Sociocast: a new network primitive for the IoT

Luigi Atzori[^], *Senior Member, IEEE,* Antonio Iera[*], *Senior Member, IEEE,* Giacomo Morabito[§]

[^]Luigi Atzori is with CNIT and the Dept. DIEE, University of Cagliari, Italy, e-mail: l.atzori@ieee.org
[*]Antonio Iera is with CNIT and the Dept. DIIES, University Mediterranea of Reggio Calabria, Italy
[§]Giacomo Morabito is with CNIT and the Dept. DIEEI, University of Catania, Italy, email: giacomo.morabito@dieei.unict.it

***Abstract* — It is a matter of fact that social ties drive communications during most of our daily activities. This observation has inspired research activities aimed at exploiting the properties of social networks to implement novel networking paradigms, applications models, or middleware platforms. In this work, we leverage social networking technologies and concepts towards the definition of a novel network primitive for the Internet of Things (IoT). The new primitive, called *Sociocast*, enables trusted group-oriented communications, in-network publish&subscribe mechanisms, dynamic and selective firewalling, flexible data casting. In this paper we illustrate the requirements that we address and present an architectural solution together with the primitive syntax. We also show some performance results and discuss the future work.**

***Index Terms* — Internet of Things, social networks, network primitives, trust evaluation, multicasting.**

I. Introduction

The Internet of the Future will surely be characterized by a population of connected devices. These will be very different from the current ones, with a lion's share taken by objects belonging to the diversified world of the Internet of Things (IoT). These will trigger an increasingly high number of sophisticated services involving heterogeneous groups of users and causing a proliferation of unicast, broadcast, and multicast traffic flows. These devices will also be characterized by multi-interfaces and by multiple addresses that will often change over time. For her part, the user will increasingly act as *prosumer* interested in: (i) generating content destined to groups of devices and delivering it in a highly flexible manner; (ii) receiving only the desired traffic from devices considered reliable, so trying to limit the risks of receiving fake and harmful data.

The current network primitives for unicasting, multicasting, and broadcasting communications are likely to be inadequate for this future scenario. Multiple unicast communications in scenarios characterized by group communications between members of social communities, including smart objects also, are not an option obviously; while multicasting and broadcasting are widely considered a nightmare for network operators because of their difficult controllability and the risks of network overload inherently connected to them.

The logical consequence is to resort to a new network primitive for group communications, specifically designed for IoT applications, that will have to manage plenty of human-to-machine and machine-to-machine data exchanges. The needed primitive should be able to support group communications in a flexible, reliable, and quickly configurable/reconfigurable way.

To address the stated research problem, this paper proposes a disruptive solution that fully exploits two aspects emerging as pillars of future IoT: the paradigm of *Social Internet of Things* and the virtualization of both network functions and physical devices. Specifically, we intend to use social ties to create a social network of devices enabling the identification of communication end-points in terms of social distance and, at the same time, to assess the trustworthiness level of the devices. The resulting network is then used to implement a social-driven networking primitive, which enables the following functions: trusted group-oriented communications, in-network publish&subscribe mechanisms, dynamic and selective firewalling, and flexible data casting.

This primitive is designed to be implemented in the core network by leveraging virtualization technologies, which are used to create end-devices' digital counterparts needed to build the social graph. The proposed primitive is called *Sociocast* and it will be presented in the remainder of this paper in its basic principles and its reference architecture. Some simulation results are also presented together with an analysis of the open research issues that must be the subject of future investigations.

II. Research background

Properties of social ties among users and devices have been studied in the past to improve the unicast, anycast and multicast communications, especially for wireless networks. These have been also adopted in the IoT domain to address the issue of handling heterogeneous communications among billions of

objects. The most relevant works are briefly reviewed in the next two subsections as the basis for introducing our research.

A. Social-aware communications

The creation of clusters of devices to share information among groups of nodes with possibly similar interests has been deeply investigated in vehicular ad hoc networks. Specifically, in [1] social-driven clusters are formed to drive the creation of the physical and logical topologies in the wireless multi-hop communications among moving vehicles. Observation of past encounters among nodes are also used in social-aware routing solutions, as in [2], where a routing algorithm is proposed, which presents also the benefit of being stateless. The "small world" properties, typical of social networks, have been extensively studied to address the routing problem in Delay Tolerant Networks (DTN), as it is proposed in [3], where the cost function used in the forwarding decision aims at limiting the number of relays and at finding the appropriate relay nodes. In [4], these properties are also used to implement the *anycast* communication service rather than *unicast*. Social properties of mobile users are also addressed in [5], to the purpose of improving device-to-device multicast communications performance in terms of throughput while guaranteeing fairly channel allocation to different multicast clusters in radio networks. Finally, in [6] authors, with reference to video content sharing, have designed the Social-aware video multiCast (SoCast) system to stimulate effective cooperation among mobile clients, by leveraging social trust and social reciprocity ties.

