IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 27 July 2022, accepted 30 September 2022, date of publication 6 October 2022, date of current version 13 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3212728

APPLIED RESEARCH

Scalability Enhancement on Software Defined Industrial Wireless Sensor Networks Over TSCH

FEDERICO OROZCO-SANTOS[1], VÍCTOR SEMPERE-PAYÁ[1,2], (Member, IEEE), JAVIER SILVESTRE-BLANES[1,3], (Member, IEEE), AND JOSE VERA-PÉREZ[1]

[1]Instituto Tecnológico de Informática (ITI), 46022 Valencia, Spain
[2]Departamento de Comunicaciones (DCOM), Universitat Politècnica de València (UPV), 46022 Valencia, Spain
[3]Departamento de Informática de Sistemas y Computadores (DISCA), Universitat Politècnica de València (UPV), 03801 Alcoy, Spain

Corresponding author: Federico Orozco-Santos (forozco@iti.es)

This work was supported in part by DAIS (https://dais-project.eu/) through the ECSEL Joint Undertaking (JU) under Grant 101007273; in part by JU through the European Union's Horizon 2020 Research and Innovation Programme and Sweden, Spain, Portugal, Belgium, Germany, Slovenia, Czech Republic, The Netherlands, Denmark, Norway, and Turkey; and in part by the Generalitat Valenciana through the Instituto Valenciano de Competitividad Empresarial—IVACE.

ABSTRACT Industry 4.0 digitization requirements have led to widespread adoption of industrial wireless sensor networks (IWSNs) in manufacturing environments. These require strict control of the traffic flowing over the network to ensure the quality of service. Software-defined networks (SDNs) can address this challenge, as their centralized operation allows routing and media access control (MAC) protocols to be optimized. In addition, in many cases a massive deployment of metering devices is necessary, therefore the network must be scalable, to allow an increase in the number and density of nodes in the network. This implies an increase in the control traffic required by SDNs, generating a significant reduction in the throughput available for data. This is because the length of the timeslot limits the throughput of the network in protocols that uses time division, such as the Time Slotted Channel Hopping (TSCH), conditioning the number of nodes that the network can have and its data sending rate. In this paper, we propose to exploit the integration and reconfiguration capabilities of SDN to extend the number of nodes in the network through the use of multi-radio sinks called virtual sinks, orchestrated from the SDN controller, and extending their use to multiple SD-IWSNs to take full advantage of the radio spectrum. To optimize bandwidth usage, the SDN controller will allocate network slices, allowing nodes to send data with a high frequency using dedicated channels. The results obtained show that the use of multi-radio sinks provides for increasing the maximum number of nodes that can be deployed while preserving the quality of service (QoS) requirements. A linear increase in nodes at least 0.7 times for each radio interface is obtained compared to an SD-IWSN with a traditional sink. In addition, the use of virtual sinks increases the packet frequency of the network by a factor equal to the number of radio interfaces. Moreover, the total number of nodes is improved through the coexistence of multiple SD-IWSNs, where up to 13.7 times more nodes can be scheduled using all the available spectrum orchestrated from the same SDN controller.

INDEX TERMS IWSN, scalability, QoS, SDN, TSCH.

I. INTRODUCTION

Wireless sensor networks have quickly become widely adopted in industrial environments due to their flexibility, rapid deployment and low cost. Because of this, they have become key enablers of Big Data where massive sources of information are required. However, with the increase of digitization elements, the complexity of the network also increases, generating challenges that are difficult to solve with traditional networks, such as mobility, quality of service, management and scalability [1].

Traditional IWSNs base a large part of their management and coordination processes on distributed protocols, storing information from other network nodes to obtain

The associate editor coordinating the review of this manuscript and approving it for publication was Razi Iqbal.

routing schedules, as in the case of Routing Protocol for Low power and lossy networks (RPL) [2] or Minimal Scheduling Function (MSF) [3] to negotiate between the nodes the minimum resources necessary for the exchange of information.

However, network growth and QoS requirements make the traditional architecture inefficient. This must be improved to address today's challenges. For this end, the scientific community is looking for disruptive alternatives to overcome current limitations. This is the case of the software-defined networking paradigm in IWSNs (SD-IWSN) [4], [5] [6], which will allow the IWSNs to go beyond the centralization of protocols independently, by completely extracting the control plane from the network equipment and centralizing all processes in a common device that has a global knowledge of the network. Complete centralization allows the execution of more demanding processes in devices external to the IWSN network itself, using a greater amount of resources and allowing a highly detailed IWSN administration, thus improving the quality of service and flexibility [7]. To achieve this global knowledge, control traffic is necessary, where each node must send information about its environment to the SDN controller.

Unlike wired networks, IWSNs have severe bandwidth limitations. In addition, due to their wireless nature and ad hoc behavior, there is no dedicated link between each node and the SDN controller. Therefore, control traffic propagates through other nodes in the network until it reaches the sink, which acts as a gateway from the IWSN to other types of networks and to the SDN controller. Therefore, the major drawback of SD-IWSNs is the flow of control traffic, which, combined with bandwidth limitations, results in an accumulation of control traffic at the nodes closest to the sink. As the number of nodes in the network increases, this effect is more noticeable and will have an impact on network performance, since the nodes, in addition to control traffic, must forward data traffic. These two types of traffic will be added through the different hops of the SD-IWSN until they saturate the system and cause packet loss. Since the sink is the only point that connects the SD-IWSN with the SDN controller, it will be the first node where this congestion occurs, limiting the total number of nodes in the network.

For this reason, this paper proposes and implements mechanisms that allow SD-IWSNs to be highly scalable, while maintaining the advantages of flexibility and quality of service guarantees offered by the combination of SDN and TSCH. The first mechanism is the implementation of sinks with multiple radio interfaces, which from now on will be referenced as virtual sinks. A virtual sink, taking advantages of the centralized SDN controller, allows multiple radio interfaces to behave as a single sink device, achieving a considerable increase in the bandwidth of the SD-IWSN network. The second mechanism is the deployment of multiple SD-IWSNs based on the virtual sink concept that has been integrated through the wired SDN network, and orchestrated from the SDN controller to take full advantage of the radio spectrum to improve the scalability of these networks. The main contributions of this paper are summarized below.

1) Evaluation and analysis of scalability problem in SD-IWSN and review state of the art.
2) Design, development and implementation of virtual sink concept and further integration with SDN TSCH framework.
3) Integration of multiple SD-IWSNs with virtual sinks, which are orchestrated from a single SDN controller.
4) Performance evaluation of virtual sink and multiple SD-IWSN, based on simulations and experimental evaluation on a testbed.

The rest of the paper is organized as follows. Section 2 briefly introduces the TSCH protocol, SDN and the basics of the Software-Defined IWSN. Moreover, there is a brief literature review. Section 3 presents the proposed methodology and the description of the scenarios used in the tests. Section 4 contains results and discussion. Conclusions and future work are shown in Section 5.

II. BACKGROUND AND LITERATURE REVIEW

This section discusses the fundamental aspects of IWSNs, the types of access to the medium and their evolution towards software-defined networks.

A. BACKGROUND

WSNs are composed of multiple nodes, specifically oriented to collect, process and send data, guaranteeing the levels of quality of service expected in industry. One of the de facto standards in this type of industry-oriented network is IEEE 802.15.4, in which transmissions are made wirelessly using up to a total of 16 channels in the 2.4GHz band, as specified in its physical layer. Initially, this standard proposed using the Carrier Sense Multiple Access (CSMA) mechanism as the Medium Access Control (MAC) protocol. However, in order to better adapt WSNs to the industrial sector, the standard was updated to IEEE 802.15.4e, incorporating three new protocols for industrial use: Low Latency Deterministic Network (LLDN), Deterministic and Synchronous Multi-channel Extension (DSME) and Time Slotted Channel Hopping (TSCH). Each of these mechanisms was defined to meet certain needs of different use cases, such as reducing power consumption, reducing latency through star networks, increasing the reliability of transmissions or achieving greater determinism in the network by eliminating much of the randomness presented by CSMA. According to [8], [9], thanks to its temporal scheduling of transmission windows, and its frequency hopping mechanism, TSCH achieves higher transfer rates while maintaining moderate power consumption, due to its low duty cycle, while obtaining high reliability by mitigating interference or signal fading characteristics in industrial scenarios. This makes TSCH the de facto medium access mechanism when deploying an IWSN that must meet high quality of service requirements in industrial environments.

1) TIME SLOTTED CHANNEL HOPPING

The TSCH protocol allows transmission and reception to be scheduled to avoid collisions and interference. For this purpose, it divides the time into a fixed number of timeslots that are grouped into Slotframes which are cyclic in time as shown in Fig. 1. The most common duration of each timeslot is 10 ms, which is enough time to transmit a frame of the maximum size (127 bytes), wait for the receiver's ACK and process the packet. The frequency division is done with the assignment of a channel offset, which is used to determine the physical channel, therefore a timeslot and a channel offset must be defined for each transmission. The combination of these two parameters is a cell or slot, which can be shared or dedicated.

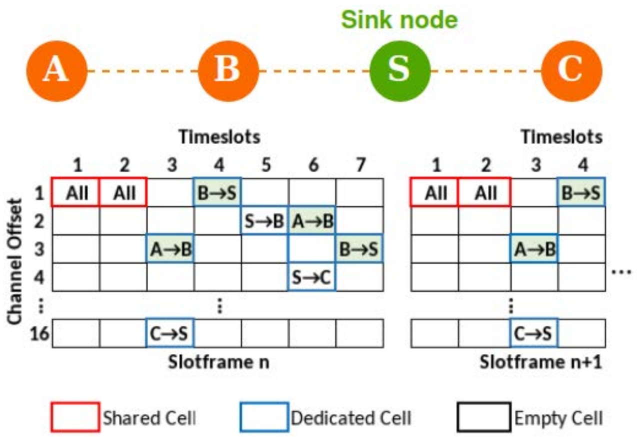

FIGURE 1. Time slotted channel hopping slotframe.

