The Alliance of SDN and MQTT for the Web of Industrial Things

Claudio Bartoli, Michele Bonanni, Francesco Chiti, *Senior Member, IEEE*, and Laura Pierucci, *Senior Member, IEEE*

***Abstract*—The modern vision of "smarter" factory processes requires the adoption of Industrial Internet of Things (IIoT) ecosystems, where the large scale use of sensors, actuators, and connected devices enable a high amount of heterogeneous data gathering and sending to data centers for analysis and monitoring purposes. However, the foundation of the Web of Industrial Things (WoIT) faces open issues due to the constraints imposed by the involved devices, the technological heterogeneity, and the complex interactions and, hence, communications patterns. Toward this paradigm, a general framework inspired by the software defined networking (SDN) principle have been here proposed, in order to jointly optimize the application requirements by minimizing end-to-end delay and, thus, prolonging the lifetime of the IIoT ecosystem. This article derives a general SDN based architecture to handle data flows in order to support industrial applications that extend the basic publish-and-subscribe scheme. Specifically, the MQTT-SN standard have been considered, since it represents a solution suited for WoIT in the presence of resources constrained devices. The proposed integrated solution has been validated in practical scenarios, pointing out remarkable performance w.r.t. the end-to-end message delivery latency, always with a limited overhead.**

***Index Terms*—MQTT-SN protocol, sensors and actuators domain, software defined networking (SDN), Web of Industrial Things (WoIT).**

I. Introduction

SEVERAL *Internet* of Things (IoT) definitions have been proposed in the recent literature [1], [2], as "*network of items-each embedded with sensors-which are connected to the Internet.*" or "*an Internet that can concurrently operate among TCP/IP and non-TCP/IP protocols, with Things that are objects identified by unique addresses*," to name a few. The mean idea behind IoT is to connect production, medical, automotive, or transportation devices (i.e., the *things*) with Information Technology (IT) and business-critical information systems in order to provide unprecedented added values in terms of innovation, efficiency, and quality [3].

Received 28 October 2024; accepted 20 January 2025. Date of publication 3 March 2025; date of current version 23 May 2025. Paper no. TII-24-5678. *(Corresponding author: Francesco Chiti.)*

The authors are with the Department of Information Engineering, University of Florence, 50121 Florence, Italy (e-mail: claudio.bartoli@unifi.it; michele.bonanni@unifi.it; francesco.chiti@unifi.it; laura.pierucci@unifi.it).

Digital Object Identifier 10.1109/TII.2025.3537291

In particular, the generally referred *Industrial Internet of Things (IIoT)* (or the *Industry 4.0*) leads towards the convergence of the IT and Operational Technology (OT) domains into an integrated cyber-physical manufacturing system aiming to support "smarter" factory processes. In this vision, the adoption of wireless sensor networks (WSNs) in an industrial scenario allows easy installations, reconfiguration, maintenance and support to mobility, with respect to wired systems. As consequence, Industrial Wireless Sensor and Actuator Networks (IWSANs) has been viewed as the concrete enablers of IoT paradigm, since they provide a heterogeneous *internet*-working and allow a more symbiotic interaction with the surrounding physical environment, gathering run-time information from and executing microtasks to modify it [4], [5].

According to an operation perspective, IIoT is supposed to perform a *chain* of tasks by iteratively involving different groups of devices, which can be arranged in a *logical* time-varying *clustered* topology. This is aligned with the fog computing (FC) principle, which supports the design of novel services beyond the current centralized Cloud concept, where their logic distributively runs in the proximity (i.e., on board of close Fog Servers), though they are still able to interact with applications running in the Cloud [6], [7].

One of the main challenges of the previously introduced distributed services is related to the set-up, maintenance, and ad hoc integration, which can be effectively achieved by adopting the so called Web of Things (WoT) approach [8]. Indeed, Web Services based on REpresentational State Transfer (REST) have been recently adopted to achieve interoperability particularly in the industrial Open Platform Communications Unified Architecture (OPC UA), that is widely considered as one of the main reference communication protocols for information exchanging among industrial applications [9], [10], [11]. OPC UA architecture integrates those protocols specifically designed for manufacturing and automation industry, such as fieldbuses, real-time Ethernet with the Web oriented protocols, i.e., Constrained Application Protocol (CoAP) and Message Queue Telemetry Transport (MQTT). CoAP is a transfer protocol first proposed to adapt Hyper Text Transfer Protocol (HTTP) for constrained devices with limited processing power, network bandwidth, and intermittent connectivity, and to enable more advanced interaction patterns w.r.t. HTTP, e.g., the support for multicast [12] and mechanisms such as resource observation. Instead, MQTT protocol was designed for monitoring applications and it is based on the publish-and-subscribe paradigm, by which publishers (e.g., sensors) transmit data messages to a broker, which in turn delivers such messages to interested entities, called subscribers [13]. It is worth noticing that OPC UA Release 14 is specifically referred to this publish-and-subscribe broker-based model. Moreover, a new version of MQTT protocol, called *MQTT for Sensor Networks* (MQTT-SN), has been specifically designed to face the constraints of a typical IoT domain with suitable architecture, interfaces, and components [14].

A crucial requirement in a realistic implementation of the WoIT paradigm is represented by the energy available to devices in order to make them to correctly operate [15]. Wireless communications implicitly require a nonwired power source, and a usual approach consists in endowing devices with batteries, whose sizing is, indeed, inevitably a tradeoff among different parameters, such as durability, performance, and system complexity or cost. Furthermore, this approach requires a specific maintenance cycle for batteries replacements.

This specific requirement can be addressed by applying software defined networking (SDN) that is an emerging paradigm to dynamic (re)design of network architecture by separating the *control* plane from the *data* forwarding plane; moreover, SDN has been recently proposed to address the specific resources-constrained nature of IoT devices [16]. According to the reference SDN architecture, a *logically centralized* controller has the knowledge of the overall network and can provide instructions to network devices, which consequently are not required to implement different protocols [17], [18]. SDN is expected to provide flexible and interoperable ad hoc architectures for a dynamic WoIT domain to accomplish complex and dynamic communications patterns, especially in the presence of heterogenous technologies and evolving applications. Specifically, SDN controller on demand defines routing policies and accordingly modifies rules and actions in the forwarding flow tables of network devices, which in turn only execute instructions and forward data traffics.

This article focuses on a system level architecture design based on SDN paradigm to a jointly data flows and power budget management. In particular, industrial devices (i.e., sensor and actuators) are logically clusterized, according to the homogeneous operations performed in terms of manufacturing and quality check. To address the possible intra- and intercluster(s) communications, we resort to the WoIT approach and we adopt the publish-and-subscribe feature of the MQTT-SN framework to setup extremely flexible end-to-end (e2e) sessions. However, their management, if performed in a distributed and reactive way, does not match the specific requirements of provided services.

