

# Performance Evaluation of a Satellite Communication-Based MEC Architecture for IoT Applications

MICHELE LUGLIO

University of Rome “Tor Vergata,” Rome, Italy

MARIO MARCHESE , Senior Member, IEEE

FABIO PATRONE , Member, IEEE

University of Genoa, Genoa, Italy

CESARE ROSETI 

FRANCESCO ZAMPOGNARO

University of Rome “Tor Vergata,” Rome, Italy

**New scenarios and use cases are raising following the birth of the fifth generation of mobile communications. The Internet of Things (IoT) is one of the main use cases which are growing, leading to a massive amount of data that need to be exchanged throughout the Internet. Satellite communication networks are essential in remote and isolated environments and can support fully connected environments by offloading the terrestrial infrastructure concerning delay-tolerant traffic flows. However, satellite network resources are limited and expensive, so they need to be carefully used in order to avoid waste and satisfy the required user performance. The multi-access edge computing (MEC) concept can be exploited in this context to allow data preprocessing at the edge, i.e., close to the users, so reducing the amount of data that has to traverse the backhaul satellite link and, in some cases, reducing data delivery times. This article analyses the performance of a satellite architecture in the IoT framework**

Manuscript received 27 October 2021; revised 2 May 2022; accepted 5 August 2022. Date of publication 17 August 2022; date of current version 11 October 2022.

DOI. No. 10.1109/TAES.2022.3199330

Refereeing of this contribution was handled by F. Granelli.

This work was supported by ESA through VIBeS project under Grant 4000122991/18/U.K./ND.

Authors' addresses: Michele Luglio, Cesare Roseti, and Francesco Zampognaro are with the Electronics Department of the University of Rome “Tor Vergata,” 00133 Rome, Italy, E-mail: (luglio@uniroma2.it; roseti@ing.uniroma2.it; zampognaro@ing.uniroma2.it); Mario Marchese and Fabio Patrone are with the Electrical, Electronics and Telecommunication Engineering and Naval Architecture Department (DITEN), University of Genoa, 16124 Genoa, Italy, E-mail: (mario.marchese@unige.it; f.patrone@edu.unige.it). (*Corresponding author: Fabio Patrone.*)

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see <https://creativecommons.org/licenses/by/4.0/>

highlighting the advantages brought by MEC, also including data aggregation and compression techniques.

## I. INTRODUCTION

When looking at the use of the Internet, the landscape of applications and communication protocols has changed substantially in the last years. Nonreal-time information exchange represented most of the Internet traffic in the past, with predominance in the use of web technology and file transfers (e.g., HTTP, e-mail, FTP, and peer-to-peer solutions). Nowadays, real-time interactions are fundamental for a whole new set of applications, from web gaming to VoIP and chat services, from conferencing platforms to cloud and Internet of Things (IoT) services. In particular, massive IoT applications show a relevant growth and are already representing a significant portion of Internet traffic, as discussed for instance in the latest Cisco Annual Internet Report [1]. Many kinds of sensor-based services fall in this category and are applied in a set of scenarios, such as home automation, smart cities, support to logistics, wearable health devices, industrial automation, environmental sensing, and critical infrastructures monitoring and protection. In all these environments, it is often important to consider large coverage and high mobility requirements. An example may be a sensor network which spans across many administrative domains and covers remote areas.

A massive growth and spread of devices, such as sensors and actuators, has been envisioned for 2021 and beyond, with a consequent big mass of data that need to be collected, processed, and, in some cases, displayed to users remotely connected to the Internet.

Satellite-based approaches, such as the space-air-ground integrated networks [2], are considered as viable solutions thanks to their ubiquitous coverage and broadcast/multicast capabilities. Considering the IoT context and the associated connectivity requirements, satellite access can play a significant role in enabling IoT services [3]. Satellite communications (SatCom) networks are particularly suitable to offer connectivity to a high number of devices located in wide areas, complementing terrestrial infrastructure thanks to the large coverage. They can support cross-the-borders IoT scenarios properly managing their available network resources to guarantee the required quality of service (QoS) [4]

In particular, Cioni et al. [3] emphasize the role of satellites in the foreseen spread of IoT devices and applications in the future 5G ecosystem focusing on the challenges related to the massive machine-type communication (mMTC) service provision, air interface, and network aspects related to the system design. Energy efficiency is another important aspect to take into account. A network coding (NC)-based approach has been proposed in [5] to properly manage the available network resources and offer reliable IoT multicast services in a 5G/satellite network. Key network challenges for satellite exploitation in future 5G and IoT satellite-based sensor networks are also presented in [6].

Assuming a massive amount of data generated by an IoT sensor network composed of thousands of IoT devices, the overall volume of raw data can lead to the rise of several issues. An example is the security and data privacy issue, which involves both legal [7] and technical matters [8] related to different aspects, such as how the network is managed and how data are stored and processed. Another example is related to how the network resources are efficiently exploited to offer the required service and avoid unpleasant situations, such as the saturation of the satellite link bandwidth.

The presence of multiaccess edge computing (MEC) functionalities close to the sensor network allows saving SatCom resources and offering enhanced performance to the users thanks to smart management of large set of data by enabling local processing and in-network functions (i.e., microservices) tailored on the specific use case.

The MEC concept, formerly mobile edge computing, has been introduced with the main aim to move the intelligence, or at least part of it, from the network core (centralized datacenters and cloud platforms) to the network edge (distributed microdatacenters and edge platforms). MEC contributes shaping the next generation network architecture and infrastructure [9], [10], thanks to its applicability in different scenarios and use cases, including IoT [11], [12].

Concerning MEC application in satellite networks, different possible network architecture configurations, challenges, and open issues have been discussed [13]. Possible solutions have been implemented and tested focusing on the integration within the 5G environment [14], based on different kinds of satellites (including nanosatellites [15]), and aimed at improving the user perceived QoS [16], [17].

IoT applications implementing satellite edge computing solutions are the object of [18], [19], [20], also considering the severe resource limitations that affect IoT devices. Latency and energy consumption reductions are the main aims of the studies [21], [22], [23], which also adopt machine learning techniques to make resource management more effective.