B. Networks of social objects

Social ties among devices are also considered in the Internet of Things to address issues related to the management and effective exploitation of huge numbers of heterogeneous devices and their potential inter-communications. There are different approaches that can be exploited when merging the social networking technologies with the IoT, which brings to different generations of social objects, as reviewed in [7]. The most advanced is called Social IoT (SIoT), which is intended as a social network where every node is an object capable of establishing social relationships with other things in an autonomous way according to rules set by the owner. Differently from previous works, herein a stable social network is created, which relies on the management of different types of social links among objects, e.g., the Ownership Object Relationship (OOR) created between objects of the same owner; the CoGeolocation Object Relationship (CGLOR), created between fixed devices located in the same place; the Parental Object Relationship (POR), created between objects of the same model, vendor and production batch [8]. In several works in the literature other inter-object relationship types have been defined and rules for their establishment have been designed [9][10].

C. Contributions

From this brief (and definitely not exhaustive) literature analysis, it derives that social ties among devices, inherited from their owners, have the potential to improve traditional communications paradigms (unicast, anycast and multicast). These ties are used either at a middleware (or even application) layer or in a cross-layer fashion by jointly considering application requirements and networking needs. The proposed solutions are usually confined to the multi-hop wireless (mainly opportunistic and delay-tolerant) communications scenarios.

Different from this body of knowledge, we intend to: i) extend these functionalities to core Future Internet networks to support communications in IoT deployments; ii) exploit these properties to increase the level of security in the communications and to create social-driven groups; iii) provide network providers with tools to control the multicast communications; iv) define a new potentially disruptive network primitive, which implements the devised functionalities.

III. The need for a new network primitive for data casting

Major reasons justify the introduction of a new network primitive for group communications that exploits social relationships among smart devices:

The need for a tool to support traditional IP multicast. IP multicast is not used by applications as much as one could imagine. Network operators are, in fact, reluctant to support IP multicast in their networks because it might be dangerous. For example, the consequences in terms of network overload and complexity of network management can be catastrophic if an excessively high number of devices establish multicast groups and many others register to such groups. One way to keep the multicast load under control could be to have a primitive available through which it is possible (i.e., for the network operator) to filter the set of nodes that can join a certain multicast group, based on their position in the social graph of devices.

The need for a method to support publish/subscribe communications. Application developers can benefit from the availability of a network primitive that allows an object to specify that it is interested in receiving packets published by all other devices within a given distance in the objects' social network, by considering only certain types of relationships. This is the case, for example, of

sensors sending alarms that should be received by all the devices owned by an industrial plant guardian (and therefore tied by relationships of type OOR) so as to be sure that the message reaches her attention.

The need for a dynamic & selective firewall. The option for any device to control which other entities are entitled to send it data by exploiting network level functions and the reciprocal position in the social network, would be a very useful feature. This should leverage the trustworthiness control policies intrinsically offered by social networks. For instance, it would be valuable, from the house security point of view, to restrict the devices able to communicate with the house control unit to those belonging to the same owner and to those owned by the rest of the family members. This could be implemented by restricting the communication to the devices with only one social hop of OOR or CGLOR types.

The need for higher flexibility in data casting. The use of metadata is a further opportunity for casting data based on device profiles. Examples are: sending queries to all "temperature sensors" in a target area, send notifications to "smartphones" of family members and friends, etc.

IV. Enabling Architecture and Functionality

In this section we first introduce the proposed framework supporting the new Sociocast network primitive along with the required functionality. Then, we briefly describe how it can be implemented in specific networks, such as legacy IPv4, OpenFlow, and Content-Centric Networks.

A. Network framework and key functions overview

Fig. 1 shows the network elements involved in the support of the new Sociocast communication primitives, i.e., Sociocast nodes, Sociocast Relationship Service, and Edge routers.

