SDN-BASED SOCIOCAST GROUP COMMUNICATIONS IN THE INTERNET OF THINGS

Luigi Atzori[1,5], Claudia Campolo[2,5], Antonio Iera[3,5], Giuseppe Massimiliano Milotta[2,5], Giacomo Morabito[4,5], Salvatore Quattropani[5]
[1]DIEE, University of Cagliari, Italy, [2]DIIES, University Mediterranea of Reggio Calabria, Italy, [3]DIMES, University of Calabria, Italy, [4]DIEEI, University of Catania, Italy, [5]CNIT - National Inter-University Consortium for Telecommunications, Italy,

NOTE: Corresponding author: Giacomo Morabito, giacomo.morabito@unict.it

Abstract – *The new applications populating the Future Internet will increasingly rely on the exchange of data between groups of devices, dynamically established according to their profile and habits (e.g., a common interest in the same software updates and services). This will definitely challenge traditional group communication solutions that lack the necessary flexibility in group management and do not support effective control policies on involved endpoints (i.e., authorized senders and intended receivers). To address the cited issues, the idea of introducing new disruptive network-layer solutions has emerged from recent literature. Among them, Sociocast has been theorized as an enabler of flexible interactions between groups of devices tied by social relationships. In this paper we start from the concept of Sociocast and propose a solution based on Software Defined Networking (SDN) for its implementation at the network layer in the Internet of Things. The performance of Sociocast is studied and compared to methods running at the application layer that provide similar features. Experimental results, achieved through an emulation-based playground, confirm that the Sociocast approach allows for a significant reduction of signaling and data packets circulating in the network with respect to traditional approaches.*

Keywords – Communication Primitives, Internet of Things, Social Internet of Things, Software Defined Networks

1. INTRODUCTION

The Internet is experiencing a rapid transformation pushed by the growing need to overcome its intrinsic limitations and ossification, which challenge network practitioners and researchers. The pressing need to come to the definition of a new Internet of the future is also motivated by the multitude of Internet of Things (IoT) applications that are recently emerging in various vertical markets [1]. Such applications are increasingly characterized by *group-based* (i.e., one-to-many, many-to-many) communications established between large sets of devices in need of simultaneously exchanging data, e.g., in the case of sensors' software updates, service advertisements, device configurations.

In human-centric communications, frequent instant messaging occurs within communities of users sharing similar interests and people largely interacting with their friends, and friends of their friends, over social networks. Similarly, groups of IoT devices are likely to interact with each other, especially if they are located in the same place (e.g., sensors/actuators in the same building), are owned by the same user (e.g., consumer devices and home appliances), share similar profiles (e.g., the same brand and type), or frequently meet each other (e.g., vehicles on a given road segment).

Support of interactions between devices raises outstanding challenges for network operators. First, IoT applications require the *dynamic and flexible management* of group-based interactions, whose scope is decided according to a given topic and to the ties existing between involved endpoints (e.g., co-locality, similarity of devices, etc.).

Second, *the communication endpoints should have the power to control* data exchanges. Indeed, a control of the enabled data receivers is strongly desired by the source device, due to the potentially confidential nature of exchanged data.

Moreover, the *massive* presence of group-based communications established by billions of IoT devices, expected to increase even at a higher pace in the near future, can cause network congestion and waste device and network resources, unless proper countermeasures are taken.

A solution is required to allow nodes to flexibly specify how to *prioritize* (*filter*) the nodes from which they want (or they do not want) to receive data, and the network to react accordingly, so to prevent the threats of Denial of Service (DoS) attacks.

Conventional multicast-based approaches [2], being mainly designed to simultaneously transmit data from

one or multiple senders to a group of (unknown) receivers, fail in *natively* achieving such objectives and in ensuring the required flexibility in group establishment and management. Clumsy patches to existing multicast solutions may further complicate their design and hinder their (already limited) deployment.

This is why in [3] authors argue in favor of a novel and future-proof comprehensive solution, named *Sociocast*, encompassing both a communication method and a data delivery scheme, going well beyond Internet Protocol (IP)-based multicast. Sociocast is theorized as a means for identifying, in a flexible manner, the intended endpoints (senders/receivers) of data exchange sessions. Groups are dynamically created according to the mutual position of endpoints in a *social network of devices* and the type of relationships among them, by means of properly defined filtering rules and policies.

This work treasures the theoretical analysis in the cited vision paper and takes a significant step forward both in terms of *practical design* and *experimental evaluation*. Herein, we argue about the actual possibility of implementing the conceived Sociocast primitive as a *network-layer* solution in IoT domains, wherein switches and routers are responsible for the efficient delivery of packets issued by IoT devices. In particular, the reference network infrastructure is deployed according to the Software-Defined Networking (SDN) technology [4].

SDN has been introduced to address a typical issue in traditional IP networks: the lack of programmability in network management and configuration. Thanks to its peculiarities, it can play a crucial role to bring the social dimension into the group data delivery procedures enforced at the network layer.

The main contributions of the work can be summarized as follows.

- The design of an architectural framework encompassing all the entities and functionalities supporting Sociocast, according to a software-defined network approach.
- The definition of the main procedures for the creation of the Sociocast packets, their forwarding and filtering, and the subscription of devices to Sociocast groups.
- The performance assessment through the widely known Mininet network emulator [5], when dealing with push-based data dissemination and deploying the Sociocast network application into the ONOS SDN controller [6]. The impact of different endpoint distribution patterns and different involved social relationships on the performance is evaluated by comparing our proposal to an alternative approach where the groups are created at the application layer. Results show that the Sociocast approach allows for a reduction of signalling and data packets by a factor of 10 and 5, respectively, in the scenario where the number of recipients is high and are close to each other.

The remainder of this paper is organized as follows. In Section 2 we survey the related literature in the field of group-based communications. In Section 3 we introduce the major Sociocast concepts and discuss the design guidelines we have considered. Section 4 discusses the conceived architectural framework along with the envisioned entities and their main role and functionalities. In Section 5 we describe the main procedures to enable the treatment (i.e., forwarding, dropping, modifying) of Sociocast packets. Then, in Section 6 we describe the playground for the evaluation, before discussing achieved results in Section 7. Finally, in Section 8 we draw some concluding remarks.

2. BACKGROUND AND MOTIVATIONS

In this section we will first overview how group communications are traditionally supported in the Internet (see Section 2.1); then, we will discuss the drawbacks of such solutions (see Section 2.2); finally, the advantages of exploiting social relationships between IoT nodes are summarized (see Section 2.3).

2.1 Multicast approaches in the literature

A large number of different applications rely on one-to-many and many-to-many data traffic exchange. They range from live video streaming, audio/video conferencing [7] and multiplayer games [8] to communications between groups of servers within data centers [9] and wide-area control in smart grids [10]. Multicasting functionality is typically leveraged in such contexts, which can be performed either at the network (IP) layer or at the application layer [2], [11] and also with the support of SDN [12], [13].

IP-based multicasting. Traditional multicast routing and management protocols, such as Protocol-Independent Multicast (PIM) [14] and Internet Group Management Protocol (IGMP) [15], effectively establish and maintain multicast communication paths between sources and receivers to enable the forwarding of packets to a multicast group. Each group is assigned a unique class D IP address. A host can send data to a multicast group by using the local network multicast capability to transmit the packet. A multicast router, upon reception of a packet, looks up its routing table and forwards the packet to the appropriate outgoing interface. Group membership is managed at the network level through

routers. When a host decides to join/leave a particular multicast group, it sends the request to the local multicast router, through IGMP [15].

IP multicast allows data to be distributed in such a way that the least amount of replicas of the same packet is placed into the network.

In its recent version, v3, IGMP allows to specify the set of senders from which a node wants to receive, in agreement with the Source-Specific Multicast (SSM) protocol [16]. In other words, the only packets that are delivered to a receiver are those originating from a specific source address requested by the same receiver. Hence, SSM is particularly well-suited to dissemination-style applications with one or more senders whose identities are known before the application begins.

Non-IP multicasting. The design of multicast solutions has also been investigated beyond IP. In application-layer solutions, group membership, multicast delivery structure construction, and data forwarding are exclusively controlled by participating end hosts, thus, the support of network nodes is not needed [11].