This article addresses the integration of satellite and terrestrial networks to interface IoT sensors and efficiently enable IoT services through the Internet. The considered solution includes a satellite backhaul link based on a geostationary earth orbit (GEO) satellite, to offload the terrestrial network of delay-tolerant data, and the employment of the MEC paradigm, to achieve and optimize a fully integrated and global IoT service, i.e., to offer an architectural and operational solution able to guarantee the typical performance requirements of IoT applications worldwide, overcoming limitations of different kinds, such as geographical and data management on vast scale. This solution is assessed in order to quantify and highlight the performance improvement obtained implementing the MEC concept to enhance the satellite transmission, by both performing data compression to save satellite bandwidth, as well as distributing and aggregating data in an efficient way in relation of the domain where it is useful (i.e., local or remote).

The rest of this article is organized as follows. Section II highlights the roles that SatCom networks can have in the IoT applications. Section III presents the used MEC-enabled satellite architecture and the considered network configurations. The obtained results are reported in Section IV. Finally, Section V concludes this article.

## II. ROLE OF SATELLITES IN IOT APPLICATIONS

Satellites can have two main alternative roles in a typical satellite/terrestrial integrated network for IoT applications: direct-access (fronthaul) or backhaul.

In the first case, sensors are directly linked to the satellites, which are the first hop of the communication path between IoT devices and users. Third Generation Partnership Project defined six different reference scenarios [24] changing the altitude of the considered satellites (GEO and non-GEO), the complexity of the carried satellite payload (transparent or regenerative), and the flexibility of the generated beams (steerable or moving beams) for direct-to-satellite access (fronthaul). The most promising approaches are mainly considering low earth orbit (LEO) and very low earth orbit satellites, even if intrinsic challenges have still to be overcome, such as the heterogeneity of the IoT solutions and the high energy consumption of IoT devices. Some ongoing activities are investigating these issues to assess the feasibility of direct-to-satellite access, such as [25] and [26].

In the second case, satellites can be adopted as backhaul to the Internet or to a dedicated service center. Sensors' data are collected within an edge cloud in the terrestrial access network and then made available through a satellite data network backhaul by using IP-based protocols. High throughput satellites and very high throughput satellites GEO platforms operating at Ka or higher frequency bands are the main options. They can considerably reduce the communication costs providing large amounts of bandwidths, making satellites a viable solution to enhance the overall network capability.

The employment of communication satellites for IoT can be beneficial for multiple applications. Several solutions have been developed considering the possible different roles of satellites and tested in different application scenarios [27]. Satellites operating as backhaul support have been considered for the smart city [28] and the road safety and autonomous ship scenarios [29]. Smart agriculture [30], smart grid [31], and maritime IoT [32] are three other typical use cases where the use of satellites has been considered for IoT applications and has been proven to be effective, not only for the lack of communication infrastructures based on different technologies.

From the application viewpoint, data generated by devices need to be transferred to receivers by using a suitable protocol. Two main communication models can be employed: "publish/subscribe" or "client/server," depending on the use case. Although in principle, it is possible to exchange such data in non-IP/proprietary formats and associated proprietary transport and application protocols,

it is worthwhile to consider the IP-based approach. IP allows enabling interoperability and reusability, leveraging well established and freely available IP-based applications and protocols. The two main IP-based solutions are Message Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP).

MQTT [33] was originally an IBM proprietary protocol now managed as a standard by the International Organization for Standardization. MQTT uses a publish/subscribe model and requires an MQTT broker to manage and route messages among MQTT nodes through point-to-point TCP permanent connections. MQTT has not been designed for “constrained” devices, so power saving optimization and tradeoffs among complexity/efficiency and bandwidth usage are usually not considered.

CoAP [34] is a lightweight IP-based protocol for sensor data communication. It is part of the Constrained RESTful Environments (CoRE) IETF working group and it aims to realize a REpresentational State Transfer (REST) architecture in a suitable form for the most constrained nodes. CoRE is a very efficient RESTful protocol based on the classic client/server model. Even if DLTS, SMS, and TCP bindings are defined as transport for CoAP, UDP-based implementations are the most commonly used.

Studies have been carried out to evaluate the employment of MQTT and CoAP in the satellite backhaul case [35]. Efficient configurations for both MQTT and CoAP have been highlighted and some modifications have been proposed. For example, MQTT and CoAP employment to support sensor data exchange on a satellite random access channel based on the DVB-RCS2 standard has been investigated in [36].

### III. MEC-ENABLED IOT ARCHITECTURE

#### A. Reference Architecture Description

The locations where MEC functionalities can be typically implemented in satellite networks are three: on the ground close to the users, on-board satellites, and on the ground close to the satellite ground stations. Implementing MEC on-board satellites require regenerative payload satellites with largely available computation and storage resources and an internal structure designed to make their allocation flexible and scalable. Companies are planning to launch this kind of satellites, which will be available in the near future. However, a first step to deploy and test satellite MEC networks could be to exploit the already deployed transparent payload satellites. It is important to quantify the achievable performance improvement in different possible IoT use cases with different and variable network and traffic flow features, such as number and density of IoT devices, distribution of generated data flows, and minimum performance requirements.

We consider exploiting the MEC concept and a satellite link to enable IoT scenarios in remote areas. In particular, we focus on GEO-based communication systems, such as Inmarsat, configured as backhaul. Unmanned aerial vehicles (UAVs) could also be considered and included in the

reference scenario to help collect data from the sensors. However, we decided to not include them since our main focus is on the backhaul link which, especially for rural and remote areas, can be more realistically assumed as a satellite link, while UAVs can be more useful for the access link. Besides, the presence of UAVs would not affect the application of our proposed MEC functionalities and the effect on the shown performance would be negligible, because the main improvement comes from a local management of data.

Fig. 1 shows the considered reference network architecture, which was validated within the European Space Agency (ESA) funded project M2MSat.<sup>1</sup> A large and widespread set of IoT devices, called sensor equipment (SE), are the data sources of this sensor network scenario. They are linked to a base station located in the remote area through wired or wireless links. The base station aims to collect the data generated by the SEs in order to send them to the Internet through the satellite backhaul link. An IoT cloud service, located on the Internet, collects and stores all SE data and offers data access to authorized users, called user equipment (UE). UEs can be located on the Internet (on the right-hand side of the satellite link, called core side) or connected directly from within the service provider network (on the left-hand side of the satellite link, called edge side). In this way, the satellite offers a backhaul solution that can completely replace the terrestrial connectivity to the cloud services.

It is assumed, as well for the reference service, that not all sensor data have the same scope and priority. In fact, it is possible to consider several use cases in which some remote data collected into a remote area can just be used for local data processing, whereas only alarms or downsampled values are delivered toward a central node. Improve satellite resource management and computation offloading is the main aim of most studies on this topic, such as [37] and [38].

