Orbital Edge Offloading on Mega-LEO Satellite Constellations for Equal Access to Computing

Pietro Cassará, Alberto Gotta, Mario Marchese, and Fabio Patrone

The authors shed some light on the possible integration of the in-network computing paradigm in mega-LEO satellite constellations. Terrestrial and/or non-terrestrial nodes can benefit from offloading the computing to an orbital edge (OE) platform reachable through the satellite constellation, exploiting its fast and distributed computational capability.

Abstract

Mega-LEO satellite constellations are becoming a concrete reality. Companies such as SpaceX, Virgin Orbit, and OneWeb have already started launching hundreds of LEO satellites and are turning their services on. Even if the aim of such LEO satellite constellations is just, for now, to offer worldwide Internet access equality, their deployment proves their feasibility and suggests usefulness for further purposes. In this article, we shed some light on the possible integration of the in-network computing paradigm in mega-LEO satellite constellations. Terrestrial and/or non-terrestrial nodes can benefit from offloading the computing to an orbital edge (OE) platform reachable through the satellite constellation, exploiting its fast and distributed computational capability. In this context, a preliminary analysis highlights that task offloading strategies can lead to performance improvements that open up novel challenges in the design and setup of OE platforms.

Introduction

New technologies often offer a window into our society to understand how they have integrated themselves into social arrangements and their effects on the development of institutions and social progress. Computing and communication technologies are primary examples of this. Asking how computing platforms will affect equality of opportunity in our society leads us to acknowledge that certain realities fall short of our ideals of life, culture, and gender equality. Therefore, providing access and computing equality is a mission of utmost importance for research.

Satellite and aerial communications have already been advocated as a viable resource to connect unconnected or poorly connected areas [1]. When dealing with mega-low Earth orbit (LEO) satellite constellations, the space industry is promising significant improvement in increasing coverage and reducing latency. Novel payloads could also allow providing data caching and cloud-like computing capabilities at the edge of the network [2]. Several applications could benefit from such a satellite computing infrastructure, namely the orbital edge (OE), in the domains of mobile Internet, the Internet of Things (IoT), and next-generation Tactile Internet. In fact, relative applications are becoming more and more mingled with artificial intelligence (AI) and machine learning (ML) algorithms requiring close location to offload computing tasks.

However, since many customers may require real-time or near-real-time computing operations, the latency to process data on the cloud cannot always be satisfactory, in particular in those regions where terrestrial connectivity is absent and satellites are the only solution. The edge computing paradigm is the new answer to such market and service needs. It allows keeping computation closer to data producer entities, limiting as much as possible the response times. It guarantees the desired quality of service (QoS) without relying on the computational capabilities and energy resources of the end devices, which can be scarce and expensive [3, 4]. The fifth-generation (5G) mobile network and its beyond 5G (B5G) evolution are foreseen as the candidate technologies to enable AI as a service (AIaaS) [5]. In fact, its standardization process also comprises a roadmap for the integration of non-terrestrial networks (NTNs), including both aerial and space segments as key innovation areas of the 3rd Generation Partnership Project (3GPP) [6].

In this article, we envision an edge computing platform that leverages the computing-as-a-service capabilities of LEO satellites to implement the in-orbit computing continuum for equal access to computing. We introduce the edge-cloud continuum concept, including a brief review of OE computing (OEC) and its feasibility. We provide a proof of concept of the reference scenario, while we describe the offloading problem related to the analyzed OE infrastructure. We provide an outlook, through simulations, of how the offloading can be beneficial for OE platforms. Final considerations and open challenges are included.

Edge-Cloud Continuum and Orbital Edge Computing Feasibility

Nowadays, technologies such as mobile, edge, and cloud computing have the potential to jointly make a computing continuum for new disruptive applications. In [7], the authors proposed a model infrastructure for the realization of the mobile-edge-cloud continuum called A3-E. The proposed infrastructure exploits the function as a service (FaaS) computing paradigm to allow stateless and

Digital Object Identifier: 10.1109/MCOM.001.2100818

Pietro Cassará and Alberto Gotta are with the Institute of Information Science and Technologies, Italian National Research Council, Italy; Mario Marchese and Fabio Patrone are with the University of Genoa, Italy.

