IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 21, 2021, accepted October 12, 2021, date of publication October 15, 2021, date of current version October 25, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3120917

Multicast Scheduling in SDN WISE to Support Mobile Nodes in Industrial Wireless Sensor Networks

FEDERICO OROZCO-SANTOS[1], **VÍCTOR SEMPERE-PAYÁ**[1,2], **(Member, IEEE), JAVIER SILVESTRE-BLANES**[1,3], **(Member, IEEE), AND TERESA ALBERO-ALBERO**[1,3]

[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 by the Ministerio de Ciencia e Innovación (MCIN)/agencia estatal de investigación (AEI)/10.13039/501100011033 and the European Regional Development Fund (ERDF) "A way of making Europe" under Grant PGC2018-094151-B-I00.

ABSTRACT The growing use of mobile nodes in manufacturing environments is increasing the Quality of Service (QoS) requirements of the communications infrastructures that support them, as is the case with Industrial Wireless Sensor Networks (IWSN). Managing networks using Software Defined Networks (SDN) addresses the challenge of handling multiple sources in a satisfactory manner. The global vision of the topology that the SDN controller has allows a logical segmentation of the network, through the allocation of dedicated resources for each information flow, thus ensuring independent service levels and complete isolation. Mobile nodes produce constant changes in topology, which lead to instability for routing protocols. Traditional routing solutions with protocols such as Routing Protocol for Low-Power and Lossy Networks (RPL) require additional time to make the parent node change. This paper proposes Mobile Multicast Forwarding with Software Defined Network (MMF-SDN), an approach using Software Defined Networking solution for WIreless SEnsor Network (SDN WISE) protocol that exploits the advantages of SDN and Time Slotted Channel Hopping (TSCH) synchronism. The mobile nodes are managed as multicast sources, through a cumulative allocation of resources from the controller. This allows reception states to be synchronized at the parent nodes to provide network stability, avoiding subsequent recalculations in the routing protocol. Parent node changes are transparent and immediate. This proposal improves on other SDN based solutions, reducing energy consumption in reception by up to 50%, 70% end-to-end delay and improving scalability with a 30% reduction in slotframe occupancy.

INDEX TERMS IWSN, mobility, multicast, QoS, SDN, TSCH.

I. INTRODUCTION

Nowadays, the industry sector needs to optimize processes and adapt quickly to technological changes. This has accelerated the evolution towards smart factories [1]. These incorporate massive measurement elements to simplify the integration of multiple variables in production lines. This transformation requires interconnectivity between all relevant systems in order to make decisions based on large volumes of information. Therefore, a range of elements must be integrated, which, combined with the dynamism of production environments generates the need to provide support for mobile elements, such as Automated Guided Vehicles (AGV) that are widely used in logistics tasks.

Industrial Wireless Sensor Networks (IWSN) have a major role in this transformation, enabling fast, simple and robust deployment to meet the requirements of industrial environments [2]. The inclusion of mobile nodes has been extensively researched in Mobile Ad-Hoc Networks (MANETSs), Flying Ad-hoc Networks (FANETS) and Vehicular Ad-hoc Networks (VANETS) [3]. However, IWSNs must guarantee strict QoS requirements to obtain the highest possible determinism. A difficult objective to achieve, considering the constant changes in the topology produced by the use of mobile nodes, which directly affect network performance.

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

The evolution towards new paradigms such as Software-Defined Network (SDN) [4] is becoming a better option to address the complexity of challenges in IWSN. These accept an increase in control traffic in exchange for additional advantages, such as: a reduction in the complexity of the processes at the nodes and a global view of the network elements. An example of this is the use of the Time Slotted Channel Hopping (TSCH) protocol for deterministic medium access control and SDN to integrate the routing process and flow descriptor for enhance de QoS. This is demonstrated in [5], where it is possible to guarantee different QoS parameters for different types of flow dynamically and with low complexity.

This work proposes Mobile Multicast Forwarding SDN (MMF-SDN) an SDN-based alternative that allows mobile nodes to improve determinism and guarantee QoS parameters with the same level of reliability as fixed nodes. It is based on the use of the global knowledge provided by SDN and the synchronization obtained with TSCH, to implement a centralized TSCH scheduler that performs a cumulative multicast resource allocation. This consists of centrally assigning a transmission cell that may have multiple receivers for each mobile node. These cells will be added to each of the parent nodes as they are discovered, and will remain active until they are removed by the controller. Therefore, with each resource allocation the connectivity of the mobile node is increased without increasing transmissions. This allocation will eliminate the problem of high signalling, because resources are allocated only the first time a parent node is added. In this multicast approach, the use of a single input cell for each mobile node and the reduction of the bandwidth required for control traffic, allows for more compact slotframes thereby increasing the maximum number of nodes.

The rest of the paper is organized as follows. Section 2 briefly introduces the TSCH protocol, SDN and the basics of the Software Defined Networking solution for WIreless SEnsor Network (SDN WISE). Moreover, there is a brief literature review highlighting the current approach to supporting mobile nodes using SDN that is Forwarding and Time-Slotted channel hopping scheduling over SDN (FTS-SDN). Section 3 presents the proposed methodology, Mobile Multicast Forwarding SDN (MMF-SDN) and the differences with FTS-SDN. Section 4 contains the description of the scenarios used in the tests as well as the results. Conclusions and future work are shown in Section 5.

II. BACKGROUND AND LITERATURE REVIEW

In this section, the fundamental aspects of Wireless Sensor Networks (WSN), their evolution towards Software Defined Networks, and how the implementation of SDN WISE provides mechanisms that allow the network to adapt to the constant changes in the topology produced by a mobile node are discussed.

A. BACKGROUND

WSNs are composed of multiple nodes in charge of collecting information from the environment and sending it to a common point (Sink), which is responsible for connecting the nodes with the point where the information is processed. Communications are carried out over a shared medium, where a medium access control protocol, such as Carrier Sense Multiple Access (CSMA), must be used to limit collisions. Wireless links provide great flexibility and scalability, allowing WSNs to be easily adapted to any scenario and facilitating rapid incorporation in industrial scenarios. However, to achieve the levels of reliability required for industrial-level applications, more robust nodes and protocols are needed. For this reason, the IEEE 802.15.4e standard incorporates medium access protocols to increase determinism in communications [6]. These are Low Latency Deterministic Network (LLDN), Deterministic and Synchronous Multi-channel Extension (DSME) and TSCH. In DSME and TSCH, the performance is adapted to industrial requirements by offering high interference tolerance and deterministic behaviour. However, the slotframe structure in TSCH is more flexible, as it is possible to add or remove a single timeslot [7].

1) TIME SLOTTED CHANNEL HOPPING

The TSCH medium access control protocol, defined in the IEEE 802.15.4e standard, is considered a suitable solution for providing real-time multi-hop transmissions in industrial environments [8]. TSCH offers guarantees in terms of latency and data delivery in IWSNs since it allows nodes to have a guaranteed space for sending their packets. To do this, it divides the time into a fixed number of timeslots that are grouped into slotframes. The behavior of the slotframes is cyclic, which means that when the last timeslot ends, the slotframe starts again with the first timeslot.

The most common duration of each timeslot is 10 ms, enough time to be able to transmit a frame of the maximum size (127 Bytes), wait for the Acknowledgement (ACK) from the receiver, 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, which can be shared or dedicated. In the case of shared, any of the nodes can transmit if they have packets to send, otherwise the node goes to receiving mode. Therefore, in the shared slot the CSMA scheme is followed with backoff exponent to reduce collisions. On the other hand, if it is dedicated, a unidirectional link is established between two nodes and only the source node can transmit during the timeslot and in the specific offset channel. The TSCH scheduling provides the action that the node will perform in each timeslot. The node will be able to receive, transmit, or turn off the radio. Time division requires that all nodes are correctly synchronized in order to ensure that the transmission and reception time slots coincide. This is achieved with the Enhanced Beacon (EB) packet exchange, which includes information about the Absolute Sequence Number (ASN) that the nodes take as a time reference and determines the current timeslot, also allows the clock drifts to be compensated. The ASN information is used with Channel offset to generate a rotation of the physical channels in each slotframe using (1). That helps reduce the points of interference and path fading, thus increasing reliability. The problem here is that a large number of channels (N_{ch}) have a longer synchronization time [9].