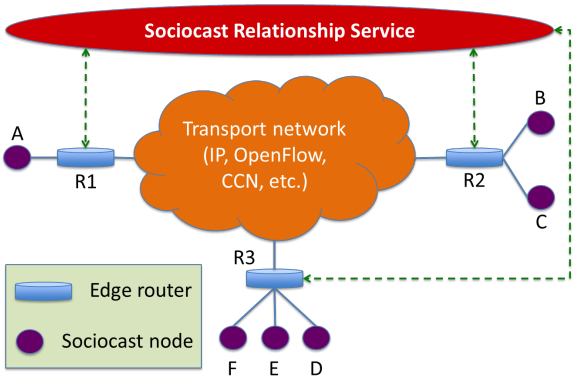

Fig. 1: Network framework supporting *Sociocast* primitives

Sociocast nodes exploit the new network primitive to deliver packets to a set of destinations on the basis of their mutual positions in the social graph. To implement the relevant functionalities, each of these nodes has a digital counterpart, which we call *Social Virtual Node* (SVN), managed by the Sociocast Relationship Service. Each SVN represents a database that stores information about the node it refers to and has an identifier, which is the current topological position of the node (i.e., the IP address or the URL). It stores some metadata describing the characteristics of the node, and a list of *friends* organized in a table named "Friends Table". For each friend in the table, the SVN keeps the type(s) of friendship(s) and a reference to the location where the SVN (SVN locator) of the friend can be found in the distributed SVN Repository (SVNR). The SVN locator may be the IP address of the server where the SVN is stored.

For example, Fig. 2 shows the Friend Table of node A, identified as ID_A, which has three "friends" identified as ID_B, ID_D, and ID_E. Relationships between ID_A and its three friends may be of different types (here we refer to those mentioned in Section II.B, but other types can be considered as well). In the example of Fig. 2, ID_A is linked with relationships of type CWOR with ID_B, type CGLOR with ID_C, and type OOR with ID_E. The IP server addresses where the corresponding SVNs can be found are 151.70.25.10 (for ID_B) and 130.40.0.56 (for ID_D and ID_E).

Node ID	Metadata	Relationship(s) type(s)	SVN locator
ID_B	logistics	CWOR	151.70.25.10
ID_D	Sport	CGLOR	130.40.0.56
ID_E	Infotainment	OOR	130.40.0.56

Fig. 2: Exemplary Friend Table.

The *Sociocast Relationship Service* implements a set of control plane functionalities relying on a distributed hardware/software infrastructure, whose major components are shown in Fig. 3. It includes: the SVN Repository (SVNR), which stores SVNs; the Relationship Manager (RM), which is responsible of establishing, updating and removing relationships; the Relationship Browser (RB), which is able to navigate the social network to find potential peers; and the Sociocast Handler (SH), which handles Sociocast communications.

The Sociocast Handler plays a crucial role in our framework as it also implements the following relevant functions: (i) handling the messages sent by Edge routers; (ii) interacting with the RB to determine the set of Sociocast nodes that should receive a given Sociocast packet, based on their position in the social network; (iii) accept/discard packets based on the position in the social network of the communicating peers.

Finally, *Edge routers* are very important in our framework as they must intercept incoming packets with certain characteristics, interpret them, interact

with the Sociocast Relationship Service, and execute corresponding commands.

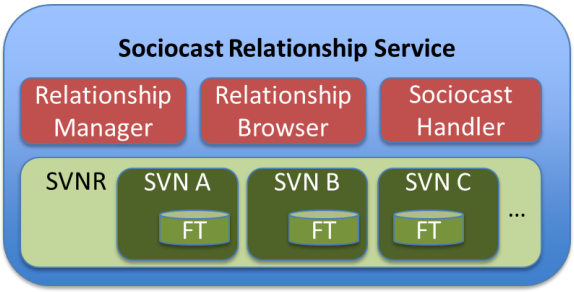

Fig. 3: Sociocast Relationship Service.

B. Support of Sociocast in different networks

In this section we briefly discuss how the framework presented in the Section IV.A can be implemented regardless of the network technology applied to the Transport Network. More specifically, we analyze three cases:

Case 1: The Core Network is an IP network – In this case, implementing the scheme described in Section IV.A is straightforward. As not many programmable network elements are available yet, in the short term, Sociocast can exploit native IP multicast to increase network efficiency. In fact, Sociocast can assign an IP multicast address to the communication and will communicate such an address to R1, R2, and R3. The above edge routers will join the multicast group and the packet will be transported in the network along the multicast tree. It is important to highlight that the multicast group includes edge routers only (hosts are excluded).