In the case of shared slots, the CSMA schedule is followed with a backoff exponent to reduce collisions. On the other hand, if it is dedicated, a unidirectional link is established between two nodes and a specific timeslot and channel offset are assigned in the TSCH scheduling, where only the source node can transmit. Multiple transmissions are possible in the same timeslot, as long as they are performed on different channel offsets and each node has a single scheduled action (Tx or Rx). Therefore, TSCH scheduling allows radio resources to be managed by establishing the actions that must be performed by the nodes in each timeslot. The node will be able to receive, transmit or turn off the radio in the case that no specific action is assigned.

The nodes within the schedule must be correctly synchronized so that the transmit and receive timeslots match. This is achieved with the Enhanced Beacon (EB) packet exchange that includes information about the Absolute Sequence Number (ASN) that the nodes take as a time reference and determines the current timeslot. In addition, the ASN is used together with the channel offset (Ch_{offset}) to perform a frequency hop according to Equation (1). Thus, the physical channel over which the transmission is performed has a rotation over the available channels (N_{ch}).

$$frequency = F[(ASN + Ch_{offset}) mod N_{ch}] \qquad (1)$$

2) SOFTWARE-DEFINED IWSN

The SDN paradigm is a network architecture characterized by separating the control and the data plane in the network devices. Furthermore, extract this control plane to place it in the SDN controller, a common entity for all network devices devices that has the ability to manage the control processes and decision making of all the network devices [7]. This allows decisions to be made based on a global knowledge of the network, resulting in optimized network processes and enabling high flexibility by having a fully programmable network that can dynamically adapt to changes at any point in the network [10].

The flexibility achieved by separating the data plane from the control plane overcomes the limitations of traditional networks [11], allowing network resources to be dynamically managed, configured and optimized by means of functions that interact with the centralized SDN controller. In addition to these control and data planes, a management plane or application layer is generated that groups together all the automated functions which modify the behavior of the network.

This management plane is located above the control plane, as shown in Fig. 2. This allows the controller's global knowledge to be reused for specific tasks. The objective of this is to simplify the development of applications, since it is not necessary for them to know in detail the operation and configuration of the network infrastructure. It will be the SDN controller that translates the requirements of the applications into configurations for each of the network elements. For this reason, the SDN controller handles a wide range of protocols, divided into Northbound (NB) and Southbound (SB). NB protocols establish communication between the management plane and the SDN controller. In contrast, SB protocols are used to communicate with the network infrastructure, the most widespread being OpenFlow developed by the Open Networking Foundation (ONF) [12].

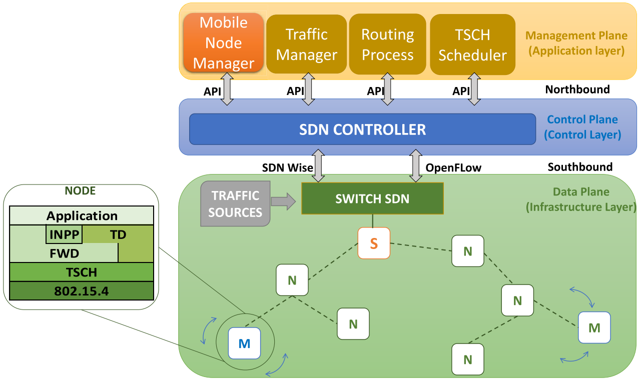

FIGURE 2. SDN WISE-TSCH general architecture [10].

Flexibility, dynamism and reconfigurability are advantages of SDN that make it possible to abstract network operation to a single point and solve problems in a simple and optimized way. For this reason, the operation of SDN has been extended, not only among the different types of wired networks, but also to other types of networks, such as IWSNs. They must be adapted to industrial environments, where tighter control over network flows is required to guarantee quality of service parameters. However, the application of the SDN approach in IWSNs is not straightforward, due to the large differences in the hardware resources of the equipment that make up the data plane. For this reason, there are different frameworks that address the integration of SDN and IWSN. One of them is SDN WISE [4], which contains a protocol stack that allows information to be exchanged with the SDN controller in an optimized way. As shown in Fig. 2, This stack is composed of the following layers: Application, In-Network Packet Processing (INPP), Topology Discovery (TD), Forwarding (FWD), the MAC level and the physical layer, which is 802.15.4. The MAC layer was replaced by TSCH in [5], where SDN WISE TSCH is proposed, which extends the functionalities of SDN WISE, to guarantee industrial level requirements with a highly deterministic system. The controller has a stack of applications that allows centralized control of all processes in the nodes (MAC, Routing, QoS, Application). The integration of these applications allows network slicing, which consists of the differentiation, isolation and logical segmentation of data flows, through the allocation of TSCH resources. This slicing makes it possible to guarantee QoS parameters for each data flow independently.

Also, to ensure the shortest possible delay, the TSCH scheduling algorithm consecutively allocates the slots for each of the hops, from the source node to the destination. It also guarantees the deadline independently of the slotframe, by means of repetitions of the flow within the same slotframe as shown in Fig. 1 where the flows C->S, S->B, S->C are performed every 70 ms, which is once per slotframe. On the other hand, the A->B->S flow is performed on average every 35 ms, because it is scheduled twice within the slotframe, in timeslots 3 and 7. The number of repetitions (*NR*) is calculated as a function of the slotframe length and the deadline.

B. LITERATURE REVIEW

Scalability in IWSNs is a widely studied topic, due to their wireless nature and autonomous operation, which allows them to adapt to changes in the physical topology. These networks have evolved to adapt to different use cases, to the point of being compatible with more critical environments, where QoS parameters such as Delay, Packet Delivery Ratio (PDR) and Deadline Satisfaction Ratio (DSR) need to be guaranteed. However, guaranteeing these QoS aspects while maintaining scalability remains a challenge in this type of network [1].

The most traditional sensor networks use the CSMA protocol as medium access protocol, performing a random access to the shared medium, which leads to the possibility of collisions and interference. In general, the scalability of this type of ad hoc network is addressed with solutions based on routing and clustering, as proposed by some works such as [13], [14], [15], [16], and [17].

Clusters are hierarchical divisions of the network by groups of nodes; in each group there is a main node that is in charge of receiving the information from the other nodes in the cluster, grouping and sending it to the sink [1]. This reduces the number of transmissions required to send the information, which translates into lower energy consumption. Some solutions, such as [14], use a distributed algorithm to choose the main node of the cluster. This node will be a critical element in the energy consumption due to the message retransmissions of its cluster. To mitigate this problem, the algorithm includes a rotation system for the main nodes. Other studies address the same problem of energy consumption of these main nodes of the cluster [17], in which the authors propose different algorithms to improve energy consumption in denser networks with longer links.

Another alternative to improve the scalability of mesh networks without having a negative impact on the quality of service, is to increase the number of sinks, including [18] in which networks with a maximum of 64 nodes are simulated. By increasing the number of sinks, the network throughput remains stable, without impairing the PDR in the network as a whole. Analyses on the coexistence of multiple asynchronous TSCH networks have also been performed [19], allowing the deployment of a more significant number of nodes but requiring centralized mechanisms to mitigate interference between TSCH networks.

However, it is challenging to guarantee QoS parameters since the probability of collisions and interference increases directly with the number of nodes. Therefore, for more critical environments, where transmissions and QoS parameters must be guaranteed, the CSMA protocol has been replaced by TDMA-based protocols such as TSCH. This protocol allows synchronization of all the nodes in the network and assignment of timeslots to each node for data transmission or reception [20]. Thus, collisions that occur in CSMA when multiple nodes transmit at similar time instants are avoided. In addition, TSCH uses multiple channels, which allows frequency hopping for each transmission, reducing interference and path fading [21]. Scalability in TSCH is an understudied topic that is not directly addressed in the current literature [22], probably due to its industrial approach in which the number of nodes has been more limited. However, current digitalization requirements mean the deployment of massive sources of information while maintaining a high degree of QoS.

The use of the TSCH protocol means that transmission scheduling must synchronize the transmission and reception states of the nodes to avoid interference. The IEEE 802.15.4e standard does not define any scheduling method, proposing only a series of recommendations for deploying a network with the minimum shared resources to establish an information exchange. Numerous studies have been carried out on different TSCH scheduling approaches, classifying these developments into centralized mechanisms [32], [33], distributed mechanisms [34] and autonomous mechanisms [35]. Centralized mechanisms maintain a global view of all network scheduling, resulting in more optimized configurations

TABLE 1. Summary of related works.

Work	Number of nodes	Simulation	CSMA	TSCH	SDN	Topics
EESRA: Energy Efficient Scalable Routing Algorithm for Wireless Sensor Networks [14]	Simulation: 100,200, 300, 400	Matlab	✓	✓	×	Cluster and routing algorithm
An Effective Relay Node Selection Technique for Energy Efficient WSN-Assisted IoT [15]	Simulation: 1000, 1500	Matlab	✓		×	Cluster and routing algorithm.
A Platform for Evaluating Clustering Strategies in Mobile IEEE 802.15.4-TSCH networks [16]	Simulation: Up to 15 nodes/cluster	Cooja		✓	×	Cluster and routing algorithm
Hybrid energy-efficient multi-path routing for wireless sensor networks [17]	Simulation: Up to 500 nodes	Matlab			×	Cluster and routing algorithm
Network Performance Enhancement of Multi-sink Enabled Low Power Lossy Networks in SDN Based Internet of Things [18]	Simulation: 16, 32, 48, 64	Cooja	✓		✓	Multi-sink placed in different locations
Coexistence analysis of multiple asynchronous IEEE 802.15.4 TSCH-based networks [19]	Simulation: Up to 60 networks	Cooja, Matlab		✓	×	Multiple TSCH networks
The Cost of Software-Defining Things: A Scalability Study of Software-Defined Sensor Networks [23]	Simulation: 289 Testbed: 10	Cooja	✓		✓	Controller positioning and multi-sink
Performance Analysis of Software-Defined Multihop Wireless Sensor Networks [24]	Simulation: 50, 100	Matlab	✓		✓	Energy consumption model for large scale SD-WSN
A Review of the Control Plane Scalability Approaches in Software Defined Networking [25]	NA	NA			✓	Review of multiple approaches to scale wired SDN.
An SDN approach to route massive data flows of sensor networks [26]	Simulation: 1000	VMWare VSphere	NA	NA	✓	Software-defined clustered sensor networks
SYNCOP: An evolutionary multi-objective placement of SDN controllers for optimizing cost and network performance in WSNs [27]	Simulation: 100,150, 200, 300	Java	NA	NA	✓	Scalability optimal controller placement
A distributed density optimized scheduling function for IEEE 802.15.4e TSCH networks [28]	Simulation: 10 : 1500	6TiSCH Simulator		✓	×	TSCH Schedulling for high density WSN, Multi-channel operation.
Decentralized broadcast-based scheduling for dense multi-hop TSCH networks [29]	Simulation: 1-1500 nodes	6TiSCH Simulator		✓	×	Scheduling algorithm
A distributed algorithm to schedule TSCH links under the SINR model [30]	Simulation: Up to 300 nodes	Sinalgo		✓	×	Scheduling algorithm
DIGEST: a decentralized divergecast scheduling algorithm for IEEE 802.15.4e TSCH in the internet of things [31]	Simulation: 30,50,70	Matlab		✓	×	TSCH Scheduling optimization
This work	Simulation: 320, 1152* Testbed: 18	Cooja		✓	✓	Multi-sink, Multi-radio and Multiple TSCH WSN

