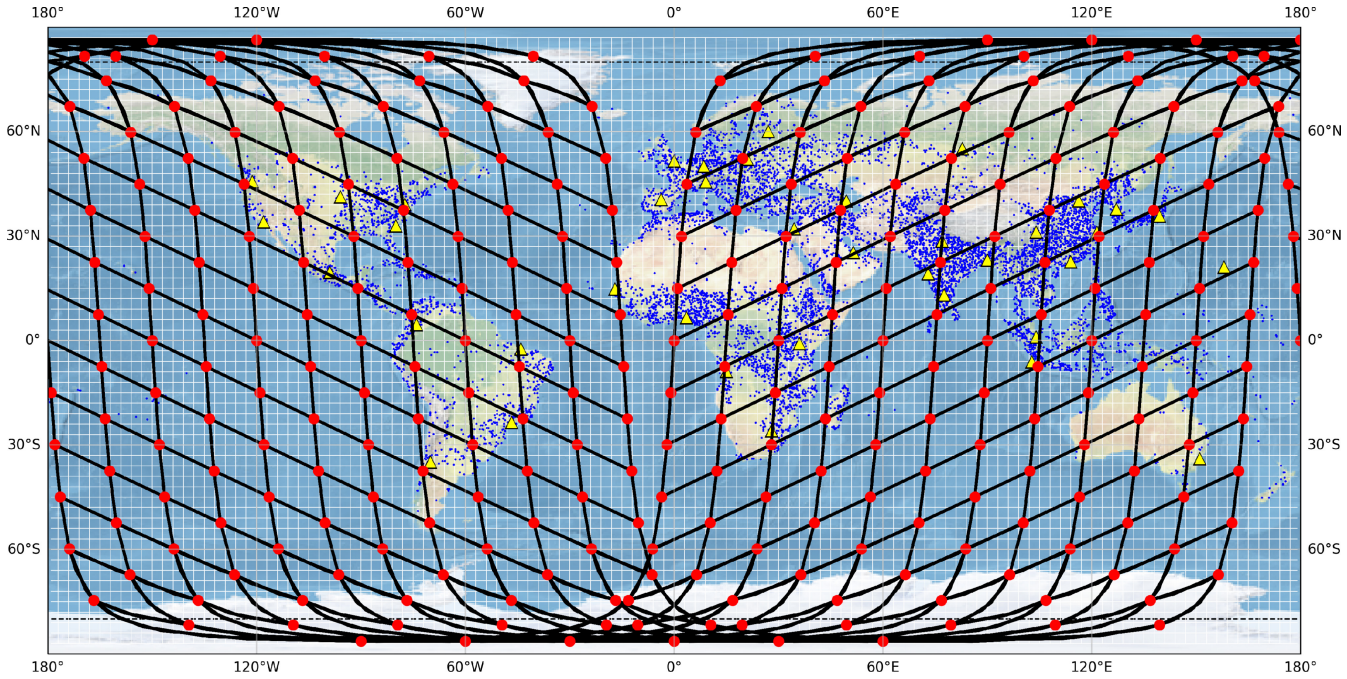

FIGURE 1 | Reference system: satellites of SCN-288 (red dots); orbital planes of SCN-288 (black lines); global distributions of UTs (blue dots) and GWs (yellow triangles); tiling into geographical areas (white grid); interplane ISL shutdown latitudes (black dashed line) [19].

TABLE 2 | Summary of the ground segment.

Ground segment characteristic	Value
Number of active user terminals	2000
User terminal minimum elevation angle (°)	30
Aggregated data rate per user terminal (Mbps)	100
Number of gateway stations	39
Gateway minimum elevation angle (°)	20
Aggregated maximum feeder uplink (Mbps)	5000
Aggregated maximum feeder downlink (Mbps)	1000

not include mobile terminals on trains, ships, or planes in this analysis.

GWs represent bridges between the nonterrestrial network (NTN) and the terrestrial Internet. We assume a deliberate distribution of GW campuses within reasonable proximity to existing infrastructure, for example, large cities, Internet exchange nodes, or data centers. The considered distribution of 39 GWs is shown in Figure 1 (GWs in yellow).

3.2.1 | Earth-Satellite Link Connectivity

Earth-satellite links (ESLs) connect the satellites with the UTs and GWs on ground. GWs are assumed to be placed strategically to enable lower minimum elevation angles. They are able to support higher ESL data rates than UTs. We assume a predictable handover scheme, which is discussed in more detail in Section 4.1. ESLs are modeled with constant data rates in this investigation, omitting aspects such as adaptive coding and modulation to isolate in-space routing effects. No assumptions are made about the antenna parameters, link budgets are considered sufficient to maintain the target ESL data rate of 100 Mbps (see Table 2). This simplification ensures that ESL dynamics do not confound the analysis of intraconstellation load-balancing. The aggregated maximum feeder uplink/downlink defines the total capacity a satellite can receive from or transmit to ground stations.

To provide resilience against local outages, we propose a minimum elevation angle which results in suitable coverage for SCN-288. Thus, we assume a minimum elevation angle of 20°, so that at least two satellites are visible at all times [25]. The coverage is shown in Figure 2. For SCN-1440, we consider a minimal elevation angle of 30°, based on similar considerations [6].

GWs profit from a variety of simultaneously visible satellites as they provide additional connectivity and routing diversity. So, for SCN-288, we assume a minimal elevation angle of 10° for the GWs. In this case, at least four satellites are visible for a GW. For SCN-1440, we assume a minimal elevation angle of 20° for the GWs, so at least six satellites are visible.

3.3 | Traffic Characteristics

Based on the requirements of specific end-user applications, suitable QoS classes have to be formulated. The considered traffic shares between these classes correspond to assumptions made in previous investigations [6, 9]. As this research primarily focuses on protocol characteristics, solely the presented set of traffic model parameters has been investigated.

As there are multiple viable use cases for such a constellation, we base our considered traffic shares on the following assumptions:

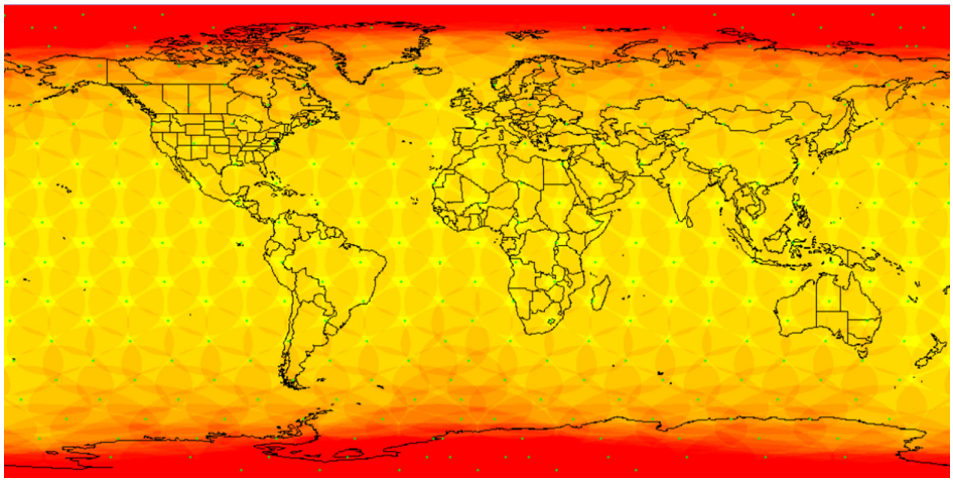

FIGURE 2 | Coverage of SCN-288 with minimum UT elevation angle of 20° [26]. At least two satellites visible everywhere.

- Most traffic, that is, > 50 %, is not delay-critical. Typically, the transfer of larger quantities of data, for example, in the form of video streaming or file sharing, is buffered. For this traffic, called "best effort," higher delay budgets are possible.
- Importantly, some traffic relies on the short end-to-end transmission delay enabled by SCNs [2]. This type has stringent delay and delay variation requirements. It is however assumed to be relatively small share due to the limited traffic volume, assumed around 10%. A limited number of frame drops for services using this priority class are assumed acceptable, given its primary focus on end-to-end latency. Thus, an elevated packet dropping rate is considered compliant.
- Some services fall between these classes. They are represented by QoS class 2. While delay-tolerant, the maximum latency is considered lower than the assumed budget for best effort. The class also includes a more relaxed jitter requirement. We assume the rest of the traffic share, that is, around 30% to 40%, belongs to this category. For signaling traffic, low latency and dropping rates are of importance. To this end, we assume that signaling packets are prioritized by the scheduler and use dedicated output buffers, which are large enough to effectively avoid packet drops for the considered signaling volume.

Naturally, for dedicated use cases or specific scenarios, more suitable traffic shares can be formulated. With the proposed assumptions we hope to reflect a plausible working example for SCNs, which allows for an in-depth analysis of QoS-based routing protocols.

The considered QoS classes are loosely based on the QoS profiles of the 5G NR networking standard [30]. In the SCN context, we consider only a simplified subset of the given profiles of interest. The proposed values are based on the end-user multimedia QoS categories of the ITU [27, 29]. The assumed latency and delay variance budgets roughly follow these recommendations, namely objectives for IP-based services, Y.1541 [28]. Some concrete values have been adapted based on observations and previous investigations into SCNs [2, 9]. From a conceptual point of view, the classifications are similar to the differentiated services (DiffServ) model [31], which is reflected in the names of the considered classes.

4 | Protocol Design

Independent Distributed SDN-based Load-Balanced routing (IDLB)

In short, the developed routing protocol, called IDLB, represents a load-balancing routing scheme based on distributed SDN tailored specifically to LEO SCNs. It is based on a space-borne sub-division of the network into clusters (different domains). These sub-networks are fixed within the topology of the constellation. Thus, from a terrestrial point of view the clusters move, the positions of the clusters are independent of the infrastructure on ground. Typically a satellite near the center of a cluster takes on the role of cluster controller. For the design of IDLB, we apply insights from a variety of routing approaches to formulate a streamlined protocol for the reference systems and traffic scenarios. IDLB consists of a distributed space-borne architecture for flexible, adaptive decision-making, which enables load-balanced intracluster routing, and heuristic intercluster routing based on geographical information.