To this purpose, we derived a solution inspired by the SDN paradigm to evaluate e2e paths among publisher(s) and subscriber(s). Specifically, we extended the general framework previously proposed in [19], toward a system level architecture design based on SDN paradigm, which allows to dynamically adapt a hierarchical communications scheme by resorting to a more distributed pattern. This concept eases the management of nonoverlapping clusters of industrial devices in order to optimize both the latency and the overall energy consumption and, consequently, maximizing the system availability and durability to support typical industrial applications.

More in details, our contributions are as follows.

1) We provide a reference WSAN architecture based on IEEE 802.15.4 Physical an MAC layers fully integrated into the SDN paradigm.
2) We develop a communication solution which applies and adapts the SDN principle to the MQTT-SN protocol in order to overcome the limitation of the centralized broker intermediation.
3) We characterize an effective objective function to be optimized, while evaluating the overall system performance of the proposed approach and comparing it with the more relevant existing communications protocols.

The article structure is as follows: after introducing the specific features and the open issue of WoIT domain in Section I, the specific literature is analyzed and discussed in Section II. The reference system model is presented in Section III in terms of both the general architecture and specific communications and control protocols. The overall performance is investigated in Section IV for different device deployments, pointing out a remarkable gain in terms of energy consumption and e2e latency, with a limited overhead, and, consequently, a good scalability. Finally, the conclusions are drawn in Section V.

II. Related Works

In the literature, several surveys present the specific challenges and applicability of the SDN principle to meet the application-specific requirements of a typical IoT domain. In particular, SDN-based architectures are coupled with network function virtualization (NFV) to cost-effectively provide the scalability and flexibility necessary for IoT services, as shown in [20]. In [21], an overview of SDN for IoT is provided, highlighting several SDN-based data aggregation and management approaches, with different architectures, i.e., edge, access, core, and data center networking, together with addressing the main requirements and issues to efficiently manage the sensor nodes resources. Moreover, future directions and open research issues are detailed to address aspects as mobility, policy enforcement, hardware platform for SDN-based solutions. Edge computing (EC) is an emerging paradigm which pushes computing tasks in proximity to mobile devices or sensors/actuators to reduce latency, save bandwidth, and improve security and privacy. Further, another recent survey [22] also discusses the issues of SDN-based EC architectures for IoT.

MQTT protocol, standardized by Oasis [23], is widely used to assist communication between servers and resource-constrained IoT devices. MQTT is a lightweight protocol based on publish/subscribe message exchanging via an intermediate device called broker. This approach allows delivery guarantees, low overhead, and power consumption. Moreover, a device can be easily added/removed to the network without perturbing the existing topology.

MQTT-SN is the version of MQTT thought specifically for sensor networks, i.e., in constrained environments and in the paper [24] a comparison with the CoAP Pub/Sub version is outlined addressing the traffic behavior of the two protocols. The experimental results shown similarity in the performance of these protocols and only small differences related to specific applications. The use of CoAP is preferred in high dynamic scenarios due to its connectionless communication mode while MQTT-SN is better for the transmissions of periodic data, such as in environmental monitoring, as it has lower topic registration overhead. The paper [25] analyzed the issue MQTT-SN cannot work without MQTT and consequently a pure UDP-based communication is not feasible as the standard does not specify a standalone broker. Therefore, a standalone MQTT-SN broker is shown together with the implementation of a prototype. The standalone broker and the UDP-based communication work more effective related to standard gateway implementations and show better performance when compared with MQTT and CoAP Pub/Sub with regards to publish delay and e2e latency parameters.

Recently, a combination of SDN and MQTT has been proposed in the literature for the management of complex and evolving networks; in particular, the SDN controller allows IoT devices to interoperate, especially in the presence of heterogeneous technologies, protocols, and standards. In [26], a set of extensions to the MQTT protocol that allow specifying real-time requirements for time-sensitive flows is proposed. The extensions were implemented and tested guaranteeing the compatibility with the standard and showing the improvement in latency due to time-sensitive traffic segregation and prioritization mechanisms especially for high loads. An architecture comprised of a master broker implemented on the SDN Controller to handle the overall system, and several slave brokers, each administrating a cluster of IoT devices, is presented in [27]. The centralized SDN controller uses a multicast scheme to distribute MQTT data across the external wireless network, thus reducing the transmission delay for massive IoT scenarios.

A multicast-based transmission scheme among publisher and subscriber is also proposed in [28], namely direct multicast-MQTT (DM-MQTT), together with the integration along an SDN architecture to minimize the transmission delays between multiple edge subnetworks. DM-MQTT relies on a hierarchical broker architecture by placing a slave broker on the very edge node, while a master broker gathers information to the SDN controller, which configures data paths among different edge networks to minimize data transfer delays.

Similarly, in [29], the SDN paradigm is applied to a wired network to modify the behavior of OpenFlow Switches in order to create direct paths between publishers and subscribers.

Besides, MQTT protocol is used for data exchanging among IoT devices by properly integrating within an SDN architecture, pointing out a remarkable performance improvement, as shown in [30].

However, we have to say that, to the best of our knowledge, the application of the SDN principle to MQTT protocol has been considered in the literature for massive IoT wired communications, while the SDN extension toward MQTT-SN, as analyzed in our paper for a typical wireless sensor network, has not been yet explored in depth. When compared with the existing contributions, our paper instead originally proposes an advanced SDN architecture for wireless WoIT scenarios, that optimizes the MQTT-SN protocol by avoiding the constraints

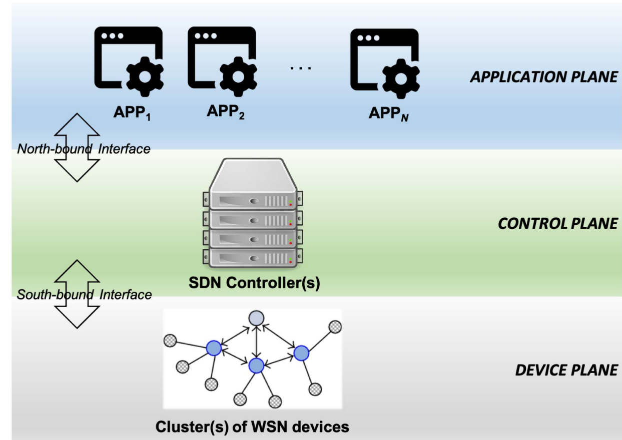


Fig. 1. Proposed architecture in terms of abstract layered model (left side) or deployment view (right side).

of the broker centered communications. For this purpose, we generalize the approach previously proposed in [19], toward an architecture based on the SDN paradigm, and also enhance the DM-MQTT approach, which is similar to our proposal, though originally conceived only for wired scenarios [28].

III. Proposal Characterization

A. Integrated Architecture

Usually, in typical WoIT scenarios, nodes are dynamically and autonomously grouped into clusters, according to the communication features and the data they collect. We considered the introduction of the SDN concept i) to manage the industrial system partition into nonoverlapping clusters and ii) to optimize the overall energy consumption and, consequently, maximizing the system availability and durability to support typical industrial applications. As shown in Fig. 1, the SDN oriented architecture layers have been designed toward these objectives and divided into: Application plane (AP), control p (CP), device plane (DP). Specifically, the devices belonging to a traditional WSN are grouped within the DP, where the processing and forwarding of the incoming packet are executed following the CP directives. The CP consists of a logically unique SDN Controller, which has a complete view of the underlying network and can directly manage the forwarding rules of the DP devices. Moreover, it can modify the networking behavior of nodes and to split the network in multiple clusters with a specific Cluster Head (CH). Finally, the AP is composed of a main network application which use the CP to reduce the network energy consumption and optimize the forwarding paths.