In the clean-slate future Internet MobilityFirst architecture [17], a context-aware delivery primitive is proposed, which generalizes multicast to groups established on the basis of attribute-based descriptors. The name service, in charge of resolution procedures between global unique identifier (GUID) and network addresses, maintains a membership set that consists of all GUIDs of devices that are subscribed to the multicast group. The sender is responsible for sending data to each of the returned addresses.

SDN-based multicasting. SDN can simplify multicast traffic engineering thanks to the centralized nature of the network control plane. Current multicast solutions employ a shortest-path tree to connect the source to the receivers which is built according to local information. Traffic engineering is difficult to be supported in a shortest-path tree. By utilizing the global view of the SDN controller, in [18] all the possible routes between the sources and each host of the multicast group are calculated in advance. In contrast with IP multicast, there are no de facto standards for SDN multicast routing. Different approaches targeting different optimization objectives can be targeted in a flexible manner and it is unlikely that a given approach is going to be dominant. SDN multicast is enabled by writing an application for the SDN controller that optimizes the traffic flows to meet the particular needs of the end user [12]. The SDN controller can build the multicast tree to meet link constraints (bandwidth consumption) or path constraints (end-to-end delay) [13]. Hence, it is a valuable solution when Quality of Service (QoS) requirements need to be ensured to a multicast flow, e.g., in the case of a multi-party video-conferencing service [19].

2.2 Weaknesses and open issues

The use of the traditional IP multicast is prone to multiple issues:

- Without the explicit join to the multicast group, a router will not forward multicast IP packets destined to end hosts. This process implies the distribution of the consent to join the multicast group among devices, increasing the signaling overhead.
- There is no way for the sender to control who subscribes to a multicast group.
- It prevents the creation of discrimination policies based on the destinations of the information within the same multicast group. Therefore, when a limitation to the distribution of packets to some entities of the same multicast group is needed, another multicast group must be created, with a consequent increase in the number of signaling packets in the network.
- All routers must be replaced with multicast-enabled routers, which could be expensive and hardly viable for the network operator, raising interoperability issues.

The poor flexibility of the IP-based multicast discourages the pursuit of such an approach for the wide variety of sender-initiated dynamic group-based communications, as demanded by future IoT deployments.
On the other hand, application-layer solutions have the drawback of a definitely worse performance in terms of end-to-end latency and efficiency compared to IP multicast. This is because end hosts have little or no knowledge of the underlying network topology.
Thanks to its programmability and global knowledge of the topology, SDN can make more efficient the creation of the multicast tree, improving forwarding procedures. However, to the best of our knowledge, the flexibility of SDN has not been investigated to manage dynamic group formation.
These issues have motivated the theorizing of a new communication method and data delivery scheme [3], able to better fit the nature of upcoming group-based communications: *Sociocast*.
This is introduced as a novel and flexible solution that allows *group-based communications in the IoT enhanced with the notion of social ties*. It inherits the strengths of IP multicast, in that it lets network nodes disseminate packets in an efficient manner: sociocast packets are assigned an IP address to facilitate their forwarding. In

addition, the proposal in [3] enables a *mutual control* of the endpoints. Not only the receiver can filter different senders, as in SSM, but also the sender can (implicitly) decide which node should belong to the set of intended receivers, by specifying the features (in terms of social relationship) of such receivers. The above capabilities are disruptive when compared to conventional IP-based multicast. Sociocast relieves the burden of group management from network nodes and of explicit join procedures from devices.
Moreover, SDN is chosen to facilitate the implementation of multicast groups with a social flavour directly at the network layer.

2.3 Advantages of a "social-oriented" approach

The use of social links to support network functionality is not new in the telecommunications landscape.
Several routing protocols in wireless ad hoc [20], mobile opportunistic and delay-tolerant networks [21, 22, 23, 24], have been designed to build upon the key concepts of social network analysis, i.e., *small world phenomenon* [25] and *centrality*. The former one, a.k.a. *community*, captures the fact that actors within a social network are separated from each other by an average number of fairly *limited hops*. The latter one shows that *some nodes in a community are the common acquaintances of other nodes*.
In the aforementioned works, the knowledge of social characteristics (e.g., node centrality, in-betweenness) is used to make better forwarding decisions and assist the relay selection when delivering data to the intended destination(s).
Many of the studied approaches involved unicast or multicast communications [26, 27, 28]. The issue of data broadcasting in a Mobile Social Network, where mobile social users physically interact with each other, is analyzed in [29].
The objective of this work is to exploit similar concepts but under a different perspective. We aim not to improve forwarding decisions by leveraging social network properties, but to better disseminate data at the network layer within dynamically created groups of socially connected devices.
The proposal has the potential of a real game changer in view of the creation of the future Internet of Things, by providing superior advantages compared to what has been done so far in the literature.
In fact, social bonds not only ensure minimum separation distances between actors, crucial for efficient and fast data propagation, but may enable data exchange within trusted groups and creation of groups that include actors belonging to different communities. In Sociocast this translates into the possibility of efficient and flexible group end-points discovery, an intrinsic possibility of implementing policies for creating trusted groups of end-points directly at the network level, and the ability to effectively and simply deal with the problem of interoperability among different IoT platforms.
Obviously, to do this we need to start from a paradigm that can provide for the establishment of pseudo-social ties between devices (to operate at the network layer). This is already available in solutions of "social networks of IoT devices", such as the Social Internet of Things (SIoT) [30] for example. However, they need to be moved from the application layer, wherein they have been initially conceived, down to the control plane of the network layer. In so doing, group establishment and data exchange among members of such groups can be managed in a tighter way, with inherent flexibility and efficiency in terms of network resource usage.

3. SOCIOCAST: OBJECTIVES AND DESIGN PRINCIPLES

In this Section we describe how we can achieve a real implementation of the Sociocast concept by relying on the capabilities of the Software Defined Networking paradigm. The resulting solution is an enabler for group communications based on social notions at the network layer.
Social ties among devices. Devices are likely to interact with other devices with similar profiles and habits, e.g., those located in the same place, owned by the same user, produced in the same company branch.
Such ties are well captured by the SIoT paradigm in [30]. There, a few basic types of social relationships, defined according to user-defined policies, are introduced: *co-ownership object relationship* (OOR), created between devices that belong to the same owner; *co-location object relationship* (CLOR), created between stationary devices located in the same place; *parental object relationship* (POR), created between devices of the same model, producer and production batch; *co-work object relationship* (CWOR), created between moving devices that meet each other at the owners' workplace; *social object relationship* (SOR), created as a consequence of frequent interactions between moving devices. The framework is quite flexible and other types of relationships can be easily added on a per use-case basis.
Applications requiring data dissemination to a social group of devices are, for instance, *software updates*: a given software patch needs to be safely delivered to all the devices or sensors of the same brand, model, batch. For this, POR relationships should be exploited. Similarly, some data needs to reach all other devices belonging to the same owner in the case of *personal bubbles*: the

OOR relationship is appropriate in this scenario. Business services may be advertised to all devices that either are currently in the same area (CLOR) or often visited the same place (SOR).

Targeted data delivery schemes. Sociocast aims to enable:

- a given sender to disseminate data in a *push*-like manner to specific nodes, which are friends over a social network of devices, according to properly defined filters and policies (i.e., the social relationship type);
- a node *to subscribe* to specific social-based topics (i.e., to receive data from friends of a given type);
- a node *to prioritize* (and not to receive) data from particular senders, e.g.: to enforce QoS; to identify the more suited and trusted communication endpoints for security reasons; to save resources.

Deployment options. To target the aforementioned objectives, the envisioned framework has *(i)* to enable nodes to indicate in an agile manner the features of the endpoints of data flows (i.e., the set of intended recipients and/or the authorized senders) based on the distance in a social network of devices, *(ii)* to properly and dynamically identify them, *(iii)* to forward data packets accordingly.