Case 2: The Core Network is an OpenFlow network – If this is the case, then the new network primitive can be implemented as a network application running on top of a Controller. In this scenario, R1, R2, and R3 are actually OpenFlow switches able to analyze Sociocast packets. With reference to the scenario explained in Section IV.A, R1 sends the first packet to the Controller which forwards it to Sociocast Handler. This identifies the destinations and provides the list to the Controller, which is responsible to instruct the Flow Tables of the switches in the Core Network.

Case 3: The Core Network is a publish-subscribe network – In this case, a major difference with respect to Case 1 is that R2 and R3 are responsible to issue the subscription messages for the packets transmitted by node A with a certain label.

V. Implementation Details

We have implemented prototypes of all components required to support Sociocast in the above Case 2, wherein the Core Network is based on OpenFlow.
In this section we provide some implementation details for such a case, which however can be easily generalized. More specifically, we first describe the syntax of the network primitive; then we provide a walkthrough of Sociocast operations in a specific situation.

A. Sociocast Primitive and packets

A sample primitive used by A to set-up a Sociocast group, may have the following syntax:

Sociocast_group_set-up_req (*SocioCastTypes*, σ_D, *Filter*, *meta*)

where: *SocioCastTypes* specifies the nature of the policy to be considered in determining the node in the group. Among the possible policies, *SocioCast_meta* consider the metadata of the nodes, whereas *SocioCast_depth* considers the distance between nodes in the social graph. The parameter σ_D indicates that the Sociocast group may include entities with distance from source *A* in the social graph lower than a given threshold σ_D called *radius*. *Filter* allows to specify which one(s) of the *L* possible relationships among devices are admissible during the set-up procedure.
It consists of an *L*-tuple $f = [f^{(1)}, f^{(2)}, \dots, f^{(L)}]$ whose $f^{(i)}$ assumes value 1 if the social relationship of type i between any couple of devices have to be considered when setting up the communication and assumes value 0 otherwise.
Meta allows for specifying metadata information associated to the devices that must be considered as admissible during the set-up procedure.
Obviously, Sociocast functioning also relies on the exchange of clearly recognizable Sociocast packets, which are generated whenever the described primitive is issued at the Sociocast node. These packets are standard IPv4/IPv6 packets. In fact, all the pieces of information required by Sociocast (e.g., filter and radius) are encoded in the destination port number. The IP destination address, instead, is a specific public IP address of a server which implements Sociocast. In this way there is the guarantee that the packet will eventually reach a Sociocast node.

B. Sociocast behavior

To clarify the role of the different elements described above we provide in the following a walkthrough in an exemplary use of the Sociocast network primitive.

As shown in Figure 4, let us suppose that at a given time instant the Sociocast node A generates, via the new communication primitive, a packet that has to be delivered to all the Sociocast nodes within a distance 2 from A in the social network graph.
When the Sociocast packet generated by *A* is received by the Edge router R1, which in this case is an OpenFlow switch, the latter does not find any relevant entry in its Flow Table and therefore queries the Controller. This interacts with the Sociocast Handler component of the Sociocast Relationship Service, which asks the Relationship Browser to find the nodes that should receive the communication according to the parameters of the originating primitive. Accordingly, the nodes B, D, and F are identified as the Sociocast nodes that must receive the packet and the list of these nodes is then passed to the Controller. This is aware of the network topology and, therefore, creates the entries for the Flow Tables needed to route packets to the intended destinations. Such entries will be sent to all the involved network elements, such as R1, R2, and R3, in our case.
Suppose that A generates a new packet meant to be delivered again to Sociocast nodes with distance in the social graph not greater than 2. Upon receiving it, R1 finds information in its Flow Table about the way in which the packet should be dealt with. Therefore, it forwards it towards R2 and R3, which, on their turn, send it to B, D and F as discussed above. Note that data communication occurs between physical nodes, i.e., not their digital counterparts.
The Sociocast packet is a new packet which is generated by the source node after the Sociocast_group_set-up_req primitive has been issued. The most important information encoded by the packet is related to the parameters of the primitive which are used by the Sociocast Relationship Service to find the target recipients by browsing the social network. As to the deployment of the Social Relationship Service, this relies on a completely distributed architecture where a server is deployed by each ISP and has peering connections with the other Services in the adjacent ISPs. Communications between one instance of this service and the others in other ISPs, happen whenever there is the need to find the location of a recipient SVN which is not included in the originating SVN friend table and the social network brings to an external service.