NA: Not Available * : Maximum allowed with QoS

but at the cost of increased control traffic. Distributed solutions avoid transmitting this control traffic to a central element, but still need to know the status of neighboring nodes. Finally, mechanisms such as [35] avoid sending this control traffic, performing scheduling autonomously, based on pseudo-random information from each node, which increases the likelihood of overlapping schedules if configurations that allocate guaranteed slots are not adopted.

Analyzing the scalability problem, the centralized mechanisms have an added complexity, and as a consequence of this, many studies have proposed new distributed mechanisms to mitigate the scalability problems of this type of network [29], [30]. However, as the number and density of deployed nodes increases, their performance also suffers, as they require more message exchanges to negotiate the resources [22]. This can be seen in [28] where a scalability-oriented TSCH scheduler is implemented, experimenting with networks of between 10 and 1500 nodes. This increase in the number of nodes results in an increase in consumption, due to the increased exchange of resource negotiation messages. The 1500 node density is achieved by employing star topologies, using all 16 channels simultaneously in all nodes of the network. This type of approach has been used in other articles [29], where, by assuming full duplex multi-radio communication at all nodes, reliability is increased in denser networks, although its implementation usually remains at the evaluation stage, given the high cost of hardware with such radio characteristics. If traditional 1-channel configurations are used, the maximum network size remains below 100 nodes.

In [31] the scalability of another distributed scheduling mechanism is analyzed, focusing on slotframe occupancy and neighbor connection rate. For a 30-node deployment, the occupancy does not exceed 85%, guaranteeing a connectivity rate of 100%. When increasing the network size to 70 nodes, these values increase to 92% and 98% respectively.

A new factor that allows the QoS and scalability challenges in IWSN to be addressed is the Software Defined Networking (SDN) paradigm. Here the IWSN nodes become programmable since all the decisions they must make are configured externally by the SDN controller. This is intended to achieve a high degree of flexibility and scalability [4], [24], [25]. An example of the flexibility of SDN is described in [5], where QoS is guaranteed in a TSCH IWSN through optimized TSCH resource allocation and network load balancing, however, scalability is not addressed. In [23] the scalability of SDN in an IWSN under the CSMA protocol is studied, in this case the network reaches 289 nodes and has a lower energy consumption than using RPL, however due to the use of the CSMA protocol the PDR decreases as the number of nodes increases.

In [26] SDN and a clustering algorithm are used to simulate a network of 1000 nodes, in this case using unrestricted nodes, to observe the effects on the SDN controller of this number of nodes in terms of CPU and memory. The position of the SDN controller and the sink nodes also has an effect on the QoS parameters, as analyzed by the authors of [27], in which the results have shown that it is an optimization problem, due to the congestion of the nodes near the sink. These approximations have also been used to modify the TSCH schedules [16], improving the agility with which these plans are modified, including cases in which mobile nodes are used, extending network capacity and therefore scalability by having more free radio resources available.

The current state of the art does not address in depth the combination of TSCH and SDN in terms of scalability.

This gap in current researchs is addressed in this paper, which, studies TSCH networks orchestrated by SDN-based solutions, paying special attention to scalability issues, using a centralized control of multi-radio virtual nodes. This enables the elimination of the bottleneck originated in the nodes close to the sink. In this way, the scalability and flexibility of these deterministic mesh networks is improved, while maintaining the QoS parameters observed in [5]. Table 1 contains a summary of related work and the most relevant aspects of each one.

III. METHODS

A. ANALYSIS OF CONTROL TRAFFIC

SDN networks rely heavily on the control information they receive from the nodes, as it is the nodes that collect information from their environment and send it to the SDN controller. This information is used by the SDN controller to build a global representation of the topology by combining information from multiple sources. In SDN WISE TSCH [5], this traffic is composed of periodic packets such as beacon and reports, and aperiodic packets which are the configuration packets (OpenPathTSCH). The beacon packets (B_i in Fig. 3) are used to discover the topology. They are transmitted in broadcast and serve to report the presence of a node in range. Therefore, this information is not retransmitted over multiple hops, but only reaches the nodes within its own coverage area, as shown in Fig. 3.

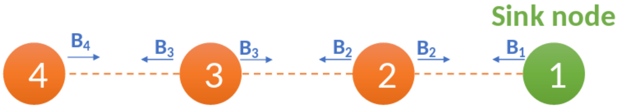

FIGURE 3. Propagación de los paquetes de Beacon (B).

Unlike the beacon packets, the report packets (R_i in Fig 4) must go from each node to the SDN controller, and include information collected by the beacons received from the rest of the neighbors in its coverage area and different statistics from the node generating the report. In most cases this packet must propagate through multiple hops until it reaches the controller, as shown in Fig. 4. Therefore, the total amount of control traffic depends mainly on the generation periods of these packets. Because SD-IWSNs have large bandwidth constraints, the total amount of control packets must be optimized, in order to minimize the impact on the bandwidth available for data traffic. Table 2, contains the notation used for the analysis of control traffic.

The throughput of a node in the TSCH protocol depends on the duration of the timeslot (T_s), as each node in the network can only perform a single action per timeslot: receive, transmit or sleep (the latter will be omitted from now on because it does not consume resources). Therefore, with a typical duration of 10 ms per T_s, a node is limited to performing a maximum of 100 actions per second ($1/T_s$), i.e. 100 receives, 100 transmits or a combination of both.

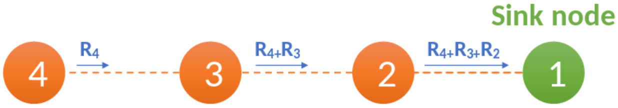

FIGURE 4. Propagación de los paquetes de reporte (R).

TABLE 2. Summary of notations used in this paper.

Notation	Definition	Units
n	Number of nodes excluding sink	nodes
C_{pk}	Control packets	packets/s
Th_D	Data Throughput	packets/s
Sh	Shared slots	timeslots
SF	Slotframe size	timeslots
H_i	Hops between sink and node i	hops
T_s	Timeslot duration	ms
T	Evaluation Period	s
B_p	Beacon Period	s
R_p	Report Period	s
P_f	Packet frequency	s
D_p	Data period ($1/P_f$)	s
n_{L1}	Number of nodes at one hop from sink	s
r	Number of radio interfaces	s
P_s	Packet size	Bytes
Ind	Indicators	Bytes
P_h	Packet header	Bytes
NR	Number of repetitions	times/slotframe

Equation (2), defines the total number of control packets (C_{pk}) in SDN WISE TSCH where n is the total number of nodes in the network not including the sink; B_p and R_p are the beacon and reporting periods; and H_i is the distance in hops from node i to the sink. The first term in this equation is the number of beacon packets generated by each node. Since they are not transmitted over multiple hops, this value only depends on the number of nodes and the generation period. In contrast, the report packets require multiple transmissions depending on the distance to the sink, consuming a greater amount of resources.

$$C_{pk} = n\frac{T}{B_p} + \sum_{i=1}^{n} \frac{T}{R_p} H_i \tag{2}$$

To isolate control and data traffic, nodes transmit control packets in shared slots while data packets are sent through dedicated slots. To ensure that the total control packets are transmitted correctly in the network, the number of shared slots must be allocated based on B_p, R_p and the number of nodes in the network (n). For the particular case of the sink, the number of control packets per second to be processed is obtained in Equation (3). The first term corresponds to the beacon packets it must receive, which depends on the nodes in the coverage area in this case n_{L1}, and the sending of its own beacon. In addition, it must receive the report packets from the n nodes in the network.

$$C^s_{pk} = \frac{n_{L1} + 1}{B_p} + \frac{n}{R_p} \tag{3}$$

To ensure that as many control packets as possible reach the sink, the number of shared slots must provide sufficient throughput for the total control packets obtained in Equation (3). Since nodes can receive a number of packets equal to the number of timeslots of the slotframe (SF), the duration of each timeslot defines the maximum flow of packets it can receive in an evaluation period (T), i.e., T/T_s. Therefore, according to [10], the number of shared slots can be obtained from Equation (4).

$$Sh = \left\lceil C^{s}_{pk} \cdot SF \cdot \frac{Ts}{T} \right\rceil \tag{4}$$

In this way, there are enough slots for all the beacon packets from the first level nodes (n_{L1}) to reach the sink, in addition to the report packets from the entire network. Although the amount of allocated resources is sufficient to transmit data to the sink without accumulating packets in the queues, there is a probability of collisions due to the operation of the shared slots.

Increasing the number of shared slots directly reduces the number of slots available for data. Therefore, the actual throughput available for data in the sink is calculated according to Equation (5).

$$Th_D(T=1) = \frac{SF - Sh}{SF \cdot T_s} = \frac{1}{T_s} - C^{s}_{pk} \tag{5}$$

B. SCALABILITY SD-IWSN

Fig. 5 shows an example of an SD-IWSN network, in which the sink is linked to the SDN controller via a wired network. The SD-IWSN uses 4 of the 16 channels available in TSCH. In each timeslot a different physical channel is used, due to the channel rotation performed by the TSCH mechanism using the ASN parameter, which is increased in each timeslot. In this case, the sink has only one radio interface regardless of the number of channels, so it can only perform one action (Rx or Tx) in each timeslot. Therefore, as it has two children, it will have to use two different timeslots for the reception of each one.