A straightforward approach to accomplish the first two features could be to rely on an *application-layer* solution. For instance, the intended set of receivers can be specified by a given sender at a high-level, e.g., through metadata. Then, the resulting request can be sent to a purpose-built proxy which is in charge of mapping such data onto IP addresses of the receivers, similar to [17]. Despite the virtue of simplicity, such an approach has the drawback of poor performance in terms of efficiency in the usage of network resources, since data forwarding to each intended destination is performed at the underlying network layer in a myopic manner.

Thus, our interest is on a *network-layer* approach, according to which the features of the intended set of receivers of a given data packet (or of a sender of unwanted data packets) can be translated into a network-layer IP address, hence treated (forwarded/dropped) by network nodes, accordingly. The type of proposed approach is inspired by the traditional IP-based multicast, with which it shares a few aspects, such as the routing of packets with a multicast address (a Sociocast address, in our case). However, multicast lacks the flexibility necessary to implement the aforementioned critical functionalities for the future Internet of billions of devices, while meeting the requirements of the end users and those of the network operators.

By overstepping the agnosticism about Sociocast traffic at the network layer, the following advantages are expected:

- data forwarding can occur in an efficient manner, e.g., by reducing the number of duplicated packets, and saving bandwidth accordingly;
- filtering procedures can be enforced in-network, as requested by potential data recipients, to limit the massive amount of generated traffic;
- network operators can benefit from traffic reduction, which is particularly crucial for their infrastructures expected to be largely overwhelmed in the near future.

Programmable packet treatment. Recent advancements in networking technologies make the deployment of Sociocast at the network layer even more viable. We identify SDN as the key enabler for Sociocast. Thanks to its programmability, which reduces the complexity of network elements, SDN can inject forwarding/dropping rules and properly manipulate headers of packets to make more efficient their forwarding.

Such policies can be defined in a network application, with no need to modify the data plane of the underlying network infrastructure.

4. THE ARCHITECTURAL FRAMEWORK

The main entities of the envisioned framework are: the *Sociocast nodes*, the *SDN network* (encompassing both switches and controller), augmented with the notion of Sociocast, and the *Sociocast Relationship Service*, as shown in Figure 1 and detailed in the following sections.

4.1 The Sociocast nodes

The *Sociocast nodes* are the endpoints of a Sociocast communication. They are legacy IoT devices (e.g., smartphones, sensors) augmented with the *Sociocast Support Layer* (ScSL) running on top of the transport layer, through which they are enabled to create, send and/or receive Sociocast packets. The ScSL exposes the Sociocast Application Programming Interfaces (APIs) to the applications that want to use the Sociocast communication configuration for data delivery. It is through this layer that Sociocast packets are created and received by the end devices.

4.2 The SDN network

The SDN network is composed of three different planes, according to the legacy deployment.

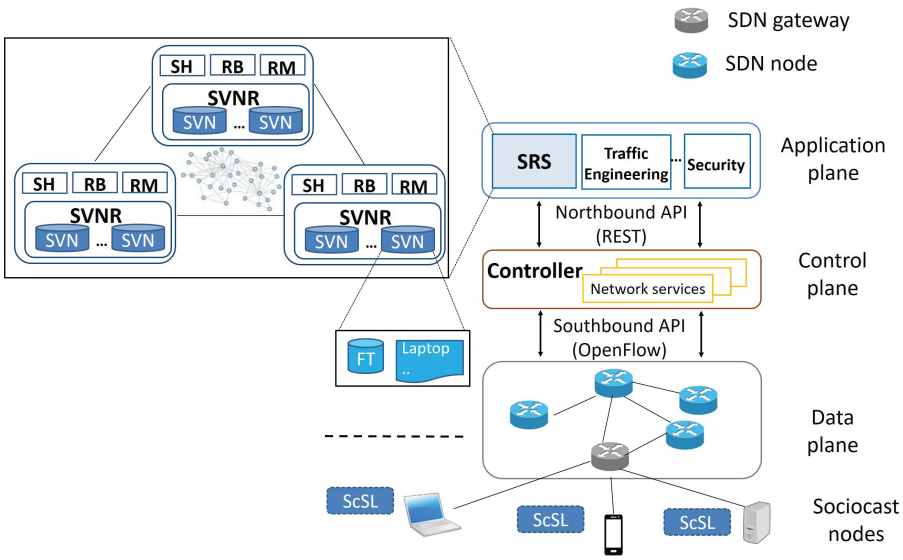


Fig. 1 – Sociocast architectural framework.

The *data plane* encompasses the *SDN switches*. They are SDN-enabled network nodes which are connected to each other and interact with the SDN controller. Between them, the *SDN gateways* are the ingress/egress nodes of the SDN network. SDN nodes interact with the SDN controller through the OpenFlow (OF) southbound interface.
The *control plane* includes the *SDN controller*, which oversees the SDN nodes, according to specific orchestration policies defined at the *application plane*. It tracks the graph of the network topology in the *Network Information Base* (NIB). According to information in the NIB and policies defined by network applications, it injects rules in the so-called *flow tables* of SDN nodes to enable the forwarding of sociocast packets through OF messages [31].

4.3 The Sociocast Relationship Service

The Sociocast Relationship Service (SRS) is implemented at the application plane, next to conventional SDN applications, and it provides the following main functionalities:

1. establishing social relationships between nodes. Without loss of generality, we inherit concepts and methodologies regarding the policies for the establishment of the social links between nodes from the well-accepted SIoT paradigm [30];

2. keeping track of the established social relationships;

3. providing interfaces towards the SDN network and to navigate the social network so to identify the nodes that belong to the set of the potential recipients/authorized senders of a Sociocast packet.

Herein, a major element is the *Social Virtual Node (SVN)*, which represents the digital counterpart of a physical device. It stores some metadata providing information about the nature of the device and a list of friends, which is organized in a table named *Friends Table*. For each friend in the table, the SVN records the type(s) of friendship(s), defined according to the SIoT paradigm and the trust level associated with each friend. The *Social Virtual Node Repository (SVNR)* stores all SVNs associated to the physical devices in a given area. Indeed, one SVNR is responsible for provididing the described services for the objects in a given area; more SVNRs are then interconnected in a distributed system. The following modules are associated to the SVNR.

- The *Relationship Manager* (RM) is responsible for the relationships' lifecyle management, i.e., detecting, creating, updating and deleting relationships[1].

- The *Relationship Browser* (RB) navigates the Friends Table to find potential recipients of a Sociocast packet, according to their position in the social network. Policies for the social network navigation are discussed in [32].

- The *Sociocast Handler (SH)*, whenever queried by the SDN controller, provides the members of a Sociocast group, after querying the RB module,

[1]For a detailed description of relationships management, the reader is referred to [32].

through a Representational State Transfer (REST) API.

SVNRs, along with relevant functionalities (i.e., RM, RB and SH), can be deployed as a peer-to-peer system, for instance building upon the one described in [32].
Our design choice is aimed at providing an implementation of SDN-based group communications based on a de-facto global IoT resource directory, which is distributed and without a single player in control of the system. Digital representations of physical IoT devices will run in distributed servers and can create autonomously social-like relationships with each other. Based on such a distributed resource directory, interactions (both point-to-point and point-to-multipoint) between IoT resources belonging to different platforms can be straightforwardly enabled. Each SVNR (or group of SVNRs) could, in fact, contain the images of the devices belonging to a given platform, it can be owned and maintained by the owner of the platform (or even the owner of the group of IoT devices), and it interacts in a peer-to-peer fashion with other SVNRs constituting the SRS.

5. SOCIOCAST IN ACTION

In the following, we detail the main steps for the creation of a Sociocast packet. Then, we describe Sociocast data delivery according to a push-based dissemination, publish/subscribe procedures to sociocast groups, as well as filtering according to sociocast rules.