0163-6804/22/$25.00 © 2022 IEEE

lightweight functions to be autonomously fetched, deployed, and exposed as micro-services by heterogeneous providers. Since distinct providers and infrastructures will not be able to autonomously coordinate and decide who should serve each client request, the A3-E infrastructure enables a mutual client-provider awareness that allows for the opportunistic and context-dependent placement of micro-services along the continuum. The idea behind the edge-cloud continuum is to extend cloud platform capabilities to the network edges, namely near edge (NE) and far edge (FE) based on the distance from the cloud. It supports data processing via the shared pool of computing resources, allowing reduction of the amount of communication data, bandwidth demand on network links, and latency of applications and services.

Traditional satellites are highly customized. The onboard resources are designed for specific applications, and their functionalities cannot be changed during their planned lifetimes, making edge computing hard to apply on them. The authors of [8] proposed an intelligent satellite, called iSat, suitable for satellite edge computing. iSat is a class of multi-purpose satellites with a powerful standardized hardware platform and a fault-tolerant expandable satellite operating system. It can load different apps and share onboard resources with other satellites on demand, providing a more robust and flexible personalized space service.

Even if the joint use of edge computing and cloud paradigms can reduce latency and accelerate computation, this solution may not be sufficient in some scenarios. For example, ubiquitous and high-data-rate sensors spanning large geographical areas may generate high data volumes that cannot be delivered unless the bandwidth from sensors to data centers is proportionally increased. This is the case of nanosatellite constellations with high-data-rate cameras where data processing is performed by a ground station in a centralized way. The ground station location and orbit parameters limit link availability, making effective data rate scalability difficult to achieve. Moreover, intermittent and often unreliable downlinks add latency between data collection and processing, requiring orbital data buffers. OEC can be an efficient alternative in this scenario. In [9], the authors proposed to equip small low-cost satellites with sensors and sophisticated processing hardware to make a CubeSat constellation able to perform data-analysis tasks, providing the performance analysis in terms of volume, mass, energy storage, power, cost, and computing performance to support sophisticated image processing with deep learning. In [2], the authors discussed the feasibility of OEC highlighting the challenges to deploy, operate, and maintain in-orbit computing services. Starting from a reference edge equipment, they assessed the feasibility of boarding such a commodity server on a Skylink payload with a modest increment of weight and volume.

In this article, we start from the findings of the related works briefly presented above, and propose an overview of an offloading strategy for in-orbit computing with a preliminary comparison of simple scheduling techniques to highlight the margins of gain that offloading policies can achieve, thus opening to future more advanced and complex techniques. The proposed comparison is done with different size of the constellation, keeping Starlink as a reference example, and accounting for a lightweight and efficient computing platform, largely adopted for ML computing tasks.

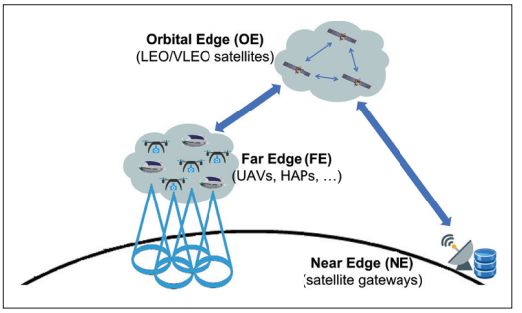

FIGURE 1. Reference scenario with the considered OE architecture implementation.

Reference Scenario

The envisioned scenario, depicted in Fig. 1, outlines the implementation of an in-network computing architecture overlaid on an NTN made up of three main computing and communication entities from the core network viewpoint:

- Autonomous vehicles (AVs), such as ground, sea, and aerial vehicles, are equipped with different kinds of sensors, such as inertial measurement units (IMUs) and cameras [10], representing the FE.
- A constellation of LEO or very low Earth orbit (VLEO) satellites are equipped with both a communication payload and a computing unit but with constrained processing capacity, representing the OE.
- A set of satellite gateways provide access to dense servers of the data centers capable of intensive processing and storage, representing the NE. This entity does not concur with the in-orbit computation but provides only connectivity and computing continuity from OE to cloud data centers.

According to the recent trend that fosters the deployment of mega-LEO satellite constellations, the complexity of facing such a high number of satellites and scheduling communication and computing tasks is outstanding. In such a scenario, each FE node can be visible to a limited number of LEO satellites at a time. Considering the high relative speed between FE nodes and LEO satellites, this set of satellites dynamically changes over time.

We assessed the possible load that satellites can receive in terms of the number of FE nodes located in each satellite coverage area in order to highlight the typical low usage of satellite resources for most of the time. Results were obtained with a C++ based simulator where satellite positions are computed following the SGP4 simplified perturbation model. 100,000 FE nodes have been considered and spread worldwide depending on the current world population dataset available in [11]. A minimum inclination angle of 40° between FE and OE nodes has been considered to decide when a couple of FE-OE nodes are visible.