In this context, MEC can foster satellite bandwidth reduction through the implementation of dedicated functionalities, such as data aggregation and local processing at the edge, with a consequent lower cost due to the reduced amount of data that need to traverse the satellite link. The proposed reference architecture may in fact include storage, computation resources, and data local breakout functionalities, which are part of the network operator infrastructure, but not necessarily present at the edge of the network and not enabled for MEC. For instance, thanks to the presence of network functions and components for the termination of the IoT protocol stack within the network operator’s core network, a possible edge computing infrastructure and the associated components can be conceived and enabled in a straightforward way. These dedicated components can be optimized for IoT and deployed dynamically, e.g., as virtual network functions, allowing to manage and terminate the local data-application protocols before the transmissions over satellite, as discussed in the next section.

<sup>1</sup>[Online]. Available: <https://artes.esa.int/projects/m2msat>

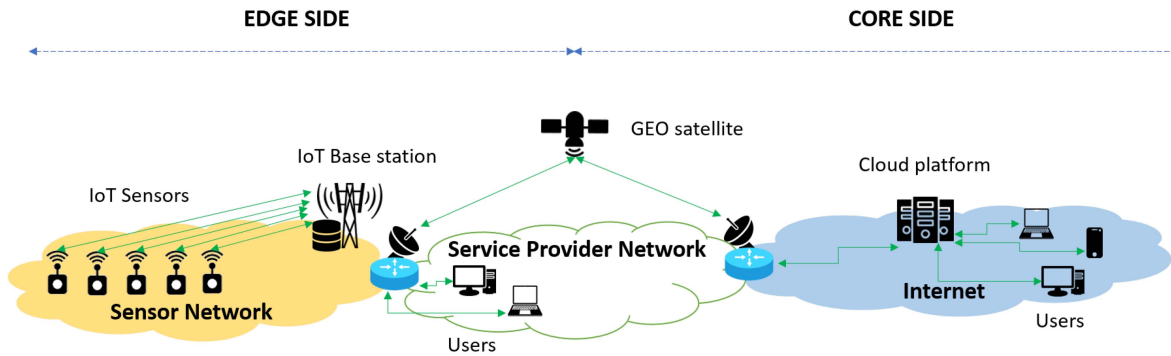


Fig. 1. Reference hybrid network to enable IoT cloud services via satellite.

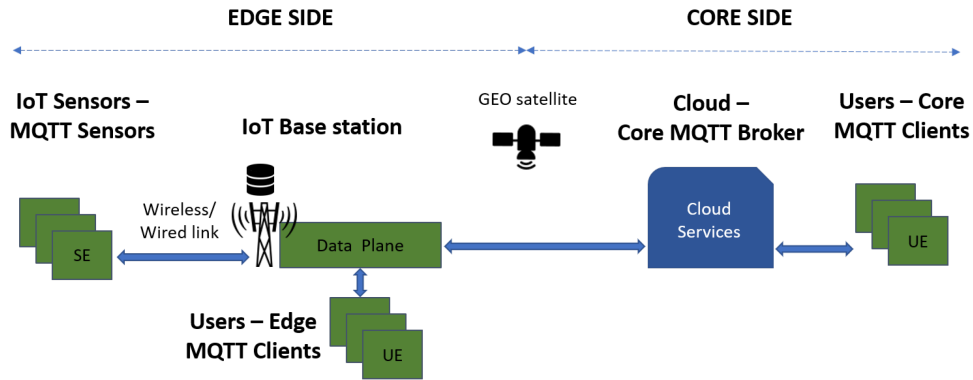


Fig. 2. IoT application level network configuration. Case 1: IoT MQTT native sensors, no edge computing. All MQTT clients access the core MQTT broker, edge MQTT clients requests and responses need to traverse the satellite link.

## B. Possible Network Configurations and MEC Operations

In relation to current standardization activities and on the basis of the state of the art and research projects [25], [39], [40], having Fig. 1 as a reference, we introduce three possible configurations, leveraging data local breakout. The local breakout means that the operator network is supporting the split of data sessions for the sensor data, delivering such data in different configurations according to the enabled MEC options (if any). The goal is to present a progressive performance improvement first due to the enabling of MEC, and then the enabling of satellite-oriented ad hoc optimizations, which are tailored on the data type and meaning. In particular data compression, aggregation, suppression, and differential distribution will be introduced as edge operations in the last configuration.

1) *Case 1 (see Fig. 2)*: SEs are IoT MQTT sensors. Due to the large number of involved sensors and users, we consider the use of MQTT because it scales better with a large population of nodes, thanks to a broker node which manages the data exchange between data sources and users. Edge computing is not included in this case. IoT MQTT sensors generate data destined to the centralized MQTT broker (core MQTT broker) located on the core side and hosted by cloud services. MQTT clients, which are located both on the edge

and core sides (edge and core MQTT clients, respectively), request data to the MQTT broker. Besides, no additional functionalities dedicated to satellite bandwidth optimization and data preprocessing are included.

2) *Case 2 (see Fig. 3)*: SEs are IoT non-MQTT sensors. They get access to the network through different interfaces (e.g., LoRaWAN) equipped with their own application layer protocols. MEC is partially added: an edge MQTT broker is introduced at the edge side with a twofold aim. On the one hand, to “convert” the generated data in the MQTT format in order to let them be stored in MQTT brokers and, consequently, allow MQTT clients to access them. On the other hand, to provide local data storage at the edge side. To guarantee the compatibility with the core MQTT broker, the edge MQTT broker has to be configured to act as a transparent proxy to the target core MQTT broker, i.e., redirecting the traffic to it after locally storing a copy of each content. Edge MQTT clients can directly ask the edge MQTT broker to provide the interested contents so reducing the amount of data that need to traverse the backhaul satellite link in both directions: MQTT requests from edge MQTT clients to the core MQTT broker and MQTT answers from the core to the edge side. Also in this case, no additional functionalities aimed at