5.1 Creating a Sociocast packet

A Sociocast packet is created whenever a device needs the services offered by the Sociocast framework, which are intended to: *(i)* disseminate data in a push-like manner; *(ii)* indicate the subscription to a Sociocast group; *(iii)* or to filter/prioritize data from particular senders. Whenever a packet is created, it has to indicate which one of these three types of service is requested. The above are the types of Sociocast services supported in the current implementation, but the set of Sociocast services can be easily extended in the future.
Let us consider a device, say A, which creates a packet with data to be sent to a Sociocast group. The application in A makes a request to the ScSL via the available APIs, providing the following information: *(i)* the type of requested Sociocast service; *(ii)* the social relationship (e.g., OOR, CLOR) according to which the Sociocast group has to be formed; *(iii)* the social distance (number of hops over the social network), which represents the scope of the Sociocast group.
The ScSL reacts to the incoming request by creating an IP packet with the following header fields:

- Source IP address: the IP address of the source device.
- Destination IP address: a fixed IP address, identified in this paper as IP_{SC}, assigned to Sociocast that allows SDN gateways to identify Sociocast packets.
- Sociocast Tag: a 2-bytes field that is carried inside the transport-layer destination port and is used to uniquely identify the type of social relationship and other appropriate filters (e.g., number of hops, possible application of Sociocast, etc.). The encoding is as follows:
  - Metadata: device metadata available for future applications.
  - RelationFilter: type of relationship (e.g., OOR, SOR, CLOR, etc.).
  - FeatureGroup: type of Sociocast services needed by the application (e.g., GroupCreation, SourceFiltering, Pub/Sub).
  - Radius: maximum distance, in number of hops, from the source.

  Fig. 2 shows some examples of Sociocast Tag configuration.

Being Sociocast packets identified through conventional layers 3 and 4 header fields, legacy matching rules can be applied, with no need to resort to OF *experimenter fields* [33]. Such design choices would facilitate the deployment of Sociocast, which candidates itself as a short-term solution to be exploited by network operators.
For the sake of simplicity, the encoding described above refers to the case where the IPv4 is used. Similar considerations hold for IPv6 packets, for which matching fields can be handled by OF since version 1.2 [33].
For those constrained IoT devices belonging to Low power and Lossy Networks (LLNs), 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) header compression methods can be used [34] over the link interconnecting the devices to the SDN gateway. For the IPv6 headers, compression methods may also affect source and destination addresses, and they vary according to the fact that the source is communicating with nodes either within or outside the WPAN. In the latter case, a 50 percent compression ratio can be still achieved by letting the full destination address, carrying the Sociocast address, be transmitted.
TCP header compression for IoT scenarios [35] is still an open issue at the standardization level [36], not part of RFC 6282 [34]. The compression foresees to avoid sending the port numbers in each packet, which however does not affect the Sociocast communications as

the port number with the SOCIOCAST TAG is reconstructed at the gateway. Indeed, decompression occurs at the SDN gateway letting Sociocast packets travel with conventional IP header fields in the SDN network. Similar operations are performed at the SDN gateways the destinations are attached to, if the latter ones belong to a WPAN.

5.2 Push-based data dissemination

Once the Sociocast packet is created with data to be disseminated, it is sent by the source device and treated in the network through the following steps.

1. The Sociocast packet reaches the SDN gateway, which the source device is connected to. Since, initially, a forwarding rule is not set in the flow table of the SDN gateway, the GOTOCONTROLLER rule applies for it. Hence, a OF PACKET_IN message is issued to be transmitted to the SDN controller.

2. Upon reading the header of the Sociocast packet[2], the SDN controller realizes that a Sociocast group must be created (FEATURE field set to GROUPCREATION). Thus, it issues a request to the SRS, to retrieve the set of devices, intended to act as recipients of the Sociocast packet.

3. The SH triggers the browsing of the social network, as specified before, and returns to the SDN controller the addresses of the set of devices of the Sociocast group.

4. The SDN controller retrieves from the NIB the SDN nodes in the shortest paths towards the intended receivers of the Sociocast group. Then, it builds the routing paths by ensuring that SDN nodes belonging to the path towards multiple receivers receive a single rule and forward the Sociocast packet only once. Hence, it injects forwarding rules in the flow table of involved SDN nodes accordingly, by sending OF FLOW_MOD messages. In particular, the SDN gateways which the Sociocast destinations are attached to, will be instructed by the SDN controller with a rule that: *(i)* matches the Sociocast-related header fields that identify the Sociocast communication and *(ii)* foresees to forward the packet to the correct physical port after changing the destination Sociocast IP address with the IP destination (unicast) address as action. This is to ensure that all devices belonging to the Sociocast group correctly receive the Sociocast packet. Other SDN nodes, instead, are instructed to forward the Sociocast packet to the physical correct ports by matching the Sociocast fields values.

[2]The entire Sociocast packet is sent by the SDN gateway, hence a PACKET_OUT is transmitted by the controller, back to the SDN gateway [33].

Once the Sociocast group is created, subsequent Sociocast packets transmitted by the source device may be handled by the SDN gateway with no need to contact the SDN controller, but rather forwarded according to rules already available in the flow table. According to the legacy SDN implementation, a timeout is applied to rules injected by the controller into SDN nodes, to prevent a rule to stay in the table for a long time and unnecessarily occupy space in the flow table [33]. Within our framework, such a timeout can be set to reflect the lifetime and frequency of interactions within the Sociocast group, the mobility patterns of nodes.

5.3 Publish/subscribe

Sociocast can be exploited to support a publish/subscribe interaction model as well. In fact, a device can *subscribe* to receive packets *published* by devices identified by their position in the social network. For example, assume that device B wants to subscribe to receive packets generated by its friends of type OOR. If this is the case, it will generate a Sociocast packet with the FEATUREGROUP field set to PUB/SUB and the RELATIONFILTER field identifying an OOR.
Such an information will reach the SDN controller which will perform the following operations:

1. It sends a query to the SH and receives the identities of the devices with position in the social network consistent with the request by device B.

2. It adds this information in a pending interest table which tracks all subscriptions received by devices. Whenever a device begins to disseminate data, the SDN controller will check whether there are devices that have subscribed to its updates (e.g., B).

3. If this is the case, the SDN controller will instruct the SDN nodes in the path to B to forward the data packets to it.

5.4 Source filtering

Sociocast allows a device to select those that are entitled to send packets to it, based on their position in the social network. Such a feature can be used both in a proactive and a reactive way. More specifically,

- *Proactive*: a device might decide to receive packets by its *friends* only, for security reasons or to save energy, computational and communication resources.

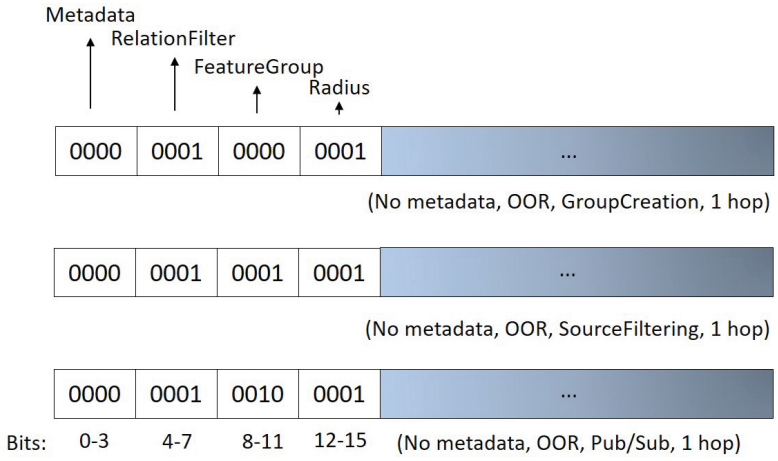


Fig. 2 – Examples of Sociocast Tag configuration.

- *Reactive*: the computational or communication load for a device may become too high, e.g., because of a DoS attack. If this is the case, the device might decide to accept packets by a subset of devices, based on their position in the social network. In this way Sociocast can be exploited to realize a firewall the policies of which change depending on the current load.

A device, say C, wishing not to receive packets from nodes with certain social properties sends a Sociocast packet by specifying in the FeatureGroup field SourceFiltering. Once the packet reaches the SDN controller, the latter one will query the SH, which will reply with the list of authorized IP addresses. Accordingly, the SDN controller will insert entries in the flow table of the SDN gateway which C is attached to, to specify the forwarding rule for packets destined to it sent from authorized senders and the dropping rule for those which are not allowed.