The first key idea of IDLB is to use the **geographical information** of the nodes on ground to approximate the position of a destination in the network topology, similar to geographical routing schemes [15, 16]. By routing towards a geographical location, no global tracking of ESL handovers is required (except at the seams). In combination with a geographical address resolution scheme, signaling overhead can be decreased [15, 16]. To facilitate this strategy, a geographical address resolution scheme is used. Destination addresses are aggregated based on their geographical position.

The second concept is **distributed network control**. The routing scheme has to be able to react and adapt to network

changes in a timely manner. Due to the physical size of the networks, end-to-end propagation delays over multiple hops of more than 100 ms are common. Outsourcing all network control to a single (or multiple) control center on ground can thus negatively impact the QoS compliance. To comply with delay-sensitive updates, the network is thus sub-divided into smaller, independent domains. For the sake of simplicity, we focus on a flat in-orbit hierarchy of clusters in this investigation. However, the protocol can also be used in a multilevel SDN hierarchy, for example, with a master SDN controller on ground.

Network information is aggregated at the dedicated SDN controller node of each cluster, enabling proactive and load-aware routing decisions. This **autonomous, space-borne decision-making** can provide improved reactivity, routing convergence and signaling overhead. The size of the cluster determines the amount of signaling and propagation delays within the domain.

To mitigate congestion, particularly for broadband traffic, the protocol is designed to enable effective **load-balancing** strategies. By efficiently using available network resources, the supported throughput can be maximized. Distributing traffic on diverse paths does not necessarily result in significantly increased latency. Because of the inherent grid structure of the network, there are often multiple paths with similar characteristics [2, 9].

The architecture of the proposed protocol is illustrated in Figure 3. The impact of these concepts is evaluated quantitatively in Section 7. Certain design elements of our protocol draw inspiration from established terrestrial protocols, such as IS-IS and BGP, as described in Section 2. However, to effectively address the unique requirements of SCNs, we have made deliberate choices to create a lightweight, efficient protocol. The key design aspects that enable the approach are outlined in the following sections.

4.1 | Handover Events

To guarantee correctness and stability of a routing solution in an SCN context, the handling of handover events is crucial. There are frequent ISL and ESL handover events which should result in new coherent path choices. For both ISLs and ESLs, we assume handovers to be triggered by the relative (elevation) angle. Typically, ISL handovers are predictable and can thus be prepared accordingly. For the considered polar constellations, ISLs are switched when a satellite enters or leaves the interplane shutdown latitudes. The resulting changes in the network topology are pre-computed by the routing solution. Updates can be timed with handovers, so that all relevant routing tables are adjusted accordingly. To provide enough time for the remaining packets which are still forwarded to the node to arrive at the destination, an interplane shutdown countdown flag is used. In our simulations, the process is triggered 1 s before a handover.

ESL handover events can be more unpredictable or ambiguous, depending on the utilized handover scheme. For the proposed distributed space network, handovers between clusters may introduce ambiguity. Due to a handover, it may be possible that a packet arrives in the apparent destination cluster, but cannot reach the egress satellite - deadlocks may occur. To resolve this, we use cell-based handovers: all UTs of a geographical cell are handed over to a new cluster at a defined moment in time. So, the geographical routing tables within the involved clusters can be adjusted accordingly simultaneously. A locally unambiguous mapping between clusters is established.

As multiple quasi-simultaneous handover events are considered, by the terminals of a geographical cell, there are potential collisions without coordination. However, based on the scaling of the geographical cells and the density of available satellites, this caused no issues during the investigations.

For the sake of simplicity, we consider a handover strategy for the ESLs based on satellite proximity and visibility. When a satellite descends below the predefined minimum elevation angle threshold, it is no longer considered visible to the ground station. At this point, a handover to a different nearby satellite is initiated.

4.2 | Routing Table Setup

To accommodate the proposed two-level routing hierarchy, IDLB applies a two-tiered routing table system. Notably, the

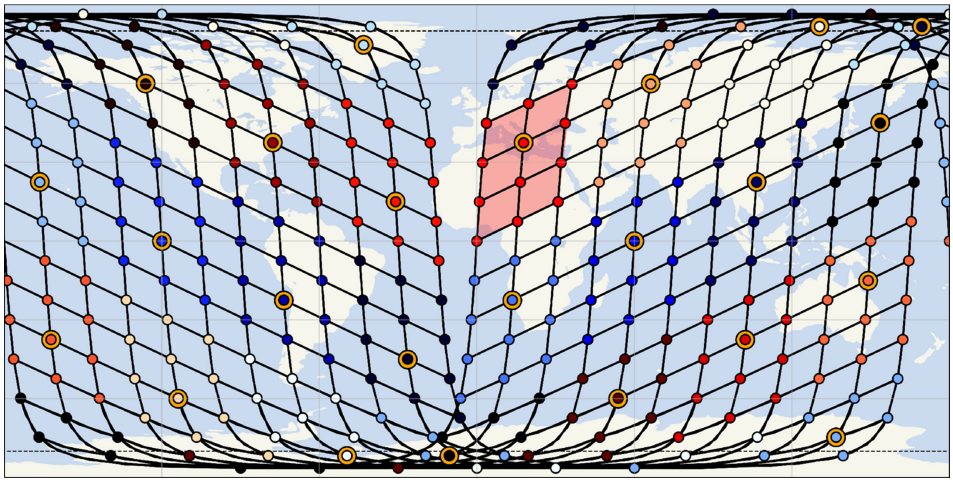

FIGURE 3 | Rectangular arrangement of clusters for SCN-288 using a cluster size of 12. Nodes of same cluster are colors identically. The region spanned by a cluster is shown (in red). Satellites with SDN controller role are highlighted (orange ring) [19].

SDN foundation of the proposed protocol allows for flexible and granular manipulation of routing tables, enhancing its adaptability and efficiency.

On the one hand, we propose a so-called "geographical routing table" which maps each geographical identifier (for each cell), with a destination node (a satellite). The geographical cells of this table should be constant. The table is used to identify whether a cell is served by the current cluster or not. In addition, it indicates over which link the cluster is to be exited. The intercluster routing utilizes a dedicated function to compute viable Next-Hop Cluster Nodes (NHCNs). These external border nodes, which connect to the given cluster but are not a part of it, are the destination nodes listed in the geographical routing table. If the packet is in the apparently correct cluster, this is indicated by a special entry. There is no global mapping of every terminal and its serving satellite.

Only local resolutions are considered to maintain minimal signaling overhead. For these mappings, cluster-specific internal routing tables are used which relate destinations to next hops. MAC addresses of GWs and UTs, which are served by the cluster, satellites within the cluster, as well as NHCNs, are mapped to suitable next hops. As this mapping only contains relevant destinations within the operational scope of a cluster, a practically manageable size is typically achieved.

An abstracted flowchart of the two-tiered routing table lookup is shown in Figure 4. As shown, after the geographical table, packets are either destined for a satellite within the cluster or an NHCN, so a satellite from a neighboring cluster. In both cases, a subsequent intrarouting table lookup is required to find the correct next hop. To maintain a coherent ESL overview within the clusters, each handover triggers a cluster-internal broadcast of a related routing table updates.

4.3 | Geographic Tiling

The geographical tiling determines the number of logical geographic areas. The length of the geographical identifier in the header of a packet has to be large enough to support this number. A high number of areas is generally preferable, as it decreases the number of potential serving satellites. However, more areas also result in longer routing tables for geographical identifiers, which in turn increases the average required lookup time. Tree-based aggregations can be used to enable faster lookups of geographical identifiers [32, 33].

For the sake of simplicity, we propose equirectangular areas of 3° latitude and 3° longitude until ±87°. At the poles, single unified cells are used. This results in 6962 areas, whose physical size varies (larger cells at equator, smaller cells at the poles). To represent this number of addresses, at least 13 bits are required for the geographical identifier. The considered tiling is shown in Figure 1 by the gray rectangular grid.

4.4 | Cluster Arrangement

Potential cluster arrangements have been mentioned in previous work [6]. The clusters should be larger than the geographical tiling in order to reduce ambiguities. For intracluster routing, a larger scope is expected to enable more efficient load-balancing. However, the principal drawback is that latency between the SDN controller and its forwarding devices increases, which negatively impacts reactivity. Signaling overhead is increased as well. To find appealing trade-offs, the upper limit of the cluster size should be determined by the maximal permissible intracluster latency. This duration determines the routing convergence within a cluster, and thus the reactivity of the protocol.

In general, to reduce the number of hops, diamond-shaped clusters are advantageous. The shape of the clusters follows the directions of the ISLs. However, due to the considered seams of the constellation, this results in irregular clusters. As we focus on identical clusters in this activity, we thus prefer a rectangular cluster shape.

For SCN-288, we assume a 4×3 arrangement resulting in 12 nodes per cluster, and 24 clusters of identical shape. Moreover, we investigate 6×4 (24 nodes per cluster) and 8×6 (48 nodes per cluster) arrangements to evaluate the described sizing trade-off. For SCN-1440, we construct clusters of a similar physical size to maintain comparable system characteristics. We analyze 30 rectangular clusters of identical shape of size 6×8, that is, 48 satellites.

4.5 | Multicast Extension

Due to the special requirements of SCNs, many traditional terrestrial multicast protocols are not well suited. A key problem is the spanning of efficient multicast trees in large-scale dynamic networks. As a light-weight extension, we propose a design based on the Protocol Independent Multicast (PIM) protocol family, which uses existing unicast routes for multicasting. As

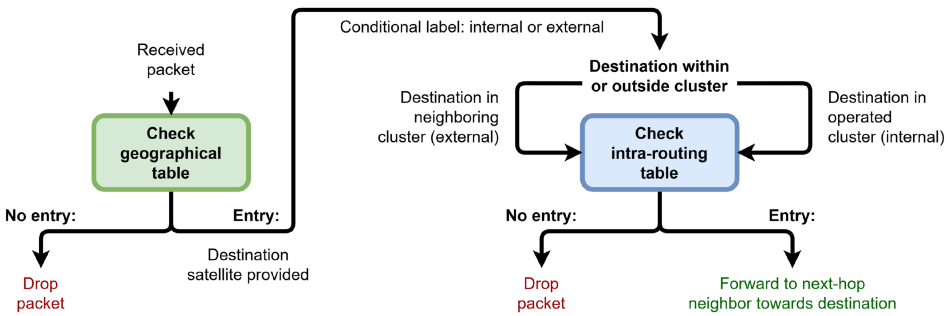

FIGURE 4 | Conceptual flowchart of packet handling: two-tiered routing table lookup. A geographical and an intrarouting table are used.

the relevant connectivity changes are considered highly predictable, a tailored PIM Sparse Mode (PIM-SM) [34] approach with timed switch-overs according to expected handovers is considered.