The communication protocol used by nodes to exchange data is based on the MQTT-SN standard, where the packets forwarding is handled by the SDN controller and optimized by the MQTT-SDN Adapter Application which is logically placed on the AP. The SDN planes and their components are described in the following.

1) SD-MQTT-SN Communication Protocol: MQTT-SN, unlike the basic MQTT protocol, has been adapted to the impairments of wireless environment (low bandwidth, high link failure, short message length, etc.) and to the constraints of typical devices in terms of low-cost, battery-powered, and limited computational and storage. Furthermore, this protocol is not necessarily based on TCP, but is completely agnostic of the underlying Transport Layer. As in the MQTT protocol, it is mainly characterized by three elements.

1) *Publisher* device which publishes messages under a certain topic.
2) *Subscriber* device which registers to a specific topic and, consequently, is willing to receive all the related messages.
3) *Broker* device which allows the communication between publishers and subscribers.

According to the MQTT-SN approach, every publish message is first sent by the publisher to the MQTT broker that, in turns, forwards them to the subscriber(s), as shown in Fig. 2(a) for a generic case. Thus, even if publisher(s) and subscriber(s) were close in terms of hops, the e2e path requested to a publish message for being delivered to the subscriber(s) could be nonoptimized, as shown in Fig. 2(b). Differently, according to the proposed software defined MQTT-SN (SD-MQTT-SN) approach, the decision to forward or not a publish message is made by the MQTT-SDN Adapter following the SDN controller indications. Specifically, a message is accordingly forwarded only if the cost associated with the path between the publisher(s) and the subscriber(s) is lower than the cost of the path connecting publisher(s), broker, and subscriber(s), as depicted in Fig. 2(c) that focuses on a message delivering to a couple of destinations with a 50% cost saving.

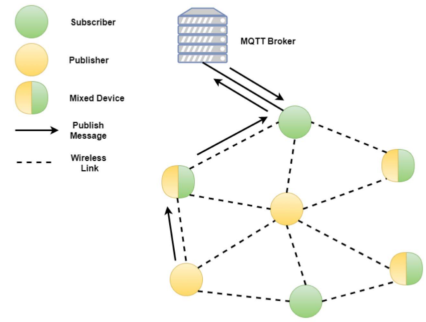
In more general terms, given a publisher P, a subscriber S, and a broker B within the considered network $\mathcal{N}$, the basic MQTT-SN approach finds the best path by solving the following constrained optimization problem:

$$\arg \min_{\pi(P,S)\in\mathcal{N}} \sum_{ij\in\pi(P,S)} w_{ij}$$
$$\text{s.t.} \quad B \in \pi(P,S) \tag{1}$$

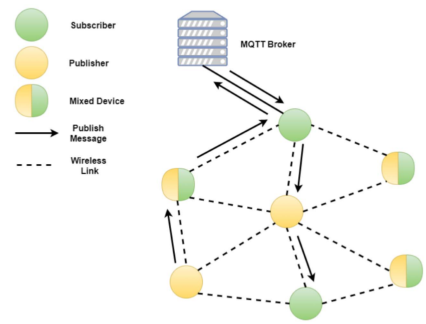
where $\pi(P,S)$ is a possible path connecting P and S, and w_{ij} the cost associated with the arc between the i and j nodes in that path. It is worth pointing out that in case of $w_{ij} = 1 \forall i,j$ the total cost to be minimized represents the path length, while in other case the cost takes into account typical IoT constraints as available battery level and communications bandwidth/delay. Conversely, in the SD-MQTT-SN, the SDN controller is able to identify a better optimum path by relaxing the previous constraint and thus focusing on the simplified problem:

$$\arg \min_{\pi(P,S)\in\mathcal{N}} \sum_{ij\in\pi(P,S)} w_{ij}. \tag{2}$$

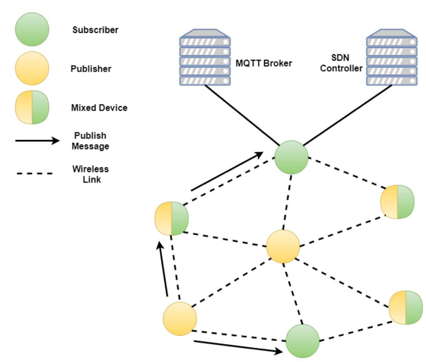
This allows the SD-MQTT-SN approach to achieve lowest cost paths and to reduce the energy consumption of the devices in the subpath connecting the broker and the subscriber, since they are not requested to store, process, and transmit unnecessary messages. In more general terms, we adapted a centralized and hierarchical Web-oriented application to an IoT domain by exploiting a more distributed communications pattern.



(a)



(b)



(c)

Fig. 2. e2e paths comparison for a publishing procedure according to MQTT-SN or SD-MQTT-SN. (a) Generic publish message forwarding to a single subscriber (MQTT-SN). (b) Unoptimized publish message forwarding to a copule of Subscribers (MQTT-SN). (c) Optimized publish message forwarding to a copule of subscribers as evaluated by the SDN controller (SD-MQTT-SN).

2) Device Plane: The DP represents the lower level of the SDN architecture and it contains the devices of a classic WSN. All these nodes are grouped into clusters and can be logically classified, according to their capabilities, in.

- *Gateway* (GW): it represents the only interface between the SDN Controller and the clusters. Moreover, it maps MQTT-SN into MQTT message allowing the use of a single Broker for both protocols.
- *Cluster Head* (CH): it communicates directly to the GW and it is capable of disseminating the Controller rules, while collecting data from the cluster nodes. For each cluster, there is only a CH and it can be chosen dynamically or statically.

TABLE I
NODE FORWARDING TABLE

MAC SOURCE ADDRESS	TOPIC ID	NEXT HOP MAC ADDRESS	TTL
C0:36:F0:22:50:6A	1	2B:1B:22:E7:E7:B0	100
0D:20:A5:10:C5:69	2	8F:37:F1:C0:10:18	100

- *Leaf Node* (LN): it only supports upstream connections, since it has no *child* nodes.
- *Relay Node* (RN): it can transmit, receive, and forward upstream and downstream packets to any destination node.

It can be pointed out that the setup of each cluster is delegated to its devices, without directly involving the SDN Controller. The upstream control messages are directed along the originated topology, usually arranged in a tree fashion to effectively support both multipoint-to-point and point-to-multipoint communications patterns, while downstream control and data messages, including MQTT publish messages, follow a path dynamically chosen by the SDN Controller. The forwarding table of every node is shown in Table I and has the following fields.