6. PERFORMANCE EVALUATION

In this section we describe the environment for the performance evaluation. More specifically, in Section 6.1 we describe the tools utilized for the performance evaluation, and in Section 6.2 we discuss the scenarios. The benchmark utilized for comparison purposes is presented in Section 6.3, whereas the considered performance metrics are identified in Section 6.4.

6.1 Tools and reference topology

The focus of the performance evaluation is to assess Sociocast in the case of *push-based data dissemination towards a group of devices*.

To this purpose, we built an emulation playground. In particular, the Mininet network emulator [5] has been used, it allows the creation of a network with thousands of nodes on the limited resources of a single (virtual) machine. In particular, it enables fast prototyping and experimental evaluation of OF-enabled networked systems. The experimental setting consists of the network topology depicted in Fig. 3. A full-mesh interconnects the core SDN nodes, which are the roots of a three-layers fat-tree topology. Up to 21 devices are attached to each SDN gateway (not all the devices are shown in the figure). ONOS has been considered as a reference SDN controller in the context of this work, due to its scalability properties and its highly modular architecture [6].

The ONOS controller interacts with an external SRS, which establishes social relationships among emulated devices, and manages them.

The ONOS controller and the Mininet network emulator are both running on the same virtual machine, while the SRS runs in a different one. Both these virtual machines are located in a physical server with an Intel Xeon(R) CPU E5-2630C v3 1.80 GHz x32 processor and 377,8 GiB of memory.

6.2 Social relationships settings and traffic patterns

The performance of the proposed solution has been evaluated with a set of representative IoT test configurations properly designed to take into account different numbers and distributions of nodes in the emulated topology, different physical distances between sources and destinations, and different types of service. This is aimed at making the obtained results as generalizable as possible and having a clear idea of the potential and limits of

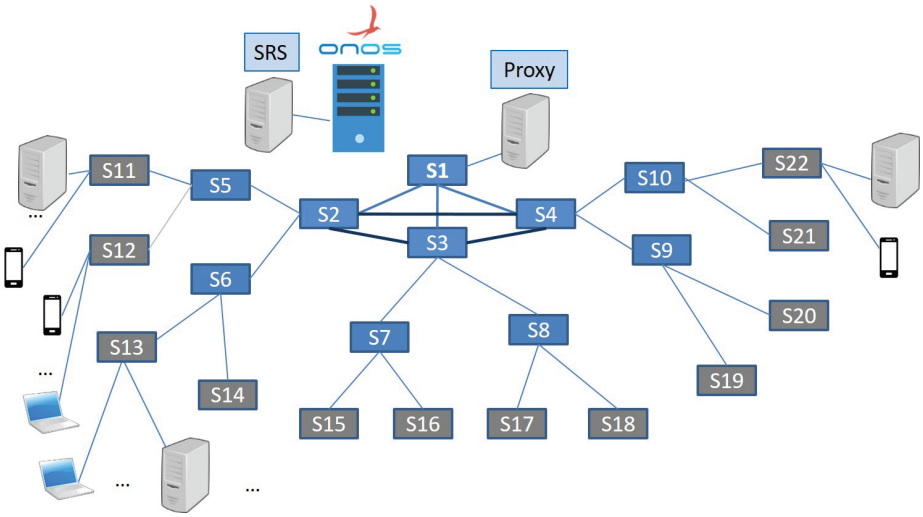


Fig. 3 – Reference topology.

Sociocast in multiple scenarios. Each of the test configurations has been mapped onto a use case characterized by the exploitation of a particular type of social relationship between the devices involved. In this way, helpful guidelines can be provided about the suitability of the proposed solution in the context of different application scenarios and, at the same time, of the effectiveness of communications based on each of the possible social-like relationships established among IoT devices. Details are given in the following subsections. Table 1 also summarizes the major features characterizing each scenario, which are: the types of social relationship (shortened as Rel.), the number of destinations (shortened as DSTs) for each communication, their distance from the source (shortened as SRC), and their position with reference to the considered network topology.

6.2.1 Scenario A: Smart industrial plant

Group communication needs: an industrial plant is equipped with several connected devices (sensors and actuators) and one of these (randomly selected) belonging to the emulated topology issues a Sociocast packet destined to *all the devices connected to the same gateway*. The group can be created, for instance, for the dissemination of alarms, for group configuration and reconfiguration, for functional testing.

Involved relationship type: CLOR.

Endpoint distribution profile: all endpoints clustered in the same area.

6.2.2 Scenario B: Smart home monitoring

Group communication needs: a randomly selected device in the emulated topology, resembling a smartphone of a user currently at office, acts as a sender and issues a Sociocast packet to create a group of recipients made of *all the smart devices connected to the (same) home gateway, which is different from the one the user's smartphone is attached to*. The group can be created, for instance, to notify devices to configure a warm welcome for the user.
Involved relationship type: OOR (ownership).
Endpoint distribution profile: sender in a location and all destinations clustered in a different (potentially) remote location.

6.2.3 Scenario C: Wireless Sensor Network (WSN) management

Group communication needs: a randomly selected device in the emulated topology acts as a sender and issues a Sociocast group creation destined to *all the devices of the same brand, uniformly distributed in the topology* to disseminate a new configuration for the device, a software update, or a new driver version.
Involved relationship type: POR (parental).
Endpoint distribution profile: uniform distribution of endpoints.

6.2.4 Scenario D: Smart mobility

Group communication needs: we assume mobile devices (e.g., smartphones, laptops) carried by people moving in a *smart city/smart campus* and interacting with

other devices met either in the neighborhood or close offices/classrooms. The type of the data exchanged within the group includes: information related to mobility applications, tourist information, data for the implementation of any Intelligent Transportation Systems application.
Involved relationship type: SOR.
Endpoint distribution profile: variable location of endpoints in the group.

6.2.5 Relationships creation

As to the creation of the relationships, these have been set in deterministic way except for the SOR. In particular, different groups of devices linked with POR, OOR, and POR relationships are created so as to have from 5 to 20 recipients for each simulated communication. However, the CLOR relationship has been created between devices that are connected to the same gateway as the co-location has to be assured. As to the SOR relationships used in Scenario D, these are established between devices in the emulated topology according to their physical distance and follow a simple probabilistic model. The principle adopted is such that the closer the devices, the higher the probability that the two devices have established a SOR relationship. Accordingly, devices attached to the same SDN gateway (i.e., an Access Point) have the highest probability to establish it. These devices are characterized by sharing the same path to reach the root node ($s1$ in Figure 3), which is made of 4 SDN nodes. We base on this number to define the notation to denote the relevant probability to create a SOR between them: $p_{soc,4}$. Following the same principle, devices sharing three, two, or one SDN nodes in the path to reach $s1$, establish a relationship with probability $p_{soc,3}$, $p_{soc,2}$, and $p_{soc,1}$, respectively. The higher j the higher the probability $p_{soc,j}$, with $j \in \{1, 2, 3, 4\}$. The setting of $p_{soc,j}$ used in the performed simulations is reported in Table 2; different configurations have been considered to evaluate the impact of different numbers of friends and their distribution in the considered topology.

6.3 Benchmark scheme

The performance of the proposal has been compared against an application-layer solution we refer to as *multiple unicast* (labeled in the plots as *M-Unicast*). Note that also for this benchmark scheme, we are focusing on the push-based data dissemination scenario. The choice of this benchmark is meant to *quantitatively* estimate the benefits of the Sociocast proposal against an application-layer solution. In the latter one, the network layer is agnostic about the communicating group, but it offers the same features in terms of sender-initiated and dynamic Sociocast group creation, hence ensuring a fair comparison. Specifically, the source node contacts a proxy in charge of interacting with a SIoT-like platform to get the set of intended destinations belonging to the Sociocast group. The latter one is described through attributes/metadata defined at the application layer, similarly to the information encoded in the tags in Sociocast packets. After retrieving the list, the proxy forwards it to the source node which sends the packet to the destinations through multiple unicast exchanges. In other words, the controller sets up distinct routing paths for each destination and some links can be shared by multiple paths towards destinations belonging to the same group. Without losing generality, we assume that the proxy is attached to the root node of the topology (i.e., $s1$ in Fig. 3).