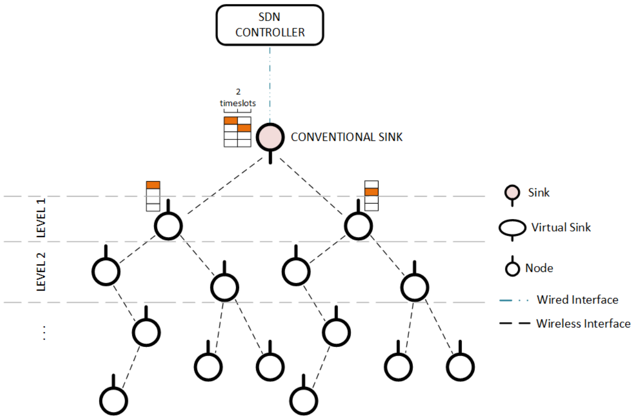

FIGURE 5. SD-IWSN traditional sink with a single radio interface.

By adding a new radio interface to the sink, as shown in Fig. 6, it is possible to receive simultaneously on the same timeslot using two different channels. In this case the sink will be composed of two conventional nodes that follow different schedules, but which the SDN controller orchestrates to behave as a single sink. This aggregation of nodes, orchestrated from the SDN controller as a single sink with multiple radio interfaces, is called virtual sink. To implement this type of node, a single device with multi-radio capability has not been used, but a logical combination of multiple simple devices, centrally orchestrated. This significantly reduces the complexity of implementing systems with simultaneous multi-radio capacity, allowing its deployment with conventional equipment.

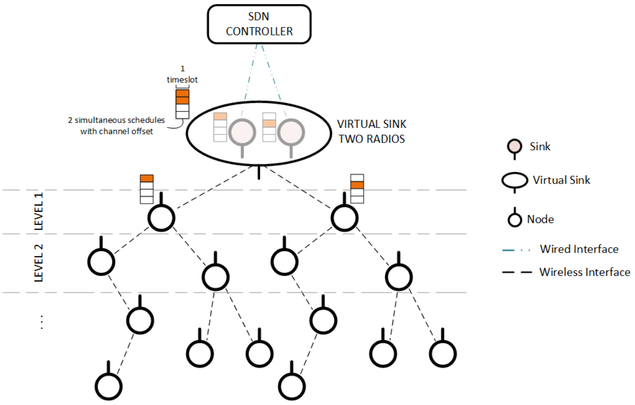

FIGURE 6. SD-IWSN virtual sink with 2 radio interfaces and 4 channels.

In the example shown in Fig. 6, two radio interfaces are used for a four-channel TSCH network. However, it is possible to create virtual sinks that add a node for each of the 16 channels available in TSCH, allowing the radio to be used simultaneously on all available channels.

This increase in radio interfaces in the sink has a direct impact on scalability, since the traditional sink is limited to performing a maximum of 100 TSCH actions per second ($1/T_s$), limiting the number of nodes the network can have and the throughput available for data. When r radio interfaces are added, the sink can perform r actions in each timeslot, which directly increases the sink capacity to r/T_s, thus reducing the bottleneck that occurs at the sink. However, since the other nodes in the network do not have multi-radio capacity, saturation will occur at nodes one hop away from the sink (n_{L1}). These nodes must receive data traffic from the other nodes in the network: $(n - n_{L1}) \cdot D_p$. In addition to receiving them, they must generate their own packets and send the total to the sink $n \cdot D_p$. On the other hand, receiving data packets from other nodes also consumes TSCH resources, reducing the maximum throughput for transmission. For these conventional nodes, the number of exchanges in reception and transmission must always be less than or equal to the throughput available for data (Th_D). Therefore, the maximum number of nodes that the network can have depends on the number of nodes in the first hop, the data packet frequency (P_f or $1/D_p$) and the available throughput, according to Equation (6).

$$\begin{aligned} (n - n_{L1}) \cdot D_p + n \cdot D_p &\leq Th_D \cdot n_{L1} \\ n &= \frac{n_{L1} \cdot R_p}{2 \cdot D_p \cdot R_p + n_{L1}} \left(\frac{1}{T_s} - \frac{n_{L1}}{B_p} + D_p \right) \end{aligned} \tag{6}$$

The above equation finds the maximum number of nodes that allow the network to use all the available throughput of each of the nodes of the first hop. According to this approach, the optimal number of radio interfaces (r_{opt}) that the virtual sink should have depends on the amount of data sent by the set of nodes in the SD-IWSN, according to Equation (7).

$$r_{opt} = \frac{n \cdot D_p}{Th_D} \tag{7}$$

Fig. 7, shows r_{opt} as a function of the packet frequency and number of nodes in the network. The number of nodes is obtained from Equation (6) using $B_p = 3$, $R_p = 6$ and $n_{L1} = 8$. The optimal number of radio interfaces increases directly as network throughput increases, i.e., when the packet frequency or the number of nodes increases. For each of these parameters, the maximum value of r_{opt} is different since it depends on the saturation of the nodes in n_{L1}. From this point on, increasing the number of radios does not represent an improvement in the network.

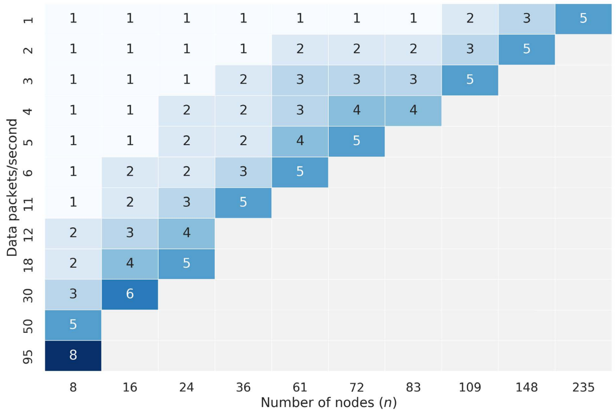

FIGURE 7. Number of radio interfaces depending on the number of nodes and the packet frequency with $n_{L1} = 8$.

This maximum value of the radio interfaces varies between n_{L1} and $n_{L1}/2$. It tends towards the lowest when the network throughput increases due to n, since the more nodes in the network, the more reception actions must be performed by the first level nodes. This reduces the amount of transmissions to the virtual sink. On the other hand, when the network throughput increases due to the packet frequency, the nodes in n_{L1} are less used in reception actions, and it is possible to perform a more significant number of transmissions, thus requiring a greater number of radio interfaces.

Fig. 7 also shows the limitations of SD-IWSN networks that use a single radio interface. The results shows how the network can grow up to a maximum of 83 nodes generating one data packet per second, while using a virtual sink with the same packet frequency, it is possible to have networks with up to 235 nodes. The limit of 235 nodes is due to the exhaustion of the throughput of the nodes at one hop from the sink, so in this case, increasing the number of radio interfaces in the virtual sink does not generate an increase in throughput. Moreover, if the number of nodes is maintained, it is possible to increase the frequency of data packet generation by a factor equal to the number of radio interfaces. For example, for 83 nodes with a single radio it is only possible to send one packet per second on each node, by increasing the number of radio interfaces to 4, it is possible to increase the packet frequency to 4 packets per second.

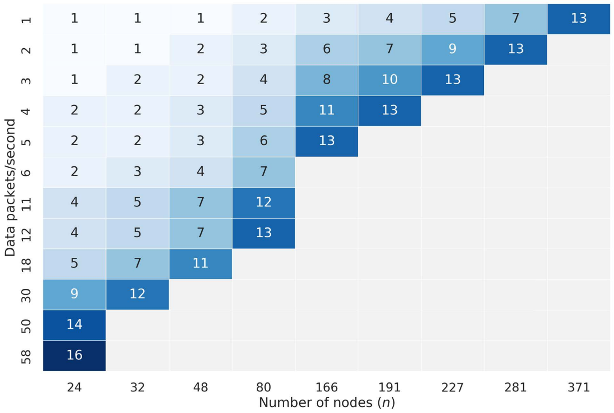

FIGURE 8. Number of radio interfaces depending on the number of nodes and packet frequency with $n_{L1} = 24$.

In Fig. 8 the number of nodes on the first level has been increased to 24. This allows the total number of nodes to be increased 371, as sending one data packet per second, 13 radio interfaces would be needed in the virtual sink. In this case, when using the minimum number of nodes, it is not possible to get the number of radios equal to the first level nodes ($r = n_{L1}$), because the available number of physical channels in the IEEE 802.15.4e standard is 16. In these figures it can also be seen how increasing the number of radios does not increase proportionally the number of nodes. This is because increasing the nodes increases the control traffic and the slotframe occupation more significantly. In addition, the maximum number of radios to be used depends entirely on the network topology, specifically on the nodes that are one hop away from the sink.

Therefore, it is more efficient to use the remaining physical channels in other SD-IWSNs operating concurrently. Following the example of Fig. 6, which shows an SD-IWSN with 2 radio interfaces and 4 channels, 12 free channels of the frequency spectrum would remain, which would allow the creation of three more SD-IWSNs, as shown in Fig. 9. In this case, up to 4 SD-IWSNs are deployed, integrated through the wired network and orchestrated by the same SDN controller. As each of the SD-IWSNs is using a frequency hopping pattern on different channels, there is no interference between them. This allows the coexistence of these networks in the same industrial environment, increasing the density of nodes without increasing the control traffic generated by the SDN controller.