- `MAC SOURCE ADDRESS` to be matched with the source mac address of the received packet.
- `TOPIC ID` to be matched with the topic identifier contained within the publish message.
- `NEXT HOP MAC ADDRESS` specifies the next hop in the e2e path to which forward the packet if the matching fields are satisfied.
- `TTL` is the time to live parameter and it is decremented by one at each hop.

When a node receives a packet, it checks the related entries in its forwarding table. If the `MAC SOURCE ADDRESS` and the `TOPIC ID` of the Publish message are verified, then the node forwards that packet to the next node indicated in the `NEXT HOP MAC ADDRESS` field and updates the `TTL` value. If no entry is satisfied, then the node forwards the packet to the CH.

3) Control Plane: The CP consists of an SDN Controller, which has a complete view of the underlying network composed of DP and PP nodes. This centralized view allows the Controller to modify the data flows of the DP devices, selecting a e2e path in order to jointly minimize the energy consumption and the delivery latency. According to the proposed approach, the SDN Controller periodically receives control messages from the DP devices indicating i) the battery level, ii) the one-hop neighboring nodes, and iii) the distance from the CH. Using this information, the SDN Controller is able to build and keep updated the view of the network and, consequently, to optimize the forwarding of MQTT-SN messages.

4) Application Plane: In the AP, the MQTT-SDN Adapter, as depicted in Fig. 3, directly communicates with the MQTT Broker, which in turn can locally or remotely run in another Server or in a Cloud. MQTT-SDN Adapter maintains a local *topic table* on which it records the Publishers list, their topics and the related Subscribers list, as shown in Table II. Moreover, it resorts to the SDN Controller to evaluate the existence of a multihop path connecting Publisher and Subscribers and its cost in terms of number of hops (or of generalized cost), according to (1) and (2). This information is then used by the MQTT-SDN Adapter to perform the decision previously described in Section III-A1. It is worth pointing out that a Subscribers can receive a Publish messages from the Publisher in a *direct* way, i.e., not involving the Broker, is labeled as *Direct Mode* (DM) in the table. Whenever a direct communication is preferred, the SDN Controller changes the forwarding rules of the nodes belonging to the path. The entire procedure is shown in Fig. 4.

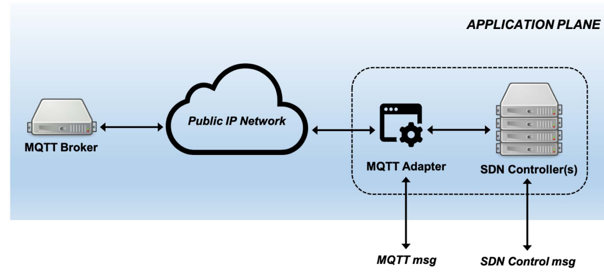


Fig. 3. Application plane architecture.

TABLE II
PUBLISHER AND SUBSCRIBER TABLE OF MQTT-SDN ADAPTER APPLICATION

Publisher	Topic Name	Subscribers	Direct Mode
P_1	Topic_1	S_1	No
		S_2	No
		S_3	Yes
		S_n	No
P_2	Topic_2	S_1	Yes
		S_3	No
		S_5	Yes
		S_m	No
P_N	Topic_N	S_i	Y/N
		$\cdots$	Y/N
		$\cdots$	Y/N
		S_M	Y/N

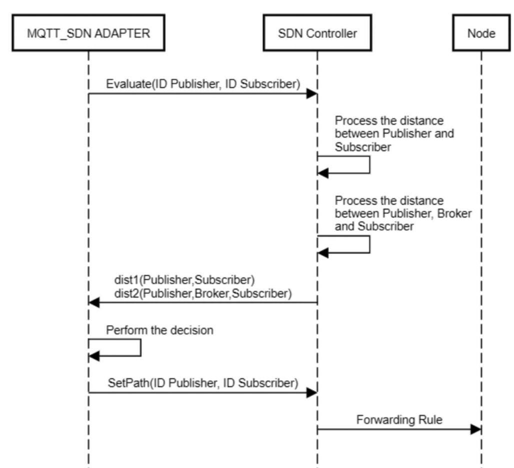


Fig. 4. Sequence diagram of MQTT-SDN application.

B. Communications Protocol Design

1) Data Flows Management and Forwarding: As previously described, the SDN controller can directly manage the forwarding rules of the DP devices in order to dynamically adapt the forwarding path for each data flow. The messages exchanged

TABLE III
NETWORK AND REPORT CONTROL PACKETS FIELDS

Packet ID	Distance to CH	Battery Level	CH ID
Network	[0 ÷ 255]	[0 ÷ 100]	CH MAC ADR
Packet ID	**Distance to CH**	**Battery Level**	**Neigbours List**
Report	[0 ÷ 255]	[0 ÷ 100]	[MAC ADR 1, ..., MAC ADR N]

TABLE IV
PATH AND CONFIG CONTROL PACKET FIELDS

Packet ID	FLOW RULE 1	...	FLOW RULE N
Path	Flow Rule Node 1	...	Flow Rule Node 1
Packet ID	**Command**	**...**	**...**
Config	Specific Command	...	...

inside the network can be divided in: i) SDN-WSN messages, and ii) MQTT-SN messages. The SDN-WSN messages are mainly used to modify the nodes flow entries according to the optimized e2e paths, and can be further categorized into:

1) *Network Control Packet:* This message is periodically sent by the CH over the broadcast wireless channel and it is mainly used by the nodes of the WSN to find the best path to the CH and to detect their neighbors. This message contains the distance in terms of hops to the CH, which is initially set to 0, its identity and the battery level of the sending node. Every time a node receives this packet it checks if the current distance to the CH is higher than the value contained in the message. In this case, the node updates the value reported in the network control packet to the current value plus one and sets its next hop toward the CH equal to the node who sent the message. Finally, it creates a new message with its battery level and its current distance to the CH and broadcasts it.
2) *Report Control Packet:* This message is periodically sent to the CH and contains the battery level of the sending node, the distance in terms of hops to the CH and the list of the neighbor nodes. CH then forwards this message to the SDN controller which uses it to build or update its network view.
3) *Path Packet:* message sent by the SDN controller to create a multihop path between publisher(s) and subscriber(s) avoiding the broker retransmission. Specifically, it is sent to the publisher and it contains all the flow entries of all the nodes along the path. Upon its reception, the publisher stores its flow entry and forwards the message to the next node on the path; the procedure is repeated until the subscriber node is reached. Each flow entry of this message contains the fields reported in Table I.
4) *Config Packet:* message sent by the SDN controller to add, modify, or delete a specific flow entry of a node. Moreover, it can be used to modify the node behavior (e.g., sleep period and sending period of report packet).

All the above mentioned packets are summarized in Tables III and IV.

Finally, the MQTT-SN messages are compliant to the standard. Since our approach is focused on the optimization of the energy consumption and on the data flows forwarding, we further specify the MQTT-SN messages as summarized in Table V. Specifically, we do not considered the *Will* messages, which are mainly used to alert the subscriber nodes whenever a topic is no more available, or the Publisher node is disconnected.