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

To allocate each of the cells in the nodes, there must be a scheduling process to synchronize the transmission and reception states. However, the IEEE 802.15.4e standard does not include any specification on how to do this. For this reason, there are different types of TSCH schedulers, that can be classified as: Distributed schedulers such as Decentralized Traffic Aware Scheduling (DeTAS) [10], and Wave [11]; others are centralized, such as Traffic Aware Scheduling Algorithm (TASA) [12] and Adaptive MUltihop Scheduling (AMUS) [13]; and autonomous scheduling algorithms such as Orchestra [14], as well as some variants of these such as Escalator [15] and Autonomous Link-based Cell scheduling (ALICE) [16].

2) SOFTWARE DEFINED NETWORKING

The SDN paradigm [17] arose in data centres, where a large amount of resources needed to be managed and optimized. It consists of managing the networks by separating the control plane and the data plane. The control plane is fully software-controlled by a centralized agent known as a controller. It is in charge of defining the forwarding rules based on a total topological knowledge of the network and the policies established for each of the flows. The data plane is composed of the physical equipment, operating in plug and play mode. This equipment receives these rules and stores them as flow tables. They only need to search within this table and execute the actions sent by the controller. This approach makes the network programmable since the controller can modify the flow tables of each device.

The flexibility achieved by separating the data plane and the control plane overcomes the limitations of traditional networks [18]. This is because it provides a global view of the network, optimizing its use and simplifying the provision of resources. With SDN it is possible to dynamically manage, configure and optimize network resources through automated programs or functions that are integrated into the controller. For this reason, several studies have been conducted on the advantages of SDN, especially in QoS where it is possible to exploit the dynamism and network management with a higher level of detail. An example of this is [19] where the controller obtains the best route for each type of traffic according to the QoS requirements using parameters such as: Delay, Jitter, Throughput and Loss.

Therefore, in addition to the control plane and the data plane, there is the application plane that groups together all the automated functions that modify the behavior of the network. The application plane is located above the control plane, as shown in Fig. 1, which allows the global knowledge of the controller to be reused for specific tasks. The objective of this is to simplify the development of applications, since it is not necessary to go into details about the operation of the network infrastructure, it will be the controller that communicates with each of the network elements. Hence, the SDN controller handles a great diversity of protocols, which are divided into Northbound (NB) and Southbound (SB). The NB protocols establish communication between the application plane and the controller. On the other hand, SB protocols are used to communicate with the network infrastructure, the most widespread being OpenFlow developed by the Open Networking Foundation (ONF) [20].

The flexibility, simplicity of deployment and management offered by SDN has driven its expansion to all points of the wired network. According to Cisco® [21], 64% of organizations had SDN solutions in their data centres, 58% as WAN technology (SD-WAN) and 40% in access networks. Currently, all infrastructure equipment manufacturers offer SDN equipment and controllers. Hence, they are also looking to incorporate the advantages of SDN into IWSNs, particularly flexibility. It is possible to generate any type of controller-oriented application, and new possibilities are opened up to address the wide range of challenges presented by IWSNs. Improvements can be added with the implementation of an SDN such as energy efficiency, routing, mobility, security, reliability and QoS [4].

3) SDN WISE PROTOCOL

The application of the SDN approach in IWSNs is not straightforward, due to the large differences in hardware resources of the equipment that make up the data plane. SDN WISE is a protocol that extends the SDN paradigm to WSNs [22]. It is based on the OpenFlow protocol, a protocol designed for wired networks. This has involved making the necessary adaptations, as there are many challenges when it is deployed in wireless environments [23]. To meet these challenges, a profound change in the OpenFlow protocol was required, such as being able to make decisions according to local states without controller intervention (Statefull) or being able to perform packet aggregation to reduce congestion. Due to the complexity of the changes, protocols have been created that allow OpenFlow concepts to operate within the restrictions of wireless environments such as SDN WISE. These changes are intended to reduce control traffic and conserve network bandwidth. Due to the wireless nature of WSNs, there is no physical isolation between transmissions. By using the CSMA medium access method, control traffic has a direct impact on network performance. However, the versatility, flexibility, and ease of management provided by an SDN deployment [22], are key elements within WSNs. In this way, traffic flows in the network can be scheduled and modified by the controller, simply by modifying the flow table at each node. The SDN WISE framework is mainly composed of the following sub-layers: Application, In-Network Packet Processing (INPP), Topology Discovery (TD), Forwarding (FWD), Medium Access Control (MAC) and the physical layer that will be 802.15.4. These layers are shown in Fig. 1

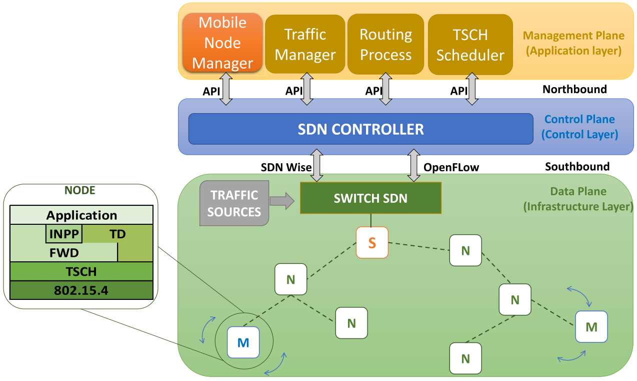


FIGURE 1. SDN general architecture and SDN WISE node layering.

where the layers in the node are detailed, using the TSCH protocol in the MAC layer.

The three sub-layers where the main operation of SDN WISE is concentrated are described below:

1) In-Network Packet Processing (INPP): Is a layer used for data aggregation. To minimize network congestion, small packets with the same destination can be combined. However, it should be noted that the maximum size of an SDN WISE packet is 116 Bytes and cannot be exceeded.
2) Topology Discovery (TD): This layer is responsible for discovering the adjacent nodes and informing the controller of its neighbours. All nodes generate beacon packets, which are sent periodically in broadcast mode. This packet contains the identity of the node that generated it, the battery level and the current distance to the sink, which is updated according to the beacons received. With the beacon packets, the node informs the other nodes within its coverage range of its presence. In this way, each node discovers its adjacent nodes. When a node receives this type of packet, it adds the sender to its neighbour table, along with its current Received Signal Strength Indicator (RSSI) and battery level. If that node is already in its table, it only updates the RSSI value and the battery level. Information from adjacent nodes is sent via a report packet to the controller. Each node sends this packet periodically every two beacon periods. The default beacon period setting is one second. The packet contains the number of neighbours and information about each one, such as the node ID and RSSI. Thanks to the reception of report packets from each node, the controller builds the global topology of the network.
3) Forwarding (FWD): Processes outgoing packets according to the WISE flow table. This table stores the flow rules sent by the controller. It is divided into three sections: matching rule, action and statistics. The matching rules contain the fields to be compared in the packets and the matching value. The packet can be inspected in any of its bytes, not only the header as is the case with OpenFlow. Up to three matching rules can be added for each flow rule. When the packet meets the rules, the action is executed. This action can be of five types: Drop, Forward, Sleep, Modify or INNP and is counted in the statistics.

These layers are specifically designed to allow the SDN controller to modify its behavior according to the network requirements. SDN WISE includes a simple controller, which contains all the network information and the Southbound protocols to communicate with the nodes. To expand the possibilities of integration with wired networks, the Southbound protocols can be added to general purpose controllers such as the ONOS controller which, due to its highly modular structure, supports the development and integration of specific protocols and applications [22].

In [5] SDN WISE TSCH is proposed, which extends the performance of SDN WISE, to guarantee industrial level requirements with a highly deterministic system. To achieve this, in the medium access control layer, the CSMA protocol was replaced by the TSCH protocol. In addition, the OpenPath packet, which installed the routing rules at each node, was modified to integrate TSCH scheduling, resulting in the OpenPathTSCH packet, which allows rapid configuration of the network by flows, with a minimal increase in control traffic. The packet generation periods are synchronized according to the schedule received. To exploit this, a system was developed in the controller that integrates three application layer processes: the Traffic Manager, Routing process and the TSCH Scheduler shown in Fig. 1. This results in a logical separation of data flows, with a specific allocation of physical resources (cells) for each type of traffic, called slicing. This network slicing makes it possible to differentiate flows and guarantee different qualities of service. The details of how slicing works and the advantages it brings in terms of quality of service are explained in depth in [5].

B. LITERATURE REVIEW

The challenge of mobile nodes in IWSN has been addressed in multiple works, where the effect of mobile nodes in RPL with Orchestra [24] and TSCH minimal [25] has been evaluated. The results obtained show how the RPL control traffic increases, due to the disconnections of the mobile nodes. Each time a mobile node disconnects, it needs to initiate a discovery process to obtain the new parent node, which takes a convergence time. Moreover, in this case the nodes depend on the timers to know that they are disconnected and start a new discovery process. This also increases packet loss and latency.