PIM-SM is based on Rendezvous Points (RPs), which are the roots of distribution trees for multicast groups. [34] In general, multicast traffic is forwarded from the source, the first-hop router, to the RP. If a receiving entity is interested in a multicasting service, it has to request the multicast service from the RP. To this end, the last-hop router sends an Internet Group Management Protocol (IGMP) request to the closest RP.

Based on the proposed distributed SDN architecture, designating the SDN controllers as RPs is practical. Multicast traffic is forwarded to the SDN controllers which act as RPs for their respective clusters. As the controllers distribute forwarding instructions with each network update, additional signaling required for multicasting is minimal. Given the nature of multicast traffic, we assume that it is generally widely available or cached. Thus, the source of the multicast traffic is assumed to be the GW which is closest to the SDN controller (the RP) of the cluster.

The considered multicast extension works as follows:

- UT requests to subscribe to multicast group
- Multicast session established between closest GW and interested UT
- If RP does not yet receive MC traffic: GW triggered to start transmission to RP
- If GW already sending MC traffic to RP: forwarding from RP to UT
- Forwarding from RP to UT follows unicast routing entries

With this setup, all RPs are relatively close to the receiving entities as well as the sources. Traffic on the connections between GWs and SDN controllers is minimized, there is a multicast gain on the trunk links. There are potential multicast gains via a duplication of packets at the SDN controller and at the egress satellite. A diagram of the intended functionality is shown in Figure 5.

5 | Routing Schemes

In order to maximize the performance of IDLB, we propose specifically tailored path computation and routing algorithms complementing the considered design. While multiple suitable options can be considered, we focus on promising low-complexity solutions which have been derived and implemented for the reference scenarios. In any case, the softwarization of SDN allows to change routing algorithms or policies by adjusting the routing logic of the controller.

5.1 | Intracluster Routing

The central goal of the intracluster routing algorithm is to balance the loads while complying with the QoS requirements of the traffic. Within a cluster, all relevant network state information is aggregated at the SDN controller. Therefore, load-aware proactive routing strategies are possible.

5.1.1 | Load-Balancing

To enable high system throughput within SCNs, an informed decision-making about which paths are best-suited to support incoming traffic is necessary. To achieve optimal performance, all of the resulting routes have to be chosen in a way which minimizes congestion. This optimization can be formulated by the Multi-Commodity Flow Problem (MCFP) [35]. It describes the problem of determining an optimal routing configuration for a given set of commodities with different source-destination pairs and an associated demand. In this context, commodities represent the different, individual data transmissions between UTs or between UTs and GWs (or vice versa). Ideally, the controller solves this underlying MCFP to distribute low-priority traffic optimally.

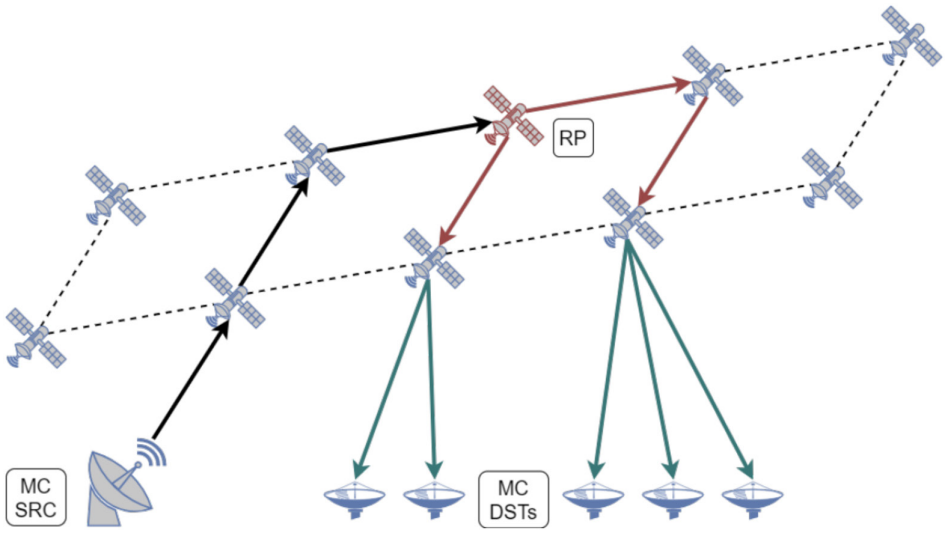

FIGURE 5 | Schematic diagram of proposed multicast extension. Nearest GW provides MC traffic to RP via trunk links (in black). RP is root of distribution tree (in red), forwarding to service nodes, which then distribute MC traffic to subscribed UTs in the downlink (in green).

The time-dependent topology of SCNs can be described by a dynamic graph model G_t, consisting of set of vertices V_t and edges E_t: $G_t = (V_t, E_t)$. Between nodes i and j, we formulate the time-dependent link utilization $u_t(i,j)$ based on the link capacity $c_t(i,j)$ as

$$u_t(i,j) = \frac{f_t(i,j)}{c_t(i,j)} = \frac{\sum_{(i,j)\in E} f_t^{\kappa}(i,j)}{c_t(i,j)} \tag{1}$$

Here, $f_t(i,j)$ describes the current aggregated flow, so the sum of all individual flows $f_t^{\kappa}(i,j)$ on this link. In order to avoid saturating links, the load-balancing objective can thus be formulated as minimizing the maximum link utilization over time. So, over all time steps the following objective function is formulated:

$$\min \max_{t} \max_{(i,j)\in E} u_t(i,j) \tag{2}$$

To highlight why formulating a solution to this problem is intractable, we formulate a corresponding value function $\mathscr{V}_t(s_t)$ for a state s_t. Besides the cost C_t of the current network state s_t, and the flow requests f_t, other aspects are relevant. As propagation delays are particularly significant in SCNs, a corresponding function $\tau(i,j,u_t(i,j))$ is considered to capture their impact. This delay function introduces nonlinearities, creating a nonlinear dynamic optimization problem. In reality, this function is also impacted by other delays for queuing and switching. Moreover, we assume *stochastic traffic models*. Therefore, future states are probabilistic in nature. An analytical formulation of suitable traffic models is exceedingly difficult. Therefore, the estimation of future states may be quite imprecise, limiting the effectiveness of approximate solutions. We use ω_t to denote these incoming random traffic events. The value function of a current state s_t thus becomes:

$$\mathscr{V}_t(s_t) = \min_{f_t} \left[C_t(s_t, f_t) + \mathbb{E}_{\omega_t} \left[\mathscr{V}_{t+1}(s_{t+1}) | s_t, f_t \right] \right] \tag{3}$$

This combination of time-dependency, nonlinearity, stochasticity, and dynamic topology represents a complex problem to solve formally. Similar problems have been determined to be unsolvable using polynomial-time algorithms [35]. The amount of parameters and their interdependencies result in a significantly large state space. The state transition function highlights this complex interplay between observed network states, control actions, delays, random events, and topology changes (represented by E_t):

$$s_{t+1} = \Phi(s_t, f_t, \tau_t, \omega_t, E_t) \tag{4}$$

Consequently, approximate solutions based on heuristics hold more promise for in-orbit application. For instance, heuristic assumptions for LEO SCNs have been formulated [14]. Due to the inherent complexity and on-board limitations of satellites, we focus on a low-complexity solution in this work. So, instead of approaching the MCFP directly, we propose a snapshot-based proactive path-finding algorithm aimed at the current cost function.

5.1.2 | Dynamic Best-of-k Algorithm

To maintain low-complexity for varied cluster sizes, we propose a dynamic best-of-k paths algorithm for intracluster routing. The approach follows similar ideas as a previously proposed top-k-paths algorithm based on SDN [36], but focuses on different metrics and objectives. By limiting our search to k suitable paths and utilizing a parameterized utility function, we achieve improved performance while reducing computational complexity by orders of magnitude. This efficiency is crucial for real-time decision making in space-based networks where computational resources are limited and latency requirements are strict.

As we rely on small values for k, the approach scales gracefully even for large cluster and constellation sizes. The approach enables QoS-aware forwarding by using a parameterized utility function which prioritizes relevant path characteristics. The proposed SDN architecture complements this approach, as it enables centralized policy definition.

A set-based approach is important, as traffic spikes may result in pressure on currently low-cost links. During a link load update period, the controller may be unaware of potential changes. If a high number of requests occur in this time frame, the routing logic has to account for the new demands and their interaction. When using sets of viable paths, choosing the best option decreases variance in path quality [37]. If paths have equal cost, we apply a tie-breaking mechanism similar to Always-Go-Left [38]. A nonuniform choice combined with such an asymmetry has been shown to improve load balancing [38].

For the different QoS classes, we consider the following parameterized utility functions $Y_{\{1,2,3\}}$ for a path p:

$$Y_1(p) = (L(p)+\epsilon)^{-1} \tag{5}$$

$$Y_2(p) = \psi_2 \cdot (N_{hops}(p)+\epsilon)^{-1} + (1-\psi_2) \cdot (U(p)+\epsilon)^{-1} \tag{6}$$

$$Y_3(p) = \psi_3 \cdot (N_{hops}(p)+\epsilon)^{-1} + (1-\psi_3) \cdot (U(p)+\epsilon)^{-1} \tag{7}$$

In these formulations, ϵ represents a small positive value to avoid division by 0. ψ_2 and ψ_3 are weighting factors for the link load. The values are chosen to comply with the QoS requirements and the system scenario. Path latency $L(p)$ is only used for the delay-critical class QoS1. The other classes focus on the number of hops N_{hops} and path utilization $U(p)$.

$$U(p) = \sum_{(i,j)\in p} u_t(i,j) \tag{8}$$

The approach corresponds to a dynamic k-shortest-path algorithm with a subsequent comparison step [39, 40]. The complexity in time for the path computation for all flows F is thus:

$$\mathcal{T}_{\text{intra, path}} = \mathcal{O}(|F| \cdot (|E| + |V| \log |V| + k)) \tag{9}$$

We compare paths according to the utility function of the respective QoS class. This additional step has the following complexity in time:

$$\mathcal{T}_{\text{intra, decision}} = \mathcal{O}(|F| \cdot k) \tag{10}$$

Tracking all possible paths can lead to exponential growth in space depending on the connectivity of the graph. For an $E \times E$ grid, which the considered clusters typically represent, a complexity in space of $\mathcal{O}(|V|^2)$ is expected. By limiting the choice to only a handful of paths, and not tracking others, the general complexity in space is reduced to

$$\mathcal{S}_{\text{intra}} = \mathcal{O}(|V| \cdot k) \tag{11}$$

These formulations assume the worst case, where every considered path has a length of at most $|V|$ nodes (a path visits every node of the sub-network). The same approach is applied by the God's Eye View routing benchmark, presented in Section 7.1, due to the prohibitive space complexity of exhaustive path evaluation.