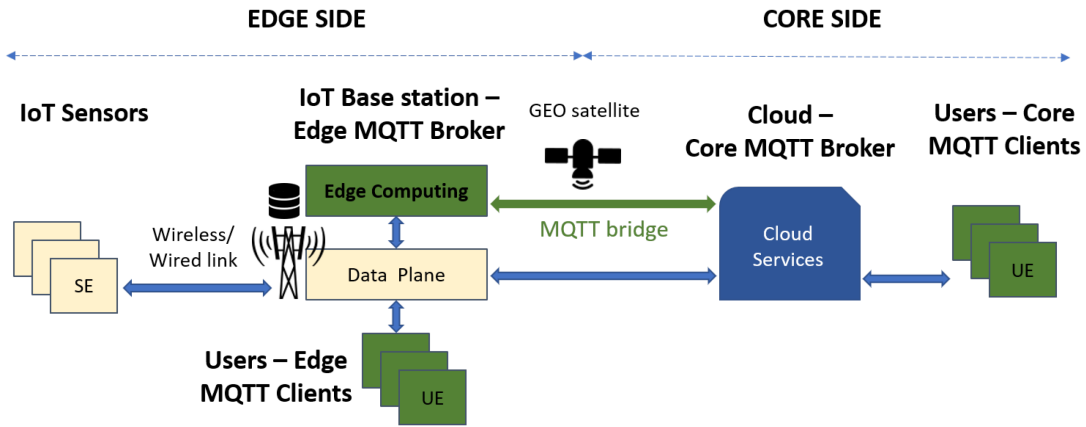


Fig. 3. IoT application level network configuration. Case 2: IoT non-MQTT sensors, edge MQTT computing in the IoT base station. Presence of an edge MQTT broker to store information made available to the edge MQTT clients that, in this way, do not need to access the core MQTT and so forward their request through the satellite link.

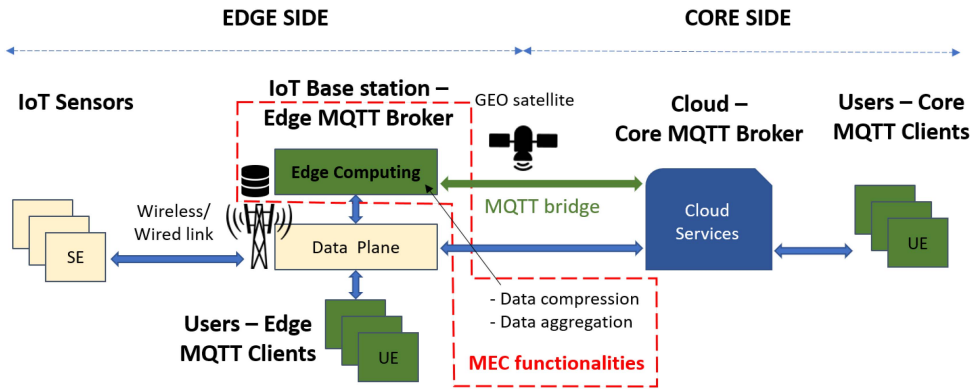


Fig. 4. IoT application level network configuration. Case 3: IoT non-MQTT sensors, edge MQTT computing in the IoT base station. Presence of an edge MQTT broker as in Case 2 + additional edge functionalities to improve satellite bandwidth utilization.

satellite bandwidth utilization improvement and data preprocessing are included. MEC functionalities are limited to the presence of the edge MQTT broker.

- 3) *Case 3 (see Fig. 4)*: SEs are IoT not-MQTT Sensors, and the network architecture includes both the core and edge MQTT brokers as in Case 2. The edge computing component also enables a data compression and data aggregation functionality. In detail, MEC functionalities include also: 1) local processing, such as downsampling and duplicated values suppression, to perform data compression; 2) data aggregation, such as aggregating different values by collecting a given amount of messages over time in a single MQTT message to reduce the protocol header overhead. This type of MEC functionalities are already proposed in [41]. As a step ahead with respect to this work, we perform an accurate performance evaluation by using real MQTT software implementation on a Linux-based testbed.

#### IV. PERFORMANCE EVALUATION

##### A. Testbed Setup

The testbed used to perform the validation over the proposed scenarios is composed of a set of Linux Virtual

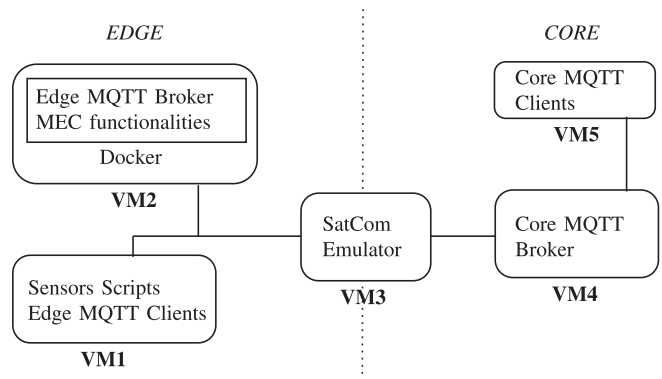


Fig. 5. Testbed configuration by using Linux-based VMs.

Machines (VMs), as shown in Fig. 5. VM3 acts as a router to emulate a GEO very small aperture terminal SatCom connection. The considered MEC functionalities are implemented as a set of Docker-based applications in VM2. The Docker image deployed to enable the edge MQTT broker is based on the Mosquitto image.<sup>2</sup> Additional configurations are added to the network in VM2 to ensure that MQTT

<sup>2</sup>[Online]. Available: [https://hub.docker.com/\\_/eclipse-mosquitto](https://hub.docker.com/_/eclipse-mosquitto)

TABLE I  
Configuration Variables and Reference Values Used in the Tests

Parameter	Variable	Reference Value
Satellite uplink (Edge to Core) datarate	$BW_{up}$	128 Kbit/s
Satellite downlink (Core to Edge) datarate	$BW_{down}$	128 Kbit/s
Satellite two-ways average delay	$RTT$	580 ms
Number of IoT or MQTT Sensors - SE	$N_S$	5000
Number of real-time sensors	$r$	2000
Real-time "local-only" sensors	$r_l$	640
Real-time "global" sensors	$r_g$	1360
Number of non-real-time sensors	$n$	3000
Non-real-time "local-only" sensors	$n_l$	960
Non-real-time "global" sensors	$n_r$	2040
Mean message size	$s_1$	150 Bytes
Mean sensor inter-generation message time	$t$	240 s
Number of Edge MQTT Clients - UE (each interested in 10% of all sensor data)	$N_{CE}$	10
Number of Core MQTT Clients - UE (each interested in 10% of all sensor data except for "local-only" type)	$N_{CC}$	40

traffic can be transparently intercepted and locally processed by the edge MQTT broker without requiring sensor reconfigurations.

At the core side, the MQTT broker is configured in VM4 and acts as the reference backhand on the cloud to collect and dispatch sensor data to the clients, which can be located both at the edge side (VM1) and at the core side (VM5).