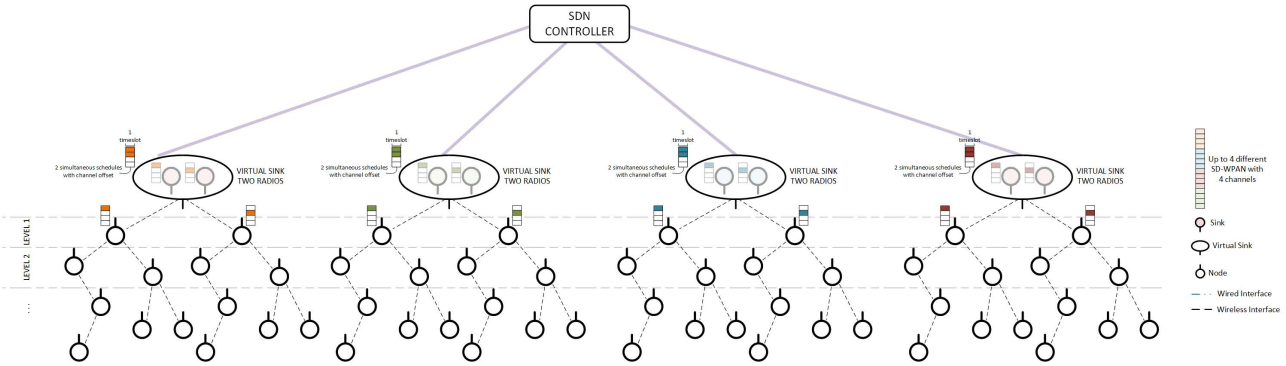
SDN CONTROLLER
1 timeslot
2 simultaneous schedules with channel offset
VIRTUAL SINK TWO RADIOS
1 timeslot
2 simultaneous schedules with channel offset
VIRTUAL SINK TWO RADIOS
1 timeslot
2 simultaneous schedules with channel offset
VIRTUAL SINK TWO RADIOS
1 timeslot
2 simultaneous schedules with channel offset
VIRTUAL SINK TWO RADIOS
LEVEL 1
LEVEL 2
...
Up to 4 different SD-WPAN with 4 channels
Sink
Virtual Sink
Node
Wired Interface
Wireless Interface

FIGURE 9. Multiple frequency-separated SD-IWSNs orchestrated by the same SDN controller.

C. TOPOLOGY

To generalise the obtained results, a square grid topology of size *mxm* has been used. The simulated nodes are 16 meters apart, and the size of the network increases regularly with an odd value of *m*, to ensure that the sink node remains at the center of the topology, as show in Fig. 10. For instance, considering a 3 x 3 topology, the IWSN network would consist of 8 nodes and the sink in the center of the topology.

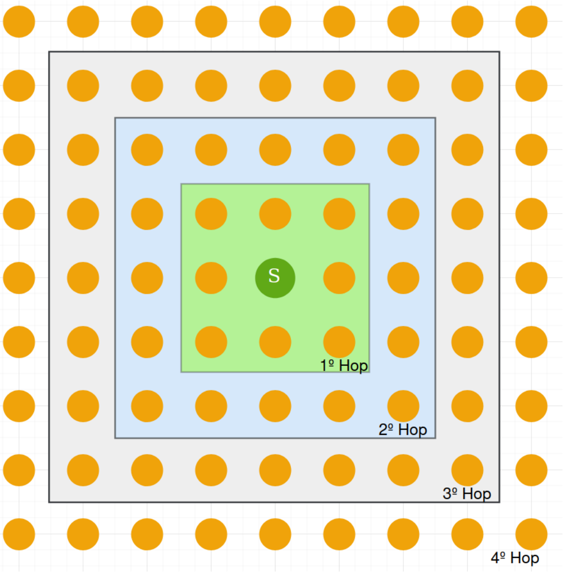

FIGURE 10. Regular square grid topology with sink in the center.

The number of nodes in each hop for the above topology is shown in Fig. 11 which shows the growth of a square grid network consisting of 1000 nodes with two coverage areas that define the nodes in the first level. For n_{L1} there would be a maximum of 16 hops for a 1000-node network, while a 100-node network does not exceed 5 hops. Increasing the node coverage or decreasing the distance between nodes increases the first level nodes, reducing the distance to the sink of all nodes. For a network of 1000 nodes with $n_{L1} = 24$ the farthest nodes are 7 hops from the sink.

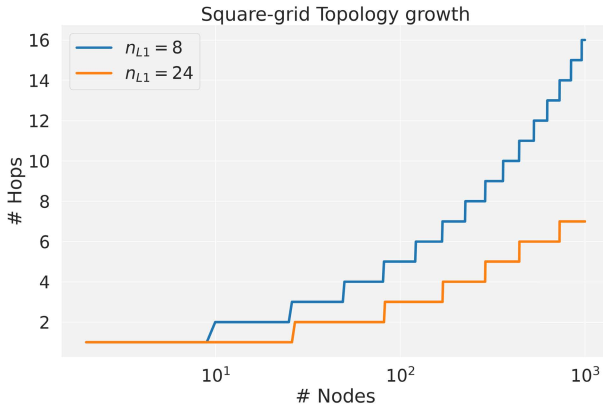

FIGURE 11. Increase in hops to sink in square grid topology depending on the number of nodes.

D. PATH CONFIGURATION

an SDN network, the SDN controller must configure the actions on each of the nodes, therefore the configuration packets must also be scalable and generate the smallest possible increase in control traffic. Therefore, to send the routing configuration and slot assignment, the SDN controller uses the OpenPathTSCH packet, which contains the rules for a specific flow, the route and the hop-by-hop configuration of the slots assigned for TSCH transmission. To avoid sending this information in bulk from the SDN controller, this type of packet has an auto-propagation mechanism, where the receiving nodes process and forward the packet to the next hop in the path, reducing the number of packets to be sent from the SDN controller. However, the size of this packet is limited to 116 Bytes, which are assigned according to Equation (8) obtained in [5]. The packet header (P_h) is common to all SDN-WISE packets and has a length of 10 Bytes, where the necessary parameters for packet processing and forwarding are specified. The indicator field (Ind) is specific to the OpenPathTSCH packet, and contains the values of NR, SF and the number of nodes in the path. These messages are used to configure the corresponding TSCH slots at each node.

$$P_s = P_h + 5 \cdot NR + Ind + 2 \cdot n + 2 \cdot NR \cdot (n - 1) \quad (8)$$

The size of this packet limits the number of hops and number of repetitions (*NR*) that a flow can have. A dashed line is shown in Fig. 12, representing the capacity of this type of configuration packet (116 Bytes). In scenarios with a high number of nodes, using multiple sendings of a flow within the slotframe (*NR*), the maximum value in hops that can configure this packet is 24 (blue curve), where a single transmission of the flow is performed per slotframe ($NR = 1$). However, as shown in Fig. 11, this number of hops is not achieved in square grid topologies, where considering a limit of 1000 nodes, the maximum distance to the sink is 16 hops. If, on the other hand, the flows require a higher frequency of data sending, it is necessary to increase the number of repetitions. In this case it is possible to configure up to 10 transmissions per slotframe for a flow with a distance of 4 hops to the sink in a single OpenPathTSCH packet.

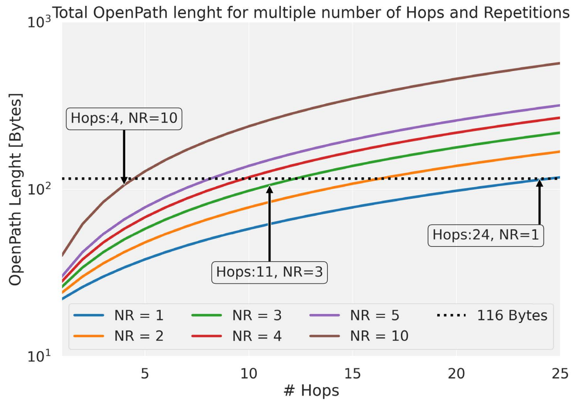

FIGURE 12. Size of the OpenPathTSCH depending on the number of nodes and repetitions within the Slotframe.

IV. RESULTS AND DISCUSSION

This section shows the results using the square grid topology with sink in the center, as shown in Fig. 10. Different types of simulations were performed using the Cooja Simulation tool, deploying different numbers of nodes. The objective of these tests is to analyze the performance and scalability limitations of a conventional sink compared to the virtual sink proposed in this paper, using different configurations with 4, 8 and 16 radio interfaces. Also, the coexistence of multiple SD-IWSN is tested to extend the number of nodes beyond the virtual sink limits.

The simulations have been carried out using the communication stack proposed in Contiki NG [36], in which the sky type motes have been used to emulate the real behavior of the equipment. A summary of the general parameters used in the simulations is shown in Table 3.

The SDN controller used is the SDN WISE-TSCH connected to the sink through a TCP port. This SDN controller includes applications for flow management, routing processes and TSCH scheduler. In addition, ONOS, a more robust industrial SDN controller with a fully modular design, was used to test the approach of multiple SD-IWSN with virtual sinks. This allows to integrate different non-standard southbound protocols, such as the one used by SDN WISE.

TABLE 3. Parameters used in simulation.

Parameter	Value
Simulation time	180 s
Number of runs	10
Data period (D_p)	500ms
Packet frequency (P_f)	2 p/s
n (Topology)	8(3x3), 24(5x5), 48(7x7), 80(9x9)
Number of Channels	4
Timeslot (T_s)	10 ms
Slotframe size (SF)	101 timeslots
Shared slots (Sh)	6, 8, 12, 18 timeslots
Beacon Period (B_p)	3 s
Report Period (R_p)	6 s

A. CONVENTIONAL SINK

Starting from the square grid and Equation (2) for all the control traffic, the average number of actions (Rx or Tx) that each node must perform to receive and retransmit the control traffic were evaluated, depending on its distance to the sink and the number of nodes in the network. As shown in Fig. 13, nodes two hops away from the sink have low saturation, only reaching the TSCH limit level when it is a network of more than 700 nodes, due to the aggregation of traffic from higher nodes. For a single hop this limit is reached at 370 nodes and for the sink at 100, therefore, it is the sink that limits the number of nodes in the network. However, these maximum values are obtained in an ideal environment, where there are no collisions in the shared slots and the traffic is equally balanced among all the nodes in the network. Therefore, the throughput available for data will be reduced by the shared slots allocated to control traffic.