TABLE V
MQTT-SN MESSAGES

MsgType Field Value	MsgType	MsgType Field Value	MsgType
0x00	ADVERTISE	0x01	SEARCHGW
0x02	GWINFO	0x04	CONNECT
0x05	CONNACK	0x0A	REGISTER
0x0B	REGACK	0x0C	PUBLISH
0x0D	PUBACK	0x0E	PUBCOMP
0x0F	PUBREC	0x10	PUBREL
0x12	SUBSCRIBE	0x13	SUBACK
0x14	UNSUBSCRIBE	0x15	UNSUBACK
0x16	PINGREQ	0x17	PINGRESP
0x18	DISCONNECT		

TABLE VI
PARAMETERS ADOPTED TO MODEL A WIRELESS SENSOR NODE IN SIMULATION CAMPAIGN

	Min	Nom	Max	Unit
Supply voltage during radio operation	2.1		3.6	V
RF Frequency Range	2400		2483.5	MHz
Transmit bit	250		250	Kbps
Current consumption: Radio Trasmitting at 0 dBm		17.4		mA
Current consumption: Radio receiving		19.7		mA
Current consumption: Radio on, Oscillator on		365		μA
Current consumption: Idle mode, Oscillator off		20		μA
Current consumption: Sleep Mode			1	μA
Active current at Vcc = 3V, 1MHz		500	600	μA
Current consumption during sensing operation		15.4		mA
Voltage regulator current draw	13	20	29	μS

IV. PERFORMANCE ANALYSIS AND DISCUSSION

The performance of the proposed SDN oriented communications and energy management protocols has been evaluated by means of an accurate numerical simulation campaigns conducted on the open source Cooja framework, where the involved nodes can be either sensors, actuators, or hybrid devices. The simulated wireless sensor nodes have been modeled considering the data sheet of a commercial sensor node as indicated in Table VI. Specifically, we considered an initial battery level of 2500 mAh, corresponding to 2 AAA batteries, a keep alive consumption of 1 μA, a radio transmission consumption of 17 mA, and finally a reception consumption of 20 mA.

The network has been partitioned in logical clusters (equal to 8 in the considered test scenario), each managed by its own CH that specifically handles the traffic from or to the GW.

The overall system performance has been evaluated in terms of the following metrics and compared with existing protocols:

1) Residual battery level: this parameter measures the normalized energy available at each node and it is proportional to the overall network lifetime.
2) Overhead: it represents the cost of maintaining up to date the flow tables, and it is generally given by the ratio between control and overall messages (including publish packets). It can be expressed for the basic case of MQTT-SN ($\eta_{MQTT-SN}$) by considering *CONFIG* messages, while if SD-MQTT-SN is adopted ($\eta_{SD-MQTT-SN}$) also *PATH* messages need to be included.

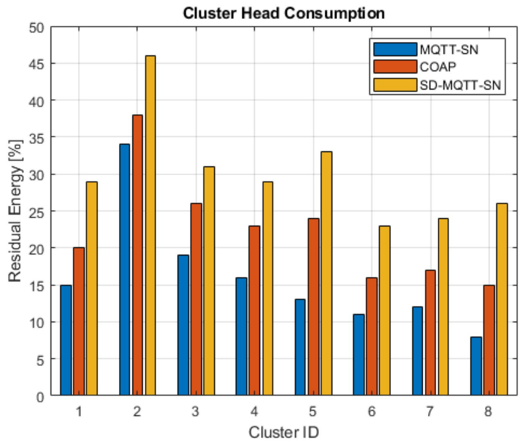


Fig. 5. CHs normalized residual battery level achieved by the MQTT-SN and SD-MQTT-SN protocols in a Random Mesh nodes deployment.

TABLE VII
OVERHEAD REQUIRED BY THE MQTT-SN AND SD-MQTT-SN PROTOCOLS IN A RANDOM MESH NODES DEPLOYMENT

Sim Run	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*Avg*
$\eta_{MQTT\text{-}SN}$	0.662	0.675	0.66	0.685	0.676	0.672	0.665	0.665	0.684	0.646	0.669
$\eta_{SD\text{-}MQTT\text{-}SN}$	0.698	0.724	0.707	0.729	0.709	0.679	0.702	0.763	0.762	0.681	0.709
η_{CoAP}	0.639	0.656	0.634	0.667	0.656	0.653	0.647	0.643	0.662	0.631	0.649

RED : max value, GREEN : min value, YELLOW : mean value

3) Equivalent Hop Number (EHN): it is intended here as the total number of transmissions needed to deliver a specific publish packet. Particularly, all the transmissions through the e2e publisher–broker and broker–subscriber paths for all the subscribers associated with a specific topic are considered. In case of direct communications, the transmissions between publisher and Subscriber not involving the broker are also taken into account.
4) Latency (δ): it measures e2e latency needed to deliver a message by considering all the transmissions delays along the path.

Moreover, the proposed architecture has been tested over three different nodes deployments: i) Random Mesh, (ii) Two-Tier, and (iii) Unbalanced Random Mesh. The overall number of devices has been assumed to be equal to 40, in compliance with most common WoIT scenarios. For the sake of comparison, we have considered the CoAP and the MQTT-SN performance as benchmark, because they represent the two most relevant WoT standards adopted in practice.

First, we evaluated the performance of the SD-MQTT-SN approach over a Random mesh topology, in which the number of hops for a specific e2e path depends on the nodes spatial distribution, while a fixed transmission range has been assumed for simplicity.

It can be preliminarily noticed that SD-MQTT-SN opportunistically resorts to the broker mediation. As a consequence, the battery consumption results more homogeneous because the majority of transmissions happen at the network edge, while traffic load is reduced on the network core, especially at the CHs side, with a consequent lifetime improvement, as shown in Fig. 5. From Table VII, it can be noticed that overhead required by the SD-MQTT-SN scheme is considerable with respect to the one of the basic MQTT and CoAP. Approximately 40% additional messages per Publish packet sent are, indeed, needed, i.e., roughly one control message out of two Publish packets. Despite this, in Table VIII is pointed out that the introduction of SD-MQTT-SN in the considered scenario allows a drastic reduction on the number of transmissions to be performed to deliver every publish packet to all the possible destinations, with performance gain of approximately 11%. On the average, it is possible to save two transmissions for every Publish Packet generated in the network, achieving a considerable reduction of the overall consumption, which completely compensate the additional control messages. Moreover, the SD-MQTT-SN scheme has a positive influence on latency, as shown in Table IX, where it can be pointed out an increasing e2e delay reduction due to the path optimization, i.e., a lower number of hops needed to reach the destination.

