On the Latency Trade-off Between Space and Terrestrial Clouds in Non-Terrestrial Networks

Camilo Rojas*, Juan A. Fraire†‡, Fabio Patrone*, Mario Marchese*

*University of Genoa, Genoa, Italy

†Inria, INSA Lyon, CITI, UR3720, 69621 Villeurbanne, France

‡CONICET - Universidad Nacional de Córdoba, Córdoba, Argentina

***Abstract*—Non-Terrestrial Networks (NTN) are poised to revolutionize 5G and 6G networks by integrating terrestrial and space-based cloud systems, enabling dynamic task allocation for optimal performance. Despite their promise, understanding the trade-offs between terrestrial and non-terrestrial edge computing architectures remains an area for improvement. This paper presents a comprehensive latency-focused trade-off analysis using a novel real-time emulation platform that accurately models terrestrial and space cloud environments. By evaluating network latency across geodesic distances from a fixed ground gateway, we delineate scenarios where terrestrial clouds excel and identify conditions under which Space Cloud architectures surpass their terrestrial counterparts. Additionally, we analyze how server placement strategies in satellite constellations impact performance, revealing the critical interplay between server distribution and latency outcomes. These findings offer actionable insights for designing and operating hybrid cloud systems, emphasizing the need for tailored architectures to maximize the potential of NTN-based edge computing.**

***Index Terms*—Multi-access Edge Computing**

I. Introduction

The sudden growth in using Low-Earth Orbit (LEO) satellite constellations for telecommunication networks has caused a transformative shift in the space industry and communication companies. Commercial Off-The-Shelf (COTS) components and standardized launch systems, like P-POD containers, have permitted massive space accessibility to new actors [1]. The new integration of increasingly complex Artificial Intelligence (AI) control systems in launch and commanding has revolutionized access to space missions [2]. These innovations have significantly reduced the barriers to entry in the space competition, enabling even small organizations with limited budgets to participate in complex space projects once monopolized by major telecommunications companies and government space agencies [3].

Recently, private companies have led the way in advancing Non-Terrestrial Networks (NTNs) by developing and deploying mega-constellations and integrating them into ground-edge segments. These constellations comprise hundreds or thousands of LEO satellites designed to provide global coverage and bandwidth equivalent to ground fiber optics networks. SpaceX's Starlink is a prominent example, with a planned constellation of over 42,000 satellites and more than 6,000 currently active in orbit. Similarly, Amazon's OneWeb plans to deploy 684 satellites at an orbital altitude of 1,200 km, with 428 satellites already in operation [4].

In recent years, the advances in space networks have gained the attention of mobile standardization organizations, including the concept of NTNs in 5G standards starting with 3GPP Release 16 [5] and advancing into further integrations for 6G and beyond. Nonetheless, to be effectively integrated into the 5G ecosystem, NTNs must meet critical Key Performance Indicators (KPIs), including capacity, latency, and service availability metrics. Deploying edge computing servers in space, with powerful embedded computing capacity [6], will play a critical role in achieving 5G KPIs, reducing the total delivery delay of space-based communication services [7].

Most constellations aim to interconnect satellites across orbital planes through Inter-Satellite Links (ISL), significantly enhancing coverage in areas where ground-based gateways cannot be accessed directly [8]. ISL connections are especially crucial in remote regions where expanding terrestrial cellular networks is either technically challenging or economically unfeasible [9].

The term *Space Cloud* refers to deploying edge servers and on-orbit clusters to deliver seamless computational services directly in space, thereby avoiding the need to route data back to Earth's Internet backbone. This concept ensures complete transparency for users or devices requesting computational workloads. It guarantees that the system will process requests holistically and ubiquitously meet specific latency requirements. This approach reimagines mega-constellations as communication networks and service platforms, aligning more closely with the paradigm of Software-as-a-Service (SaaS) cloud providers.

Traditionally, research on mega-constellations has focused on their role as basic communication infrastructures rather than platforms capable of delivering added-value services. Efforts such as those by Zhang et al. [10], Rago et al. [11], and Xie et al. [12] highlight innovative architectures and strategies to enhance QoS, IoT connectivity, and multi-node task scheduling. The growing interest in NTNs reflects the broader industry and academic push to leverage satellite constellations for global, low-latency computational services. However, the concept of a space cloud has received limited attention despite its great potential.

In this context, the specific contributions of this work are:

1) **A realistic cloud emulation environment** to achieve realistic instantiation and isolation for Terrestrial and Space Cloud systems. MeteorNet is based on micro-

services containers replicating operational communication conditions to facilitate the assessment of real-world edge computing scenarios.
2) **A comprehensive and realistic trade-off analysis** between Terrestrial and Space Clouds, providing the first quantitative identifications of the limitations where each orchestration approach is better.

The remainder of this paper is organized as follows. Section II revises related works in the context of NTN edge computing. Section III presents MeteorNet: a cloud emulation tool for modeling space and terrestrial cloud. Section IV introduces the system model specifying the space and terrestrial cloud assumptions. Section V analyzes the results and quantitatively assesses the trade-off between space and terrestrial cloud. Finally, Section VI concludes the paper.