6.4 Metrics

The following metrics have been considered to evaluate the performance of the compared schemes in the creation of a Sociocast group and data exchange among its members:

- the *number of OF signaling packets* exchanged between SDN nodes and controller to build routing paths towards the intended Sociocast destinations. The metric only refers to the control packets exchanged to process incoming requests from sociocast nodes at the SDN gateway, namely PACKET_IN, PACKET_OUT and FLOW_MOD. The background (periodic) signaling exchanged between the controller and the SDN nodes is not considered;

- the *number of data packets* exchanged into the network to reach all the intended destinations of the communicating group, once it has been created; the metric considers the number of transmitted packets per link and are represented by either Sociocast or M-Unicast packets.

For the benchmark scheme, the request packets issued by the source towards the proxy as well as the signaling messages required to instruct the relevant SDN nodes towards it are also considered.
The above metrics have been measured through the well-known Wireshark network protocol analyzer[3].
Comparison experiments have been conducted when varying the number of destinations (or relevant probability settings) and are averaged over 20 runs.

[3]Please notice that the analysis of the signaling incurred for the creation of social relationships between devices is outside the scope of this paper and is peculiar of the conceived SIoT implementation. An interested reader is referred to [37].

Table 1 – Summary of the main social relationships settings.

Scenario	Use case	Rel.	#DSTs	SRC-DSTs Distance	Position of DSTs
A	Smart industry	CLOR	5-20	1 hop for all destinations	Attached to the same SDN gateway (= sender)
B	Smart home	OOR	5-20	Fixed for a given set of destinations	Attached to the same SDN gateway (≠ sender)
C	WSN management	POR	5-20	1-7 hops	Uniformly distributed in the topology
D	Smart mobility	SOR	*	*	*

Table 2 – Probabilities of SOR establishment.

Sim-ID	#Destinations	$p_{soc,1}$	$p_{soc,2}$	$p_{soc,3}$	$p_{soc,4}$
1	4.5	0.1	0.2	0.3	0.4
2	5.5	0.2	0.3	0.4	0.5
3	8.4	0.3	0.4	0.5	0.6
4	11.2	0.4	0.5	0.6	0.7
5	12.7	0.5	0.6	0.7	0.8
6	15.3	0.6	0.7	0.8	0.9
7	18.4	0.7	0.8	0.9	1

7. EXPERIMENTAL RESULTS

In this section we show the performance results. More specifically, in Section 7.1 we assess Sociocast in terms of generated signaling packets; whereas in Section 7.2 we will focus on data packets.

7.1 Signaling packets

The first set of results aims to analyze the control plane signaling footprint incurred by the proposal w.r.t. the benchmark scheme. Fig. 4 reports the number of exchanged OF packets when varying the number of destinations of the Sociocast group for the scenarios A-C, whereas the results for scenario D are shown in Fig. 5(a). It can be clearly observed that for the M-Unicast approach the metric significantly increases with the number of destinations, in all the considered scenarios. Such a trend is due to the fact that the end-to-end communication path towards each *single* destination needs to be discovered with the support of the SDN controller. In other words, an SDN node receives a number of M-Unicast packets to forward equal to the number of destinations it allows to reach. For each of them, it contacts the controller by generating a Packet_In message and waits for the corresponding Packet_Out and Flow_Mod with instructions about the forwarding behaviour.

For a given number of destinations, the highest number of OF packets are exchanged in case of Scenario C. In the latter one, indeed, the destinations are spread over the topology and the routing path towards them may involve several SDN nodes (and gateways). Scenario B follows with a lower number of exchanged OF packets. In Scenario A, instead, only a single SDN gateway is in charge of Sociocast packet forwarding. It is the only SDN node transmitting and receiving OF packets.

In the proposed Sociocast solution, the controller is in charge of building routing paths towards them so to avoid the forwarding of the same Sociocast packet over the same link.

Hence, unlike the benchmark scheme, in our proposal, those SDN nodes which belong to the paths towards different destinations receive only a single Sociocast packet to forward and a single FLOW_MOD from the controller. The gain of Sociocast w.r.t. M-Unicast in terms of exchanged OF packets gets more remarkable as the number of destinations increases. For instance, in Scenario C, it passes from a factor of around 6 for five destinations to a factor of more than 14 for twenty destinations.

It is worth observing that, in Sociocast, a single FLOW_MOD message may convey multiple rules to be injected into an SDN node. In particular, Table 3 reports, for Scenario A, the size of the FLOW_MOD message, as measured at the SDN gateway, which the source and the destinations are both attached to. For the Sociocast proposal, the size reasonably increases with the number of destinations to accommodate the action rule for each of them. The rule specifies the physical output port as well as the change of the IP address from Sociocast to unicast. For M-Unicast, each FLOW_MOD carries a single rule, since its injection is issued per each M-Unicast packet traversing an SDN node. The size increases of less than a factor of 3 for the Sociocast approach compared to M-Unicast, in the case of twenty destinations.

Despite the larger size of FLOW_MOD packets, it can be easily inferred that, overall, the OF signaling footprint of the proposal, in terms of number of exchanged bytes, is significantly lower than M-Unicast. Also, the proposal better scales with the size of the Sociocast group.

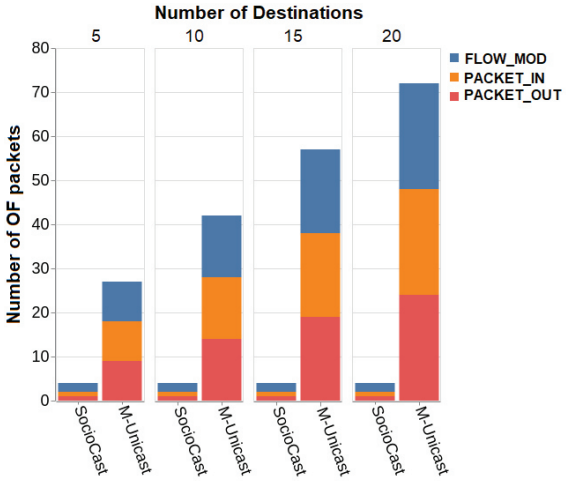
Similar to the benchmark scheme, the proposal experiences the largest signaling in Scenario C, wherein multiple SDN nodes, involved in forwarding Sociocast packets to destinations, spread over the topology, need to be instructed.

Similar considerations hold for Scenario D, Fig. 5(a). Also in such a case, the proposed Sociocast solution is less sensitive to the simulation settings (i.e., size of Sociocast group and its configuration in terms of proximity of destinations w.r.t. the source) than the benchmark.

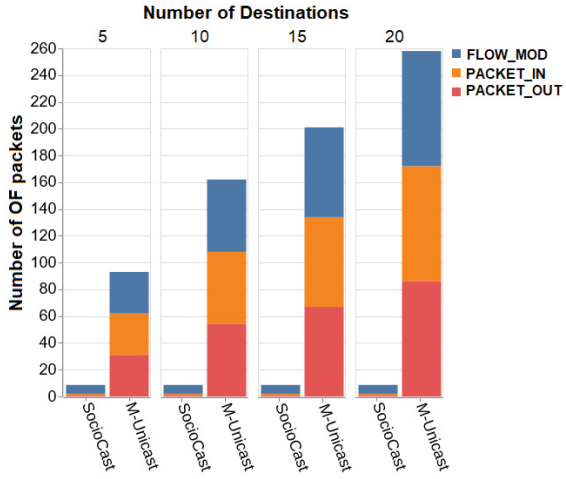
7.2 Data packets

Results in Fig. 6 shed further light into the efficiency of the compared schemes in delivering the data packets. Similar to the OF signaling, also the number of exchanged Sociocast packets increases with the number of destinations; the highest values are experienced for Scenario C and the lowest ones in Scenario A.