5.1.3 | Per-Packet Considerations

In addition to per-flow operations, per-packet forwarding is also a potentially useful approach in the given context. Stateful information about flows is not required, which can lead to a less complex routing logic.

So packets are forwarded based on generic routing instructions independent of their flow (including details about source or demand). If a load-balancing update is triggered, all packets toward a certain destination adjust their path accordingly. Because high-priority traffic is not load-balanced, this poses no problem in terms of jitter for delay-critical traffic. Nevertheless, periodic updates may lead to path instability for load-balanced traffic. Link load hysteresis and thresholds have to be implemented to mitigate potential route flapping.

Given these considerations, flow-based routing is expected to enable improved decision-making regarding network congestion compared with per-packet forwarding [36]. Effective load-balancing approximating the MCFP becomes challenging, due to the lack of distinction between flows. This assumption is evaluated in Section 7.4.1, where the QoS compliance under high network loads is compared. To distinguish per-packet IDLB, it is called *P-IDLB* in the following. For clarity, per-flow IDLB is called *F-IDLB*.

5.2 | Intercluster Routing

For intercluster routing we primarily focus on limiting signaling overhead, while avoiding potential instabilities and congestion. Because geographical routing approaches for SCNs have the ability to strike attractive trade-offs w.r.t. signaling [16], a related approach is considered.

Based on the proximity of clusters to the geographical areas, the destination cluster is determined. Accordingly, an Shortest Path Tree (SPT) is computed on a cluster-level providing all viable neighboring clusters. The key problem is now to identify on which specific link to exit the current one. The cluster-external node which is reached by this link, the NHCN, is used as the destination in the geographical routing table. The actual serving satellite is resolved in the destination cluster.

The NHCN choice is QoS-dependent, similar to intracluster routing. Delay-critical traffic should follow the shortest path, while any low-load link is valid for best effort traffic. Notably, the choice should be randomized for low-priority flows to mitigate bottlenecks. Algorithm 1 represents such an NHCN function.

For smaller cluster sizes, topological scenarios arise in which few viable links to leave the cluster may be available. For instance,

ALGORITHM 1 | IDLB intercluster routing: Finding next-hop-cluster nodes.

Algorithm 1 IDLB inter-cluster routing: finding next-hop-cluster nodes

```
Require: QoS class, destination cell, destination cluster
Ensure: Destination cluster is not current cluster
next_clusters ← current_cluster.next_hops(dst_cluster)
best_value ← max_float
best_nhcn ← null
candidates ← list()
for each node nhcn ∈ current_cluster.nhcns() do
    ngbr ← get_neighbor_in_cluster(nhcn, current_cluster)
    if (nhcn ∉ next_clusters) ∨ isl_shutdown(ngbr, nhcn) ∨
  isl_load_threshold(ngbr, nhcn, qos) then
        continue                          ▷ Skip unsuited options
    end if
    dist ← distance(nhcn, dst_cell)
    link_load ← isl_link_load(ngbr, nhcn)
    value ← cost(dist, link_load, qos)
    if value < best_value then
        best_value ← value
        best_nhcn ← nhcn
    end if
    if to_randomize(qos) then
        candidates.add(nhcn)        ▷ Add to candidates list
    end if
end for
if best_nhcn = null then
    return null        ▷ No suitable next-hop cluster node
end if
if to_randomize(qos) then
    return sample(candidates)
else
    return best_nhcn
end if
```

near the poles, because of ISL shutdowns, bursty traffic spikes may result in saturated links to the next cluster. Naturally, forgoing end-to-end route estimations results in some zig-zagging between clusters. Nevertheless, the resulting end-to-end latency is acceptable for the given QoS classes.

The complexity of the intercluster routing procedures is dominated by the cluster-based path selection and the NHCN choice. In both cases, the set size is relatively small. For SCN-288, there are at most 24 individual clusters. The complexity in time for a per-flow approach with $|F|$ flows, and $|\mathscr{C}|$ clusters (and $|\mathscr{E}|$ virtual links between them) is

$$\mathcal{T}_{\text{inter}} = \mathcal{O}\big(|F| \cdot \big(|\mathcal{E}| + |\mathcal{C}| \log |\mathcal{C}| + N_{\text{NHCN}}\big)\big) \tag{12}$$

The complexity in space linearly depends on the number of clusters and NHCNs. As we also have a candidate list, space required of size $|N_{\text{NHCN}}|$ is counted twice. Depending on the cluster size, $|\mathscr{C}|$ or $|N_{\text{NHCN}}|$ is larger.

$$\mathcal{S}_{\text{inter}} = \mathcal{O}(|\mathscr{C}|) + \mathcal{O}(|N_{\text{NHCN}}|) + \mathcal{O}(|N_{\text{NHCN}}|) = \mathcal{O}(|N_{\text{NHCN}}|) \tag{13}$$

Overall, the resulting complexity in time and space is considered feasible for on-board processors of next-generation satellite systems.

5.2.1 | Per-Flow Considerations

For the NHCN choice, per-flow and per-area decisions are possible. The former results in an NHCN computation for every flow. The latter periodically decides on suitable NHCNs for every geographical cell.

Due to the considered scenario, the per-area approach is expected to result in worse performance. While 6962 areas are used, the ones containing GWs or multiple UTs are the focus of a significant traffic share. Consequently, determining a single NHCN for all flows with the same destination area can result in bottlenecks. The assumption is investigated in Section 7.4.2. In a per-flow approach, this issue is avoided by selecting from a set of valid NHCNs, even for the same destination.

6 | System-Level Simulator

6.1 | Simulator Design

The utilized network simulator was extended and adapted specifically for this investigation [23]. It is programmed in C++ and Python, building upon software and tools used in previous research [6, 9, 16]. The simulator is designed to efficiently simulate routing in SCNs with ISLs on a packet level. Faster than real-time simulations are possible for a variety of scenarios, including complex satellite constellations with thousands satellites and of terminals on ground. In addition to the discrete event-based network simulator written in C++, there is a suite of Python pre- and post-processing scripts. These generate simulation input files, and visualize the intended system as well as the simulation results. Relevant performance metrics, for example, link utilization and packet drops, are written into output files during a simulation run. Satellite constellations, UT and GW distributions, as seen in Figure 1, are thus prepared as inputs.

The simulator generates C++ objects for the satellite nodes, user nodes (i.e., the terminals), and gateway nodes according to configurable parameters. For different scenarios, dedicated configuration XML files can be created or edited to simulate with the desired parameters. The movement of the constellation, as well as the creation, handling, and forwarding of packets is realized by timed events. Using recursive functions, it is possible to include periodic events in the time-ordered event loop. This allows for the tracking of packet objects propagating and queuing on nodes, while measuring metrics such as end-to-end latency.

The generated traffic is based on sessions. A session represents the transmission between two terminals on ground, that is, UTs and GWs. While the simulator includes both on-off as well as constant bit rate sessions, we focus on the latter for more coherent results in this investigation. Overall a session is defined by a unique identifier, its start time, its duration, the involved terminals, and its QoS class (determining its priority). The starting time is sampled uniformly across the simulated time. An exponential distribution is used for the end of the session with the defined average session duration. The priority of sessions is distributed to reflect the considered traffic shares. The sets of UTs and GWs are sampled accordingly. The parameters determining this sampling are based on the inputs defined in Section 3.3. Using this format, a diverse traffic model emerges with temporary spikes, and potential geographic hot spots depending on the terminal distribution on ground. Nevertheless, over longer simulated time periods, which significantly exceed the average session duration, the overall network load does not vary too strongly. So, comparable traffic requirements are maintained for different simulation runs.

6.2 | Key Performance Indicators

We consider several core Key Performance Indicators (KPIs) for the performance evaluation. The relevant metrics are listed in Table 3. The KPIs are based on the defined QoS requirements (as described in Table 4).

Because fast responsiveness is a key argument for space-borne routing, we include routing convergence as a core parameter. Due to the physical size of the network, the resulting end-to-end latency can be larger than 130 ms for SCN-288 (will be shown in Section 7). Therefore, we assume that the average delay of any signaling packet within a cluster and the installation of its instructions should not exceed this latency.

For the signaling overhead, we assume that the ratio of resulting control plane packets to all forwarded packets (so control and data plane) should not exceed 5%. Naturally, this metric depends on the applied traffic volume. If there is little to no data transmitted, it will be outweighed by periodic updates of control plane traffic. As we evaluate this metric for average network loads exceeding 6.9 Gbps, a meaningful statement is expected.

TABLE 3 | Relevant key performance indicators for the performance evaluation.

Indicator	Unit	Comment	Requirements
End-to-end latency	ms	Delay between sending on uplink ESL to reception after downlink ESL.	See Table 4
Jitter	ms	Delay variation between two subsequent packets.	See Table 4
Packet dropping ratio	ratio	Ratio of dropped packets relative to successfully forwarded packets.	See Table 4
Routing convergence	ms	Delay of forwarding and installation of routing instructions.	< 130 ms
Control plane signaling	ratio	Signaling overhead of control plane traffic.	< 5%

TABLE 4 | Summary of considered QoS classes.

QoS class	Priority	Packet delay budget	Packet loss rate	Delay variation	Example services	Relative traffic share
0: Signaling	4	<150 ms	$< 10^{-6}$	< 30 ms	Control plane signaling.	N/A
1: Delay-critical forwarding	3	150 ms [27–29]	10^{-2} [28, 29]	< 30 ms [28]	Conversational voice, real-time applications.	10%
2: Delay-tolerant forwarding	2	200 ms	10^{-4} [28, 29]	< 50 ms [28]	Nonconversational video (e.g., live-streams).	34%
3: Best effort	1	300 ms [28, 29]	10^{-6} [28]	N/A	Video (e.g., buffered streaming), file sharing, mail, web activity, multicast.	56%

Note: Some values from standards have been slightly adapted based on observations and previous investigations into SCNs [2, 9].