Clients are coded in C by using open-source Mosquitto libraries. Sensors are modeled with scripts that generate data messages with uniform distribution and settable mean value for data size and intergeneration interval. Data messages from sensors then can be either directly included within MQTT messages by the SEs (Case 1) or encapsulated in MQTT messages by the edge MQTT broker before transmitting them to the Cloud (Cases 2 and 3).

In our tests, we define four different sensor categories: real-time, nonreal-time, local-only, and global sensors. Real-time sensors generate time critical data (e.g., alarms) that should be delivered as fast as possible without additional processing. On the contrary, nonreal-time sensors provide data whose validity is acceptable also in the case of deferred or partial delivery (e.g., a noncritical temperature value). Both real-time and nonreal-time sensor data can be of local-only type, i.e., they will be required only by edge MQTT clients, or global type, i.e., they will be required by both edge and core MQTT clients.

Table I reports the reference values of the configuration variables. Such values are chosen to get realistic configurations. The basic reference is the IoT scenario reported in the Ericsson mobility report [42] with additional assumptions to implement the satellite scenario.

## B. Assessing the Proposed Optimizations

The obvious expectation is that enabling the edge MQTT broker as an edge function will reduce the amount of traffic forwarded through the satellite link. Enabling the additional MEC functionalities should reduce even more the impact of sensor traffic on the satellite link, due to: 1) the reduction of the amount of data that have to reach the

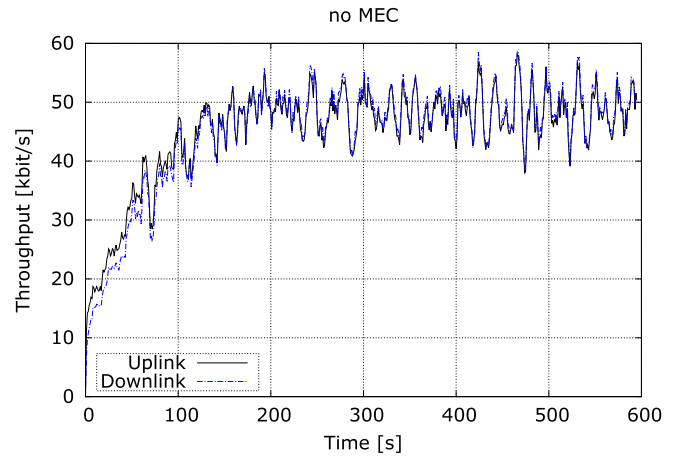


Fig. 6. Satellite link bandwidth utilization for Case 1: MEC is not applied.

core MQTT broker and then go back to the edge MQTT clients owing to the edge MQTT brokering functions; 2) data compression and aggregation functions. Compression is very important for all real-time and very-low latency services where sensor data must be quickly available avoiding unnecessary data forwarding through the high-latency satellite link; aggregation is significant for delay-tolerant data. The interesting aspect is to quantify these improvements in a realistic environment.

We have performed real-time tests whose duration is 10 min for each of the three cases defined in Section III and configured as detailed in Table I. The two considered output performance are given as follows.

- 1) *Message delivery time*: The time elapsed between the data message generation by the SE and its reception by the interested MQTT client.
- 2) *Satellite link bandwidth utilization, also called throughput*: The amount of data that traverse the satellite uplink (from edge side to core side) and downlink (from core side to edge side).

In each test, the MQTT clients gradually join the network, which leads to about 200 s of transient time before reaching the regime values of the satellite link bandwidth utilization. For each of the three cases, we show the overall satellite link bandwidth utilization and the distribution of the message delivery separate for edge and core MQTT clients.

The first test, referred to Case 1, allows introducing a benchmark reference for the proposed application. The satellite link bandwidth utilization is reported in Fig. 6.

We can see an almost symmetric use of the channel since all sensor data must reach the only MQTT broker available in the network (the core MQTT broker at the core side) and go back to the edge clients. The values for the average throughput after the initial ramp time are 47.9 kb/s for the uplink and 48.6 kb/s for the downlink. The throughput over time provides further insights on the traffic pattern: after the gradual entry of the sensors, we can see wide variations that must be taken into account to design and dimension



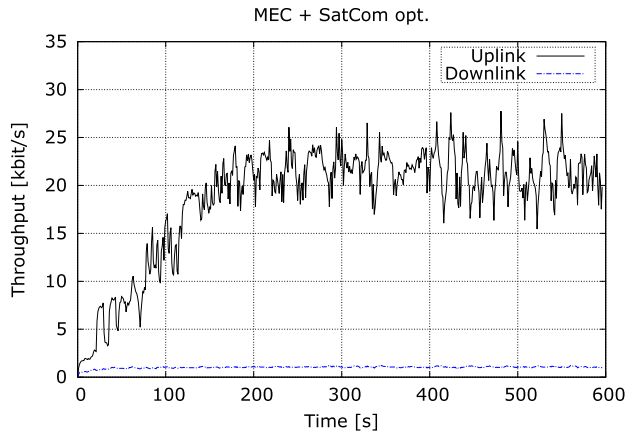


Fig. 10. Satellite link bandwidth utilization for Case 3: MEC is applied, including also data compression to reduce satellite traffic.

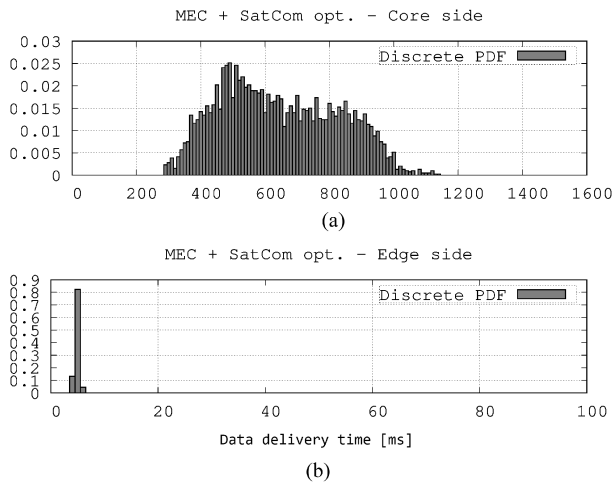


Fig. 11. Delivery time distribution of data destined to the edge and core MQTT clients for Case 3.

times, such as the same temperature value, the data are just sent once, i.e., only the data variations are sent instead of always sending each data; 2) data reformatting—data message format is changed from text to binary in order to add some lossless compression. These functions, colocated with the edge MQTT broker, are so able to reduce redundant information and aggregate sensor data in a much more compact form by using larger TCP packets. These additional actions lead to a further satellite uplink utilization reduction of 12%, with an average value of 21.1 kb/s. At the same time, core side data delivery time is not so much affected, with an average value increased to 647 ms.