TABLE VIII
EHN ACHIEVED BY THE MQTT-SN AND SD-MQTT-SN PROTOCOLS IN A RANDOM MESH NODES DEPLOYMENT

Sim Run	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*Avg*
$EHN_{MQTT\text{-}SN}$	13.79	16.93	18.43	17.91	16.14	14.87	15.33	18.95	19.54	15.52	16.741
$EHN_{SD\text{-}MQTT\text{-}SN}$	13.41	15.26	15.70	15.96	13.96	13.17	13.10	16.52	17.57	13.79	14.844
η_{CoAP}	13.63	15.84	16.38	16.56	14.63	13.79	13.69	17.05	18.17	14.41	15.42

RED : max value, GREEN : min value, YELLOW : mean value

TABLE IX
LATENCY ACHIEVED BY THE MQTT-SN AND SD-MQTT-SN PROTOCOLS IN A RANDOM MESH NODES DEPLOYMENT

Sim Run	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*Avg*
$\delta_{MQTT\text{-}SN}$ [ms]	0.130	0.179	0.237	0.336	0.197	0.408	0.396	0.441	0.394	0.465	0.318
$\delta_{SD\text{-}MQTT\text{-}SN}$ [ms]	0.112	0.138	0.156	0.266	0.188	0.358	0.307	0.372	0.327	0.418	0.264
η_{CoAP}	0.121	0.152	0.192	0.302	0.193	0.386	0.356	0.398	0.362	447	0.291

RED : max value, GREEN : min value, YELLOW : mean value

An alternative scenario that has been investigated consists in a two-tier star-based nodes deployment, in which the GW is located at the barycenter of the network surrounded by a first tier of CHs, where all the other nodes are one-hop distant from them. Simulations performed on this scenario have highlighted that the amount of configuration messages necessary to keep the flow tables updated is considerably reduced compared to the random mesh topology, since those messages are forwarded only by one intermediate node, i.e., the CH. When applied, the SD-MQTT-SN can optimize intercluster or, less frequently, intracluster paths. Furthermore, even though SD-MQTT-SN approach reduces CHs energy consumption and, consequently, increases network lifetime, these improvements are more limited than those ones achievable in the random mesh configuration, as depicted in Fig. 6. It is interesting to note that, despite the considerable reduction of control messages, the lifetime gain is reduced too: this depends on the node spatial distribution, as nodes that represent bottlenecks, i.e., CHs, are more stressed. As a matter of fact, the traffic from/to a single cluster is entirely managed by its CH, without the possibility of redistributing the workload according to the nodes battery level. Moreover, the control messages overhead is illustrated in Table X. As already mentioned, since any node is at most two hops far from the Broker, there is a limited number of messages necessary for the setup of direct routes between two nodes; as a consequence, the overhead is less than the one needed to the random mesh deployment, but still greater then CoAP protocol. Besides, the energetic consumptions are clearly unbalanced, with consequent performance degradation. The equivalent hop

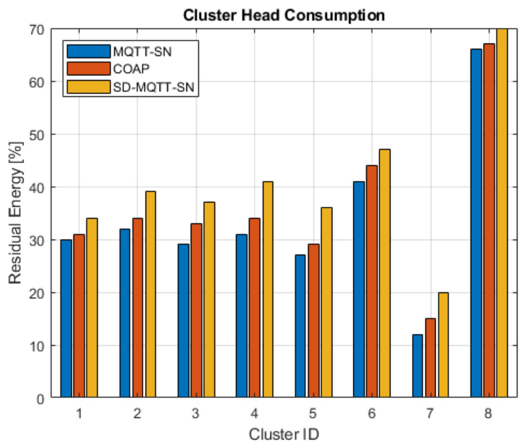


Fig. 6. CHs normalized residual battery level achieved by the MQTT-SN and SD-MQTT-SN protocols in a two-tier nodes deployment.

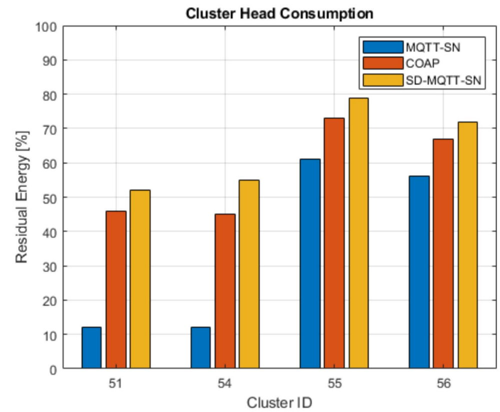


Fig. 7. Normalized residual battery level of the one-hop from GW nodes achieved by the MQTT-SN and SD-MQTT-SN protocols in an unbalanced random mesh topology nodes deployment.

TABLE X
OVERHEAD REQUIRED BY THE MQTT-SN AND SD-MQTT-SN PROTOCOLS IN A TWO-TIER NODES DEPLOYMENT

Sim Run	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*Avg*
$\eta_{MQTT\text{-}SN}$	0.27	0.303	0.248	0.3	0.306	0.274	0.21	0.234	0.297	0.213	0.265
$\eta_{SD\text{-}MQTT\text{-}SN}$	0.33	0.359	0.335	0.362	0.379	0.349	0.327	0.336	0.352	0.317	0.345
η_{CoAP}	0.22	0.294	0.239	0.296	0.304	0.268	0.17	0.229	0.292	0.210	0.252

RED : max value, GREEN : min value, YELLOW : mean value

TABLE XI
EHN ACHIEVED BY THE MQTT-SN AND SD-MQTT-SN PROTOCOLS IN A TWO-TIER NODES DEPLOYMENT

Sim Run	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*Avg*
$EHN_{MQTT\text{-}SN}$	11.55	13.16	13.19	14.67	12.11	11.52	11.98	14.24	14.26	12.07	12.875
$EHN_{SD\text{-}MQTT\text{-}SN}$	10.83	12.67	12.55	13.93	11.54	10.99	11.17	13.45	13.61	11.54	12.453
η_{CoAP}	11.21	12.97	12.91	14.28	12.04	12.49	11.53	13.89	13.98	11.91	12.721

RED : max value, GREEN : min value, YELLOW : mean value

TABLE XII
LATENCY ACHIEVED BY THE MQTT-SN AND SD-MQTT-SN PROTOCOLS IN A TWO-TIER NODES DEPLOYMENT

Sim Run	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*Avg*
$\delta_{MQTT\text{-}SN}$ [ms]	0.13	0.14	0.185	0.265	0.175	0.275	0.27	0.315	0.27	0.32	0.234
$\delta_{SD\text{-}MQTT\text{-}SN}$ [ms]	0.13	0.14	0.185	0.251	0.175	0.257	0.246	0.315	0.27	0.296	0.226
η_{CoAP}	0.13	0.14	0.185	0.260	0.175	0.268	0.253	0.315	0.27	0.316	0.231

RED : max value, GREEN : min value, YELLOW : mean value

TABLE XIII
PERFORMANCE IMPROVEMENTS DUE TO THE INTRODUCTION OF SD-MQTT-SN APPROACH IN AN UNBALANCED RANDOM MESH NODES DEPLOYMENT

	CoAP	*MQTT-SN*	*SD-MQTT-SN*
η [%]	0.845	0.851	0.856
EHN [hop]	20.7	27.5	17.1
δ [ms]	0.309	0.466	0.257

RED : max value, GREEN : min value

number is considerably reduced, and this depends on the fact that there are at most two hops between a publisher and the broker, and the same distance between the broker and subsequent subscriber. As shown in Table XI, the introduction of direct communications does not bring substantial improvements, due to limited number of direct messages exchanged. Finally, from the results shown in Table XII, it can be seen that latency is significantly lower than the one occurring in the random mesh configuration, even though the SD-MQTT-SN approach does not lead to any remarkable improvement because of the limited number of direct transmissions.