7 | Performance Evaluation

7.1 | Benchmarks

As an upper benchmark, we consider *God's Eye View Routing (GEVR)*. The approach is based on instantaneous perfect knowledge of all network states everywhere, while exploiting a set of suitable paths. There is no signaling traffic or delay. Naturally, such an algorithm cannot be implemented in the real world. Due to the inherent structure of the network, there is a diverse set of paths satisfying the QoS requirements for most ingress-egress pairs. To reduce the complexity in space, not all possible paths are tracked by the algorithm described in 5.1.2.

For a state-of-the-art benchmark solution, a *dynamic source-routed scheme* is used. It is based on Dijkstra's algorithm to provide the fastest routes in noncongested networks. As link load information is not shared, it is prone to packet drops in more demanding scenarios. It serves as a lower benchmark to quantify the advantages of load-balancing approaches in terms of latency and dropping rate. Comparisons with other state-of-the-art approaches can be difficult, as the routing schemes require significant design and implementation efforts. Given the packet-based nature of the simulator, implementing diverse protocols is rather complex. Moreover, terrestrial state-of-the-art protocols would also require significant adjustments to work seamlessly for the considered satellite networks scenario (e.g., rapid topology changes, lack of static backbone links as described in Section 2). Thus, we focused on source routing, which represents a common baseline and mirrors the deterministic path computation used in many existing systems. This choice aligns with our goal to isolate the impact of load-balancing and QoS-aware routing.

7.2 | Protocol Performance

Firstly, we want to establish that the protocol design results in correct routing and forwarding behavior. So, we investigate core KPIs of IDLB and compare the results with the GEVR benchmark solution. Overall, the protocol typically complies with QoS requirements. We focus on latency, jitter, and later packet dropping characteristics. We analyze the results of all sessions, which includes the sampled UT-to-UT transmissions as well as UT-to-GW and vice versa.

The observed end-to-end latency characteristics are primarily due to the varied geographical distances between terminals. As the underlying terminal distribution is based on population density, it is difficult to formulate traffic models mathematically which can precisely estimate the resulting delays. To formally describe latency, we denote the time-dependent per-hop propagation delay by $l_{p,t}$, and the queuing delay by $l_{q,t}$. Given the chosen buffer sizes, the queuing delays was observed to be in the domain of milliseconds. We also include the per-hop routing delay l_r, which represents the time required to determine the next hop of a packet, and the switching delay l_s, which is the typical time required to switch a packet from the input buffer to the corresponding output buffer. Therefore, the end-to-end latency $L_t(p)$ of path p can be described by (observed time domain below):

$$L_t(p) = \sum_{(i,j)\in p} \left[\mu s] \underbrace{l_s(i)} + \mu s] \underbrace{l_r(i)} + \mu s] \underbrace{l_{q,t}(i)} + ms] \underbrace{l_{p,t}(i,j)} \right] \tag{14}$$

End-to-end latency represents a fundamental drawback of the independent cluster design. While shortest paths are possible, we expect additional delay on average. Due to the independent load-balancing of clusters and the geographical intercluster routing, some zig-zagging is possible. Nevertheless, the comparison of end-to-end latency in Figure 6 shows that most sessions comply with the QoS requirements. For QoS class 1 and 2, there are less than 1% of sessions exceeding the maximum delay (150 ms for QoS1 and 200 ms for QoS2). Moreover, the QoS-based forwarding is illustrated, as best effort traffic of QoS3 tends to follow longer paths, but still within the defined limit of 300 ms. The bars at higher latencies (>150 ms) are a clear indication that network resources outside of the shortest paths are used. For GEVR, this pattern is even more succinct: there is are clear latency distributions ending at the maximum end-to-end latency of each QoS class. For QoS class 3, in particular, longer end-to-end paths are visible. As the path computation of GEVR is able to evaluate candidate paths on an end-to-end basis, any path which fulfills the latency requirement of the QoS class is an option. Thus, the approach can distribute best effort sessions among paths with low congestion.

We also consider the maximum instantaneous jitter J_{inst}, which describes the difference in interarrival time between subsequent packets. In the given context, this is the maximum latency variation of subsequent packets, so the difference between $L_{\rho-1}$ and L_ρ for the M packets of a session:

$$J_{\text{inst, max}} = \max_{\rho \in \{2,\ldots,M\}} \left| L_\rho - L_{\rho-1} \right| \tag{15}$$

Based on this formulation, it is apparent that the largest delay variances occur when paths are changed, for example, by rerouting, resulting in different propagation delays (representing the largest contributor to latency).

A comparison between the F-IDLB approach with a cluster size of 48 and GEVR is provided in Figure 7. Because a resulting path of IDLB is under the control of multiple control entities, more frequent adjustments are expected. Cell handovers between clusters may result in path changes which are not considered by end-to-end schemes such as GEVR. Nevertheless, the impact is well within the requirements. For QoS class 1, the average jitter

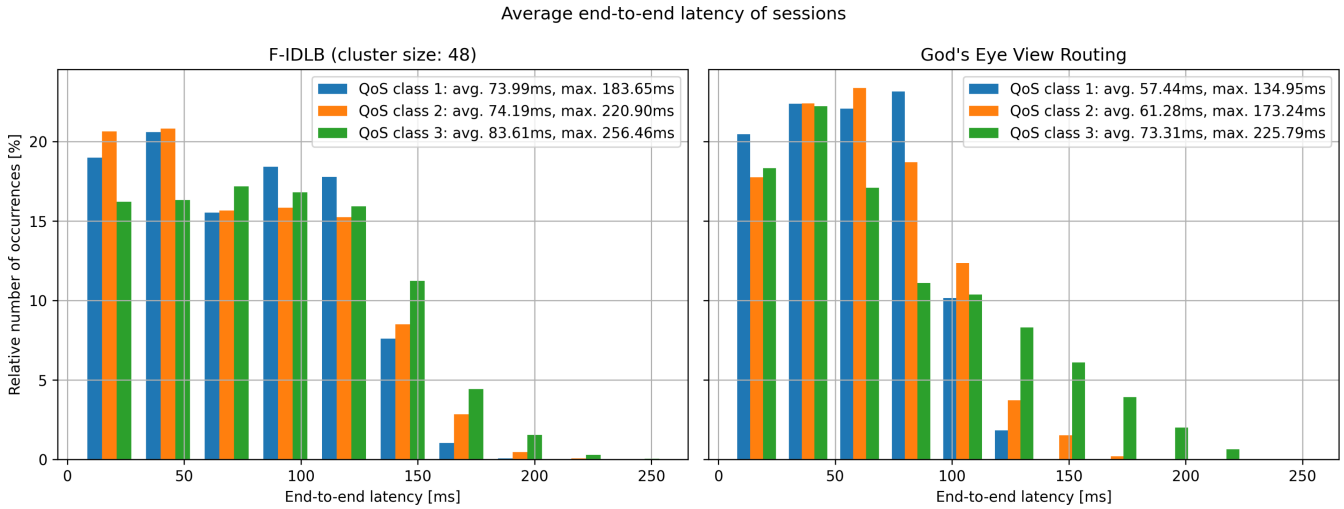

FIGURE 6 | End-to-end latencies of sessions: SCN-288 wideband scenario.

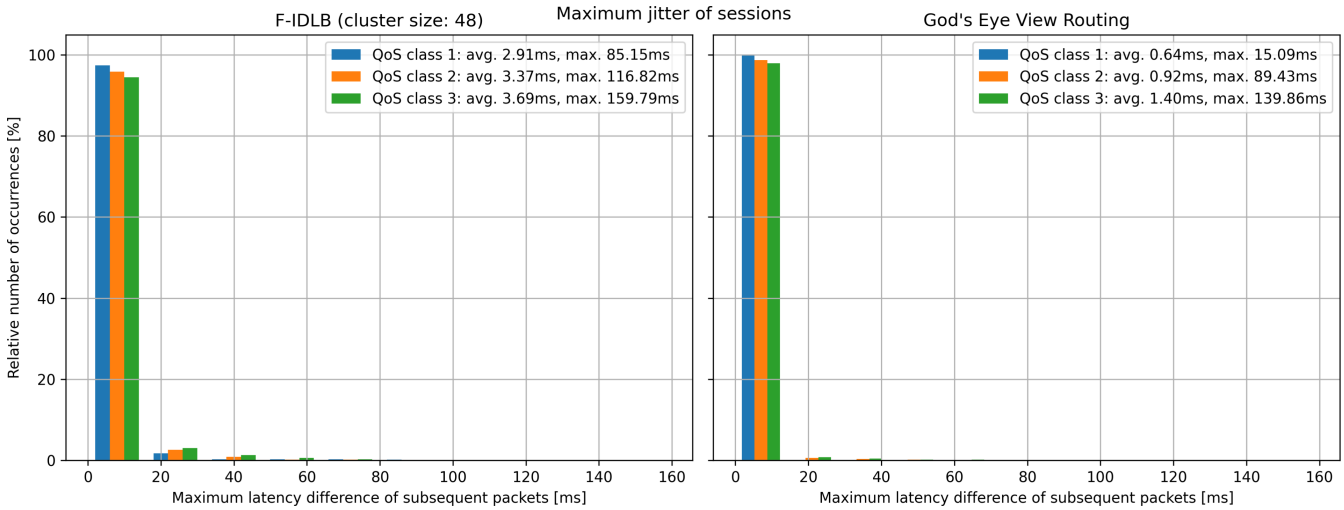

FIGURE 7 | Delay variance of sessions: SCN-288 wideband scenario.

is 2.91 ms, and the maximum observed 85.15 ms. While this is significantly larger than the GEVR counter-part (on average 0.64 ms, maximum 15.09 ms), these variances are acceptable in the reference scenario. It is important to note that even when GEVR is used, high jitter occurs for QoS class 3. The cause are handover events at the seams which coincide with path adjustments due to high link loads. As the routing of this best effort traffic considers paths with valid end-to-end latency, but does not rank them, paths of varying length occur.