II. Related Works

Some state-of-the-art research papers have tackled deploying edge servers in NTN. For example, Zhang et al. [10] explore the integration of Multi-access Edge Computing (MEC) into high-speed satellite-terrestrial networks, and they propose a schedule offloading model to enhance the Quality of Service (QoS) and improve performance in the network. Similarly, Rago et al. [11] examine the integration of Terrestrial Networks (TNs) and NTNs to deliver three-layer dimensional wireless connectivity for 6G networks, enhancing coverage and service reliability.

Liang et al. [13] explore a UAV-Satellite architecture designed to enhance the localization and communication of IoT systems. They use NTNs, leveraging UAVs as aerial relays and satellites for positioning and communication backhaul to ensure seamless connectivity for IoT devices. They also suggest areas for future research to advance this integrated architecture.

Hosseinian et al. [14] review satellite networks and their potential to provide global coverage with minimal reliance on ground infrastructure. They describe some critical issues, such as high costs, long propagation delays, and dependence on line-of-sight (LoS), that hinder their competitiveness with terrestrial networks. They also discuss the renewed interest in satellite communication due to new mega-constellation projects and advancements in high- and high-throughput satellites (HTS/VHTS) in geostationary orbits.

Kim et al. [15] explores the potential of satellite edge computing for extending 5G and 6G IoT services, leveraging satellites to provide global coverage and support 3D mobility. IoT devices are cost-effective solutions for future networks due to their relatively low computational demands. They also propose a network architecture and scheduling strategy addressing latency, computational power, and transmission power challenges.

Xie et al. [12] introduce a multi-layer edge computing architecture within satellite-terrestrial networks to support diverse future network services. They incorporate direct task processing on satellites and heterogeneous edge computing clusters, addressing challenges such as ensuring Quality of Experience (QoE), cooperative computation offloading, multi-node task scheduling, mobility management, and fault recovery.

This paper focuses on integrating edge computing paradigms within a LEO satellite constellation. We compare ground-based and satellite-based edge computing networks to assess which quantitatively delivers lower compute latency in varying conditions. To the authors' knowledge, this is the first paper comparing space and ground clouds. Our analysis is unique in the sense that it leverages a realistic containerized emulation tool that accurately replicates the cloud conditions in space and ground systems [16]–[18]. The following section introduces the MeteorNet tool.

III. Cloud Emulation Tool

In this section, we introduce our tool MeteorNet [17], [19] designed to test various edge computing strategies within space network environments. MeteorNet is an open-source, continuous-time emulation platform for satellite constellation networks. It stands out from existing tools by integrating network and operating system virtualization, utilizing production-grade kernel and network management components. Unlike traditional discrete-time simulation tools, MeteorNet offers real-time fidelity, ensuring highly accurate deployment tests and supporting scalable, multi-threaded development. This real-time emulation capability closely mirrors real-world scenarios, ensuring that computing services under test must handle the precise timing constraints of a production-grade space cloud system.

A. MeteorNet Architecture

MeteorNet platform accurately replicates a cloud computing environment deployed in space as a LEO satellite constellation, incorporating the following core modules:

1) **Orbit Propagation Module**: simulates satellite motion and orbital dynamics;
2) **Software-Defined Networking (SDN) Controller**: centralizes network management by using the Mininet software [20];
3) **Containerized Software**: leverages Docker for OS-level virtualization to allow isolated software instantiation [21];
4) **Network Performance Monitoring**: collects critical performance metrics for analysis and optimization.

Figure 1 shows the interaction of two virtual network nodes in MeteorNet; here, a ground and satellite node are connected by a communication link between switches managed by Mininet. MeteorNet can change the link quality of service parameters like capacity, delay, and loss and disable the link if the ground node loses the satellite's line of sight. Every node in the network runs a Docker container with a local operating system and software library. Satellite nodes can also work only as host-less switches, routing packets using an SDN framework compatible with Openflow.

B. MeteorNet Cloud Models

MeteorNet employs Mininet to create a realistic virtual network environment, integrating production-level Linux kernels,