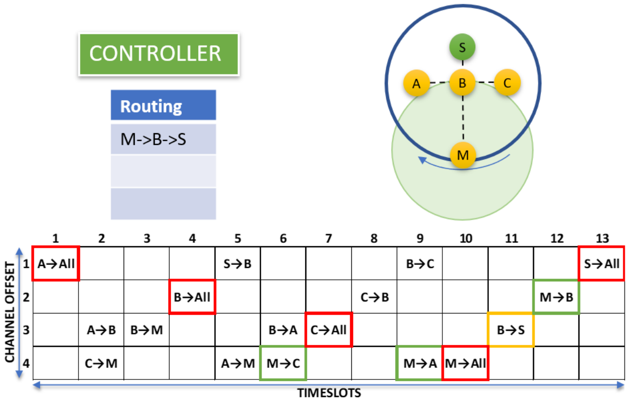
An attempt to overcome this difficulty was made in [26] with the use of timers that allow the parent nodes that leave the coverage area to be quickly updated. Also, different techniques have been tested to optimize the routing process, such as segmenting the network into circular areas defined by the Directed Acyclic Graph (DAG) roots [27] or using Kalman filters to improve RPL decisions with position predictions [28]. In addition, distributed scheduling functions have been improved, including prediction of position, queue size and packet generation rate to negotiate the number of cells required [29]. This negotiation process has limitations in mobility environments, when a parent change occurs the node needs time to discover the new neighbours and start the negotiation process. This can be reduced by reserving dedicated cells for negotiation and adding multiple control packets to detect topology changes [30]. However, although they represent an improvement over traditional RPL, they are far from offering the same determinism as fixed nodes.

These previous studies show how the support of mobile nodes considerably increases the complexity of decisions in the nodes and the amount of signalling packets. Therefore, the implementation of Software Defined Networks (SDN) in IWSNs is becoming a better option, as it assumes an increase in control traffic in exchange for additional advantages, such as: reducing the complexity of processes at the nodes and a global view of the network elements. The combination of SDN and Time Slotted Channel Hopping (TSCH) increases the reliability, determinism and flexibility of the network. This is demonstrated in [5], where it is possible to guarantee different QoS parameters for different types of flow dynamically and with low complexity compared to RPL with Orchestra.

To support nodes that are moving using SDN over TSCH [31] propose FTS-SDN. The objective is to increase the frequency and reliability in the exchange of control packets. In order to decrease the topology update time, the controller will send the new routing rules with each parent node change. To adapt to the changes, the controller must update the topology each time the mobile node changes its parent node. In this case, the nodes must report their existence through beacon messages that are broadcast to be received by all nodes in the coverage range. In addition, they must send the information of the received beacons to the controller. This is done by sending report packets, which include all the nodes from which beacon messages have been received. Therefore, the greater the number of control packets, the more updated the topology will be, and the faster it will adapt to changes in the mobile node. In fact, the speed at which the mobile node moves is a parameter that can affect this approach. Each time the controller updates the topology, it must generate the routes and inform the nodes of the new route with an Open-Path packet. In FTS-SDN the TSCH scheduling is fixed and pre-configured, and is performed according to the following rules:

- Minimum 1 uplink cell for each node.
- Minimum 1 downlink cell for each node.
- 1 broadcast cell for each node (Shared Slot).

This solution is highly dependent on control traffic. Higher frequency and reliability of this traffic increases the performance of the protocol. Therefore, in order to guarantee the forwarding of control traffic, a particular case of shared slots is used, where a transmission shared slot is assigned exclusively to each node, eliminating the probability of collision. This means that in each shared slot only one node will be transmitting and all the others will be receiving. In these timeslots it is not possible to plan parallel communications. The result is a significant increase in slotframe length and occupancy, which is reflected in an increase in energy consumption, a reduction in data bandwidth and more complex network management due to the need for longer slotframes.



	1	2	3	4	5	6	7	8	9	10	11	12	13
1	A→All				S→B				B→C				S→All
2				B→All				C→B				M→B	
3		A→B	B→M			B→A	C→All				B→S		
4		C→M			A→M	M→C			M→A	M→All			

FIGURE 2. Topology and slotframe used in [31].

For example, Fig. 2 shows the first scenario used in [31], a simple topology with 5 nodes (Sink or S, A, B, C and the mobile M). The trajectory of M is circular (blue) and the only node that can communicate with the Sink is B. The scheduling [31] consists of a slotframe composed of 13 timeslots and 4 channels. The controller has assigned only the route M->B->S, according to the current coverage range. In this scheduling shown in Fig. 2, 5 shared slots can be seen, 1 for each node (red). In addition, multiple inputs from the Mobile node (M) at different points of the slotframe (6, 9, 12; green) and a single arrival timeslot at the Sink node (11; yellow). Therefore, this solution uses 5 shared slots in a slotframe of 13. The end-to-end delay varies depending on the input node. It will be 150 ms, 120 ms and 50 ms for inputs A, B and C respectively. These differences are due to the distribution of the slots in the slotframe made in [31], which for inputs from nodes A and B need to wait until the next slotframe to complete the transmission.

III. PROPOSED METHODOLOGY: MOBILE MULTICAST FORWARDING SDN

This paper proposes Mobile Multicast Forwarding SDN (MMF-SDN), which exploits the advantages of SDN WISE and TSCH, to improve support for mobile nodes and bring them closer to determinism levels that are similar to those of fixed nodes. MMF-SDN allows a parent node change to be performed instantaneously and transparently. Because of this, it has a low dependency on control traffic, so there are no current limitations due to parent switching or saturation with control traffic.

Due to the global knowledge of the network in SDN, the extensive resources available at the controller and the synchronization obtained by the use of the TSCH MAC layer, mobile nodes can have a high level of determinism.

MMF-SDN involves managing the mobile nodes as a multicast source, meaning that the mobile node will have multiple receivers for the same packet. These receiving states for the mobile node will be synchronized in the same TSCH scheduling cell for all its parent nodes. Therefore, factors affecting the topology such as the position and velocity of the mobile node will be less relevant. Thus, it will be possible to increase the beacon and report periods, which reduces control traffic and simplifies the management of the mobile nodes. To achieve this, the report packets coming from a mobile node must be differentiated from the report packets from the fixed nodes. Thus, when the controller receives a report from a mobile node it will update the topology, but will not make changes to the routing or TSCH scheduling. Therefore, the management of these processes is carried out by implementing a new application *Mobile Node Manager* which, being located in the application layer, has direct communication with the SDN controller as shown in Fig. 1. This application will be in charge of storing the different parent nodes of a mobile, determining which nodes should be multicast receivers and requesting that the controller install the necessary resources (Routing + TSCH) for these nodes. The resources assigned by request of *Mobile Node Manager* are accumulated. This means that, when assigning resources to the new parent, the resources assigned to the previous parents are maintained. One of the installed resources is the multicast receiver cell, which is the same in all the parent nodes of a mobile node, so that the number of multicast receivers increases with each resource allocation. With this approach, it is possible to have all the inputs of the mobile node scheduled, without generating an overload of frequent updates and configurations. Therefore, it is possible to have stable behaviour with mobile nodes, after an initial discovery phase. This is the time it takes for the controller to know all the entries of a mobile node and to allocate the routing and TSCH scheduling resources. The duration of this phase depends on the length, trajectory and movement speed of the mobile node. In a defined industrial environment, it corresponds to the first displacement of the mobile node through the complete trajectory.

Since the topology construction in SDN WISE is done through the information arriving from the nodes, the update rate also depends on the capacity and reliability of the network to send the control traffic. Since in FTS-SDN the route corresponding to each change in the topology is sent, it is necessary to ensure high reliability in the control packets. For this reason, MMF-SDN seeks to reduce the dependency on control traffic through an accumulated multicast resource allocation.

This reduced dependency on control traffic increases robustness, which allows the frequency of such traffic to be reduced. Therefore, in MMF-SDN it is possible to use the default operation of shared slots, where nodes can transmit in any of the shared slots if they have packets in queue or, in the opposite case, are in receive state. This makes it possible to reduce the number of shared slots that are assigned. However, there is now one medium per contention, which increases the probability of collision. To reduce the number of collisions, the number of shared slots allocated must be optimized, which depends directly on the amount of control traffic generated, which makes it necessary to perform an analysis of this traffic.