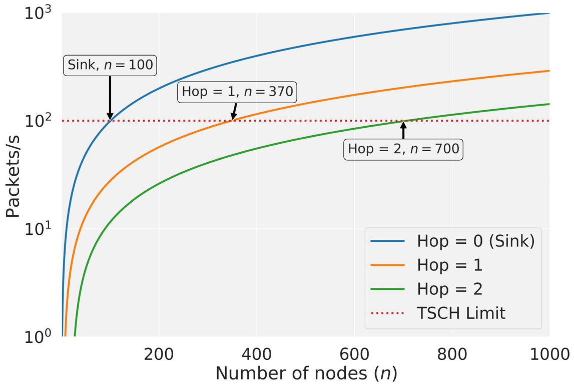

FIGURE 13. Average number of received/transmitted packets by each node as a function of the distance to the sink node.

To have a higher throughput available for data, $B_p = 3s$ and $R_p = 6s$ were used. Adequate to maintain an intermediate network dynamism without a significant throughput reduction due to control traffic [10]. In this case, for 81 nodes, the control traffic requires 20% of the total bandwidth.

Fig. 14 (a) shows how the total amount of data packets increases linearly with the number of nodes, until the maximum throughput of the network is reached. At that moment, packets begin to be stored in the queues until they overflow and are discarded by the node. Therefore, with 49 and 81 nodes, the total amount of data packets does not vary significantly. This packet loss can be seen in Fig. 14 (b), where for the first two networks the PDR is 100%, for the 48-node network, throughput has already exceeded, and the PDR drops to 80%. This situation worsens for 80 nodes, where the PDR decreases to 45%.

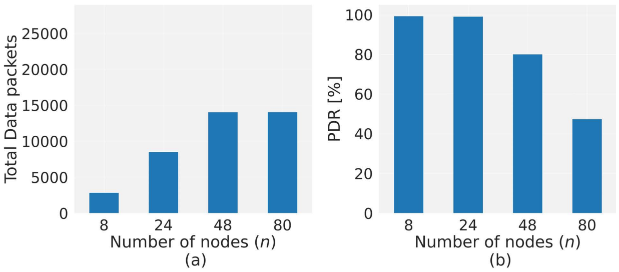

FIGURE 14. **Total data packets in simulations with common sink.**

Fig. 15 shows the proportion of timeslots being used in transmission or reception by the sink, depending on the number of nodes. The first case shows the results for the 3 × 3 topology, where the sink is located in the center and the 8 nodes one hop away from it. The slotframe occupancy is not significant, since most of this occupancy is due to the number of shared slots used to send control traffic. When the number of nodes is increased to 24, the occupancy has a significant impact on the sink compared to the previous result, since the occupancy is over 50%. In the following topologies, it can be seen how the sink reaches the maximum occupancy of the slotframe. Therefore, the network generates more packets than the sink node can receive, creating a bottleneck that ends up saturating the queue of nodes when the network throughput exceeds 80 packets per second or 40-nodes network sending two data packet per second, also limiting the scalability capabilities of the SD-IWSN.

B. VIRTUAL SINK

In accordance with the previous section, the use of a single sink limits the scalability capabilities of the SD-IWSN, because it is the only entry point for the n nodes in the network. The saturation level of the network depends mainly on two factors: the number of nodes and the frequency of sending packets (including data and control messages) from each node, both related to the occupation of the TSCH slotframe and the total throughput of the SD-IWSN.

Increasing any of these variables has a direct impact on the amount of TSCH resources to be allocated, especially at the sink. Because these resources are limited, the sink loses packets when the product $n \cdot D_p$ exceeds the throughput imposed by the TSCH protocol. To reduce this limitation, the proposed virtual sink deployments have been used, in which the use of multiple radio interfaces (r) allows each of these devices to perform multiple actions on different channels and at the same instant of time. This allows the limitation in terms of scalability of the TSCH mechanism to be changed, increasing it by a factor that will depend on the number of radio interfaces used by the virtual sink. This factor ranging from $1/T_s$ to r/T_s, with a limit of $r = 16$, the maximum number of channels in 2.4GHz defined in the IEEE 802.15.4e standard.

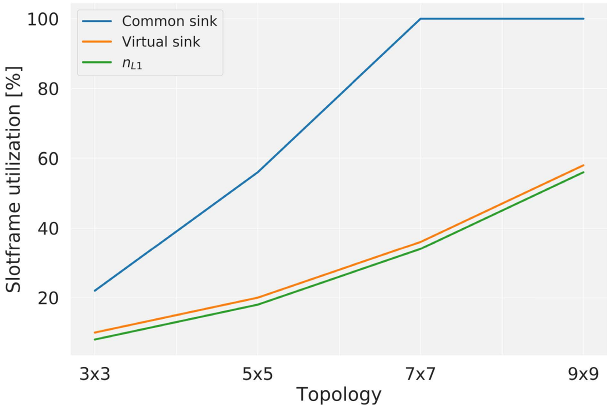

FIGURE 15. **Slotframe occupancy depending on the number of nodes.**

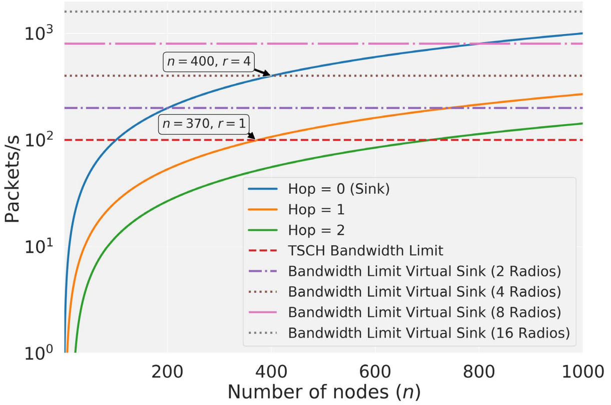

FIGURE 16. **Average packets received/transmitted by the nodes depending on the distance to the sink.**

Fig. 16 shows the distribution of the load on each node, depending on the distance to the virtual sink, and the number of nodes in the network. In addition, virtual sinks with 2, 4, 8 and 16 radio interfaces have been added. In the case of the virtual sink with 16 radio interfaces, the traffic of more than 1000 nodes can be supported, since for this value it does not reach the bandwitdth limit. However, in a network of 1000 nodes, the nodes at the first level would be saturated, preventing traffic from reaching the virtual sink. This behavior is similar for the virtual sink with 4 and 8 radio interfaces, where the bottleneck is the first hop nodes. Having a high number of radio interfaces significantly increases the throughput in the virtual sink, which has a direct impact on the scalability capabilities of the SD-IWSN, although it will be limited by the maximum throughput that can be generated by the first-hop nodes (n_{L1}).

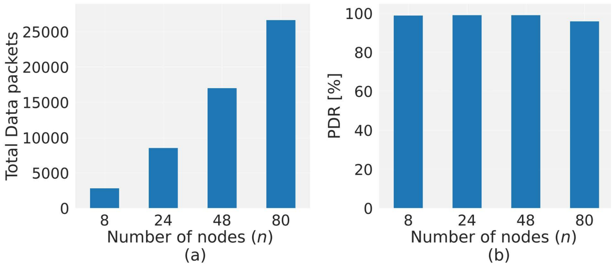

FIGURE 17. **Total data packets in simulations with virtual sink composed of 2 radio interfaces.**

The results of aggregation of radio interfaces can be seen in Fig. 17 (a) where, under the same conditions as the conventional sink, there is a constant growth in the total number of packets. In this case, the difference between the 49-node and 81-node network is clearly observed, where it reaches 27000 data packets, doubling the previous throughput. The PDR shown in Fig. 17 (b) remains above 99% in all cases, so there is no saturation.

Since there are now 4 radio interfaces with independent schedules, the slotframe occupancy in the sink is drastically reduced, as can be seen in Fig. 15, where for the virtual sink it remains below 60% in all cases. Moreover, the curve of the virtual sink is completely adjusted with the use of the slotframe in the first level nodes. As these results show, the aggregation of radio interfaces in virtual sinks considerably increases the bandwidth, avoiding congestion and allowing all the available throughput of the first-hop nodes to be used. These performance improvements not only optimize the SD-IWSNs by making better use of radio resources, but also help to increase the number of deployed nodes and the data sending period.

Finally, if the energy is analyzed, considering the use of a virtual sink with multiple radio interfaces, the impact on the rest of the network nodes is transparent, since for these devices there is only one sink with which to communicate. As for the virtual sink with multiple radio interfaces, the power consumption will be higher, since it must have several of these devices. However, since it must remain connected to the SDN controller, it has a continuous power supply. Therefore, this higher consumption will not result in any negative effect on the lifetime of the network, as is the case with the consumption of the network with a conventional sink.

C. MULTIPLE SD-IWSN USING VIRTUAL SINKS

Using virtual sink-based solutions, it is evident that it is possible to improve the total bandwidth capabilities of an SD-IWSN, the impact of which translates into improved network scalability, as well as the transmission frequency of the different nodes of the SD-IWSN. However, in some cases only 4 of the 16 channels available in the 2.4 GHz band are used. To further improve the scalability of the SD-IWSN and take full advantage of the TSCH radio spectrum, it is possible to replicate these SD-IWSNs using a virtual sink, but using a tuple of 4 different channels, which will remain completely isolated in different interference domains. The SDN controller is designed to aggregate multiple networks, even of different types, so the only limitation of the control plane will be the CPU and memory usage of the SDN controller per managed node. Therefore, there is no difference at the SDN controller level between increasing the number of devices on one network, or across multiple networks. If this network growth were to be carried out taking into account the single network option, it would have a considerable negative impact, since control traffic is forwarded through shared slots. The shared slots are common to all nodes in the network, therefore, when a node must send control information it can use any of the configured shared slots to make the transmission. This behavior increases the probability of control traffic collisions as the number of nodes in the same network increases. However, using multiple SD-IWSNs not only mitigates this problem, but also takes full advantage of the available spectrum. Accordingly, the number of nodes in each SD-IWSN can be increased as long as the control and data bandwidth requirements do not exceed the TSCH limit, and thereafter must be increased through other coexisting SD-IWSNs. SDN allows different SD-IWSNs to be managed from the same SDN controller, reducing the administration and configuration difficulties of deploying multiple SD-IWSNs.