It is important to note that signaling traffic has been omitted from this analysis given that it does not represent data plane traffic. Signaling does not follow end-to-end paths from and to terminals on ground, thus its latency is not comparable to data plane traffic. As we consider individual signaling messages and not flows, there is no jitter. In general, signaling packets follow the same paths which have been computed for QoS class 1. However, the packets are scheduled using a dedicated signaling buffer which prevents packet losses for the encountered network update frequencies. The latency characteristics correspond to the convergence delays shown in Table 5.

7.3 | Increasing Network Loads

To evaluate the load-balancing capacities of the proposed protocol, we investigate its behavior when encountering high traffic volumes. So, in this comparison, we test the resilience of the approaches in terms of QoS compliance to increasingly high network loads. In this context, the increase in packet drops represents the main issue. In order to maximize throughput, the protocol has to balance the elevated traffic load to maintain QoS compliance. We assume that a QoS compliance of >95% is considered acceptable.

As we investigate a large set of simulations, we consider a shorter simulation window with the same average network load is used. Here, network load is defined as the aggregated data entering the SCN. So, every packet generated at a UT or GW and sent up to a connected satellite is counted in a sliding window. This constellation-wide uplink of packets to the constellation in terms of amount of data is then divided by the duration of the observation window. For a given window size T_{window}, using N_{uplink} to describe all the packets which were sent up and $\zeta(\rho_i)$ to denote the size of the i-th packet entering the SCN, the network load at t is thus:

$$\text{Network load}(t) = \frac{1}{T_{\text{window}}} \sum_{i}^{N_{\text{uplink}}(t)} \zeta(\rho_i) \tag{16}$$

While it is possible that packets have different sizes, in the presented simulations we assumed a general packet size of 12 kbit. Analogously, the system throughput describes the packets sent down from the SCN in the current window. The resulting system load and system throughput for the link load-agnostic source-routing approach is shown in Figure 8. Due to the ramp up and down of the traffic models, higher peak loads occur in the shorter simulation window. So, protocol performance may be even slightly better when evaluating longer windows. Nevertheless, we focus on the average network load and system throughput for this comparison. Notably, the

TABLE 5 | Comparison of convergence rate and overall packet dropping rates for different cluster sizes in SCN-288.

SCN-288 cluster size	Control plane signaling share	Convergence delay [ms]: average, maximum		Packet dropping rates: QoS 1, QoS 2, QoS 3		
12 (arranged 4 × 3)	0.062%	56.968	104.782	1.044×10^{-4}	3.571×10^{-4}	3.374×10^{-4}
24 (arranged 6 × 4)	0.112%	117.338	285.276	2.948×10^{-6}	7.050×10^{-5}	1.344×10^{-4}
48 (arranged 8 × 6)	0.208%	273.526	747.552	8.942×10^{-7}	2.557×10^{-6}	2.285×10^{-6}

Note: Network load of 13.2 Gbps, which corresponds to 9500 sessions of an average duration of 100 s.

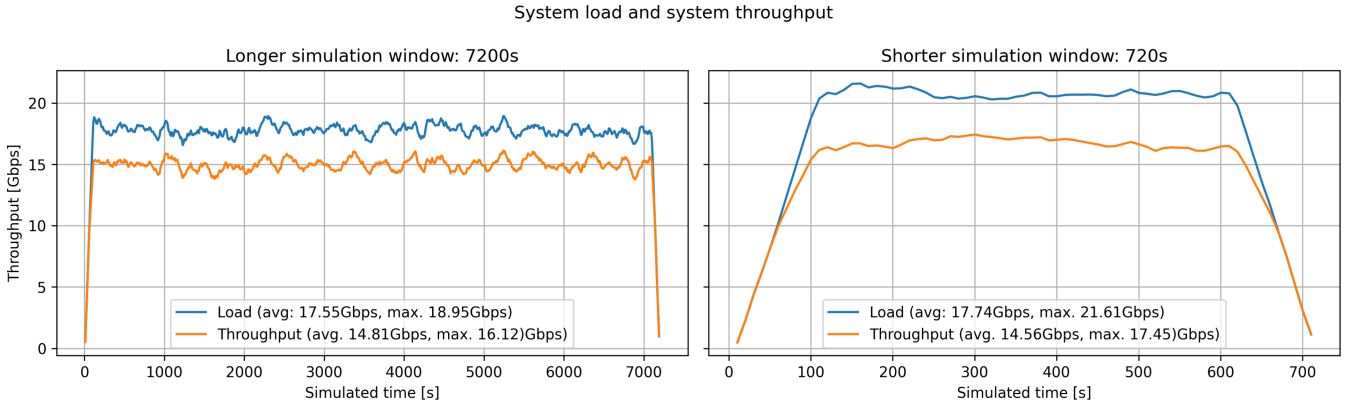

FIGURE 8 | Network load and system throughput for source-routing in SCN-288. Longer window with a simulated time of 7200 s, and shorter window with simulated time of 720 s are shown. Due to uniform session distribution and margins dictating simulation start/end times, shorter window results in slightly higher loads.

throughput is significantly lower in Figure 8, hinting at packet losses for the used source-routed approach.

The comparison in Figure 9 shows that, in terms of QoS compliance for increasing network load, the proposed approach (F-IDLB) is able to outperform source routing. Looking at the overall QoS comparison, source-routing falls below the 95% compliance threshold at around 7.6 Gbps in this scenario. Using IDLB with 12-node clusters, enables such a compliance up to 12.5 Gbps, an increase of 64.5%. Larger cluster sizes improve load-balancing further. At network loads of approximately 15.0 Gbps and 16.2 Gbps, the 24-node and 48-node cluster approaches fall below the threshold respectively. Compared with source-routing, this corresponds to improvements of 97.4% and 113.2%, respectively.

These results highlight the load-balancing capabilities of IDLB for QoS-based routing. As all of the routing and path finding logic is within the controller software, it is possible to adjust or change intracluster and intercluster algorithms by providing software updates to the controller logic. Therefore, solutions suitable to the system specifics and processing power of the satellites can be applied to IDLB. A further analysis of relevant design aspects is provided in the following.

7.3.1 | Cluster Size Comparison

As expected, increasingly large cluster sizes result in improved load-balancing, as longer end-to-end paths in a larger scope are possible. However, there appear to be diminishing returns, as illustrated in Figure 9. Further KPIs for the different cluster sizes, under a network load of 13.2 Gbps, are summarized in Table 5.

The observed signaling overheads are significantly below the required ratio of 5% for the given traffic load. As expected, there is a clear increase in the relative share of signaling packets for larger clusters. Most notably, Table 5 shows a significant increase in convergence delay. This metric is based on the latency of signaling packets within a cluster. For every packet its creation time at the source and moment of arrival at the destination are tracked by the simulator. Importantly, the delay does not solely represent signaling to and from the controller. For instance, for ESL handovers, updates have to be forwarded to all other nodes of the cluster to adapt their intrarouting tables accordingly. As this can include two satellites at opposite ends of a cluster, the resulting convergence latency can thus be longer. As the physical distances in large clusters introduce high propagation delays, the average increases from 56.968 ms for the 12-node cluster to around 273.526 ms for the 48-node setup.

For the given working point, the difference between approaches in terms of packet dropping ratios is significant. While the protocol is still able to comply with QoS requirements for most sessions, but there are more apparent bottlenecks resulting in increased dropping rates when using smaller clusters. For QoS3, a rate of 1.344×10^{-4} occurs in the 24-node cluster, while 2.285×10^{-6} is observed for the largest one. As the simulation was configured to avoid packet drops for all QoS classes, there are similar dropping rates between them. Rare packet drops were occasionally observed in certain handover edge cases. While recorded separately, these events are included in the reported dropping rates to ensure completeness. Nevertheless, their influence on the overall results is considered negligible, with the maximum relative share being less than 5% of observed drops.

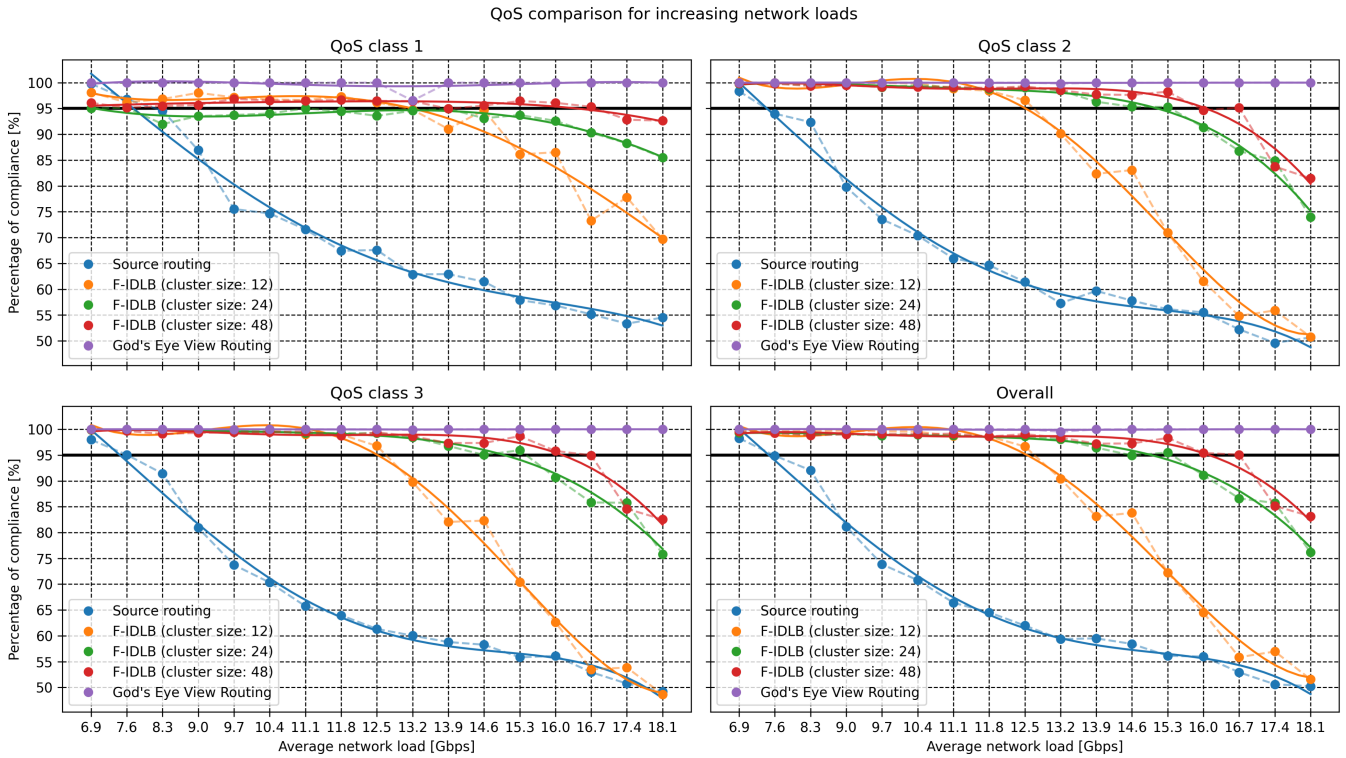

FIGURE 9 | QoS comparisons for increasing network loads for IDLB with a cluster size of 24, source-routing, and GEVR. Individual sub-plots for each QoS class: delay-critical forwarding (QoS1), forwarding (QoS2), best effort (QoS3). Source-routing approach is not QoS-based.