A. ANALYSIS OF CONTROL TRAFFIC

Control traffic is essential for the operation of an SDN, since it is used by the controller to build the topology and send the flow rules to the nodes. In SDN WISE TSCH [5], this traffic is composed of periodic packets such as beacon and report, and aperiodic packets which are the configuration packets (OpenPathTSCH). The total amount of control traffic depends mainly on the generation period of the beacon and report packets. With a higher generation frequency, a more up-to-date network is obtained, allowing it to be more adaptable to changes. However, this type of traffic is a considerable overhead for the network and can affect data flows. Therefore, measures must be taken to minimize the impact on the network. In MMF-SDN the measures used are:

- Reduce: Optimize beacon (Bp) and report periods (Rp)
- Limit: Send this type of traffic only through the shared slots.
- Optimize: Allocate an optimal number of shared slots to maintain control traffic reliability.

TABLE 1. Notations.

Notation	Definition	Units
C_{pk}	Control packets	packets
C_{BW}	Control Bandwidth	packets
Sh_{SF}	Shared slots	timeslots
SF_{sz}	Slotframe size	timeslots
H_i	Hops between Sink and node i	Hops
Ts	Timeslot	ms
T	Evaluation Period	s
B_p	Beacon Period	s
R_p	Report Period	s

In Table 1 the notation used is shown. The total amount of control traffic (C_{pk}) for SDN WISE TSCH over a period of time T seconds is determined by (2), where n is the number of nodes in the network; Bp and Rp are beacon and report periods; and H_i distance in hops to the Sink from each node i. The first term of (2), $\frac{T}{Bp}$, is the number of beacon packets generated by a node. These packets are not forwarded by the nodes, so there is no factor that depends on the distance to the Sink, unlike the second term, which are the report packets that must pass through multiple nodes to reach the controller.

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

Because control packets will be sent through the shared slots, the number of shared slots allocated per slotframe must be optimized. A low number of shared slots will create a high collision rate and prevent the controller from updating the topology. Conversely, a very high number of shared slots

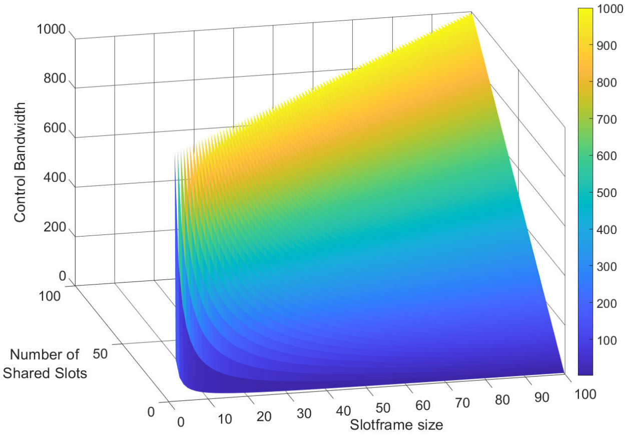


FIGURE 3. Effect of occupancy ratio on the amount of control traffic.

will increase power consumption and reduce the bandwidth available for data. To calculate the Control Bandwidth (C_{BW}) (3) is used, where the bandwidth is obtained as a function of the number of shared slots in the slotframe (Sh_{SF}) and the slotframe size (SF_{sz}) in a period of T seconds, taking into account the duration of the timeslot Ts. The first term $\frac{Sh_{SF}}{SF_{sz}}$, is the ratio of slotframe occupancy to shared slots. An increase in the number of shared slots means a higher occupancy and a greater bandwidth for control. The same bandwidth can be achieved with different slotframe lengths while maintaining the occupancy ratio.

$$C_{BW} = \frac{Sh_{SF}}{SF_{sz}} \cdot \frac{T}{Ts} \tag{3}$$

The optimum number of shared slots is obtained from (4), a variation of (3) which guarantees the minimum value of the control bandwidth. By replacing the total number of control packets obtained in (2) and the slotframe parameters, a whole number that corresponds to the number of shared slots is obtained. Fig. 3 represents the amount of control traffic that can be sent, varying the length of the slotframe and the number of shared slots assigned.

$$Sh_{SF} = \left\lceil C_{pk} \cdot SF_{sz} \cdot \frac{Ts}{T} \right\rceil \tag{4}$$

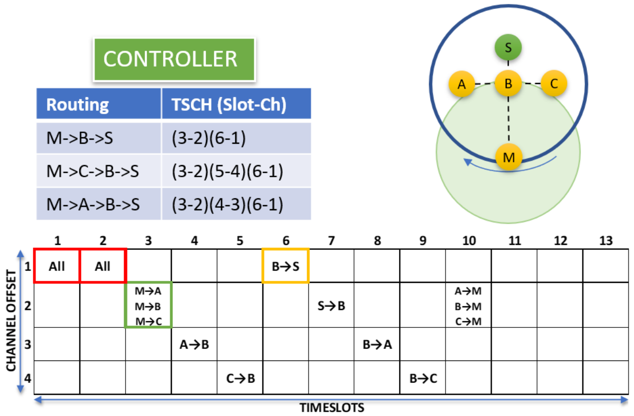
B. SCHEDULE ALLOCATION

In MMF-SDN the scheduling is flow-based, meaning that cells are mapped from source to destination and the result is sent for each flow through a single OpenPathTSCH packet. This allows more control over the cells that are assigned, by being able to assign or delete specific flows, instead of having to modify the entire schedule of a node. In MMF-SDN, multicast cells have been included, which contain multiple receivers in the same cell for the data packets sent by the mobile node. For this, a global multicast address has been added to the nodes, which is used by the mobile nodes to transmit their data packets and by the receivers to receive in the same cell. Thus, the transmission of mobile nodes is not made to a specific node and a dedicated queue is guaranteed for multicast traffic, which increases determinism. Another difference with respect to TSCH scheduling in FTS-SDN is the way in which the number of shared slots is allocated. In MMF-SDN, an optimal number of shared slots is used, which is obtained according to (4). This reduces the total number of shared slots, but increases the number of shared slots in which a node can transmit. This increase is due to the default operation of the shared slots, where nodes can transmit in any of them if they have packets in queue. Thus, as each node has more transmission opportunities, the bandwidth for control traffic per node increases, even though the overall bandwidth is reduced. This operation allows the nodes to dynamically adapt to the traffic needs of the network. For example, the nodes farther away from the sink will only send their beacon and report packets. Therefore, they perform fewer transmissions than the nodes closer to the sink, which must forward the control traffic received from other nodes. The allocation of dedicated slots can congest the central nodes, unlike the optimized allocation where the maximum traffic that can be generated is taken into account and allows each node to access the amount of resources it requires. For this reason, the scheduling of the dedicated cells was calculated based on the scheduler used in [5]. This scheduler uses the routes calculated by Dijkstra's algorithm in the routing manager and allocates the necessary cells for each hop in the path. The algorithms presented in [5] have a computational complexity of quadratic order $O(n^2)$, where n is the number of nodes. This has little impact on the proposed MMF-SDN, as we are dealing with a typical industrial system as defined in IEC 62264 (ANSI/ISA 95), where the number of nodes is generally limited between 10 and 30.

The example of the Fig. 2 generates 85 control packets (C_{pk}) every 10 seconds, using beacon and report periods of 1 and 2 seconds respectively. With this, the number of shared slots obtained with (4) is $\lceil 1.1 \rceil$, and therefore 2 shared slots are used for a slotframe length of 13 timeslots.

With the MMF-SDN approach, there are 2 shared slots instead of 5, as shown in Fig. 4. In addition, in MMF-SDN the behaviour of the mobile nodes is stabilized after the initial discovery phase, where the controller knows all the input nodes used by the mobile node and has the routing and TSCH resources assigned for all the routes. In this schedule, there is no variation of the end-to-end delay, this will be the minimum (30 ms) regardless of the input node. The number of allocated cells is reduced from 17 to 10 cells (3 shared slots and 4 dedicated cells) compared to FTS-SDN. Therefore, the slotframe occupancy goes from 61% to 31%. This 30% reduction in occupancy is largely (23%) due to the reduction in shared slots.