TABLE 4. **ONOS RAM and CPU usage with multiple SD-IWSN.**

SD-IWSN	0	1	2	3	4
CPU	20%	20.7%	21.5%	22.1%	23%
RAM	4 Gb	6.67 Gb	9.21 Gb	11.92 Gb	14.55 Gb

To test the coexistence of multiple SD-IWSNs, the ONOS controller is used, deploying 4 SD-IWSNs each with 4 different channels, in which the topology of each one is a square grid of 80 nodes. The results obtained in terms of total packets, PDR and slotframe occupancy have the same behavior as that obtained in Fig. 17 and Fig. 15 for each of the networks, since groups of 4 different channels have been used, thus taking advantage of the entire spectrum of the IEEE 802.15.4e standard. The computational consumption in terms of CPU and RAM can be seen in Tab. 4, where the consumption of the SDN controller without connected devices is shown, and with up to 4SD-IWSN of 80 nodes each, where the virtual sinks are formed by 4 radio interfaces with different channels, which in total allows a network of 320 nodes to be served.

D. TESTBED

To test the performance with real hardware, a testbed was set up in which two frequency-isolated SD-IWSNs were deployed, using groups of 4 different channels. The networks have a 3 × 3 square-grid topology and the power of the nodes was limited to -15 dB to perform the tests in a laboratory environment. In this case, each network would be composed of a sink node and 8 nodes that perform periodic data forwarding. The sink is connected to a Raspberry Pi via USB, to give it access to the wired SDN network, where the SDN controller is located, as shown in Fig. 18. The tests were carried out using the two types of sink, first the two SD-IWSN with a conventional sink and finally both SD-IWSN with a virtual sink with 2 radio interfaces, formed using two openmote-B managed by the SDN controller so that the other nodes in the network perceive it as a single device.

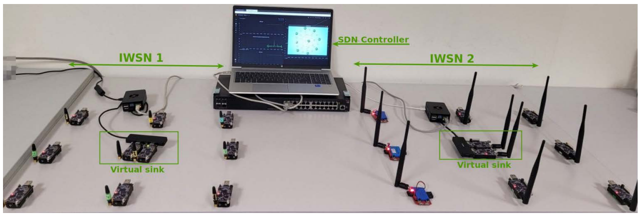

FIGURE 18. Testbed: 2 SD-IWSN with square-grid topology, virtual sink with two radio interfaces.

In order to observe the effect of saturation with a limited number of nodes, different packet generation frequencies were used, ranging from two packets per second (500 ms) as in the simulations, to 20 packets per second (50 ms). The operating system of the nodes was Contiki NG, on which the firmware for the Openmote-b and REmote hardware was developed. The configuration parameters used are listed in Tab. 5.

TABLE 5. Parameters used in testbed.

Parameter	Value
Simulation time	180 s
Number of runs	5
Data period (D_p)	500 ms, 250 ms, 100ms, 50ms
Packet frequency (P_f)	2 p/s, 4 p/s, 10 p/s, 20 p/s
n (Topology)	8(3x3)
Number of Channels	4
Transmision power	-15 dB
Timeslot (T_s)	10 ms
Slotframe size (SF)	101 timeslots
Shared slots (Sh)	6 timeslots
Beacon Period (B_p)	3 s
Report Period (R_p)	6 s

The results obtained after injecting data traffic from all nodes are shown in Fig. 19, where the total number of packets received by SD-IWSN for different data sending periods is shown. Using the 500 ms and 250 ms periods, the behavior obtained is similar to that of the simulations, where for a small number of nodes there is no difference between using hardware with one radio interface or two. For the 100 ms and 50 ms cases, it is evident that the network with the conventional sink reaches its maximum bandwidth at 100 ms, since the 50 ms and 100 ms total packets columns do not show a significant difference. On the contrary, in the virtual sink, where its two radio interfaces allow a higher packet flow, a slightly higher total number of packets than the conventional sink is observed in the 100ms period. When the period is reduced to 50 ms, the virtual sink doubles the total number of packets, while the conventional sink receives the same number of packets as with the 100 ms period.

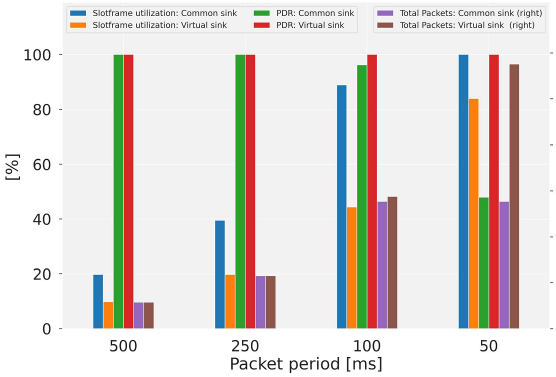

FIGURE 19. PDR and Slotframe utilization in the testbed, with different packet frequency.

The results show that, thanks to the higher bandwidth capacity in the virtual sink, it is possible to guarantee a higher PDR in conditions with high traffic flow, which has a direct impact on the scalability of the network and the maximum data transfer frequency. As can be seen in the green and red columns of Fig. 19, a PDR close to 100% is obtained for the first two cases, both with the conventional sink and the virtual sink. However, from this point on, the conventional sink starts to lose packets, until it loses more than 50% for the tests with a period of 50 ms, where the virtual sink maintains the PDR above 99%. This packet loss is caused by the lack of TSCH resources, as shown in the blue and orange columns of Fig. 19, which increase progressively when the period is reduced, as in the simulation in which the number of nodes was increased. The conventional sink has 100% of the timeslots assigned to it, where each of the transmitter nodes has 30 slots assigned to it, about 30% of the slotframe. In the virtual sink, no saturation is observed in the use of the slotframe since, although it has the same number of timeslots, it can perform two receptions in each one, and in this way it does not lose packets and the use of slotframe does not reach 100% of occupancy. These results can be extrapolated to networks with a larger number of nodes, since high packet generation frequencies have been used to observe the behavior of the network with a high amount of data flow. For example, a packet generation rate of 20 packets per second ($D_p = 50$ ms) in a 9-node network generates the same amount of data packets as a 180-node network with a generation rate of 1 packet per second.

V. CONCLUSION

Scalability in IWSNs has traditionally been addressed by routing and clustering to reduce network energy consumption. However, to implement these networks in industrial sectors, a QoS-oriented approach is required, which is not possible to achieve with traditional networks. Hence, a different approach is required, such as the use of SDN networks, which provide greater flexibility, ease of administration and optimized network configurations to ensure different levels of QoS.

In this paper, the scalability limitations of traditional IWSN and SD-IWSN networks have identified. It has been proven through simulation and testbed, that thanks to SDN technology it is possible to create a logical grouping of single nodes and manage them through the SDN controller as a single virtual sink, which physically will have multiple radio interfaces that allow the radio interaction capabilities to be multiplied.

Thanks to the use of these virtual sinks, it is possible to increase the throughput of the SD-IWSN, also allowing the number of nodes in the network to be increased and the negative effects of control traffic associated with this increase in the number of nodes to be reduced. This behavior is especially significant in the sink, where increasing the size of the network causes a bottleneck due to the aggregation of traffic from all nodes in the topology. Allowing to increase the total number of nodes at least 0.7 times for each radio interface with respect to a SD-IWSN with a traditional sink. Passing from networks of 83 nodes to networks of 144 nodes (1.73 more nodes, 0.86 per radio interface) or 232 nodes (2.8 more nodes, 0.7 per radio interface). Also, maintaining the number of nodes, the packet frequency can be incremented proportionally to the number of radio interfaces.

Furthermore, thanks to the centralized control of multiple networks, optimizing the network distribution from a single point, where it is possible to aggregate multiple SD-IWSNs, separated by physical channels, to avoid interference. Among the different advantages provided by SDN, the results of this paper highlight that, thanks to the combination of different methods orchestrated by the SDN controller, it is possible to deploy 8 SD-IWSNs with two radio interfaces that allow 144 nodes per SD-IWSN. For a total of 1152-node SD-IWSNs in which a PDR higher than 99% is guaranteed even under high traffic loads. This means that using virtual sinks in a combination of multiple networks can increase by up to 13.7 times in node capacity achieved concerning a conventional IWSN.

As future work, the use of Machine Learning applications will be considered to interact with the SDN controller. That guarantees the throughput and scalability in the multiple SD-IWSN approach, through an analysis of the channels used by each network and their performance. This will allow not only to select the best channels for each IWSN but also to automatically optimize the segmentation of the networks by choosing the IWSN with the optimal channel configuration for each node. Also, flexibility is enhanced by allowing the nodes to move between different IWSNs according to the network performance.

REFERENCES

[1] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: Challenges and design requirements," *IEEE Access*, vol. 5, pp. 1872–1899, 2017.

[2] H. Kim, J. Ko, D. E. Culler, and J. Paek, "Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2502–2525, 4th Quart., 2017.

[3] D. Hauweele, R.-A. Koutsiamanis, B. Quoitin, and G. Z. Papadopoulos, "Pushing 6TiSCH minimal scheduling function (MSF) to the limits," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2020, pp. 1–7.

[4] A.-C. Anadiotis, L. Galluccio, S. Milardo, G. Morabito, and S. Palazzo, "SD-WISE: A software-defined WIreless SEnsor network," *Comput. Netw.*, vol. 159, pp. 84–95, Aug. 2019.

[5] F. Orozco-Santos, V. Sempere-Payá, T. Albero-Albero, and J. Silvestre-Blanes, "Enhancing SDN WISE with slicing over TSCH," *Sensors*, vol. 21, no. 4, p. 1075, Feb. 2021.

[6] M. Baddeley, R. Nejabati, G. Oikonomou, M. Sooriyabandara, and D. Simeonidou, "Evolving SDN for low-power IoT networks," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Jun. 2018, pp. 71–79.

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

[8] S. Kharb and A. Singhrova, "A survey on network formation and scheduling algorithms for time slotted channel hopping in industrial networks," *J. Netw. Comput. Appl.*, vol. 126, pp. 59–87, Jan. 2019.