Overall, the comparison illustrates the trade-offs for cluster sizes. In the 12-node cluster, load-balancing was limited. With larger cluster sizes, traffic was distributed more effectively and higher network loads supported. However, given the design of the approach, there were diminishing returns. The 48-node cluster only outperformed the 24-node cluster by 12% more load supported (with > 95% compliance). Moreover, significant increases in maximum routing convergence delay are observed. As improved reactivity represents a core feature of the space-borne design, overly large cluster may not be of interest. Given this trade-off, the proposed protocol is considered particularly valuable for larger constellations.

7.4 | Design Aspects

There are various design aspects which have been evaluated in the context of this investigation. We focus on the most relevant aspects which have been discussed in the description of the design of the IDLB protocol. These include a comparison of per-packet and per-flow intracluster routing as well as an evaluation of per-area and per-flow NHCN choice.

7.4.1 | Per-Packet Versus Per-Flow IDLB

To quantify the viability of a per-packet approach, it is compared with the per-flow version. The results are shown in Figure 10. The P-IDLB approach is clearly outperformed by its F-IDLB counterpart. The former crosses the 95% compliance threshold at around 8.6 Gbps, which is better than source-routing. Additionally, performance degrades significantly. F-IDLB represents an improvement of 45.3% in terms of supported network load. This results clearly demonstrates the advantage of using flow-based SDN strategies.

As described in Section 5.1.3, P-IDLB sets nonflow-specific routing table entries. Therefore, it is more likely to the create bottlenecks for flows with similar destinations. In-depth analyses show that this traffic pattern is actually prevalent in some clusters, resulting in packet drops.

Notably, the choice of NHCN is also limited by this design. One specific NHCN is chosen for each destination cell, which can lead to congestion at adjacent links. The impact of generic and flow-based NHCN choice is investigated in the following section.

7.4.2 | Intercluster Routing: NHCN Choice

As described in Section 5.2, a specific NHCN mapping to each destination cell is expected to cause bottlenecks. Thus, we compare this design with a per-flow NHCN choice. The results are shown in Figure 11. As expected, the performance of per-area choice is significantly worse and varies much more at higher network loads. An analysis of the results showed significant drops occurring at the borders of the clusters.

It is important to note, that the shown 48-node cluster approach with the per-area mapping also shows buffer limitations. Using the same small buffer size of 360 kbit caused issues for signaling in larger clusters. When specific instructions are forwarded to 47 different nodes, a larger buffer or careful timing is required to avoid overflows. Signaling traffic is prioritized by the scheduler, so packets of other QoS classes may be dropped. Figure 11 illustrates this systematic error consisting of periodic drops due to peaks in high-priority signaling traffic. Consequently, this faulty approach rarely achieves more than 97% QoS compliance. For better results, a buffer size of 1.08 kbit was used for all other simulations using 48-node clusters.

7.5 | SCN-1440 Comparison

While this study focuses on a 288-node satellite constellation, we include a comparison with the larger reference system consisting of 1440 satellites. The same configurations are used. However, for this constellation, a cluster of similar size to the 12-node cluster now consists of 48 nodes. Therefore, comparable intracluster propagation delays are expected. Accordingly, the observed maximum routing convergence is 222.165 ms. The comparison of QoS compliance for increasing network loads is

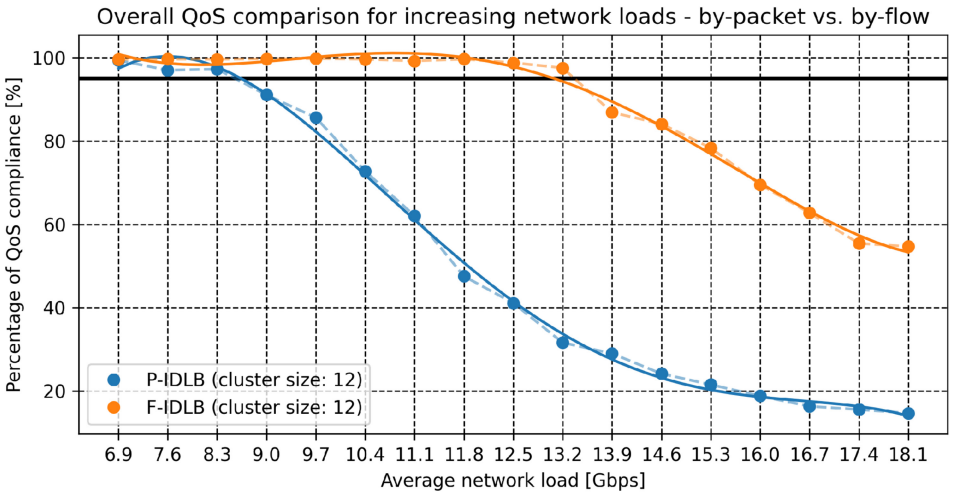

FIGURE 10 | QoS comparisons for increasing network loads for per-packet IDLB (P-IDLB in blue) with a cluster size of 12, and per-flow IDLB (F-IDLB in orange) with a cluster size of 12.

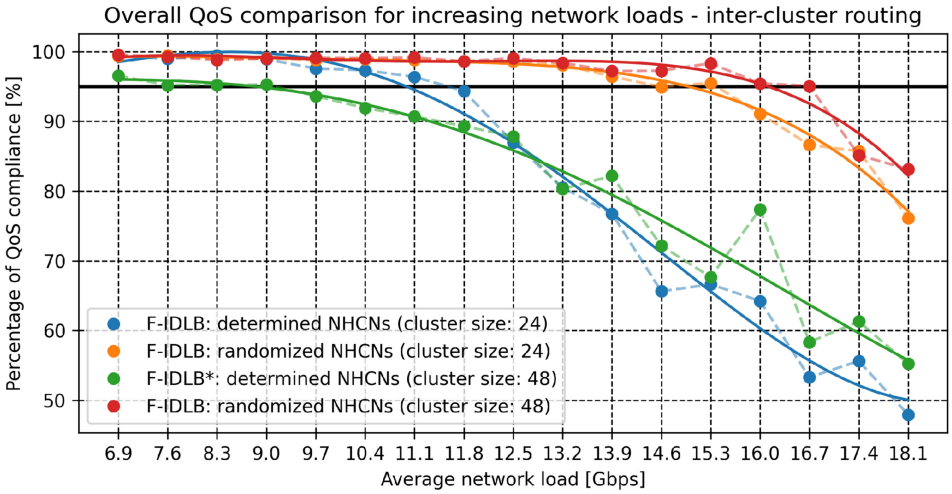

FIGURE 11 | QoS comparisons for increasing network loads for a per-area determined NHCN mapping with a cluster sizes of 24 (in blue) and of 48 (in green), as well as randomized per-flow NHCN mapping with a cluster size of 24 (in orange) and of 48 (in red). The determined mapping with a cluster size of 48 also shows buffer limitations (F-IDLB* in green).

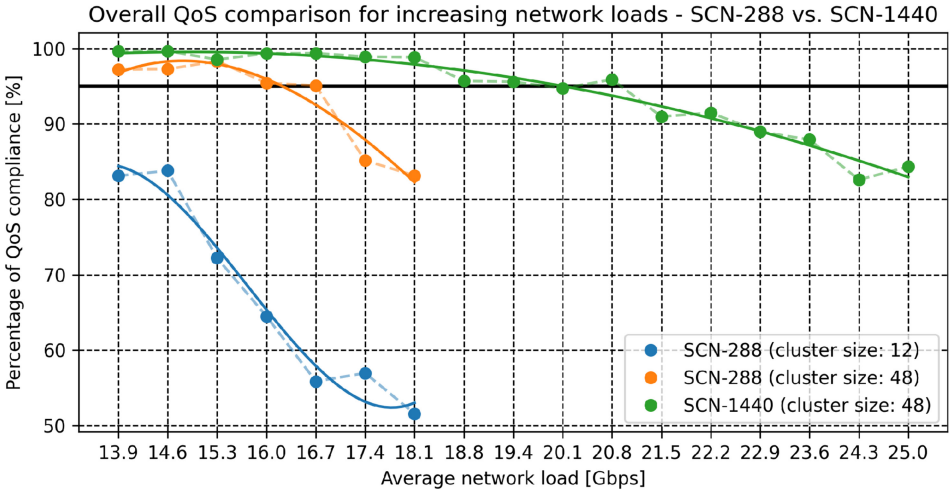

FIGURE 12 | QoS comparisons for increasing network loads for SCN-288 and SCN-1440 constellations. SCN-1440 constellation uses 48-node clusters (in green), while for SCN-288 both 48-node clusters (in orange) and 12-node clusters (in blue) are used. The latter included due to the similar physical size to 48-node clusters in SCN-1440.

shown in Figure 12. Importantly, the x-axis is now shifted: we observe network loads from 13.9 Gbps to 25.0 Gbps.

Given the increased number of satellites and links, much higher network loads can be achieved in the SCN-1440 constellation. However, given that the same cluster size and structure are used, the comparison is of interest. With 95% compliance, a data rate of approximately 20.2 Gbps was supported. Thus, an increase of approximately 25% is achieved for the same number of nodes in a cluster. Compared with a similarly sized cluster, which in the case of SCN-288 is a cluster size of 12, the increase is even more significant: approximately 61%. In SCN-288, F-IDLB with a cluster size of 12 drops below 95% at 12.5 Gbps (see Figure 9).