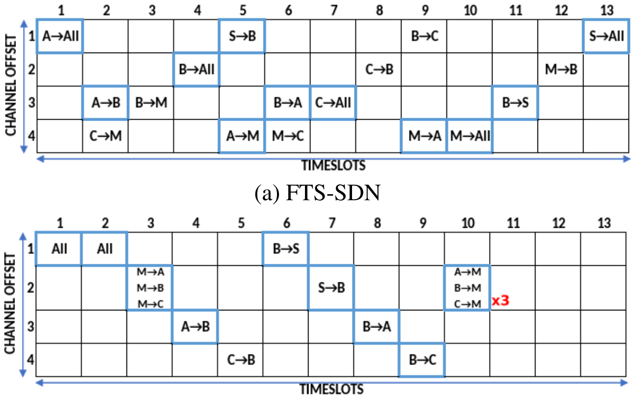
The energy consumption can be estimated using the number of transmitting and receiving states at each node. For the transmitter nodes, the worst case is used, which corresponds



Routing	TSCH (Slot-Ch)
M->B->S	(3-2)(6-1)
M->C->B->S	(3-2)(5-4)(6-1)
M->A->B->S	(3-2)(4-3)(6-1)

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	All	All				B->S							
2			M->A M->B M->C				S->B			A->M B->M C->M			
3				A->B				B->A					
4					C->B				B->C				

FIGURE 4. Topology and slotframe using MMF-SDN.



	1	2	3	4	5	6	7	8	9	10	11	12	13
1	A→All				S→B				B→C				S→All
2				B→All				C→B				M→B	
3		A→B	B→M			B→A	C→All				B→S		
4		C→M			A→M	M→C			M→A	M→All			

(a) FTS-SDN

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	All	All				B→S							
2			M→A M→B M→C				S→B			A→M B→M C→M	x3		
3				A→B				B→A					
4					C→B				B→C				

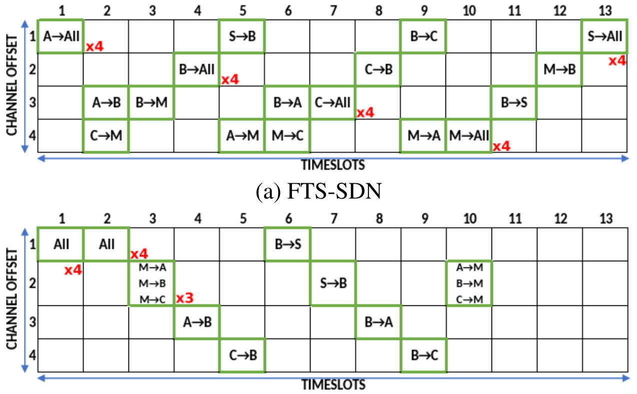
(b) MMF-SDN

FIGURE 5. Number of transmission states.

to the longest route (M->A->B->S). Fig. 5 shows in blue all cells used to transmit in the FTS-SDN slotframe, Fig. 5 (a) and for MMF-SDN, Fig. 5 (b). In both cases there are a total of 11 transmission states per slotframe.

In reception, all the scheduled slots are taken into account, since, unlike transmission, the nodes must listen whenever it is planned. In addition, in the shared slots, all the nodes except the transmitting node (n-1) will listen, so each shared slot will be multiplied by 4. In Fig. 6 in total there are 32 receive states per slotframe for case (a) and 18 for case (b). If the shared slots are omitted, there would be 12 and 10 receive states. In this case, a 40% reduction in power consumption in reception is achieved. Most of this is due to the reduction in the number of shared slots, since it is a factor that means additional consumption in all nodes.

The analysis of control traffic and the use of multicast cells illustrate that FTS-SDN can be improved in terms of quality of service (delay, jitter), energy consumption and scalability (slotframe length and occupancy). These aspects were not targeted by [31] which instead focused on having a low packet loss at the mobile node through a very high update rate. This means allocating a very high number of shared slots to ensure that all control packets reach the controller. The authors of FTS-SDN proposed in [31] the use of clustering methods to overcome this limitation. Thus, the objective of MMF-SDN is to limit the constant topology updates due to the movement of the mobile node. In addition, it is intended to provide the same deterministic behaviour that fixed nodes have. In MMF-SDN increasing the size of the slotframe does not lead to an increase in delays, since the repetition of flows within the slotframe is implemented. Improvements will be observed in the following aspects:



	1	2	3	4	5	6	7	8	9	10	11	12	13
1	A→All	x4			S→B				B→C				S→All
2				B→All	x4			C→B				M→B	x4
3		A→B	B→M			B→A	C→All	x4			B→S		
4		C→M			A→M	M→C			M→A	M→All	x4		

(a) FTS-SDN

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	All	All	x4			B→S							
2		x4	M→A M→B M→C	x3			S→B			A→M B→M C→M			
3				A→B				B→A					
4					C→B				B→C				

(b) MMF-SDN

FIGURE 6. Number of reception states.

1) End-to-End delay: Lower and deterministic.
2) Control traffic independent of mobile topology: The amount of control traffic does not depend on the number of mobile nodes or their speed, since it is not based on constant updates.
3) Reduced control traffic: Scheduling and routing changes are sent by flow with a single packet to all nodes in the path, instead of a specific schedule for each node.
4) Simplified scheduling: Since it does not depend on control traffic, it is not necessary to assign a shared slot per node.
5) Greater scalability: Due to the reduction in slotframe length and the ease of modifying and adding slots.
6) Lower energy consumption: Due to a lower number of shared slots.

IV. PERFORMANCE EVALUATION

This section shows the experimental results for three scenarios using the COOJA simulator under the Contiki NG operating system. The radio medium is setting as Multi-path Ray-tracer (MRM) medium, with the same parameters used in FTS-SDN $n = 2.04$ and $\sigma = 6.7$. The hardware used for simulation is a intel core i7 with 16 GB of RAM and the nodes are modeled as OpenMote nodes. A summary of the general parameters used in the simulations is shown in Table 2.

The SDN controller used is the SDN WISE controller connected with the sink through a serial socket. This controller was modified to handle MAC TSCH and include the Mobile Node Manager, Traffic Manager, Routing process and the

TABLE 2. Summary of parameters used in Cooja.

Parameter	Value
Radio Medium	MRM ($n = 2.04$, $\sigma = 6.7$)
Number of Channels	4 Channels
Timeslot (Ts)	10 ms

TSCH Scheduler. The first two scenarios are those used in FTS-SDN [31]. As a more representative proposal for an industrial environment, a third scenario has been used and this is shown in Fig. 16. Results are expressed in terms of power consumption, end-to-end delay and network overhead. In the experiments, simulations have been carried out with the values used in [31], sending 500 data packets from the mobile node to the sink, sending frequencies of one packet per second and one packet every three seconds. In all three cases, these parameters are sufficient to observe the behaviour of these approaches.

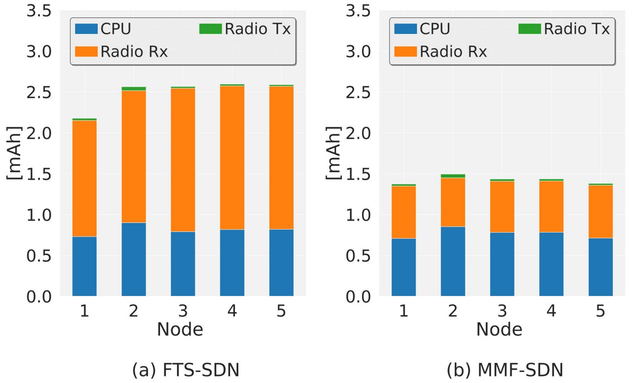
A. SCENARIO 1

This is the simplest scenario, the topology of which is shown in Fig. 2 [31]. The mobile node sends a packet to the sink every second. It moves at different speeds and follows a circular trajectory of 16 m radius. Node B is located at the centre of this trajectory, so that it is always within the coverage area of node M. A summary of the parameters used is shown in Table 3.

TABLE 3. Parameters used in scenario 1.