[9] H.-T. Wu, K.-W. Ke, S.-F. Wang, P.-H. Chen, G.-D. Lee, C.-Y. Tseng, and C.-Y. Ho, "The implementation of wireless industrial Internet of Things (IIoT) based upon IEEE 802.15.4–2015 TSCH access mode," in *Proc. IEEE Int. Conf. Dependable, Autonomic Secure Comput., Int. Conf. Pervasive Intell. Comput., Int. Conf. Cloud Big Data Comput., Int. Conf. Cyber. Sci. Technol. Congr. (DASC/PiCom/CBDCom/CyberSciTech)*, Aug. 2019, pp. 367–369.

[10] F. Orozco-Santos, V. Sempere-Paya, J. Silvestre-Blanes, and T. Albero-Albero, "Multicast scheduling in SDN WISE to support mobile nodes in industrial wireless sensor networks," *IEEE Access*, vol. 9, pp. 141651–141666, 2021.

[11] M. Karakus and A. Durresi, "Quality of service (QoS) in software defined networking (SDN): A survey," *J. Netw. Comput. Appl.*, vol. 80, pp. 200–218, Feb. 2016.

[12] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, "A roadmap for traffic engineering in SDN-OpenFlow networks," *Comput. Netw.*, vol. 71, pp. 1–30, Oct. 2014.

[13] K. Singh, "WSN LEACH based protocols: A structural analysis," in *Proc. Int. Conf. Workshop Comput. Commun. (IEMCON)*, Oct. 2015, pp. 1–7.

[14] E. F. A. Elsmany, M. A. Omar, T.-C. Wan, and A. A. Altahir, "EESRA: Energy efficient scalable routing algorithm for wireless sensor networks," *IEEE Access*, vol. 7, pp. 96974–96983, 2019.

[15] A. Shukla and S. Tripathi, "An effective relay node selection technique for energy efficient WSN-assisted IoT," *Wireless Pers. Commun.*, vol. 112, no. 4, pp. 2611–2641, Jun. 2020.

[16] L. L. Bello, A. Lombardo, G. Patti, and M. Reno, "A platform for evaluating clustering strategies in mobile IEEE 802.15.4-TSCH networks," in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2019, pp. 987–992.

[17] M. Sajwan, D. Gosain, and A. K. Sharma, "Hybrid energy-efficient multipath routing for wireless sensor networks," *Comput. Elect. Eng.*, vol. 67, pp. 96–113, Apr. 2018.

[18] G. Shabbir, A. Akram, M. M. Iqbal, S. Jabbar, M. Alfawair, and J. Chaudhry, "Network performance enhancement of multi-sink enabled low power lossy networks in SDN based Internet of Things," *Int. J. Parallel Program.*, vol. 48, no. 2, pp. 367–398, Dec. 2018.

[19] F. Veisi, M. Nabi, and H. Saidi, "Coexistence analysis of multiple asynchronous IEEE 802.15.4 TSCH-based networks," *IEEE Access*, vol. 8, pp. 150573–150585, 2020.

[20] *IEEE Standard for Low-Rate Wireless Networks*, IEEE Standard 802.15.4-2020, (Revision of IEEE Standard 802.15.4-2015), 2020, pp. 1–800.

[21] J. Vera-Pérez, D. Todolí-Ferrandis, S. Santonja-Climent, J. Silvestre-Blanes, and V. Sempere-Payá, "A joining procedure and synchronization for TSCH-RPL wireless sensor networks," *Sensors*, vol. 18, no. 10, p. 3556, Oct. 2018.

[22] A. R. Urke, Ø. Kure, and K. Øvsthus, "A survey of 802.15.4 TSCH schedulers for a standardized industrial Internet of Things," *Sensors*, vol. 22, no. 1, p. 15, Dec. 2021.

[23] R. C. A. Alves, D. A. G. Oliveira, G. A. N. Segura, and C. B. Margi, "The cost of software-defining things: A scalability study of software-defined sensor networks," *IEEE Access*, vol. 7, pp. 115093–115108, 2019.

[24] F. F. Jurado-Lasso, K. Clarke, and A. Nirmalathas, "Performance analysis of software-defined multihop wireless sensor networks," *IEEE Syst. J.*, vol. 14, no. 4, pp. 4653–4662, Dec. 2020.

[25] A. Abuarqoub, "A review of the control plane scalability approaches in software defined networking," *Future Internet*, vol. 12, no. 3, p. 49, Mar. 2020.

[26] O. Flauzac, C. J. G. Santamaria, F. Nolot, and I. Woungang, "An SDN approach to route massive data flows of sensor networks," *Int. J. Commun. Syst.*, vol. 33, no. 7, p. e4309, May 2020.

[27] S. Tahmasebi, N. Rasouli, A. H. Kashefi, E. Rezabeyk, and H. R. Faragardi, "SYNCOP: An evolutionary multi-objective placement of SDN controllers for optimizing cost and network performance in WSNs," *Comput. Netw.*, vol. 185, Feb. 2021, Art. no. 107727.

[28] E. Municio, K. Spaey, and S. Latré, "A distributed density optimized scheduling function for IEEE 802.15.4e TSCH networks," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 7, p. e3420, Jul. 2018.

[29] E. Municio and S. Latré, "Decentralized broadcast-based scheduling for dense multi-hop TSCH networks," in *Proc. Workshop Mobility Evolving Internet Archit.*, Oct. 2016, pp. 19–24.

[30] J. C. Da Silva and F. Assis, "A distributed algorithm to schedule TSCH links under the SINR model," *Des. Autom. Embedded Syst.*, vol. 23, nos. 1–2, pp. 21–39, Jun. 2019.

[31] A. Demir and S. Bilgili, "DIGEST: A decentralized divergecast scheduling algorithm for IEEE 802.15.4e TSCH in the Internet of Things," *Wireless Netw.*, vol. 27, pp. 1–16, Oct. 2021.

[32] M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, and T. Engel, "On optimal scheduling in duty-cycled industrial IoT applications using IEEE802.15.4e TSCH," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3655–3666, Oct. 2013.

[33] Y. Jin, P. Kulkarni, J. Wilcox, and M. Sooriyabandara, "A centralized scheduling algorithm for IEEE 802.15.4e TSCH based industrial low power wireless networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2016, pp. 1–6.

[34] N. Accettura, M. R. Palattella, G. Boggia, L. A. Grieco, and M. Dohler, "Decentralized traffic aware scheduling for multi-hop low power lossy networks in the Internet of Things," in *Proc. IEEE 14th Int. Symp. World Wireless, Mobile Multimedia Netw.*, Jun. 2013, pp. 1–6.

[35] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust mesh networks through autonomously scheduled TSCH," in *Proc. 13th ACM Conf. Embedded Networked Sensor Syst.* New York, NY, USA: Association for Computing Machinery, Nov. 2015, pp. 337–350.

[36] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes, "The Contiki-NG open source operating system for next generation IoT devices," *SoftwareX*, vol. 18, Jun. 2022, Art. no. 101089.

FEDERICO OROZCO-SANTOS received the master's degree in telecommunications engineering from the Universitat Politècnica de València (UPV), Spain, in 2019, where he is currently pursuing the Ph.D. degree. In January 2020, he joined at the Advanced Communications and Industrial Informatics Group, Informatic Technological Institute (ITI). His research interest includes the application of dynamic management approaches in wireless sensor networks. The focus of this research is mainly on the end-to-end quality of service, using slicing, virtualized networks functions, cloud computing, and software defined networks.

VÍCTOR SEMPERE-PAYÁ (Member, IEEE) received the Graduate degree in industrial electronics and computer engineering and the Ph.D. degree in telecommunications engineering from the Universitat Politècnica de València (UPV), Valencia, Spain, in 1987, 1993, and 1998, respectively. Currently, he is an Associate Professor with the Department of Communications, UPV, where he teaches industrial communications and public access networks. His current research interests include factory communications, real time communications, and quality of service (QoS) in networks. He has served as a program committee member for several conferences in the area of factory communications. Since 1996, he has authored or coauthored more than 60 technical papers in journals and international conferences. He has managed more than 50 research and technological projects. Currently, he is also the Director of the Advanced Communications and Industrial Informatics Group, Informatic Technological Institute (ITI).

JAVIER SILVESTRE-BLANES (Member, IEEE) was born in Alcoy, Valencia, Spain, in 1965. He received the M.S. and Ph.D. degrees in computer architecture, in 1999 and 2003, respectively. In 1999, he joined the Department of Computer Architecture, Universitat Politècnica of València, as a part-time Associate Lecturer, having been promoted to an Adjunct Professor, in 2001. He was promoted to an Assistant Lecturer, in 2004, and since 2010, he has been working as an Associate Professor. He joined the ITI, in 2009, in the research and development area (industrial computing, communications, and image processing), participating in research and development projects both individually, and in cooperation with other entities. He has more than 32 papers in international peer-reviewed conferences, 14 articles published in journals of impact JCR and eight book chapters, related to the sector of industrial communications, heterogeneous networks, multimedia networks, image processing, and computer vision. He has carried out research stays abroad, at the University of Aveiro, Portugal, and Anglia Ruskin University, Cambridge, U.K. He is a member of the Industrial Electronics Society (IES), belonging to the IEEE, where he also belongs to the Sub-Committee FA10 Computer Vision and Human–Machine Interaction in Industrial and Factory Automation, belonging to the IES-IEEE Factory Automation Committee. Since 2005, he has been participated as a member of the organizing committee or a member of the Program Committee of numerous international conferences of the IEEE Industrial Electronics Society.

JOSE VERA-PÉREZ received the B.S. and M.S. degrees in telecommunications engineering from the Universitat Politècnica de València (UPV), Spain, in 2017, and the Ph.D. degree in telecommunication engineering with his research on optimization mechanism for industrial wireless sensor networks. Currently, he is a Research and Development Engineer, part of the Advanced Communication and Industrial Informatics Group, ITI, Valencia. He is the author of several recent research related to wireless sensor networks for industrial communications. His research interests include private 5G networks, low power wireless communications, industry 4.0, industrial automation, and cybersecurity.