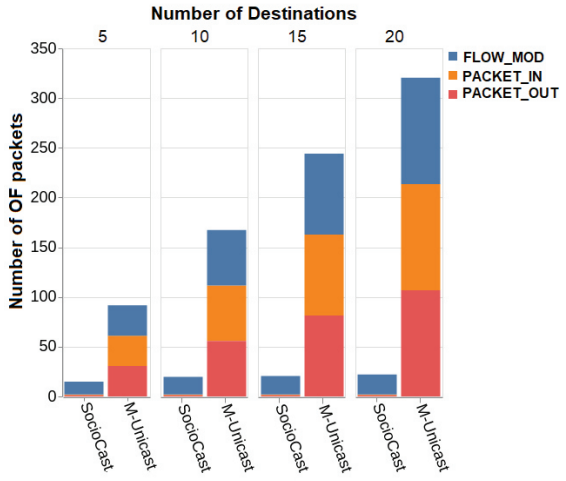
As a general remark, it can be observed that the proposal is less sensitive to increases in the number of destinations when compared to the benchmark. This happens because the controller builds the routing paths to avoid that packets are redundantly transmitted over a given



(a) Scenario A

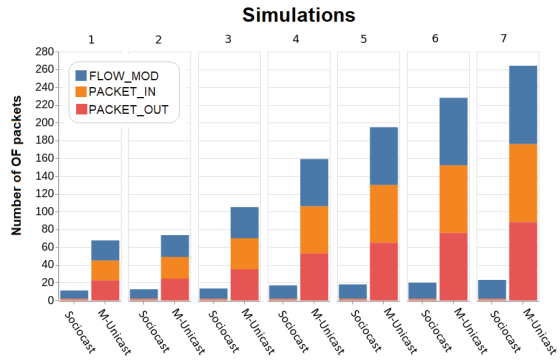


(b) Scenario B

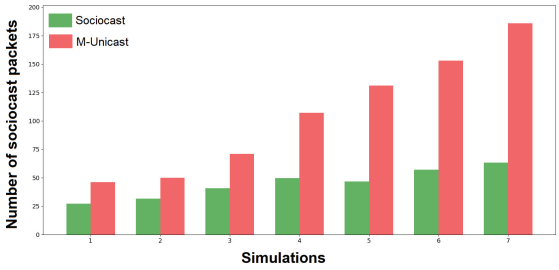


(c) Scenario C

Fig. 4 – Sociocast Vs. Multiple Unicast: Exchanged OF packets for Sociocast group creation when varying the number of destinations, in different scenarios.



(a) OF packets exchanged to create the Sociocast group



(b) Exchanged data plane packets in the topology

Fig. 5 – Sociocast Vs. Multiple Unicast: performance metrics under different simulation settings for Scenario D. The different simulation runs (from 1 to 7) correspond to the different Sim-IDs of Table 2.

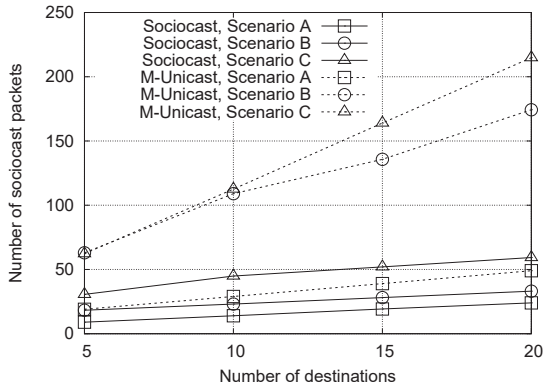


Fig. 6 – Sociocast Vs. Multiple Unicast: Exchanged data plane packets when varying the number of receivers, in different scenarios.

Table 3 – Size (in bytes) of the FLOW_MOD packet for Scenario A.

#Destinations	M-Unicast	Sociocast
1	172	172
5	172	252
10	172	332
15	172	412
20	172	492

link shared by more destinations.

This is not the case for the M-Unicast solution where forwarding decisions are separately taken for each data packet, according to the address of the intended destination.

When referring to Scenario A, the M-Unicast approach always sends twice as many data packets as the proposal. This is an obvious consequence of the fact that, after receiving the destinations list, for the M-Unicast approach there are two packets, for each destination, traveling into the topology. One packet travels from the source to the SDN gateway, and the other one from the SDN gateway to the corresponding destination. This does not apply for the Sociocast approach, where there is only the data packet from the SDN gateway to each destination.

Improvements get larger for other scenarios.

In Scenario B, more SDN nodes are involved in the routing path, despite the fact that all the destinations are connected to the same SDN gateway. Hence, more data packets travel into the network, especially for the M-Unicast solution. Such a trend is more remarkable in Scenario C, due to the larger spread of destinations over the topology. A similar trend is observed for Scenario D in Fig. 5(b).

Not surprisingly, improvements of Sociocast w.r.t. M-Unicast are greater in Scenario B compared to Scenario C. Indeed, in Scenario B the path towards all intended destinations is the same from the source to the SDN gateway. Hence, in Sociocast, the SDN controller judiciously issues rules that prevent from forwarding duplicated packets over the same links.

8. DISCUSSION AND CONCLUSION

In this paper we have proposed and analyzed the behaviour of an architectural framework encompassing all the entities, functionalities, and procedures that support a fresh new network-layer group dissemination method, i.e., Sociocast, by leveraging a software-defined network approach.

Results achieved through an emulation testbed show the better scalability of the proposal in terms of OF signaling and data packet redundancy when compared to an application-layer benchmark scheme, under different representative IoT scenarios.

Improvements are achieved by leveraging a purpose-built network application in the controller *(i)* in charge of identifying the set of Sociocast destinations by interacting with an external SIoT platform (feature implemented at the application layer by the benchmark scheme) and *(i)* responsible for smartly building routing paths towards multiple receivers so as to avoid packet duplication over links. SDN allows to manage the implementation of such functionalities at the control plane in a flexible and programmable manner, with no changes in the forwarding elements, hence making the devised framework practically viable at a low implementation cost.

Benefits of the proposal are definitely large when big groups of destination devices are clustered together, as witnessed by results referring to Scenario B: the OF signaling is reduced by a factor higher than 10 and the number of exchanged data packets shrinks by more than a factor of 5 (for twenty destinations). The lower gains for Sociocast packets w.r.t. OF signaling are due to the fact that Sociocast also resorts to multiple unicasts forwarding in the last hop from the SDN gateway towards the intended destinations, to ensure successful reception at the application layer. It can be further easily inferred (although not shown in results) that improvements get even larger as the distance between the source and the set of destinations increases.

Overall, the proposal is especially suited for push-based data dissemination to large Sociocast groups highly clustered and far from the source, which well resembles the case of multiple devices of a smart home (e.g., appliances) to be remotely configured by the user's smartphone.

In the other cases, the gains are also significant and always higher than a factor of 2.

The achieved encouraging results motivate us to further explore this fertile research area which has large room for improvements. The effectiveness of the proposal in handling other Sociocast features, like source filtering and publish/subscribe, needs to be practically explored. As a further challenge, IoT devices belonging to Sociocast groups may move long distances between different access points. Hence, tracking their positions at the virtual counterparts (SVN and SVNR), as well as managing the forwarding rules associated to them in the SDN nodes, become very difficult and entail proper workarounds which will be a subject matter of future investigations.

ACKNOWLEDGEMENT

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] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," *IEEE network*, vol. 14, no. 1, pp. 78–88, 2000.

[3] L. Atzori, A. Iera, and G. Morabito, "Sociocast: A new network primitive for IoT," *IEEE Communications Magazine*, vol. 57, no. 6, pp. 62–67, 2019.

[4] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[5] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proc. of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, p. 19, ACM, 2010.

[6] "ON.LAB, "Introducing ONOS - a SDN network operating system for service providers," 2014.

[7] S.-H. Shen, "Efficient SVC multicast streaming for video conferencing with SDN control," *IEEE Transactions on Network and Service Management*, 2019.

[8] B. Knutsson, H. Lu, W. Xu, and B. Hopkins, "Peer-to-peer support for massively multiplayer games," in *IEEE INFOCOM 2004*, vol. 1, IEEE, 2004.

[9] X. S. Sun, Y. Xia, S. Dzinamarira, X. S. Huang, D. Wu, and T. E. Ng, "Republic: Data multicast meets hybrid rack-level interconnections in data center," in *2018 IEEE 26th International Conference on Network Protocols (ICNP)*, pp. 77–87, IEEE, 2018.