Parameter	Value
Number of Packets	500
Sending frequency	1 packet/s
Speed	1 m/s, 2 m/s, 4 m/s
Nodes	4 fixed and 1 mobile
Beacon Period (B_p)	1 s, 4 s
Report Period (R_p)	2 s, 8 s

The energy consumption depends on the states contained in the TSCH scheduling. In Fig. 5 and 6 the number of slots used in transmission and reception for FTS-SDN and MMF-SDN is shown. Transmissions do not represent an additional power consumption, since if a node has no data to transmit, the radio is not turned on. However, on receptions the radio must be turned on for a minimum of 2.2 ms, the default time to determine that there is no traffic to receive. When in a shared slot no node has data to transmit, all nodes will be in receiving state during this 2.2 ms, which means additional power consumption. For the worst case, where all nodes are receiving, the proposed MMF-SDN schedule will have 5 nodes receiving in each shared slot. In the case of FTS-SDN, 4 nodes will be in receiving state, since one of the nodes is defined in transmission. In total for control traffic a slotframe will have 10 receiving states for MMF-SDN scheduling and 20 for FTS-SDN scheduling. This means a 50% decrease in power consumption in reception in the worst case. To verify this, the simulation of the topology shown in Fig. 2 was carried out only with control traffic for the two schedules using $B_p = 1$ s and $R_p = 2$ s. The simulation was run for one hour, time enough time to observe the long-term network behavior when collisions effects and buffer issues affect performance. The result is shown in Fig. 7, where the power consumption at each node is shown according to CPU, Radio Tx and Radio Rx usage for the same topology. In Fig. 7 (a) for FTS-SDN 5 shared slots are used, the consumption in this case is about 2.5 mAh for all nodes and most of this energy is used in reception. In Fig. 7 (b) MMF-SDN uses 2 shared slots, the total consumption is below 1.5 mAh for all nodes and there is a clear reduction in the energy used for reception. This reduction is 62%, and is produced when FTS-SDN has a consumption of 1.66 mAh and MMF-SDN of 0.63 mAh as is shown in Fig. 7 (a) and in Fig. 7 (b) respectively. There are no considerable differences in the CPU and Radio Tx values.



(a) FTS-SDN

(b) MMF-SDN

FIGURE 7. Energy consumption Scenario 1.

The operation of shared slots in FTS-SDN is very specific, due to the fact that there is only one transmitter node for each shared slot. This requires an additional step to configure the corresponding shared slots on each node. With this method, transmission is assured, unlike MMF-SDN where multiple nodes can transmit in the same shared slot, which may cause collisions. Fig. 8 shows the number of control packets sent during one hour of simulation using $B_p = 1$ s and $R_p = 2$ s in the two protocols. The SDN WISE beacon values are slightly lower in MMF-SDN and the number of report packets is higher and does not exactly match the number of ACKs. This is due to collisions in the shared slots, which affect all control packets. Since the beacon packets are sent in broadcast, no retransmissions are made, unlike the report packets which, as they are unicast, are retransmitted when a collision is detected. For this reason, in MMF-SDN there are 17% more reports than ACKs. In the case of FTS-SDN, the lost report packets are due to changes in the parent of the mobile node and correspond to 5% of the report packets.

The collisions that occur in the shared slots do not have a considerable impact on MMF-SDN performance, since the MMF-SDN does not require an immediate topology update. This reduced dependency on control packets also allows

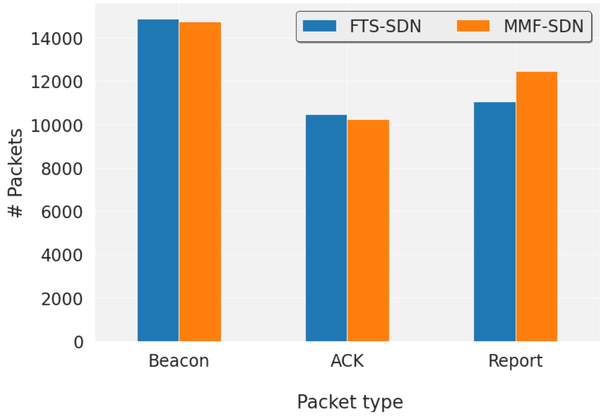


FIGURE 8. Number of control packets sent during 1 hour of simulation.

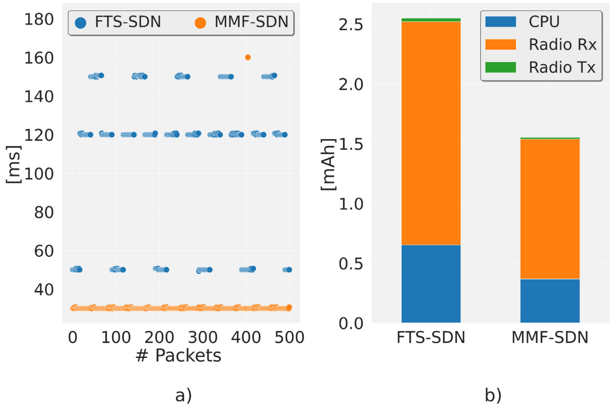


FIGURE 9. (a) End-to-End delay FTS-SDN and MMF-SDN. (b) Average energy consumption.

reduction in the frequency of beacon and report packets without affecting performance. Fig. 9 shows the end-to-end delay when the speed of movement of the mobile node is 1 m/s, and the schedules in Fig. 5 are used. In Fig. 9 (a) the end-to-end delay produced by the FTS-SDN algorithm is observed with $B_p = 1$ s and $R_p = 2$ s and by the MMF-SDN algorithm with $B_p = 4$ s and $R_p = 8$ s. In this case, the two protocols have the expected performance according to TSCH scheduling and a PDR of 100%. However, the optimization of the TSCH scheduling used in MMF-SDN reduces the average end-to-end delay by 70%, from 106 ms in FTS-SDN to 30 ms. The beacon and report periods used in MMF-SDN are 4 times higher than those used in FTS-SDN, so that the same PDR is obtained with a smaller number of control packets.

The energy consumption is plotted in Fig. 9 (b). At reception the power consumption has been reduced by 50% compared to that of FTS-SDN, since the number of shared slots is lower. Nodes will spend more time in reception, because they must forward a smaller amount of control traffic. Fewer transmissions reduce the processing load, which requires less CPU consumption. This proves that it is possible to use a smaller amount of control traffic and that the loss of this type of packet does not lead to a loss of performance.

Adapting to topology changes produced by a mobile node is complex, since the speed and trajectory of the node may change. Therefore, the effect of speed on MMF-SDN and FTS-SDN performance has been analyzed. A high dependency on control traffic means that a failure can occur at any time, unlike the cumulative resource allocation, where stable behaviour is achieved by keeping the routes assigned, which limits the amount of configurations to be sent by the controller. Fig. 10 shows the end-to-end delay obtained with different travel speeds for the mobile node. In 0 the mobile moves at a speed of 1 m/s, and both FTS-SDN and MMF-SDN adapt to the movement without any loss. When increasing the speed to 2 m/s, Fig. 10 (c) and (d), the FTS-SDN protocol has a high performance in the transmission of the first packets, but after 100 packets it starts to have losses. This is because the routes start to become out of date due to the accumulation of queued packets, unlike MMF-SDN, which has losses in the first parent change and then remains stable. With a high movement speed, 4 m/s, the controller does not manage to update the routes fast enough and in the case of FTS-SDN Fig. 10 (e), constant losses occur (around 20%). In contrast, with MMF-SDN, Fig. 10 (f) the configuration of the routes only affects the first packets. After this, the behaviour is totally stable as no additional configurations are required, allowing better performance in permanent regime.

Fig. 11 summarizes the behaviour of the two protocols for the three travel speeds of the mobile node. In Fig. 11 (a) the PDR for FTS-SDN remains at 100% for a velocity of 1 m/s. For travel speeds of 2 m/s and 4 m/s it is 98.6% and 76.6%. However, the PDR starts to decrease steadily. Because the controller cannot adapt to rapid changes in the topology, it is not possible to achieve a stable system. Therefore, the speeds of 2 m/s and 4 m/s have a negative trend, the higher the speed the more noticeable this effect. Fig. 11 (b) shows the PDR for the MMF-SDN protocol, at the end of sending the 500 packets there is a PDR of 100%, 99.8% and 97% respectively. With the accumulated resource allocation, there are losses and delays in the initial discovery phase while the controller configures the inputs, but after this allocation the throughput remains stable. Therefore, in all cases the trend is positive.

B. SCENARIO 2

This section shows the results after assessing the scenario of Fig. 12 [31]. It is composed of 43 nodes, 3 of which are mobile (Nodes 41, 42 and 43), and which send packets to the sink every 3 seconds, as they move with a circular trajectory at 1 m/s. Unlike scenario 1, where the mobile node had a fixed distance to one of its parents (B), this scenario has constant changes of parent node and a larger distance to the sink (Node 1). Therefore, the impact of control traffic will be more significant. Table 4 shows the parameters used for this scenario.