### C. Further Testing Configurations

We performed additional tests to determine the effectiveness of the MEC architecture considering alternative configurations. These configurations have been designed with the aim of presenting some limit/stress cases, in which we want to show the scalability of the service, and the validity of the approach when the number of local-only sensors or clients is reduced. Furthermore, even when local data are managed at the edge and distributed only locally, processing

delays and enqueueing times for the MEC processing and MQTT are still present, and are interesting for a large-scale deployment of the proposed system in real networks.

The following parameters have been changed, keeping all the other parameters set as reported in Table I:

- 1)  $N_{CC} = 100$  (higher number of core MQTT clients);
- 2)  $N_{CE} = 0$  (no edge MQTT clients);
- 3)  $N_S = 50\,000$  (10x sensors, keeping the same percentage of real-time and nonreal-time, global and local-only sensors defined in Table I);
- 4)  $r_l = n_l = 0$  (no local-only sensors).

Table II includes the obtained mean results of the additional tests for each of the proposed MEC configurations. The mean results of the tests, already described in Section IV-B, labelled as *default*, have been included too for comparison.

Concerning the increase of  $N_{CC}$ , there is no significant change in the results compared to the default configuration due to the publisher–subscriber nature of the MQTT protocol. The amount of sensor data collected at the core MQTT broker through the satellite link does not depend on the number of core MQTT clients. Therefore, the results of this configuration are not included in Table II.

Without edge MQTT clients interested in sensor data ( $N_{CE} = 0$  and, consequently, no edge side data delivery time results), there is a significant reduction of the satellite downlink bandwidth utilization for Case 1. Since there is no MEC processing, the absence of edge MQTT clients reduces the amount of traffic in the core to edge direction since there is no more requests. For Cases 2 and 3, there is no significant difference in the satellite link bandwidth utilization since all traffic at the edge is handled locally, regardless of the number of interested edge MQTT clients. Besides, in Cases 2 and 3, the amount of traffic delivered through the MQTT bridge from the edge to the core side is the same. For this reason, also core side data delivery time is not affected.

In the second additional test, we introduce a high degree of congestion by increasing the number of sensors by ten times ( $N_S = 50\,000$ ). The uplink becomes quickly congested in all cases, but the introduction of MEC provides a substantial difference. Without MEC, also the satellite downlink is highly occupied, with consequent very high mean data delivery time values both at core and edge sides. Several sensors suffered also disconnections due to the excessive packet loss and TCP retransmissions due to retransmission timeout expiration. As soon as MEC is enabled, the core side performance is guaranteed, obtaining an edge side data delivery time mean value comparable with the default configuration. Case 3 shows a further better performance of the core side data delivery time. This specific configuration allows assessing the reaction of the system in case of congestion: the increase of sensors from 5000 to 50 000 does not lead to a proportional increase of the data delivery time, which instead increases of about 100 times. This conclusion is not applicable, as already discussed, for the edge side data delivery time with MEC enabled, which is not affected.



TABLE II  
Mean Results From the Additional Performed Tests

Variable configuration	Scenario	Uplink bandwidth utilization [Kbit/s]	Downlink bandwidth utilization [Kbit/s]	Core side data delivery time [ms]	Edge side data delivery time [ms]
Default	Case 1	47.9	48.6	325	868
	Case 2	24.6	0.6	587	4.9
	Case 3	21.1	0.7	647	4.9
$N_{CE} = 0$	Case 1	46.6	12.6	296	n.a.
	Case 2	24.6	0.5	588	n.a.
	Case 3	21.4	0.6	648	n.a.
$N_S = 50000$	Case 1	128	93.1	56779	53661
	Case 2	128	3.2	71916	4.8
	Case 3	128	2.9	40547	4.8
$r_l = n_l = 0$	Case 1	33.1	33.6	314	904
	Case 2	24.6	0.5	584	4.8
	Case 3	21.4	0.6	638	4.7

Finally, without local-only real-time and nonreal-time sensors ( $r_l = n_l = 0$ ), we can observe a proportional reduction of the satellite link bandwidth utilization in Case 1. Concerning Cases 2 and 3, there are no significant changes since the local-only traffic flows do not traverse the satellite link. This test allows in particular to confirm the effectiveness of data local management.

The execution of additional runs showed the main benefit of the MEC-based approach in several configurations. According to the specific applications and associated requirements in terms of satellite link utilization and latency, a service operator can decide which is the best architectural configuration depending on the numerical results shown in this article.

## V. CONCLUSION

The huge amount of raw IoT data could stress the limited and expensive SatCom resources, which have to be carefully managed. Equipping the edge of the network with local processing MEC functionalities, such as data compression and aggregation, allow improving end-to-end performance and QoS in communications when satellite links are exploited.

This article has analyzed a SatCom-based MEC network architecture where a GEO satellite acts as a backhaul node between an IoT sensor network and the Internet. A distributed MQTT framework is defined where local instances of MQTT brokers and data aggregation and compression techniques are deployed as Virtual Functions at the edge of the network. Performance evaluation is carried out through an ad hoc designed testbed comparing the obtained performance with three different network configurations in terms of the amount of data that need to traverse the satellite link and data delivery time. The employment of MEC functionalities allows obtaining a significant reduction of the satellite link bandwidth utilization and of the achievable data delivery time in most of the considered cases. The use of data aggregation techniques leads to a higher delivery time in some cases, which can affect delay-tolerant data, so implying a careful design.

Considering the rapid growth of LEO SatCom systems, a second performance evaluation will be performed considering an additional reference scenario based on a LEO satellite constellation. This further analysis aims to show the impact of aspects, such as satellite handover and

possible link disruptions between the IoT base station and satellites on the shown performance in all the considered cases, so to highlight the improvement achievable with the proposed MEC functionalities also in the case of LEO satellite backhaul links.

## ACKNOWLEDGMENT

Responsibility of the content resides with the authors.