For the sake of completeness, the proposed solution have been tested over an unbalanced random mesh nodes deployment, where the GW is positioned at the edge of the playground. It is worth noticing that this represents a worst-case scenario, where performances obviously degrades, but it is interesting to evaluate the effects of SD-MQTT-SN. The impact of an extremely unbalanced topology can be noticed. First of all, there are fewer nodes one hop away from the GW than in the balanced Random Mesh deployment previously introduced, but they still manage a similar amount of traffic. Furthermore, there are less chances to optimize the paths in the proximity of the GW and, therefore, lifetime is reduced. On the other hand, in the zone opposite to the GW, it is more likely to optimize the routes and this implies an overhead increasing need for the flow tables configuration, with a consequent increase of the EHN, due to the greater (average) distance between nodes and the GW, and also of the e2e latency. Despite this, it is observed that SD-MQTT-SN is beneficial for energy consumption, as highlighted in Fig. 7. Besides, the performance in terms of overhead, EHN and e2e latency are represented in Table XIII, where it can be clearly pointed out the improvements provided by the SD-MQTT-SN protocol for what concerns latency and the EHN. This is achieved with a comparable overhead, while in the previously investigated deployments, the SD-MQTT-SN scheme requires to an overhead of about 6% (Random Mesh) or even 30% (Two-Tier). In Fig. 8, the normalized average residual battery level for CoAP, MQTT-SN, and SD-MQTT-SN schemes is presented, pointing out a progressive improvement over time, which is directly correlated with the overall consumption and the network lifetime.

To conclude the performance analysis, additional metrics could be qualitatively addressed based on the previous results discussion. Focusing on SD-MQTT-SN protocol scalability, it can be noticed that in the presence of both high payload size and message transmission rate, the proposed approach outperforms the other alternatives in managing the traffic volume especially at

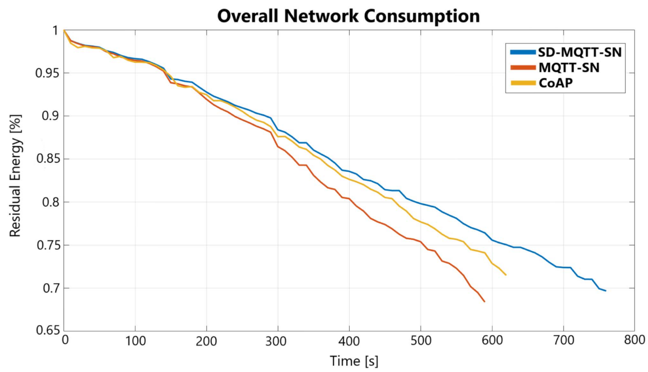


Fig. 8. Normalized average residual battery level achieved by the MQTT-SN and SD-MQTT-SN protocols in an unbalanced random mesh topology nodes deployment.

the increasing of the number of nodes. Another relevant property is represented by the system reliability, which is strictly dependent on the availability and reachability of the SDN controller. To address this issue, it could be adopted a redundant design with hot (or cold) standby SDN controller to ensure continuity in the event of an outage or even a failure. However, it is important to note that, according to the proposed SD-MQTT-SN protocol, the SDN Controller configures specific paths only when direct (east-west) paths need to be activated. Consequently, in case of SDN controller temporary or permanent fault, the communications scheme switches to the classical MQTT approach.

V. Conclusion

In this article, a comprehensive architecture has been proposed targeted to IoT ecosystem for industrial applications. We adopted a Web-oriented communications paradigm to enable complex interaction among sensors and actuators in the presence of technological heterogeneity. In order to provide service availability and reliability, despite the limited energetic budget, a general framework inspired by the SDN principle have been proposed. It has been accurately presented in terms of component and communications/control protocols with a specific focus on the SDN controller design by characterizing its specific functionalities to handle data flows and to manage power budget distribution. This allowed the extension of the basic MQTT-SN protocol toward the SD-MQTT-SN approach, that is able to jointly optimize the QoS requirements and the network lifetime. The integrated solution have been validated in practical scenarios, always pointing out remarkable performance w.r.t. the e2e latency and the overall power consumption, with a limited overhead, and, consequently, an improved network lifetime.

References

[1] International Telecommunication Union, "The Internet of Things. Executive summary," ITU, Geneva, ITU Internet Rep., 2005.

[2] J. M. Lee, P. Park, N. Kong, and N. Crespi, "The Internet of Things: Concept and problem statement," Internet Res. Task Force, Internet Draft, 2012.

[3] M. Weyrich and C. Ebert, "Reference architectures for the Internet of Things," *IEEE Softw.*, vol. 33, no. 1, pp. 112–116, Jan./Feb. 2016.

[4] P. Eugster, V. Sundaram, and X. Zhang, "Debugging the Internet of Things: The case of wireless sensor networks," *IEEE Softw.*, vol. 32, no. 1, pp. 38–49, Jan./Feb. 2015.

[5] K. Xu, Y. Qu, and K. Yang, "A tutorial on the Internet of Things: From a heterogeneous network integration perspective," *IEEE Netw.*, vol. 30, no. 2, pp. 102–108, Mar./Apr. 2016.

[6] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, 2016.

[7] S. M. A. Oteafy and H. S. Hassanein, "IoT in the fog: A roadmap for data-centric IoT development," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 157–163, Mar. 2018.

[8] D. Raggett, "The Web of Things: Challenges and opportunities," *Computer*, vol. 48, no. 5, pp. 26–32, 2015.

[9] M. Damm, S.-H. Leitner, and W. Mahnke, *Opc Unified Architecture*, Berlin, Germany: Springer, 2009.

[10] S. Cavalieri, "A proposal to improve interoperability in the Industry 4.0 based on the open platform communications unified architecture standard," *Computers*, vol. 10, no. 6, 2021, Art. no. 70.

[11] S. Grüner, J. Pfrommer, and F. Palm, "RESTful industrial communication with OPC UA," *IEEE Trans. Ind. Informat.*, vol. 12, no. 5, pp. 1832–1841, Oct. 2016.

[12] C. Bormann, A. P. Castellani, and Z. Shelby, "CoAP: An application protocol for billions of tiny internet nodes," *IEEE Internet Comput.*, vol. 16, no. 2, pp. 62–67, Mar./Apr. 2012.

[13] C. Gomez, A. Arcia-Moret, and J. Crowcroft, "TCP in the Internet of Things: From ostracism to prominence," *IEEE Internet Comput.*, vol. 22, no. 1, pp. 29–41, Jan./Feb. 2018.