VI. Performance analysis

We have utilized the prototype described in the previous Section V to assess the performance of the proposed approach. Results of such an assessment are shown in **Errore. L'origine riferimento non è stata trovata.**5.
The values represented in the figure have been obtained by assuming that there are 40 Sociocast nodes divided in 4 sets, each containing 10 nodes connected to the same OpenFlow switch.
Indeed, the network consists of 4 OpenFlow switches that are connected according to a ring topology.
The Controller and the Sociocast Relationship Service run in the same server which is attached to one of the OpenFlow switches. Delay estimates have been obtained by using Mininet.

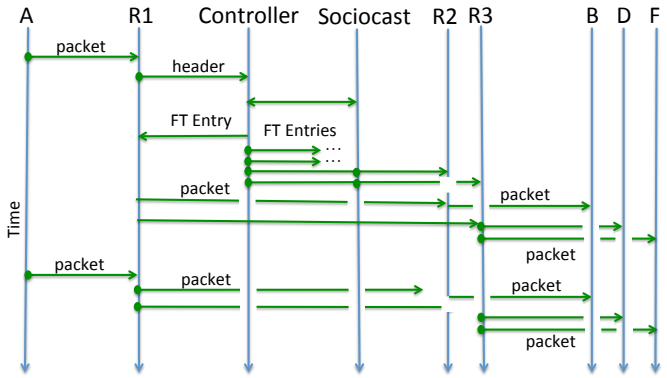

Fig. 4: Exemplary exchange of messages.

For the sake of comparison, **Errore. L'origine riferimento non è stata trovata.**5 plots the *delay gain* of Sociocast vs. two alternatives that implement the same functionality at the application layer. More specifically, as alternatives we consider two different options:
Option 1: The source node queries a server which provides the list of nodes which should receive the packet and, then, the source sends a copy of the packet to each node.
Option 2: The source node sends the packet to a server, say S_A. The server S_A obtains the list of nodes which should receive such a packet from another server and then forwards a copy of the packet to each intended destination.
The plotted delay gain is the ratio between the delay obtained by using one of the alternative solutions and the delay obtained by applying Sociocast. Curves are represented versus the *hit probability,* i.e., the probability that a packet arrives at an edge router where a flow entry is available that specifies how to treat the packet and therefore, it is unnecessary to inquire the controller.
Note that for most configurations the gain delay is higher than one (thus, Sociocast experiences a lower delay). Exceptions are cases in which Option 1 is applied and the hit probability is low (lower than

0.2). Nevertheless, if Option 1 is utilized then the source node must transmit one packet for each intended destination, which implies a higher consumption of both energy and communication resources that are scarce resources in most IoT scenarios. In Sociocast, instead, such as in the Option 2, only one packet is transmitted by the source node.

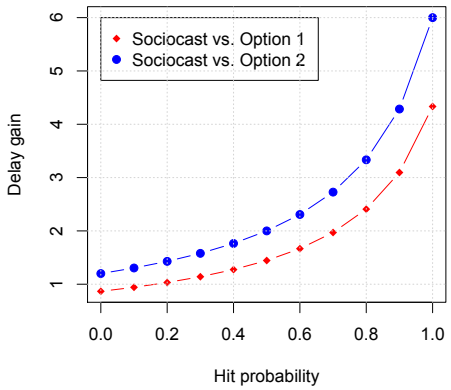

Fig. 5: Delay comparison between Sociocast and alternative solutions.

VII. THE ROAD AHEAD

Key several research issues still need to be addressed:

Social graph and social ties- we need to define suitable policies for the creation, maintenance and updating of social bonds between devices; possibly by providing the option of triggering both stable and temporary bonds. In this direction, research results are already available from the literature, but it remains to investigate which rules to use to optimize the behaviour of Sociocast procedures at the network layer

Data plan – in this initial paper we have given hints on how the new control plane functionality can be coupled with existing transport networks. In the future, we will need to provide a data plane that enables data transfer from source to destinations by using the Sociocast primitive natively. This transition towards a new data plane shall be implemented in a non-disruptive way with reference to existing protocols.

Social distance – setting the social distance between devices within the social group is crucial for the good functioning of the primitive as it will be exploited by users or network operators to control and configure Sociocast sessions. Therefore, it will be necessary to define appropriate schemes for the identification of the appropriate value of the social distance between source and destinations as well as algorithms for efficient and effective navigations of the social graph, which is crucial for Sociocast implementation.