[10] X. Li, Y.-C. Tian, G. Ledwich, Y. Mishra, X. Han, and C. Zhou, "Constrained optimization of multicast routing for wide area control of smart grid," *IEEE Transactions on Smart Grid*, 2018.

[11] M. Hosseini, D. T. Ahmed, S. Shirmohammadi, and N. D. Georganas, "A survey of application-layer multicast protocols," *IEEE Communications Surveys and Tutorials*, vol. 9, no. 1-4, pp. 58–74, 2007.

[12] S. Islam, N. Muslim, and J. W. Atwood, "A survey on multicasting in software-defined networking," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 355–387, 2017.

[13] Z. AlSaeed, I. Ahmad, and I. Hussain, "Multicasting in software defined networks: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 104, pp. 61–77, 2018.

[14] A. Adams, J. Nicholas, and W. Siadak, "RFC 3973, protocol independent multicast-dense mode (PIM-DM): Protocol specification (revised)," tech. rep., 2005.

[15] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan, "RFC 3376, Internet Group Management Protocol, version 3," tech. rep., August 2006.

[16] H. Holbrook and B. Cain, "RFC 4607, Source-Specific Multicast for IP," tech. rep., August 2006.

[17] A. Venkataramani *et al.*, "MobilityFirst: a mobility-centric and trustworthy internet architecture," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 74–80, 2014.

[18] C. A. Marcondes, T. P. Santos, A. P. Godoy, C. C. Viel, and C. A. Teixeira, "Castflow: Clean-slate multicast approach using in-advance path processing in programmable networks," in *2012 IEEE Symposium on Computers and Communications (ISCC)*, pp. 000094–000101, IEEE, 2012.

[19] M. Zhao, B. Jia, M. Wu, H. Yu, and Y. Xu, "Software defined network-enabled multicast for multi-party video conferencing systems," in *2014 IEEE International Conference on Communications (ICC)*, pp. 1729–1735, IEEE, 2014.

[20] D. Katsaros, N. Dimokas, and L. Tassiulas, "Social network analysis concepts in the design of wireless ad hoc network protocols," *IEEE network*, vol. 24, no. 6, pp. 23–29, 2010.

[21] Y. Zhu, B. Xu, X. Shi, and Y. Wang, "A survey of social-based routing in delay tolerant networks: Positive and negative social effects," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 387–401, 2012.

[22] K. W. *et al.*, "Exploiting small world properties for message forwarding in delay tolerant networks," *IEEE Transactions on Computers*, vol. 64, no. 10, pp. 2809–2818, 2015.

[23] K. Wei, X. Liang, and K. Xu, "A survey of social-aware routing protocols in delay tolerant networks: applications, taxonomy and design-related issues," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 556–578, 2013.

[24] M. Xiao, J. Wu, and L. Huang, "Community-aware opportunistic routing in mobile social networks," *IEEE Transactions on Computers*, vol. 63, no. 7, pp. 1682–1695, 2013.

[25] D. J. Watts, "Networks, dynamics, and the small-world phenomenon," *American Journal of sociology*, vol. 105, no. 2, pp. 493–527, 1999.

[26] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *Proceedings of ACM MobiHoc*, pp. 299–308, ACM, 2009.

[27] W. Gao, Q. Li, B. Zhao, and G. Cao, "Social-aware multicast in disruption-tolerant networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 5, pp. 1553–1566, 2012.

[28] X. Hu, T. H. Chu, V. C. Leung, E. C.-H. Ngai, P. Kruchten, and H. C. Chan, "A survey on mobile social networks: Applications, platforms, system architectures, and future research directions," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1557–1581, 2015.

[29] J. Fan, J. Chen, Y. Du, W. Gao, J. Wu, and Y. Sun, "Geocommunity-based broadcasting for data dissemination in mobile social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 734–743, 2013.

[30] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (SIoT)–when social networks meet the internet of things: Concept, architecture and network characterization," *Computer networks*, vol. 56, no. 16, pp. 3594–3608, 2012.

[31] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[32] L. Atzori, C. Campolo, B. Da, R. Girau, A. Iera, G. Morabito, and S. Quattropani, "Enhancing identifier/locator splitting through social internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2974–2985, 2018.

[33] "Openflow switch specification - version 1.3.1 Open Networking Foundation (ONF)," September 2012.

[34] J. Hui and P. Thubert, "Compression format for IPv6 datagrams over ieee 802.15.4-based networks, RFC 6282," September 2011.

[35] A. Ayadi, D. Ros, and L. Toutain, "TCP header compression for 6LoWPAN," *Internet Draft (draft-aayadi-olowpan-tcphc-00), work in progress*, 2010.

[36] C. Gomez, A. Arcia-Moret, and J. Crowcroft, "TCP in the Internet of Things: from ostracism to prominence," *IEEE Internet Computing*, vol. 22, no. 1, pp. 29–41, 2018.

[37] L. Atzori, C. Campolo, B. Da, R. Girau, A. Iera, G. Morabito, and S. Quattropani, "Smart devices in the social loops: Criteria and algorithms for the creation of the social links," *Future Generation Computer Systems*, vol. 97, pp. 327–339, 2019.

AUTHORS

Luigi Atzori (PhD, 2000) is professor of Telecommunications at the University of Cagliari, where he leads the activities of the MCLab laboratory (Multimedia Communications) with around 20 affiliated researchers. Since 2018, he has been the coordinator of the master degree course in Internet Technology Engineering at the University of Cagliari. His research interests fall in the area of Internet of Things (IoT), with particular reference to the design of effective algorithms for the realization of social networks among connected devices to create the Social IoT paradigm. His interests also falls in the area of Quality of Experience (QoE), with particular application to the management of services and resources in new generation networks for multimedia communications. Lately, he also applies the study of QoE to IoT services. He serves regularly in the conference organizing committee of the sector and as associate and guest editor of several international journals (IEEE IoT journal, Ad Hoc Networks, IEEE IEEE Open Journal of the Communications Society, IEEE Communications Magazine, etc.).

Claudia Campolo (Senior Member, IEEE) received the master's and Ph.D. degrees in telecommunications engineering from the Mediter- ranea University of Reggio Calabria, in 2007 and 2011, respectively. In 2008, she was a Visiting Ph.D. Student with the Politecnico di Torino and a DAAD Fellow with the University of Pader- born, Germany, in 2015. She is currently an Associate Professor of telecommunications with the Mediterranea University of Reggio Calabria. Her main research interests include vehicular networking, 5G, and edge computing.

Antonio Iera is professor of Telecommunications at the University of Calabria, Italy. He 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 from 1997 to 2019 with the University of Reggio Calabria. He has published more than 300 papers in high-quality journals and conferences and has given several Tutorials and invited speeches at international events on the topics of IoT, Social-IoT, and 5G networks. He is also serving as Editor in Chief for Computer Networks, Elsevier.His research interests include wireless and mobile 5G networks and Internet of Things.

Giuseppe Massimiliano Milotta was born in Catania, Sicily (Italy) on May 07, 1986. He received his master degree at "Università degli studi di Catania" in 2017 after a six-month collaboration with "Télécom Paris Tech", Paris(France) in 2016, which led to the realization of his master degree thesis. He is currently enrolled at the last year of PhD in Information Engineering at "Università Mediterranea di

Reggio Calabria", Reggio Calabria (Italy) where is still cooperating with the CNIT research unit of Catania. His main research activities focus on SDN and the IoT's world, with a particular interest in the UAVs and the security systems.

Giacomo Morabito received the laurea degree cum laude in Electronic Engineering and the Ph.D from the University of Catania (Italy) in 1996 and 2000, respectively. Since 1999 to 2001 he was a research engineer at Georgia Tech (Atlanta, USA). Since 2001 he is with the Department of Electric, Electronic, and Computer Engineering of the University of Catania where he is currently full professor of telecommunications. His research interests focus on analysis and design of wireless networks and Internet of Things.

Salvatore Quattropani received the BSc. and MSc. degrees in Computer Engineering in 2016 and 2017, respectively, from University of Catania, Italy. Since August 2017 he is with the Consorzio Nazionale Interuniversitario per le Telecomunicazioni as a Research Engineer. His research interests focus on IoT connectivity and computing approaches.