## REFERENCES

- [1] Cisco, "Cisco annual Internet report (2018-2023)," San Jose, CA, USA, Cisco, White Paper, 2020. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>
- [2] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2714–2741, Fourth quarter 2018.
- [3] S. Cioni, R. De Gaudenzi, O. D. R. Herrero, and N. Girault, "On the satellite role in the era of 5G massive machine type communications," *IEEE Netw.*, vol. 32, no. 5, pp. 54–61, Sep./Oct. 2018.
- [4] N. Kato et al., "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Commun.*, vol. 26, no. 4, pp. 140–147, Aug. 2019.
- [5] F. Chiti, R. Fantacci, and L. Pierucci, "Energy efficient communications for reliable IoT multicast 5G/Satellite services," *Future Internet*, vol. 11, no. 8, pp. 164–176, 2019.
- [6] M. Bacco et al., "Networking challenges for non-terrestrial networks exploitation in 5G," in *Proc. IEEE 2nd 5G World Forum*, 2019, pp. 623–628.
- [7] L.-A. Turner and H. Jahankhani, "An investigation into an approach to updating the governance of satellite communications to enhance cyber security," in *Proc. Cybersecur. Privacy Freedom Protection Connected World*, 2021, pp. 23–33.
- [8] M. Lin et al., "Integrated 5G-satellite networks: A perspective on physical layer reliability and security," *IEEE Wireless Commun.*, vol. 27, no. 6, pp. 152–159, Dec. 2020.
- [9] Q.-V. Pham et al., "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020.
- [10] S. Kekki et al., "MEC in 5G networks," Sophia Antipolis, France, ETSI, White Paper, 2018.
- [11] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for Internet of Things realization," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2961–2991, Fourth quarter 2018.
- [12] T. K. Rodrigues, K. Suto, and N. Kato, "Edge cloud server deployment with transmission power control through machine learning for 6G Internet of Things," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 4, pp. 2099–2108, Oct.–Dec. 2021.

- [13] R. Xie, Q. Tang, Q. Wang, X. Liu, F. R. Yu, and T. Huang, "Satellite-terrestrial integrated edge computing networks: Architecture, challenges, and open issues," *IEEE Netw.*, vol. 34, no. 3, pp. 224–231, May/June 2020.
- [14] L. Yan et al., "SatEC: A 5G satellite edge computing framework based on microservice architecture," *Sensors*, vol. 19, no. 4, 2019, Art. no. 831.
- [15] B. Denby and B. Lucia, "Orbital edge computing: Nanosatellite constellations as a new class of computer system," in *Proc. 25th Int. Conf. Architectural Support Program. Lang. Operating Syst.*, 2020, pp. 939–954.
- [16] Z. Zhang, W. Zhang, and F.-H. Tseng, "Satellite mobile edge computing: Improving QoS of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE Netw.*, vol. 33, no. 1, pp. 70–76, Jan./Feb. 2019.
- [17] T. De Cola, M. Marchese, M. Mongelli, and F. Patrone, "A unified optimisation framework for QoS management and congestion control in VHTS systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11619–11631, Oct. 2020.
- [18] J. Wei, J. Han, and S. Cao, "Satellite IoT edge intelligent computing: A research on architecture," *Electronics*, vol. 8, no. 11, 2019, Art. no. 1247.
- [19] Y. Wang, J. Yang, X. Guo, and Z. Qu, "Satellite edge computing for the Internet of Things in aerospace," *Sensors*, vol. 19, no. 20, 2019, Art. no. 4375.
- [20] D. Chen et al., "Resource cube: Multi-virtual resource management for integrated satellite-terrestrial industrial IoT networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 11963–11974, Oct. 2020.
- [21] N. Cheng et al., "Space/Aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.
- [22] G. Cui, X. Li, L. Xu, and W. Wang, "Latency and energy optimization for MEC enhanced SAT-IoT networks," *IEEE Access*, vol. 8, pp. 55915–55926, 2020.
- [23] Y. Cao, Y. Shi, J. Liu, and N. Kato, "Optimal satellite gateway placement in space-ground integrated network for latency minimization with reliability guarantee," *IEEE Wireless Commun. Lett.*, vol. 7, no. 2, pp. 174–177, Apr. 2018.
- [24] The Third Generation Partnership Project (3GPP), "Solutions for NR to support non-terrestrial networks (NTN)," Tech. Rep. 38.821 v16.1.0, Jun. 2021.
- [25] ESA ARTES AT Project, "NB-IoT4Space - 3GPP Narrow-Band Internet-of-Things (NB-IoT) user sensor integration into satellite," 2020. Accessed: Aug. 18, 2022. [Online]. Available: <https://artes.esa.int/projects/nbiot4space>
- [26] A. Guidotti et al., "Architectures and key technical challenges for 5G systems incorporating satellites," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2624–2639, Mar. 2019.
- [27] M. Centenaro, C. E. Costa, F. Granelli, C. Sacchi, and L. Vangelista, "A survey on technologies, standards and open challenges in satellite IoT," *IEEE Commun. Surv. Tut.*, vol. 23, no. 3, pp. 1693–1720, Third quarter 2021.
- [28] R. Giuliano, F. Mazzenga, and A. Vizzari, "Satellite-based capillary 5G-mMTC networks for environmental applications," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 34, no. 10, pp. 40–48, Oct. 2019.
- [29] M. Höyhty, T. Ojanperä, J. Mäkelä, S. Ruponen, and P. Järvensivu, "Integrated 5G satellite-terrestrial systems: Use cases for road safety and autonomous ships," in *Proc. 23rd Ka Broadband Commun. Conf.*, 2017, pp. 16–19.
- [30] J. V. Y. Martnez, A. F. Skarmeta, M. A. Zamora-Izquierdo, and A. P. Ramallo-Gonzlez, "IoT-based data management for smart agriculture," in *Proc. 2nd Int. Conf. Embedded Distrib. Syst.*, 2020, pp. 41–46.
- [31] K. Sohraby, D. Minoli, B. Occhiogrosso, and W. Wang, "A review of wireless and satellite-based M2M/IoT services in support of smart grids," *Mobile Netw. Appl.*, vol. 23, no. 4, pp. 881–895, 2018.
- [32] T. Wei, W. Feng, Y. Chen, C.-X. Wang, N. Ge, and J. Lu, "Hybrid satellite-terrestrial communication networks for the maritime Internet of Things: Key technologies, opportunities, and challenges," *IEEE Internet of Things J.*, vol. 8, no. 11, pp. 8910–8934, Jun. 2021.
- [33] OASIS, "Message queuing telemetry transport (MQTT) version 3.1.1," 2015. Accessed: Aug. 18, 2022. [Online]. Available: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.pdf>
- [34] Z. Shelby, K. Hartke, C. Bormann, and B. Frank, "The constrained application protocol (CoAP)," 2014. Accessed: Aug. 18, 2022. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc7252>
- [35] R. Soua, M. R. Palattella, and T. Engel, "IoT application protocols optimisation for future integrated M2M-Satellite networks," in *Proc. IEEE Glob. Inf. Infrastructure Netw. Symp.*, 2018, pp. 1–5.
- [36] M. Bacco, P. Cassarà, M. Colucci, and A. Gotta, "Modeling reliable M2M/IoT traffic over random access satellite links in non-saturated conditions," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 5, pp. 1042–1051, May 2018.
- [37] F. Wang, D. Jiang, S. Qi, C. Qiao, and H. Song, "Fine-grained resource management for edge computing satellite networks," in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–6.
- [38] C. Qiu, H. Yao, F. R. Yu, F. Xu, and C. Zhao, "Deep Q-learning aided networking, caching, and computing resources allocation in software-defined satellite-terrestrial networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5871–5883, Jun. 2019.
- [39] ESA ARTES AT Project, "VIBES - Implementation of Virtualised network functions (VNFs) for broadband satellite networks," 2018. Accessed: Aug. 18, 2022. [Online]. Available: <https://artes.esa.int/projects/vibes>
- [40] ESA ARTES AT Project, "M2MSAT - Demonstrator of light-weight application and transport protocols for future M2M applications," 2017. Accessed: Aug. 18, 2022. [Online]. Available: <https://artes.esa.int/projects/m2msat>
- [41] M. R. Palattella, R. Soua, A. Stemper, and T. Engel, "Aggregation of MQTT topics over integrated satellite-terrestrial networks," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, pp. 96–97, 2019.
- [42] Ericsson, "Massive IoT in the city," Mobility Report, 2016. Accessed: Aug. 18, 2022. [Online]. Available: <https://www.ericsson.com/en/mobility-report/articles/massive-iot-in-the-city>