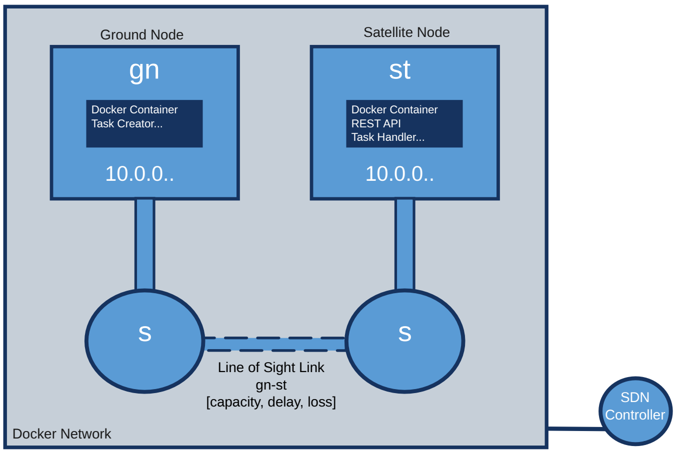


Figure 1: MeteorNet Architecture

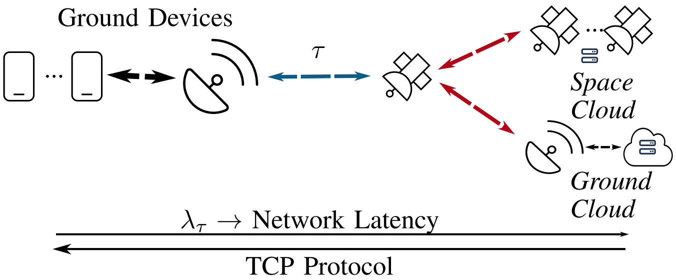


Figure 2: MeteorNet Cloud Models

switches, and the OpenFlow standard [22]. Mininet enables dynamic deployment and configuration of custom virtual networks and communication links, closely mimicking real-world cloud networks by utilizing production-grade technologies. Through Mininet, MeteorNet emulates network interfaces, routing behavior, and compute resources, focusing on the second layer and higher of the OSI model. Instead of directly modeling physical layer equations, MeteorNet approximates key statistical metrics, such as channel availability, capacity, and propagation delays, to reflect real-world communication link characteristics.

MeteorNet also provides task edge computing capabilities in addition to the network features described before. Figure 2 shows the two cloud edge computing models that MeteorNet supports:

1) terrestrial cloud, when the computing server is located on the ground;
2) space cloud, when one or more computing servers are located on satellite nodes.

For both cloud orchestrations, space makes the first connection link with the first satellite in line of sight of the ground device. In addition, we consider intersatellite links and connections between satellites to route data to and from the clouds. We leverage ONOS SDN software to dynamically calculate the best network path, using their reactive routing algorithm [23], which creates the best route in the network graph using Dijkstra.

Figure 3: Ground nodes on the map. Green points are device locations, and purple point is the location of the ground gateway.

IV. System Model

In this section, we use MeteorNet features for emulating ground and space edge computing systems to create experimental setups where we can see the performance of both networks.

A. Space Segment Model

We consider a mutual constellation of $S = 100$ satellites in the Walker Delta pattern with ten individual planes distributed uniformly along the half of the equatorial circumference ($180°$, leaving an inclination of $18°$ between planes and ten satellites distributed uniformly per the $360°$ orbital plane. We assume ISLs between neighbor satellites within the same orbital plane and between neighbor orbital planes, leaving each satellite with a direct connection with the four adjacent satellites (two in the same orbit, two in the adjacent orbits).

B. Ground Segment Model

We test the performance of *ground* and *space* edge computing systems using the satellite constellation to route and process computation tasks demanded by IoT devices deployed on the Earth's surface. We consider that the devices are not directly connected to the Internet cloud and cannot process their tasks directly. Consequently, the first network hop requires a space link to the nearest satellite passing by in a line-of-sight condition. Figures 3 and 4 show the geographic locations of ground devices related to satellite nodes' initial coordinates. Table I provides the ground nodes' exact latitude and longitude positions. The distance to the gateway (located in Italy) is also listed.

C. Space and Terrestrial Cloud Models

In total, we tested four different edge computing systems: one ground-based and three satellite-based. The principal difference between ground-edge and space-edge computing architectures is the location of the computing server. In particular, we consider:

1) **Terrestrial Cloud**: exploits a ground gateway located in Italy connected to the Internet to forward satellite data

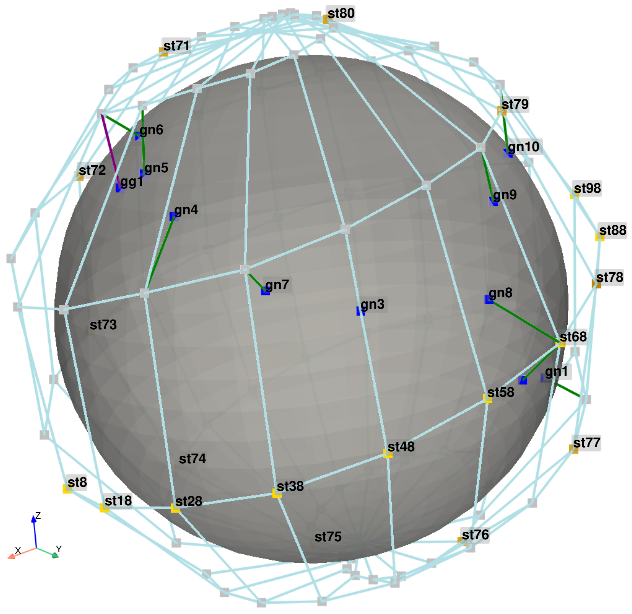


Figure 4: Network nodes on 3D view.

Table I: Ground nodes locations

Location	Latitude [°]	Longitude [°]	Name	Distance [Km]
Italy	44.41	8.93	gg1	-
Australia	-27.4550	153.0351	gn1	16321
Indonesia	-6.1744	106.8294	gn2	11117
India	19.017	72.857	gn3	6489
Turkey	39.9272	32.8644	gn4	2027
Austria	48.2	16.3666	gn5	709
UK	55.9483	-3.2191	gn6	1543
U. Arab Emirates	25.23	55.28	gn7	4660
Thailand	13.75	100.5166	gn8	9069
China	30.58	114.27	gn9	8770
Japan	35.685	139.7514	gn10	9826

to and from a data center assumed directly connected to the station. It is assumed a link latency of 5 ms between gateway and the Internet.