Figure 2 shows the distribution of FE nodes: the greater the radius, the greater the node density based on the user density over the world map. This figure provides a qualitative idea of how the satellites can be affected by the FE data traffic.

We can infer from Fig. 2 that at least 70 percent of the satellites are in no or low usage due

We assessed the possible load that satellites can receive in terms of the number of FE nodes located in each satellite coverage area in order to highlight the typical low usage of satellite resources for most of the time. Results have been obtained with a C++ based simulator where satellite positions are computed following the SGP4 simplified perturbation model.

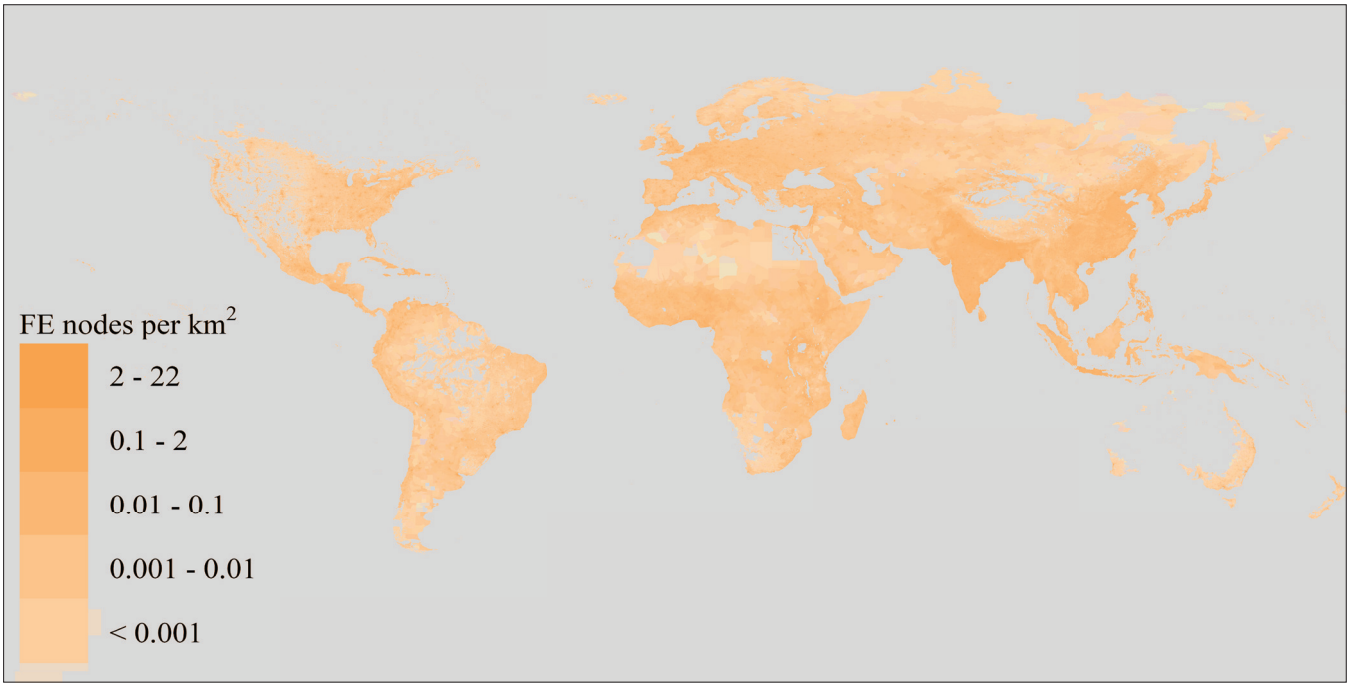

FIGURE 2. Areas where satellites would be active in terms of number of underlying FE nodes per square kilometer (100,000 FE nodes in total).

to the presence of large areas without, or with a low number of, possible users. During all the time that satellites are not travelling above FE nodes, their computational capabilities are not directly exploited. On the contrary, when they pass above crowded areas, their capabilities are highly stressed and might turn out to be not enough to satisfy all requests, leading OE nodes, in turn, to offload tasks to the cloud through the NE nodes. Such a waste of resources is also mentioned in [2], where the authors provided a coverage picture of the Starlink topology.

We prove that task offloading among neighbouring satellites is feasible and leads to better exploitation and a more homogeneous distribution of all tasks among OE nodes.