**Michele Luglio** received the Laurea degree in electronic engineering and the Ph.D. degree in telecommunications from the University of Rome "Tor Vergata," Rome, Italy, in 1990 and 1994, respectively.

From August to December 1992, he was a Visiting Staff Engineer with Comsat Laboratories, Clarksburg, MD, USA. From 1995 to 2004, he was a Research and Teaching Assistant with the University of Rome "Tor Vergata." In 2001 and 2002, he was a Visiting Professor with the Computer Science Department, University of California Los Angeles, Los Angeles, CA, USA, to teach Satellite Networks class. Since 2004, he has been an Associate Professor of telecommunications with the University of Rome "Tor Vergata." He teaches "Internet via Satellite" and "Telecommunications Fundamentals". He works on designing innovative satellite communications systems for multimedia services, both mobile and fixed. He coordinates the Laboratory of the Satellite Multimedia Group, University of Rome Tor Vergata, he is the President of the University spin-off RomARS, and was the President of the Research Interuniversity consortium NITEL. He is author of more than 170 international publications. His research interests include network protocols, resource management, heterogeneous networks, and 5G development with particular regard to satellite systems.

Dr. Luglio was the General Co-Chair of IEEE ISNCC 2018, ISAECT 2019, IEEE ISNCC 2018, ISAECT 2019, ISAECT 2020, and ISAECT 2021. He is a Member of the Editorial Board of *China Communications* and *IAENG Engineering Letters*. He is an Associate Editor for *Space Communications* section of *Frontiers in Space Technologies*. He was affiliated with the Italian Space Agency and has been the Italian expert delegate and Co-Chair of the advisory committee SJAC of ESA.



**Mario Marchese** (Senior Member, IEEE) received the “Laurea” degree (*cum laude*) and the Ph.D. degree in telecommunications from the University of Genoa, Genoa, Italy, in 1992 and 1997, respectively.

From 1999 to January 2005, he was with the Italian Consortium of Telecommunications (CNIT), Parma, Italy, where he was the Head of research. From February 2005 to January 2016, he was an Associate Professor, and since February 2016, he has been a Full Professor with

the University of Genoa, where, since December 2020, he has been Rector’s Delegate to Doctoral Studies. He is the author of the book *Quality of Service over Heterogeneous Networks* (John Wiley Sons, 2007), and has authored or coauthored more than 300 scientific works, including international magazines, international conferences, and book chapters. His research interests include networking, quality of service over heterogeneous networks, software-defined networking, satellite networks, network security, critical infrastructure security and intrusion detection systems.



**Fabio Patrone** (Member, IEEE) received the M.Sc. degree in telecommunication engineering and the Ph.D. degree in telecommunications from the University of Genoa, Genoa, Italy, in 2013 and 2016, respectively.

He is currently an Assistant Professor with the Satellite Communications and Heterogeneous Networking Laboratory, University of Genoa, Genoa, Italy. His research interests include 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.



**Cesare Roseti** graduated (*cum laude*) in TLC engineering and the Ph.D. degree in space systems and technologies from the University of Rome “Tor Vergata,” Rome, Italy, in 2007.

Since 2012, he has been an Assistant Professor with the University of Rome “Tor Vergata,” teaching technologies for the communications and broadcast technologies. He is the Co-Founder of RomARS srl, Rome, Italy, an innovative startup dealing with advanced telecommunication technologies. He is technical and/or

scientific responsible of several international research projects addressing converged satellite and 5G networks, and he actively participated to several funded international research programmes, with the role of WP leader. He is the coauthor of about 95 scientific publications for international conferences, workshops, and journals.



**Francesco Zampognaro** received the M.Sc. degree in telecommunication engineering from the University of Rome “La Sapienza,” Rome, Italy, in 2004, and the Ph.D. degree in space systems and technologies from University of Rome “Tor Vergata” Rome, Italy, in 2010.

He is currently a Researcher with the University of Rome “Tor Vergata,” and a Professor in space systems at University Marconi, Rome, Italy. He teaches with Linux Management and Configuration Laboratory, MATLAB for Signal

theory and Satellite Systems classes. He is Co-Founder of RomARS srl, Rome, Italy, an innovative startup dealing with advanced telecommunication technologies. His research interests include satellite systems, supporting many research funded projects, and study and simulation of satellite systems, in particular DVB-RCS, covering protocols optimization, services provision and QoS, virtualization, security, resource allocation, integrated/hybrid architectures, and 5G.

Open Access provided by ‘Università degli Studi di Genova’ within the CRUI CARE Agreement