2) **Space Cloud, Single Distribution**: Space clouds exploit servers located in space and deployed onboard satellites with three different server orchestrations. The single distribution one assumes one server located at satellite n° 78;
3) **Space Cloud, Orbital Plane Distribution**: 10 servers belonging to the same orbital plane, these satellites have equal orbital elements set except for the mean anomaly where they have a difference of 36 ° between orbital plane neighbors, satellites 71-80 from Figure 4; and
4) **Space Cloud, Equatorial Plane Distribution**: 10 servers placed on neighbors orbital planes with similar element sets, but separated by 18° of the Right ascension of the ascending node angle (Ω) between neighbors [24], satellites 8, 18, 28 ... 98 from Figure 4.

D. Compute Model

To compare test cases, we fixed the constellation parameters and task generation statistics for all orchestration systems as indicated in Table II. IoT devices create tasks following a Poisson distribution with a generation rate of $r_P = 2$ per minute per device and a total of $D = 20$ devices fixed at each ground location. This distribution accurately represents the random nature of task arrivals with independent arrival probabilities. We choose an emulation duration of $T = 45$ minutes equal to about half of the orbital period, which approximately means a total number of tasks of $N_\tau \approx r_P \cdot D \cdot T = 1800$ tasks. We execute ten emulations for each edge system, one for each device location. On every emulation, we measure the task network latency, λ_τ for each task τ, defined as the time between the task transmission from the device and the reception back of the processed data. We compare the test mentioned above cases by using mean and standard deviation metrics over the variable λ_τ:

$$\overline{\lambda}_\tau = \mu(\lambda_\tau) = \frac{1}{N_\tau} \sum_{i=1}^{N_\tau} \lambda_i \tag{1}$$

$$\sigma(\lambda_\tau) = \sqrt{\frac{1}{N_\tau - 1} \sum_{i=1}^{N_\tau} (\lambda_i - \overline{\lambda}_\tau)^2} \tag{2}$$

The proposed test environment allows us to find groups in the solution space where space cloud systems could behave better than terrestrial cloud systems.

Table II: Constellation parameters for test cases.

Parameter	Value	Description
S	100	Number of satellites
D	20	Number of devices per ground location
T	45 minutes	Total emulation duration
r_P	2 tasks/minute	Task generation rate per device
N_τ	$\approx$ 1800	Total number of tasks generated

V. Performance Evaluation

In this section, we evaluate the performance of space and terrestrial clouds by quantitatively assessing the network latency of user task requests from remote devices worldwide. The proposed environments are emulated within MeteorNet in real-time using a regular computer with an Intel Core i7, 8th generation, and 16 GigaBytes of RAM. Using a single ground gateway located in Italy, we compare the results obtained from space clouds against a terrestrial cloud similar to existing mega constellations, like Starlink or OneWeb. The results are organized by devices in ascending order of geodesic distance from the ground gateway, providing a clearer perspective on performance variations.

A. Absolute Latency Analysis

Figure 5 shows $\overline{\lambda}_\tau$ versus device geodesic distances to the ground gateway. The numerical values for each specific node location, ordered by distance, are presented in Table III. We can see an expected behavior for the terrestrial cloud with an increasing function as the distance increases. By analyzing node locations, it surfaces that increasing the distance from the ground gateway also increases the route distance and ISL hops to the server, so the latency increases accordingly. On

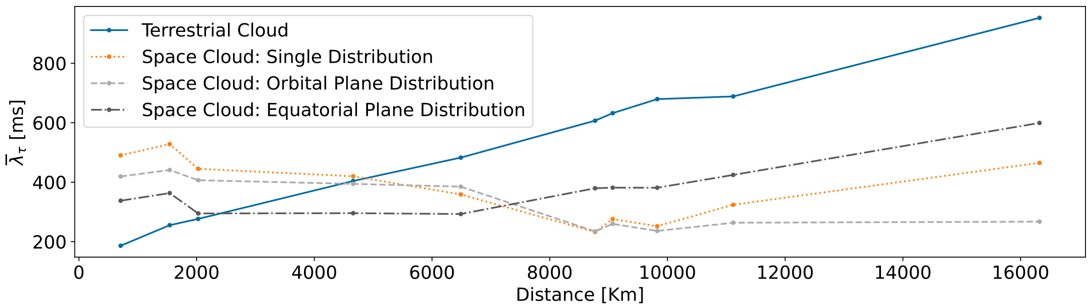


Figure 5: Mean network latency for edge computing systems leveraging terrestrial and space cloud configurations. Distance is the distance between the device requesting the compute service and the ground gateway used by the satellite network to access the Internet.

Table III: Latency results for edge computing clouds.