Offloading Strategy

Exploiting the available information about current network status and possible estimations of its evolution in the near future can help the task offloading process to better exploit the overall available distributed resources. This aspect is a matter of primary importance, especially when the satellite constellation size increases and the planned maximum number of supported users is higher.

To assess this, we consider four scenarios with different LEO satellite constellation networks:

1. 66 satellites equally spaced in 6 circular orbits with 11 satellites each, satellite altitude 781 km, orbital plane inclination 86°
2. 180 satellites equally spaced in 18 circular orbits with 10 satellites each, satellite altitude 1000 km, orbital plane inclination 86°
3. 360 satellites equally spaced in 18 circular orbits with 20 satellites each, satellite altitude 1000 km, orbital plane inclination 86°
4. 1584 satellites equally spaced in 72 circular orbits with 22 satellites each, satellite altitude 550 km, orbital plane inclination 53°

The constellation of Scenario 1 is set with the same number of satellites and orbital parameters as the Iridium constellation. This choice aims to assess the possible obtainable performance of the traditional satellite constellations if they were equipped with in-network computing capabilities. On the other hand, the constellation of Scenario 4 is set with the same planned number of satellites and orbital parameters as the Starlink phase 1 constellation as a realistic example of near-future satellite constellations. Both Scenarios 2 and 3 have a number of satellites and orbital planes between Scenarios 1 and 4 in order to provide insights on intermediate configurations.

In each of these scenarios, the FE nodes generating tasks have been considered proportional to the number of satellites and spread throughout the world in line with the distribution information shown in Fig. 2.

For the task offloading process, we consider three different task offloading strategies:

- Round-robin (RR): Satellites offload the tasks directly received from FE nodes to one of the four neighbour satellites at one-hop distance following a simple RR policy. Offloading events take place only when a satellite cannot process the received task by itself because it currently does not have enough available resources.
- Full offloading (FO): Satellites always offload the tasks directly received from FE nodes to one of the four neighbor satellites at one-hop distance following a simple RR policy. Offloading events take place even when the satellites that directly receive tasks from FE nodes have enough available resources.
- Fuzzy: Satellites offload the tasks directly received from FE nodes to one of the four neighbor satellites at one-hop distance following a fuzzy-logic-based policy [12]. Offloading events take place following the indications of the fuzzy logic that are related to the current status of the network and its estimated evolution.

We decided to use fuzzy logic as a first step to exploit knowledge of the network in terms of different parameters. Without going into too much detail, with the fuzzy-logic-based offloading strategy we consider exploiting information about the satellite's overall and currently available resources to estimate the delivery latency of each task considering all the possible offloading choices. This information is related to knowledge about the currently available CPU computation, storage memory, and energy consumption of each selectable satellite and the estimation of the evolution of these variables in the near future. The consequent output fuzzy variables indicate which of the four one-hop neighbor satellites is the most suitable to guarantee the minimum latency (i.e., will be able to process the task before the other satellites and will have enough available resources at the estimated task processing time).

Performance Analysis

The performance analysis is based on the four scenarios with the three offloading solutions described earlier, where the number of FE nodes that generate tasks are proportional to the number of satellites and spread throughout the world in line with the distribution information shown in Fig. 2. Only 30 percent of the satellites collect tasks from the relative FE nodes, while the other 70 percent are left available to compute tasks eventually received from one of the four neighbor satellites at one-hop distance through inter-satellite links. Each FE source node generates tasks following a Poisson distribution with different λ_i parameter for each of the three considered applications (APP_i, i = 1, 2, 3). The simulation design parameters are shown in Table 1. Note that the computing resources are equal for each satellite in every scenario.

In order to properly show both the QoS guaranteed for the user and the consequent network resource consumption, we consider the following three metrics:

- *Latency*: the average time between the task generation and the reception of the processing result by the task generating node
- *CPU utilization*: the average utilization of the CPU of the satellites that receive tasks to process directly from the underlying FE nodes
- *Data rate*: the average data rate of the inter-satellite links considering the transmissions of both tasks to processing and post-processing results

Such metrics have been chosen referring to a lightweight and efficient constrained hardware already used for computing tasks, for example, on unmanned aerial vehicles (UAVs), such as RaspberryPi or nVidia Jetson boards. Indeed, these two boards significantly differ from any other hardware in the use of resources and computing power. However, the present study does not intend to provide exact performance metrics for specific hardware. It aims to show a proof of concept of computing offloading on mass market hardware that could be embedded onboard a satellite for free.