Potential adjustments to the configuration of the algorithms can be made, which potentially further improves performance. However, to enable a direct comparison, the same parameters used for SCN-288 were applied. A more detailed analysis of this larger and varied constellations is left for future work.

The performance of IDLB in SCN-1440 demonstrates its capacity to scale to dense satellite networks, a critical requirement for next-generation mega-constellations. The hierarchical design of the protocol is extensible to multiorbit, multilayer systems. Naturally, the impact of cluster sizing and other design choices may vary depending on the architecture and orbital layer. Future work should investigate the performance in operational architectures like Starlink or OneWeb [1]. For these constellations, adjustments to handover policies and cluster boundaries may be required to align the protocol with different and shell-based topologies. Comparative studies of load-balancing strategies across these systems would also further validate its versatility.

8 | Conclusion

This paper presents a comprehensive design and evaluation of a load-balanced distributed SDN-based routing protocol for autonomous satellite constellation networks, using in-space

cluster-based network control. Dedicated routing schemes are proposed for QoS-compliant forwarding within and between clusters. For intracluster routing, a tailored dynamic best-of-k-paths algorithm is proposed. Given its low complexity, the scheme is expected to be feasible for on-board processing. For intercluster routing, geographical information is used to find suitable links between clusters with reduced signaling.

Using a system-level simulator, we provide in-depth performance analyses of the proposed protocol. The QoS compliance of IDLB is investigated under increasing network loads to evaluate its load-balancing capacities, which significantly outperform a source-routing benchmark.

Additionally, we evaluate the impact of various design aspects. The analyses include comparisons between per-flow and per-packet routing, intercluster link choices, and cluster sizes. The impact of the added capacity of a larger satellite constellation using IDLB is also quantified.

Overall, our evaluation demonstrates the effectiveness and scalability of flow-based IDLB in supporting QoS-compliant routing in satellite systems. The proposed protocol is also suitable for multilayer constellations, and multilevel SDN hierarchies. It offers an extensible baseline for future research, for instance as an enabler of machine learning-enhanced traffic engineering. The architecture is well-suited for integration into future standards, making it a viable option for autonomous next-generation satellite networks.

Acknowledgments

This work has been supported by the Advanced Research in Telecommunications Systems Programme of the European Space Agency (ESA), activity code 3A.117, https://connectivity.esa.int/projects/ropro. Responsibility for the contents of this publication rests with the authors. Open Access funding enabled and organized by Projekt DEAL.

Conflicts of Interest

The authors declare no conflicts of interest.

References

1. T. Butash, P. Garland, and B. Evans, "Non-Geostationary Satellite Orbit Communications Satellite Constellations History," *International Journal of Satellite Communications and Networking* 39, no. 1 (2021): 1–5, https://doi.org/10.1002/sat.1375.

2. M. Handley, "Delay Is Not an Option: Low Latency Routing in Space," in *HotNets'18* (New York, NY, USA: Association for Computing Machinery, 2018): 85–91.

3. G. Giambene, S. Kota, and P. Pillai, "Satellite-5G Integration: A Network Perspective," *IEEE Network* 32, no. 5 (2018): 25–31, https://doi.org/10.1109/MNET.2018.1800037.

4. F. Bannour, S. Souihi, and A. Mellouk, "Distributed SDN Control: Survey, Taxonomy, and Challenges," *IEEE Communication Surveys and Tutorials* 20, no. 1 (2018): 333–354, https://doi.org/10.1109/COMST.2017.2782482.

5. T. De Cola, *Enabling Effective Multi-Link Data Distribution in NTN-Based 6G Networks* (VDE, 2023): 1–5.

6. M. M. H. Roth, H. Brandt, H. Bischl, "Distributed SDN-Based Load-Balanced Routing for Low Earth Orbit Satellite Constellation Networks". (2022): 1–8.

7. X. Cao, Y. Li, X. Xiong, and J. Wang, "Dynamic Routings in Satellite Networks: An Overview," *Sensors* 22, no. 12 (2022): 4552, https://doi.org/10.3390/s22124552.

8. E. Papapetrou and F. N. Pavlidou, "Distributed Load-Aware Routing in LEO Satellite Networks". (2008): 1–5.

9. M. M. H. Roth, "Analyzing Source-Routed Approaches for Low Earth Orbit Satellite Constellation Networks," in *LEO-NET'23* (New York, NY, USA: Association for Computing Machinery, 2023): 43–48.

10. L. Bertaux, S. Medjiah, P. Berthou, et al., "Software Defined Networking and Virtualization for Broadband Satellite Networks," *IEEE Communications Magazine* 53, no. 3 (2015): 54–60.

11. A. Papa, T. De Cola, P. Vizarreta, M. He, C. Mas Machuca, W. Kellerer, "Dynamic SDN Controller Placement in a LEO Constellation Satellite Network". 2018: 206–212.

12. A. Papa, T. De Cola, P. Vizarreta, M. He, C. Mas-Machuca, and W. Kellerer, "Design and Evaluation of Reconfigurable SDN LEO Constellations," *IEEE Transactions on Network and Service Management* 17, no. 3 (2020): 1432–1445, https://doi.org/10.1109/TNSM.2020.2993400.

13. M. Corici, H. Buhr, H. Zope, M. Zaboub, "An SDN-Based Solution for Mega-Constellation Routing". 2024: 1–6.

14. P. Grislain, N. Pelissier, F. Lamothe, et al., "Rethinking LEO Constellations Routing With the Unsplittable Multi-Commodity Flows Problem". (2022): 1–8.

15. H. Tsunoda, K. Ohta, N. Kato, and Y. Nemoto, "Supporting IP/LEO Satellite Networks by Handover-Independent IP Mobility Management," *IEEE Journal on Selected Areas in Communications* 22, no. 2 (2004): 300–307, https://doi.org/10.1109/JSAC.2003.819977.

16. M. M. H. Roth, H. Brandt, and H. Bischl, "Implementation of a Geographical Routing Scheme for Low Earth Orbiting Satellite Constellations Using Intersatellite Links," *International Journal of Satellite Communications and Networking* 39, no. 1 (2021): 92–107, https://doi.org/10.1002/sat.1361.

17. T. Li, "A Routing Architecture for Satellite Networks. Request for Comments RFC 9717, Internet Engineering Task Force"; 2025.

18. L. Davoli, L. Veltri, P. L. Ventre, G. Siracusano, S. Salsano, "Traffic Engineering With Segment Routing: SDN-Based Architectural Design and Open Source Implementation". 2015.

19. M. Office, *Cartopy: A Cartographic Python Library With a Matplotlib Interface* (Devon: Exeter, 2010).

20. J. G. Walker, "Satellite Constellations," *Journal of the British Interplanetary Society* 37 (1984): 559.

21. R. Leopold and A. Miller, "The IRIDIUM Communications System," *IEEE Potentials* 12, no. 2 (1993): 6–9, https://doi.org/10.1109/45.283817.

22. C. Fossa, R. Raines, G. Gunsch, and M. Temple, "An Overview of the IRIDIUM (R) Low Earth Orbit (LEO) Satellite System". (1998). 152–159.

23. ESA. "Routing and Management Protocols for Large Constellations With Inter-Satellite Links (ROPRO)". (2025). https://connectivity.esa.int/projects/ropro.

24. Center for International Earth Science Information Network - CIESIN - Columbia University. "Gridded Population of the World, Version 4 (GPWv4): Population Density, Revision 11". 2018.

25. E. Lutz, M. Werner, and A. Jahn, *Satellite Systems for Personal and Broadband Communications* (Springer Science & Business Media, 2012).

26. L. Wood, "SaVi: Satellite Constellation Visualization". 2012.

27. ITU, "G.114: One-Way Transmission Time". g.114, ITU-T; 2003.

28. ITU, "Y.1541: Network Performance Objectives for IP-based Services. y.1541, ITU-T". 2011.

29. N. Seitz, "ITU-T QoS Standards for IP-Based Networks," *IEEE Communications Magazine* 41, no. 6 (2003): 82–89, https://doi.org/10.1109/MCOM.2003.1204752.

30. 3GPP. "Solutions for NR to Support Non-Terrestrial Networks (NTN)". Tech. Rep. TR 38.821 V16.0.0, 3GPP; 2019.

31. F. Baker, D. L. Black, K. Nichols, S. L. Blake, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. Request for Comments RFC 2474, Internet Engineering Task Force". 1998.

32. C. Maihofer, "A Survey of Geocast Routing Protocols," *IEEE Communication Surveys and Tutorials* 6, no. 2 (2004): 32–42, https://doi.org/10.1109/COMST.2004.5342238.

33. B. Meijerink, M. Baratchi, and G. Heijenk, "An Efficient Geographical Addressing Scheme for the Internet," in *Wired/Wireless Internet CommunicationsLecture Notes in Computer Science*, eds. L. Mamatas, I. Matta, P. Papadimitriou, and Y. Koucheryavy (Cham: Springer International Publishing, 2016): 78–90.

34. B. Fenner, M. J. Handley, I. Kouvelas, and H. Holbrook, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised). Request for Comments RFC 4601, Internet Engineering Task Force". 2006.

35. C. Larsson, *Design of Modern Communication Networks: Methods and Applications* (Academic Press, 2014).

36. J. Li, X. Chang, Y. Ren, Z. Zhang, and G. Wang, "An Effective Path Load Balancing Mechanism Based on SDN". 2014: 527–533.

37. Y. Azar, A. Z. Broder, A. R. Karlin, E. Upfal. *Balanced Allocations (Extended Abstract)*. In: ACM Press; 1994; Montreal, Quebec, Canada: 593–602.

38. V. B. How, "Asymmetry Helps Load Balancing," *Journal of the ACM* 50, no. 4 (2003): 568–589, https://doi.org/10.1145/792538.792546.

39. D. Eppstein, "Finding the k Shortest Paths," *SIAM Journal on Computing* 28.2 (1997): 26.

40. S. Hoceini, A. Mellouk, and Y. Amirat, "K-Shortest Paths Q-Routing: A New QoS Routing Algorithm in Telecommunication Networks," in *Networking - ICN 2005 Lecture Notes in Computer Science*, eds. P. Lorenz and P. Dini (Berlin, Heidelberg: Springer, 2005): 164–172.