Sociocast programming – in order to enable hosts to join Sociocast groups, suitable socket APIs shall be defined with relevant socket options.

Performance analysis - the performance of Sociocast will have to be carefully evaluated by considering the dynamics of network topologies, user mobility, and social ties.

VIII. CONCLUSIONS

In this paper we have introduced Sociocast a new network primitive that supports novel group communication schemes in the IoT. In Sociocast the IoT nodes involved in a group communication session are identified through their mutual position in a social graph built in accordance to the Social Internet of Things (SIoT) paradigms.
Sociocast main components have been described and a preliminary performance assessment has been carried out.

IX. ACKNOWLEDGEMENTS

This work was partially supported by the European Union's Horizon 2020 research and innovation program under the COG-LO project (grant agreement no. 769141).

REFERENCES

[1] L.A. Maglaras, D. Katsaros, "Social Clustering of Vehicles Based on Semi-Markov Processes," *IEEE Trans. Vehicular Technology*, vol. 65, No. 1, pp. 318 – 332, 2016
[2] A. Mei, et al., "Social-Aware Stateless Routing in Pocket Switched Networks," *IEEE Trans. on Par. and Distr. Syst.*, Vol. 26, No. 1, 2015
[3] K. Wei, et al., "Exploiting Small World Properties for Message Forwarding in Delay Tolerant Networks," IEEE Trans. on Computers, Vol. 64, No. 10, pp 2809 – 2818, 2015
[4] T. Le, M. Gerla, "Social-Distance based anycast routing in Delay Tolerant Networks," in Proc. Med-Hoc-Net Workshop, 2016
[5] P. Zhao, et al., "A Social-Aware Resource Allocation for 5G Device-to-Device Multicast Communication," IEEE Access, Vol. 5, 2017
[6] Y. Cao, et al., "Social-Aware Video Multicast Based on Device-to-Device Communications", IEEE Transactions on Mobile Computing, Vol.: 15, No: 6, pp 1528 – 1539, 2016
[7] L. Atzori, et al., "From "smart objects" to "social objects": The next evolutionary step of the IoT", IEEE Comm. Magazine, 2014
[8] L. Atzori, et Al., "The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization", Computer networks, 56(16), 3594-3608, 2012

[9] K.M Alam, M. Saini, A. El Saddik, "Toward social internet of vehicles: concept, architecture, and applications", IEEE Access, 3, 2015.
[10] O. Voutryas, et al., "An Architecture supporting Knoweledge flow in Social Internet of Things Systems", IEEE WF-IoT 2014.

Luigi Atzori (SM'09) is Associate Professor at the Department of Electrical and Electronic Engineering at the University of Cagliari (Italy), where he leads the laboratory of Multimedia and Communications. His interests are in: multimedia communications, NGN service management, and IoT. He is the coordinator of the Marie Curie Initial Training Network on QoE for multimedia services (qoenet-itn.eu), which involves ten European Institutions in Europe and one in South Korea. He is member of the steering committee for the IEEE Trans. on Multimedia, member of the editorial board of the IEEE IoT, the Elsevier Ad Hoc Networks and the Elsevier Digital Communications and Networks journals.

Antonio Iera (SM'07) Graduated in Computer Engineering at the University of Calabria, Italy, and received a Master Diploma in Information Technology from CEFRIEL/Politecnico di Milano, Italy, and a Ph.D. degree from the University of Calabria. From 1994 to 1995 he has been with Siemens AG in Munich, Germany, and since 1997 with the University of Reggio Calabria, where he is currently a professor of Telecommunications and director of the Laboratory for Advanced Research into Telecommunication Systems (www.arts.unirc.it). His research interests include wireless and mobile 5G networks, RFID systems, and Internet of Things.

Giacomo Morabito received the laurea degree in Electronic Engineering and the Ph.D in Electronic, Computer and Communication Engineering from the University of Catania (Italy) in 1996 and 2000, respectively. From 1999 to 2001 he was research engineer at the Broadband and Wireless Networking Laboratory of the Georgia Institute of Technology. Since 2001 he is with the University of Catania where he is currently professor of telecommunications. Giacomo Morabito has served in the Editorial Boards of Computer Networks, IEEE Wireless Communications, Wireless Networks, and IEEE Networking Letters. His research interests include analysis and design of solutions for wireless networks and the Internet of Things.