Tests have been performed using Sat-EdgeSim, software developed to model and simulate satellite edge computing environments [13]. The results of the performance evaluation achieved for the different apps and the different numbers of satellites per constellation are collected in Table 2.

Parameter	APP1	APP2	APP3
Input task size (MB)	10	20	5
Output task size (MB)	1	2	0.5
Poisson λ	1	2	0.5
Operations per task (MI)	50,000	100,000	25,000
Inter-satellite max data-rate (Gb/s)	1		
CPU capacity (MIPS)	50,000		
Number of core per CPU	8		
Storage capacity (TB)	1		
Battery capacity (Wh)	20		
FE nodes per scenario	(300, 818, 1636, 7200)		
Simulation duration (h)	1		

TABLE 1. Configured simulation parameters.

General Considerations and Open Challenges

Results in Table 2 show that a technique based on an optimization logic, like fuzzy, compared to simple deterministic logic, like RR, can enhance the performance of a mega-LEO satellite constellation network in terms of the three metrics considered in this work. But this is more evident looking at the latency metric for different numbers of satellites per constellation. An optimization based on the considered fuzzy logic is able to reduce the latency between 38 and 51 percent for APP1, between 41 and 51 percent for APP2, and between 16 and 40 percent for APP3. From Table 2, we can argue that this kind of control policy allows significantly reducing the task processing latency. Even if it is not trivial to understand, given the dynamism of both the mega-LEO satellite network topology and its links' status, adopting a control policy for tuning both the task offloading and load computing, we can achieve minimization of latency.

Improving the computing sharing in a distributed platform can allow achieving both a significant reduction of satellite launches and a lower computational load on ground resources. This is also going to affect the economic aspect of deploying an edge computing satellite constellation. Satellites are becoming cheaper and cheaper to build and launch thanks to the miniaturization of electronics that allows producers to build objects with the same, or even more, available resources than before but lower-weight. By deploying a lower number of satellites and guaranteeing the same service quality is another money-saving factor, and also of primary importance from the sustainability viewpoint. The importance of efficient techniques based on AI mainly lies in the possibility of allowing these resource-constrained devices to cooperate among themselves and with terrestrial gateways to ensure quality and efficiency of services to a foreseen increasing number of users.

Such an outcome fosters the investigation of other stochastic or learning-based techniques, such as actor-critic and deep reinforcement learning [14, 15] in such an application scenario, keeping the work presented here as a reference baseline. In addition, more objective functions could be consid-

Constellation	Approach	Latency (s)			CPU utilization (%)			Data rate (Mb/s)		
		APP. 1	APP. 2	APP. 3	APP. 1	APP.2	APP. 3	APP. 1	APP. 2	APP. 3
66	RR	9.3	11.04	7.64	52.36	64	43.68	265	290	221
	FO	7.6	9.58	5.64	40.73	48.24	26.46	331	362	276
	FU	4.53	5.43	4.69	34.17	38.21	17.65	277	290	245
180	RR	9.51	11.51	7.89	53.25	65.48	44.29	263	292	219
	FO	8.2	10.01	5.77	40.9	49.93	27.16	329	365	274
	FU	4.84	5.68	4.73	34.34	40.5	17.95	277	312	253
360	RR	9.68	12.5	8.24	53.46	67	45.21	267	288	220
	FO	8.47	10.31	5.93	41.89	51.45	27.69	334	360	275
	FU	5.12	6.04	4.92	33.69	42.54	18.11	283	308	247
1584	RR	9.86	12.7	8.45	54.69	68.16	46.23	269	286	220
	FO	8.73	10.64	6.03	42.02	53.09	28.37	336	358	275
	FU	5.36	6.25	5.06	33.94	43.87	18.23	286	309	252

TABLE 2. Obtained numerical results.

ered to further optimize the pay-off of the services, and different layers of a space information network could further share the computational tasks, aiming to further improve the service performance by exploiting a wider set of available network resources and interconnections among nodes.

References

[1] I. del Portillo *et al.*, "Connecting the Other Half: Exploring Options for the 50% of the Population Unconnected to the Internet," *Telecommun. Policy*, vol. 45, no. 3, 2021, p. 102,092.

[2] D. Bhattacherjee *et al.*, "In-Orbit Computing: An Outlandish Thought Experiment?," *ACM 19th Wksp. Hot Topics in Networks*, 2020, pp. 197–204.

[3] L. Bittencourt *et al.*, "The Internet of Things, Fog and Cloud Continuum: Integration and Challenges," *Internet of Things*, vol. 3, 2018, pp. 134–55.