The effect of the beacon and report periods on the amount of control traffic is shown in Fig. 13. A topology with

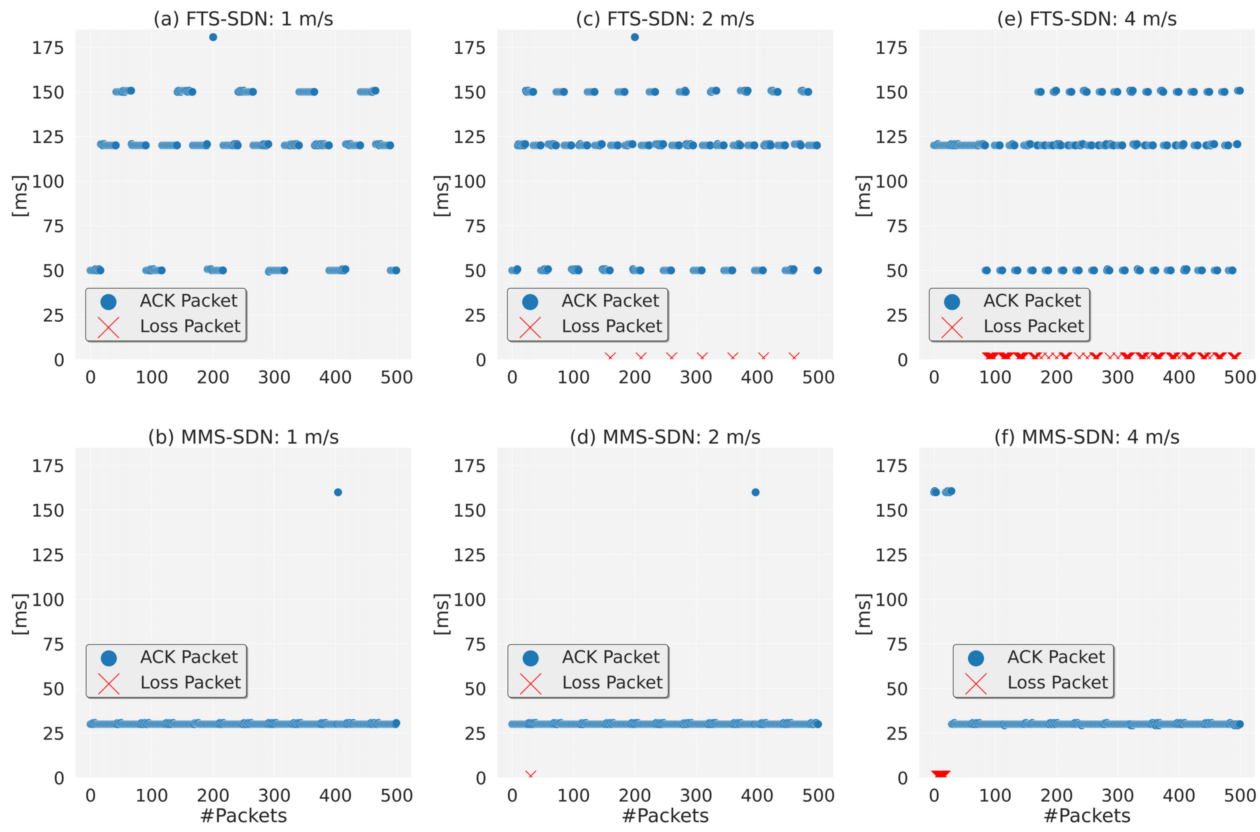
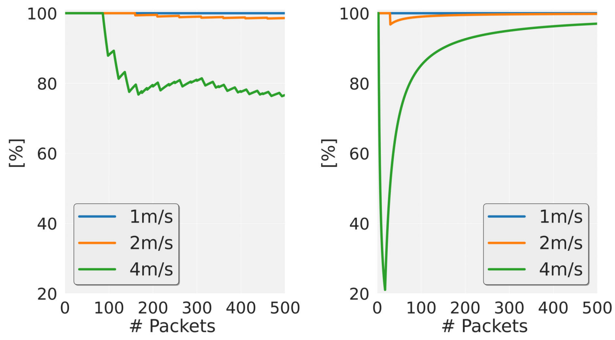


FIGURE 10. MMF-SDN and FTS-SDN comparison at different mobile node speeds.



(a) FTS-SDN

(b) MMF-SDN

FIGURE 11. PDR for both protocols with different speeds of the mobile node.

TABLE 4. Parameters used in scenario 2.

Parameter	Value
Number of Packets	500
Sending frequency	1 packet/3s
Speed	1 m/s
Nodes	40 fixed and 3 mobile
Beacon Period (B_p)	3 s, 5 s
Report Period (R_p)	6 s, 10 s

n nodes and a maximum of 4 hops to the sink has been used. The number of nodes in each hop is distributed in a similar way to that shown in Fig. 12. The dashed line represents the bandwidth limit, which is achieved by having all slots as shared according to (3). Therefore, the closer the control traffic line is to the bandwidth limit, the higher the occupancy rate of the slotframe with shared slots and the lower the capacity of the network to send data flows. The FTS-SDN solution needs a high update frequency, so the authors use $B_p = 1$ s and $R_p = 2$ s. These parameters correspond to the line (Blue) that approaches the bandwidth limit the fastest. The proposed MMF-SDN is based on a cumulative resource allocation, so it is not necessary to update with such a low period to determine the current parent of the mobile node. This allows a reduction the control traffic by using longer beacon and report periods, where the control traffic is reduced from 50-80% for a configuration of $B_p = 5$ s and $R_p = 10$ s. This reduction in the amount of control traffic also allows the network to be more scalable. Therefore, it can be seen that the limit for the values of B_p and R_p used by FTS-SDN is close to 40 nodes, this can be extended by reducing the amount of control traffic, with higher B_p and R_p periods.

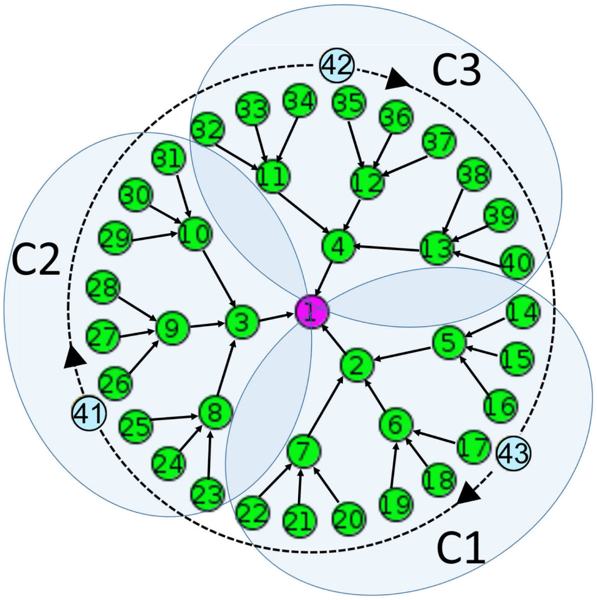


FIGURE 12. Topology with 43 nodes, 3 mobile nodes [31].

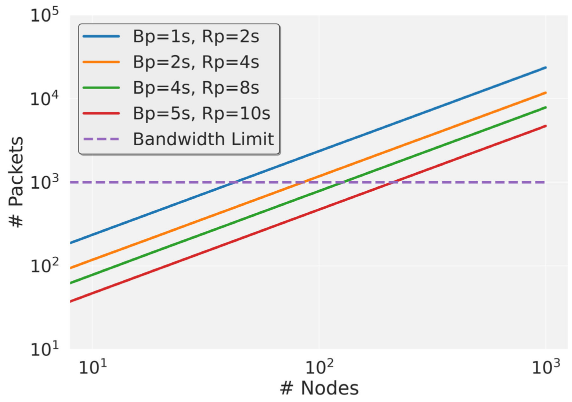


FIGURE 13. Total control packets as a function of the number of nodes for $T = 10$ s, with different beacon periods (B_p) and report periods (R_p).