				Space Cloud					
		Terrestrial Cloud		Single		Orbital		Equatorial	
Name	Distances [km]	$\mu(\lambda_\tau)$[ms]	$\sigma(\lambda_\tau)$[ms]	$\mu(\lambda_\tau)$[ms]	$\sigma(\lambda_\tau)$[ms]	$\mu(\lambda_\tau)$[ms]	$\sigma(\lambda_\tau)$[ms]	$\mu(\lambda_\tau)$[ms]	$\sigma(\lambda_\tau)$[ms]
gn5	709	186	64	490	104	419	86	338	160
gn6	1543	255	96	528	103	441	109	363	151
gn4	2027	277	85	445	83	407	72	295	165
gn7	4660	404	66	420	79	394	80	296	180
gn3	6489	483	68	359	86	385	74	293	177
gn9	8770	607	98	233	88	234	86	380	237
gn8	9069	632	87	276	118	260	84	382	234
gn10	9826	679	64	252	100	236	72	381	238
gn2	11117	688	66	324	115	264	76	424	257
gn1	16321	953	88	465	130	267	75	599	311

the other hand, we observe that space clouds' performance is relatively insensitive to the distance to the ground gateway; consequently, they appear stable across user device locations. As a result, terrestrial clouds perform better than space-based ones when the distance between the device and the ground gateway is less than 2000 km. However, for distances beyond 5000 km, all space cloud configurations exhibit lower latency. The best cloud approach depends on the topological configuration and space cloud variant within the [2000, 5000] km range.

B. Relative Latency Analysis

Figure 6 presents the latency data by showing the difference between ground and space clouds:

$$\Delta\overline{\lambda}_\tau = \overline{\lambda}_\tau^G - \overline{\lambda}_\tau^{Si}. \quad (3)$$

The value is shown on a red scale when negative, meaning the terrestrial cloud beats the space cloud, and on a green scale when positive, meaning the space cloud beats the ground one. We can see a group of distances ranging between 2000 km and 5000 km where the network latency difference between space and terrestrial clouds is in the order of 100 ms.

C. Space Cloud Configuration Analysis

Analyzing the results of the three configurations tested for the space cloud, no particular server distribution performs better for all device locations. The *Equatorial* distribution performs the best for ground devices gn5, gn6, gn4, and gn7. However, the *Orbital* distribution performs best for the rest of the locations. Interestingly, for some device nodes (gn7 and gn3), the mean latency of *Single* and *Orbital* configurations are pretty close. This suggests that choosing a specific edge server count and placement in the space cloud can be critical. Indeed, similar KPIs are obtained for the space cloud with one and ten operative servers in these cases.

D. Static Hop Count Analysis

This section analyzes the average number of hops the packets travel to reach the computing server in space or the terrestrial cloud. For example, Figure 7 shows the paths for device locations gn4 (2027 km from the gateway) and gn10 (9826 km from the gateway) for each cloud system at their initial positions in the simulation. For gn4, the terrestrial solution requires five hops. In contrast, the space cloud utilizes seven hops in both the *Single* and *Orbital* plane distributions and just two hops in the equatorial distribution. Based on the initial path results, the *Equatorial* configuration is expected to exhibit lower latency than the other distributions for gn4. However, the averaged latency results above revealed that the terrestrial solution outperforms all others for gn4 at 2027 km, with only a slight margin over the equatorial distribution.

E. Dynamic Hop Count Analysis

To investigate this further, Figure 8a disaggregates the number of hops calculated every 5 minutes of simulation for gn4. By observing these results, we can see that the *Equatorial* distribution offers as low as two hops count at the beginning of the simulation time. This can be confirmed

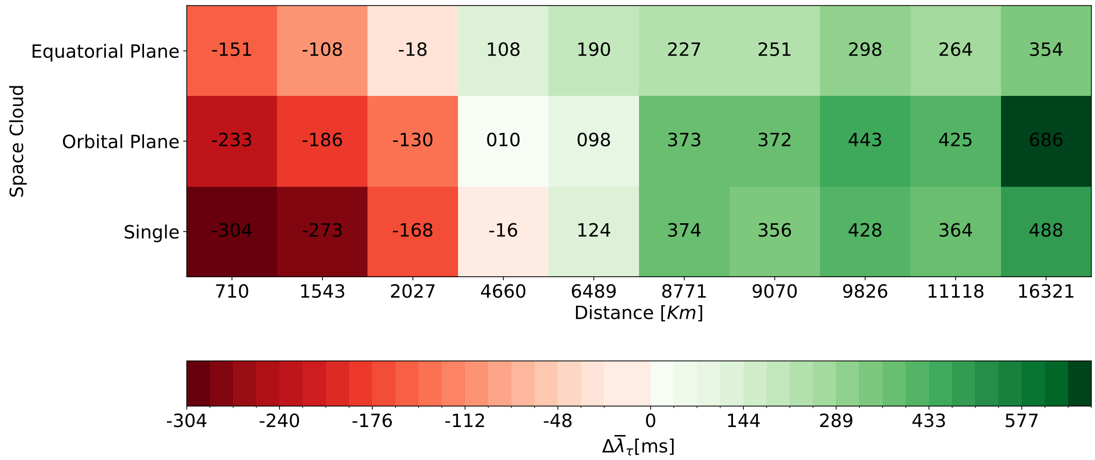
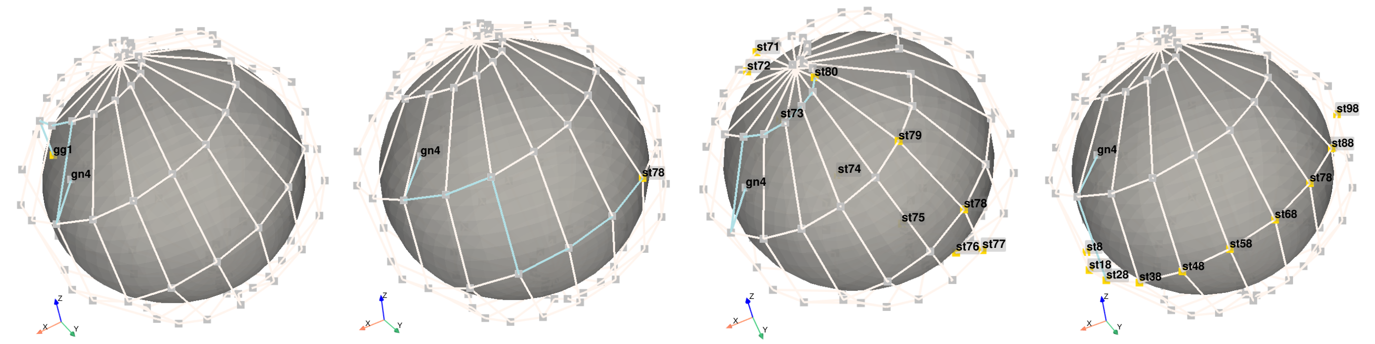