[4] T. Qiu *et al.*, "Edge Computing in Industrial Internet of Things: Architecture, Advances and Challenges," *IEEE Commun. Surveys & Tutorials*, vol. 22, no. 4, 2020, pp. 2462–88.

[5] M. Bacco *et al.*, "Networking Challenges for Non-Terrestrial Networks Exploitation in 5G," *IEEE 2nd 5G World Forum*, 2019, pp. 623–28.

[6] P. Saxena *et al.*, "Resilient Hybrid SatCom and Terrestrial Networking for Unmanned Aerial Vehicles," *IEEE INFOCOM Wksps.*, 2020, pp. 418–23.

[7] L. Baresi *et al.*, "A Unified Model for the Mobile-EdgeCloud Continuum," *ACM Trans. Internet Technology*, vol. 19, no. 2, 2019, pp. 1–21.

[8] Y. Wang *et al.*, "Satellite Edge Computing for the Internet of Things in Aerospace," *Sensors*, vol. 19, no. 20, 2019, p. 4375.

[9] B. Denby and B. Lucia, "Orbital Edge Computing: Machine Inference in Space," *IEEE Computer Architecture Letters*, vol. 18, no. 1, 2019, pp. 59–62.

[10] M. Bacco *et al.*, "UAVs and UAV Swarms for Civilian Applications: Communications and Image Processing in the SCIADRO Project," *Int'l. Conf. Wireless and Satellite Systems*, Springer, 2017, pp. 115–24.

[11] NASA Socioeconomic Data and Application Center, "Gridded Population of the World, Version 4"; https://doi.org/10.7927/H4JW8BX5, 2020.

[12] C. Sonmez, A. Ozgovde, and C. Ersoy, "Fuzzy Workload Orchestration for Edge Computing," *IEEE Trans. Network and Service Management*, vol. 16, no. 2, 2019, pp. 769–82.

[13] J. Wei *et al.*, "SatEdgeSim: A Toolkit for Modeling and Simulation of Performance Evaluation in Satellite Edge Computing Environments," *IEEE 12th Int'l. Conf. Commun. Software and Networks*, 2020, pp. 307–13.

[14] M. Á. Vázquez *et al.*, "Machine Learning for Satellite Communications Operations," *IEEE Commun. Mag.*, vol. 59, no. 2, Feb. 2021, pp. 22–27.

[15] G. F. Anastasi *et al.*, "A Hybrid Cross-Entropy Cognitive-Based Algorithm for Resource Allocation in Cloud Environments," *IEEE 8th Int'l. Conf. Self-Adaptive and Self-Organizing Systems*, 2014, pp. 11–20.

Biographies

Pietro Cassarà (pietro.cassara@isti.cnr.it) (M.Sc. 2005, Ph.D. 2010) is a staff member of the Institute of Science and Information Technologies (ISTI) at the National Research Council (CNR), Pisa, Italy, and since 2017 he has been a temporary staff member of the CMRE lab at NATO of La Spezia. His research interests include machine learning and optimization theory for wireless sensor network and IoT communications. He has been participating in European, ESA, and national funded projects.

Alberto Gotta (alberto.gotta@isti.cnr.it) (M.Sc. 2002, Ph.D. 2007) is a researcher at ISTI, CNR. He is member of the IEEE International Network Generations Roadmap (INGR) Satellite Working Group. His research interests include traffic engineering applied to satellite, machine-to-machine, and UAV networks. He has participated and leaded several EU, national, and ESA funded R&D projects. He has co-authored more than 80 papers and has served on the TPCs of flagship ComSoc conferences.

Mario Marchese (mario.marchese@unige.it) (M.Sc. 1992, Ph.D. 1997) is a full professor of telecommunication networking at the University of Genoa, Italy. He is author/co-author of more than 350 scientific works including international magazines, international conferences, book chapters, patents, and books. His main research activity concerns: networking, quality of service over heterogeneous networks, software defined networking, satellite DTN and nanosatellite networks, and networking security.

Fabio Patrone (f.patrone@edu.unige.it) (M.Sc. 2013, Ph.D. 2016] is an assistant professor in the Satellite Communications and Heterogeneous Networking Laboratory at the University of Genoa. His main research activity involves routing, scheduling, and congestion control algorithms in satellite, vehicular, and sensor networks, and the employment of networking technologies, such as network function virtualization and software defined networking for the integration of these networks with the terrestrial infrastructure within 5G.