[14] S. Quincozes, T. Emilio, and J. Kazienko, "MQTT protocol: Fundamentals, tools and future directions," *IEEE Latin America Trans.*, vol. 17, no. 09, pp. 1439–1448, Sep. 2019.

[15] C.-L. I, S. Han, Z. Xu, S. Wang, Q. Sun, and Y. Chen, "New paradigm of 5G wireless internet," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 474–482, Mar. 2016.

[16] K. Sood, S. Yu, and Y. Xiang, "Software-defined wireless networking opportunities and challenges for Internet-of-Things: A review," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 453–463, 2016.

[17] C. J. Bernardos et al., "An architecture for software defined wireless networking," *IEEE Wireless Commun.*, vol. 21, no. 3, pp. 52–61, Jun. 2014.

[18] J. H. Cox et al., "Advancing software-defined networks: A survey," *IEEE Access*, vol. 5, pp. 25487–25526, 2017.

[19] C. Bartoli et al., "Toward the web of industrial things: A publish-subscribe oriented architecture for data and power management," *Sensors*, vol. 22, 2022, Art. no. 4882.

[20] N. Bizanis and F. A. Kuipers, "SDN and virtualization solutions for the Internet of Things: A survey," *IEEE Access*, vol. 4, pp. 5591–5606, 2016.

[21] S. Bera, S. Misra, and A. V. Vasilakos, "Software-defined networking for Internet of Things: A survey," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 1994–2008, Dec. 2017.

[22] S. Jazaeri et al., "Edge computing in SDN-IoT networks: A systematic review of issues, challenges and solutions," *Cluster Comput*, vol. 24, pp. 3187–3228, 2021.

[23] A. Banks, E. Briggs, K. Borgendale, and R. Gupta, "MQTT version 5.0," *OASIS Standard*, 2019. [Online]. Available: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html

[24] F. Palmese, E. Longo, A. E. C. Redondi, and M. Cesana, "CoAP vs. MQTT-SN: Comparison and performance evaluation in publish-subscribe environments," in *Proc. IEEE 7th World Forum Internet Things*, 2021, pp. 153–158.

[25] M. Nast, F. Golatowski, and D. Timmermann, "Design and performance evaluation of a standalone MQTT for sensor networks (MQTT-SN) broker," in *Proc. IEEE 19th Int. Conf. Factory Commun. Syst.*, 2023, pp. 1–8.

[26] E. Shahri, P. Pedreiras, and L. Almeida, "Extending MQTT with real-time communication services based on SDN," *Sensors*, vol. 22, no. 9, 2022, Art. no. 3162.

[27] R. Tamria and S. Rakrak, "The MI-SDN system to manage MQTT data in an interoperable IoT wireless network," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 5, pp. 1031–1036, 2021.

[28] J.-H. Park, H.-S. Kim, and W.-T. Kim, "DM-MQTT: An efficient MQTT based on SDN multicast for massive IoT communications," *Sensors*, vol. 18, no. 9, 2018, Art. no. 3071.

[29] J. H. Park, S. Lee, S. Yun, H. Kim, and W.-T. Kim, "Dependable fire detection system with multifunctional artificial intelligence framework," *Sensors*, vol. 19, no. 9, 2019, Art. no. 2025.

[30] M. Raikar, M. S. M., and M. Mulla, "Software defined Internet of Things using lightweight protocol," *Procedia Comput. Sci.*, vol. 171, pp. 1409–1418, 2020.

Claudio Bartoli received the B.S. degree in electronics and telecommunications engineering and the M.S. degree in electronic systems engineering from University of Florence, Florence, Italy, in 2021 and 2024, respectively.

During his studies, he received a one-year research grant from the Information Engineering Department of the University of Florence aimed at designing a novel Industrial IoT architecture based on 5G technologies within the scope of the RI.B.AT. project funded by Regione Toscana. During his internship at Leonardo Electronics Division, Montevarchi, Italy, he explored the topic of the arc fault detection in avionics systems applying the wavelet transform in order to find anomalies related to such events. He currently works as an Electronic Engineer in the Electronic Equipment and Mixed Technology Group, Leonardo Electronics Division, Montevarchi, Italy.

Michele Bonanni received the bachelor's degree in telecommunications engineering from the University of Pisa, Pisa, Italy, in 2015, and the master's degree in telecommunications engineering and the Ph.D. degree in information engineering from the University of Florence, Florence, Italy, in 2018 and 2023, respectively.

In 2019, he was awarded with a one-year research grant for the Industry 4.0 Project HYDROCONTROLLER, where he contributed to the implementation of an advanced sensor network integrated with SDN. During his Ph.D., he analyzed the benefits and challenges of integrating SDN with sensor networks, worked on optimizing their energy efficiency, and investigated the combined use of SDN and NFV for live migration optimization. His work resulted in several publications in national and international journals. He is currently the Director of the R&D Department at SS4I S.R.L. and actively collaborates with the University of Florence.

Francesco Chiti (Senior Member, IEEE) received the M.Sc. degree in telecommunications engineering and the Ph.D. degree in information engineering from the University of Florence, Florence, Italy, in 2000 and 2004, respectively.

He is an Associate Professor with the Department of Information Engineering, University of Florence. His research is presently devoted to the synergies between Internet of Things and Software Defined Networking/Network Function Virtualization paradigms with application to 6G systems, 5G Vehicular Networks, Industry4.0 and Smart Cities. In addition, his research is devoted to the foundation of Quantum Networking for nonterrestrial scenarios. He participated in national initiatives (PRIN SIPT-VARIS, PRIN PATTERNS, PRIN WORLD) and EU (NextGenerationEU NRRP "RESTART," COST 22168, COST 289, IP "GoodFood," NoE "NEWCOM," CRUISE NoE, STREP "DustBot," REGPOT AgroSense, EDA "MEDUSA"). He was with the IEEE Communications and Information Security Technical Committee as Secretary/ Vice Chair/ Chair (2013-2021). He has been involved as ÒCommunication & Information System SecurityÓ Symposium Chair for IEEE GlobecomÕ21 and IEEE GlobecomÕ16.

Laura Pierucci (Senior Member, IEEE) received the M.Sc. degree in electronics engineering from the University of Florence, Florence, Italy, in 1986.

She is an Associate Professor with the Department of Information Engineering, University of Florence, Florence, Italy. She co-invented a patent on the use of RFID for health application. She has been involved in several national and European research projects on satellite communications, telemedicine systems, wireless systems, and radar signal processing. She has been the Scientific Coordinator of the EU COST Action 252. She was an Expert for European Committee in the area of satellite communications. She has coauthored of several international papers. Her current research interests include wireless communication systems especially in the topics of 5G/6G systems, resource management, multiple-input multiple-output antenna systems, security for wireless sensor networks and the IoT, software defined networking and Quantum satellite networking.

Dr. Pierucci was a recipient of the IWCMC 2016 Best Paper Award and the Globecom 2016 Best Paper Award. She is an Associate Editor IEEE COMMUNICATIONS MAGAZINE, and IEEE Internet of Things Journal.