Figure 6: Latency difference between space and terrestrial clouds for different geodesic distances between the device and the target compute node. Red means the Terrestrial Cloud performs better; Green means the Space Cloud (either Equatorial plane, Orbital plane, or Single server) performs better.

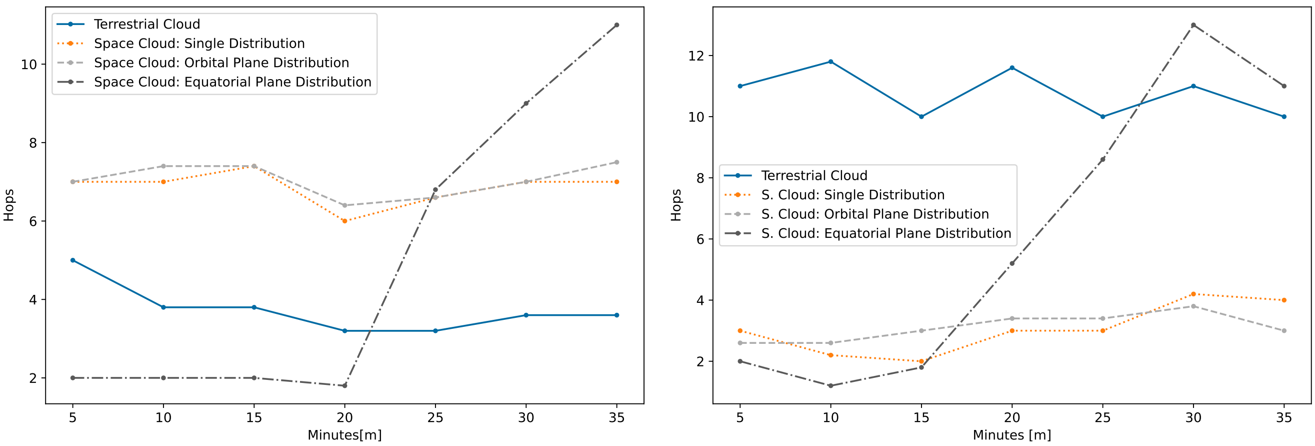


(a) gn4 - TC - t = 0 (b) gn4 - SC (Single) - t = 0 (c) gn4 - SC (Plane) - t = 0 (d) gn4 - SC (Equatorial) - t = 0

(e) gn10 - TC - t = 0 (f) gn10 - SC (Single) - t = 0 (g) gn10 - SC (Plane) - t = 0 (h) gn10 - SC (Equatorial) - t = 0

Figure 7: Route paths for gn4 and gn10 for each terrestrial cloud (TC) and space cloud (SC) configuration at the start of the simulation (t = 0). The servers are colored in yellow, the ground nodes are gray, and the chosen multi-hop path is in light blue.

with Figure 7d, as only two hops can connect gn4 with the space cloud servers in the equator. However, from minute 20 onwards, the *Equatorial* configuration suddenly rises to more than ten hops in total. This corresponds to the topological configuration in Figure 9a, where the space cloud servers are near the North Pole. Considering that the server selection criteria honor the geometrically closest one, gn4 requires a long route that circulates all polar satellites before reaching the selected server. A similar analysis can be made for gn10 by observing Figure 7h when *Equatorial* space cloud servers are in the equator and Figure 9b when they are over the pole. This is an even more dramatic case as gn10 is even further away from the gateway station.



(a) Dynamic Hop Count for gn4.

(b) Dynamic Hop Count for gn10.

Figure 8: Time-evolving average number of hops sampled every 5 minutes for gn4 (2027 km away from the gateway) and gn10 (9826 km away from the gateway). Both ground device presents a notable hop count step at 20 minutes of emulation time, which corresponds with the *Equatorial* servers located over the poles (see Figure 9).

F. Discussion and Evaluation Takeaway

These insights underscore the highly dynamic nature of space networks, where the optimal packet travel path can shift dramatically from the best to the worst-case scenario. The realistic emulation tool presented in this study allowed us to quantify and identify two key categories of challenges in the context of space clouds:

1) **Design Phase Challenges**: Determining where to place space cloud servers within the satellite network and deciding the optimal number of servers required to balance coverage and performance.
2) **Operations Phase Challenges**: Establishing criteria for devices to intelligently select the most suitable server among the available options to minimize latency.

Addressing these challenges necessitates novel design and operational techniques that enable space clouds to outperform terrestrial clouds, even for scenarios with relatively short distances between devices and ground gateways. By tackling these questions, space clouds can achieve their full potential as high-performance, globally distributed computing platforms.

VI. Conclusions

This study examines innovative edge computing scenarios where servers are distributed in LEO satellite constellations and compared to ground-based clouds. Our evaluation of network latency highlights that ground-based systems excel for user devices within 2000 km of a gateway, but their performance degrades as distances grow. Beyond 5000 km, space-based solutions consistently outperform terrestrial systems due to their flexibility in server distribution.

The number of network hops plays a critical role in performance. For closer locations, terrestrial solutions maintain lower latency despite higher initial hops, while for distant locations, space cloud configurations, particularly the single and orbital plane distributions, offer fewer hops and better performance. The equatorial distribution initially shows an advantage but degrades in dynamic environments, underscoring the need for adaptive strategies.

These findings emphasize the importance of optimizing server placement and count during the design phase and establishing intelligent server selection criteria during operations. By addressing these challenges, space clouds can surpass terrestrial systems even at shorter distances, making them a viable solution for globally distributed networks. Further research into constellation distributions and integrated ground-space systems is essential to fully harness the potential of space cloud architectures for diverse applications.

Acknowledgement

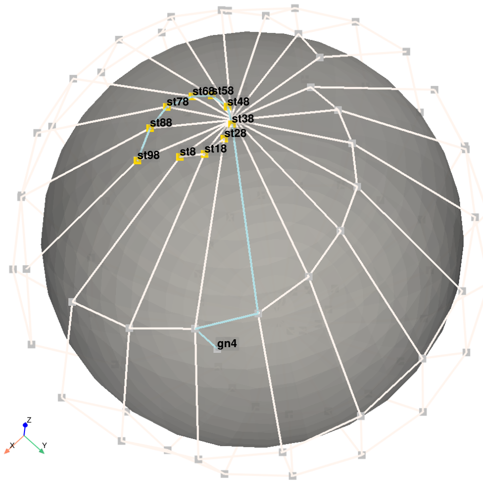
This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 101008233 (MISSION) and the French National Research Agency (ANR) under the project ANR-22-CE25-0014-01. The European Union also supported this work under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, a partnership on "Telecommunications of the Future" (PE00000001—program "RESTART").

References

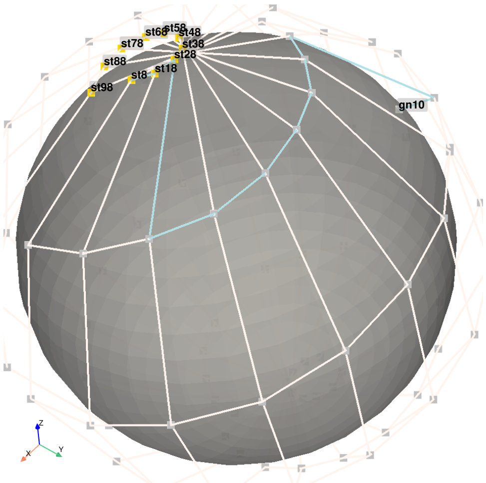
[1] S. Lee, A. Hutputanasin, A. Toorian, W. Lan, R. Munakata, J. Carnahan, D. Pignatelli, A. Mehrparvar, Cubesat design specification rev. 13 (2014).

[2] H. Yao, L. Wang, X. Wang, Z. Lu, Y. Liu, The Space-Terrestrial Integrated Network: An Overview, IEEE Communications Magazine 56 (9) (2018) 178–185. `doi:10.1109/MCOM.2018.1700038`.
URL https://ieeexplore.ieee.org/document/8338471/

[3] D. Paikowsky, What Is New Space? The Changing Ecosystem of Global Space Activity, New Space 5 (2) (2017) 84–88. `doi:10.1089/space.2016.0027`.
URL https://www.liebertpub.com/doi/10.1089/space.2016.0027



(a) gn4 - SC (Equatorial) - t = 20 min



(b) gn10 - SC (Equatorial)- t = 20 min

Figure 9: Route paths for gn4 and gn10 for the Equatorial space cloud (SC) configuration when servers are at the pole (t = 20 min).

[4] J. Zhang, Y. Cai, C. Xue, Z. Xue, H. Cai, LEO Mega Constellations: Review of Development, Impact, Surveillance, and Governance, Space: Science & Technology 2022, _eprint: https://spj.science.org/doi/pdf/10.34133/2022/9865174 (2022). doi:10.34133/2022/9865174.
URL https://spj.science.org/doi/abs/10.34133/2022/9865174

[5] 3Gpp, Solutions for NR to support Non-Terrestrial Networks (NTN), 3Gpp Release 16 (2019) 1–28doi:TR38.821.

[6] G. Lentaris, K. Maragos, I. Stratakos, L. Papadopoulos, O. Papanikolaou, D. Soudris, M. Lourakis, X. Zabulis, D. Gonzalez-Arjona, G. Furano, High-Performance Embedded Computing in Space: Evaluation of Platforms for Vision-Based Navigation, Journal of Aerospace Information Systems 15 (4) (2018) 178–192. doi:10.2514/1.I010555.
URL https://arc.aiaa.org/doi/10.2514/1.I010555

[7] Q. Li, S. Wang, X. Ma, Q. Sun, H. Wang, S. Cao, F. Yang, Service Coverage for Satellite Edge Computing, IEEE Internet of Things Journal 9 (1) (2022) 695–705. doi:10.1109/JIOT.2021.3085129.
URL https://ieeexplore.ieee.org/document/9444334/

[8] I. del Portillo, B. G. Cameron, E. F. Crawley, A technical comparison of three low earth orbit satellite constellation systems to provide global broadband, Acta Astronautica 159 (2019) 123–135. doi:10.1016/j.actaastro.2019.03.040.
URL https://linkinghub.elsevier.com/retrieve/pii/S0094576518320368

[9] I. d. Portillo, S. Eiskowitz, E. F. Crawley, B. G. Cameron, Connecting the other half: Exploring options for the 50% of the population unconnected to the internet, Telecommunications Policy 45 (3) (2021) 102092. doi:https://doi.org/10.1016/j.telpol.2020.102092.
URL https://www.sciencedirect.com/science/article/pii/S0308596120301828

[10] Z. Zhang, W. Zhang, F.-H. Tseng, Satellite Mobile Edge Computing: Improving QoS of High-Speed Satellite-Terrestrial Networks Using Edge Computing Techniques, IEEE Network 33 (1) (2019) 70–76. doi:10.1109/MNET.2018.1800172.
URL https://ieeexplore.ieee.org/document/8610431/

[11] A. Rago, A. Guidotti, G. Piro, E. Cianca, A. Vanelli-Coralli, S. Morosi, G. Virone, F. Brasca, M. Troscia, M. Settembre, L. Pierucci, F. Matera, M. De Sanctis, S. Pizzi, L. A. Grieco, Multi-layer NTN architectures toward 6G: The ITA-NTN view, Computer Networks 254 (2024) 110725. doi:10.1016/j.comnet.2024.110725.
URL https://linkinghub.elsevier.com/retrieve/pii/S1389128624005577

[12] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, T. Huang, Satellite-Terrestrial Integrated Edge Computing Networks: Architecture, Challenges, and Open Issues, IEEE Network 34 (3) (2020) 224–231. doi:10.1109/MNET.011.1900369.
URL https://ieeexplore.ieee.org/document/9048610/

[13] T. Liang, T. Zhang, Q. Zhang, Toward Seamless Localization and Communication: A Satellite-UAV NTN Architecture, IEEE Network 38 (4) (2024) 103–110. doi:10.1109/MNET.2024.3384298.
URL https://ieeexplore.ieee.org/document/10488448/

[14] M. Hosseinian, J. P. Choi, S.-H. Chang, J. Lee, Review of 5G NTN Standards Development and Technical Challenges for Satellite Integration With the 5G Network, IEEE Aerospace and Electronic Systems Magazine 36 (8) (2021) 22–31. doi:10.1109/MAES.2021.3072690.
URL https://ieeexplore.ieee.org/document/9508471/

[15] T. Kim, J. Kwak, J. P. Choi, Satellite Edge Computing Architecture and Network Slice Scheduling for IoT Support, IEEE Internet of Things Journal 9 (16) (2022) 14938–14951. doi:10.1109/JIOT.2021.3132171.
URL https://ieeexplore.ieee.org/document/9632811/

[16] C. Rojas, J. Fraire, F. Patrone, A. Gotta, M. Marchese, Continuous Time Emulation for Software-Defined Non-Terrestrial Edge Computing Networks, in: European Wireless (EW) Conference, IEEE, 2023, pp. 1–6.

[17] C. Rojas, J. Fraire, F. Patrone, M. Marchese, Fuzzy Logic-Based Orchestration of Multi-Access Edge Computing in LEO Satellite Constellations, in: IEEE International Conference on Communications, In Press, IEEE, 2024, pp. 1–6.

[18] C. Rojas, J. Fraire, F. Patrone, M. Marchese, Advanced Constellation Emulation and Synthetic Datasets Generation for Non-Terrestrial Networks, in: IEEE International Mediterranean Conference on Communications and Networking, In Press, IEEE, 2024, pp. 1–7.

[19] Satellite Communication and heterogeneous Networking Laboratory (SCNL), University of GEnoa, Italy, Meteornet (2024).
URL https://gitlab.com/camilo.rojas/meteornet

[20] L. Yan, N. McKeown, Learning Networking by Reproducing Research Results, ACM SIGCOMM Computer Communication Review 47 (2) (2017).

[21] S. Singh, N. Singh, Containers & Docker: Emerging roles & future of Cloud technology, in: 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2016, pp. 804–807. doi:10.1109/ICATCCT.2016.7912109.

[22] K. Benzekki, A. El Fergougui, A. Elbelrhiti Elalaoui, Software-defined networking (SDN): a survey, Security and Communication Networks 9 (18) (2016) 5803–5833, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/sec.1737. doi:https://doi.org/10.1002/sec.1737.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1737

[23] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, et al., Onos: towards an open, distributed sdn os, in: Proceedings of the third workshop on Hot topics in software defined networking, 2014, pp. 1–6.

[24] F. R. Hoots, Spacetrack report no. 3, models for propagation of norad element sets, http://www. itc. nl/-bakker/orbit. html (1980).