In Fig. 14 the number of shared slots used by FTS-SDN as a function of the number of nodes are compared with the optimal number of shared slots used by MMF-SDN for 4 different configurations of B_p and R_p. In all cases, the amount of shared slots is much lower than the 1 shared slot per node rule used in FTS-SDN, since using a reduced number of shared slots allows a reduction in the slotframe size. For example, for 40 nodes FTS-SDN uses 40 shared slots and a slotframe of 53 timeslots while MMF-SDN can use 13 shared slots in a slotframe of 23 timeslots. This means an increase in bandwidth, since due to the cyclic operation of the slotframes, the distance between the shared slots will be smaller. Therefore, in MMF-SDN, schedules with better utilization of network resources are obtained, even in conditions with a high update frequency. Hence, increasing the number of nodes also increases the performance difference, since the number of shared slots allocated in FTS-SDN does not adapt to the bandwidth requirements of the network. For example, the topology shown in Fig. 12 [31], is composed of 40 fixed nodes **(N)** and 3 mobile nodes **(M)** and a maximum of 4 hops. In the 4th hop there are 3 mobile nodes, in the 3rd hop there are 27 fixed nodes, in the 2nd hop 9 fixed nodes and in the first hop 3 fixed nodes. Therefore, for the fixed nodes the following are needed, ***N-1*** uplink cells and ***N-1*** downlink cells, which is equivalent to 39 in both cases. In addition, the mobile nodes need a link to each outside node. This is 27 cells per mobile node for uplink, plus another 27 cells per mobile node for downlink, i.e. 54 cells per mobile node. Therefore, the slotframe must have 162 cells assigned for the mobile nodes and 78 cells for the fixed nodes, i.e. a total of 240 cells. This value is defined in [31] as the *minimum number of unicast links*, which is equal to twice the number of fixed nodes multiplied by the number of mobile nodes ($2 \cdot N \cdot M$). For FTS-SDN, a slotframe of 91 timeslots and 4 channels was used. By removing the 43 timeslots that are assigned to the shared slots, 48 timeslots remain available. Using 4 channels there are 192 cells available in total, where it is not possible to place the required 240 cells. By using the MMF-SDN multicast approach, it is not necessary to use a different cell for each of the 27 receivers of the mobile node. Hence, the 162 cells used for the unicast links of the mobile nodes can be reduced to only 6, two per mobile node. Therefore, using the same parameters, the slotframe would go from 100% to 70% occupancy. When evaluating the amount of control traffic generated by the topology of Fig. 12, using (2) with $Bp = 1$ s and $Rp = 2$ s there are 1000 control packets in a period T equal to 10 seconds:

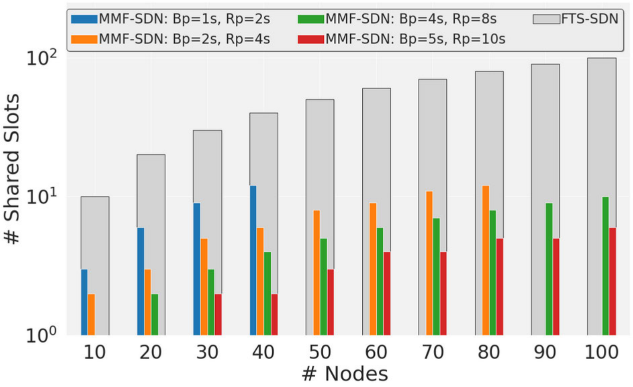
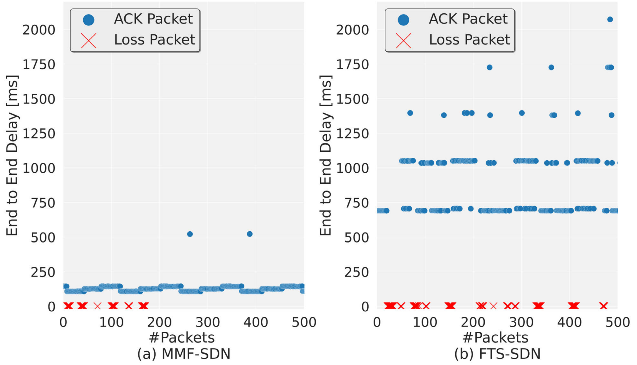


FIGURE 14. Number of shared slots to use depending on the control traffic.

$$C_{pk} = 43\frac{T}{Bp} + \frac{T}{Rp}((3 \cdot 4) + (27 \cdot 3) + (9 \cdot 2) + (3 \cdot 1)) = 1000$$

The bandwidth required to send this number of control packets is only achieved by using all the slots of the slotframe as shared slots, i.e., according to (3) $\frac{Sh_{SF}}{SF_{sz}} = 1$. Therefore, the beacon and report period must be increased. In this scenario, a slotframe with the same size as in the FTS-SDN [31] tests is used, 91 timeslots and 4 channels. The optimal number



(a) MMF-SDN

(b) FTS-SDN

FIGURE 15. End-to-End delay Topology with 43 nodes.

of shared slots for this slotframe length is 30 timeslots, using $B_p = 3$ s and $R_p = 6$ s.

Under these conditions MMF-SDN performs better in end-to-end delay and determinism, as can be seen from Fig. 15 (a). MMF-SDN can guarantee a delay below 200 ms in 99% of the packets sent during the initial discovery phase. This coincides with a full rotation of the mobile node, which for the speed and path length in this scenario is close to 200 packets. In the same scenario for FTS-SDN Fig. 15 (b), there is a high variability in the delays and losses during the whole simulation. The results obtained in [31] show delays greater than 1.5 s, and when this network is segmented into clusters, the authors manage to reduce these delays to around 500 ms. This significant performance difference is due to the fact that in FTS-SDN there is a saturation of the central nodes, since they must forward the control traffic of the leaf nodes, and having only one shared slot for each slotframe the bandwidth is reduced by increasing the length of the slotframe. During the simulation there is an average of 6 packets in the buffer of the central nodes (2, 3 and 4) for the FTS-SDN implementation with $Bp = 1$ s and $Rp = 2$ s. Therefore, there is a mismatch between the topology of the controller and the real one, which affects the reliability of the routes. For this reason, $B_p = 3$ s and $R_p = 6$ s are used where it is reduced to 2 average buffered packets for FTS-SDN and 1 packet for MMF-SDN. The latter manages to have a lower amount of buffered packets, because the nodes have a higher amount of available shared slots per slotframe.

C. SCENARIO 3

In this case, a typical scenario of the industrial sector is simulated, where there are two AGVs covering different routes within the plant. In total there are 27 nodes, 25 fixed and 2 mobile, all of them with a coverage range of 20 m. The mobile nodes are located in the AGVs that are in continuous movement at a speed of 1.5 m/s. These nodes send a data packet to the sink every second. As shown in Fig. 16 the mobile nodes move along the two central corridors, one in the horizontal direction (purple) for 64 m, and the other in the vertical direction (green), with a distance of 49 m. The sink is located at the top of the vertical central corridor. According to the coverage range, the traffic of the entire WSN reaches it through the 4 nodes located at the top of the central corridor. The parameters used are shown in Table 5. Compared to the two previous scenarios, the complexity is increased due to the displacement of the mobile nodes through the mobile of the industrial building used as a scenario, which produces changes in a greater number of nodes simultaneously.

TABLE 5. Parameters used in scenario 3.

Parameter	Value
Number of Packets	3600
Sending frequency	1 packet/s
Speed	1.5 m/s
Nodes	25 fixed and 2 mobile
Beacon Period (B_p)	1 s
Report Period (R_p)	2 s

Fig. 17 shows the end-to-end delays obtained using the two approaches during the start-up phase. Fig. 17 (a) shows how MMF-SDN achieves an end-to-end delay that slightly exceeds 300 ms for both vertical and horizontal movement. Between 50 and 150 packets there is a higher delay. This is due to the fact that in the fixed nodes there is only one dedicated link to send the data traffic to the sink. Therefore, the end-to-end delay increases when the routes of the two mobile nodes coincide and one of the packets must remain in queue. With MMF-SDN, packet loss occurs only during the initial discovery phase, when resources are not allocated. Fig. 17 (b) the results using the FTS-SDN implementation are shown. For horizontal movement the end-to-end delay can be kept below 200 ms, however, for vertical movement there are different levels of end-to-end delay up to a maximum of 800 ms. In FTS-SDN packet loss can occur at any time during the test. The delays and losses are the effect of the queuing that occurs.

The packet loss over the entire path is plotted in Fig. 18 (a). MMF-SDN has no losses in the vertical movement, it is in the horizontal movement where losses occur in the initial discovery phase, specifically during the sending of the first 50 packets. It is during this period that the first path traversal occurs and the losses are due to the delay in the allocation of resources. After this, a constant increase in the PDR is observed. For FTS-SDN the probability of losses is equal at any instant of time. This is due to the saturation of the nodes closest to the sink, which send the control traffic of all leaf nodes. When the bandwidth is exceeded, packets are buffered and sent when resources become available again. This increases the lag between the actual topology and the topology used by the controller to make decisions. Therefore, the rules sent by the controller to the mobile nodes will produce losses. Fig. 18 (b) shows the energy consumption of the two approaches, in this case the difference between the two is less than that obtained in scenarios 1 and 2. This is because in MMF-SDN due to the multicast nature, duplicate traffic is sent when two multicast receivers receive the packet from

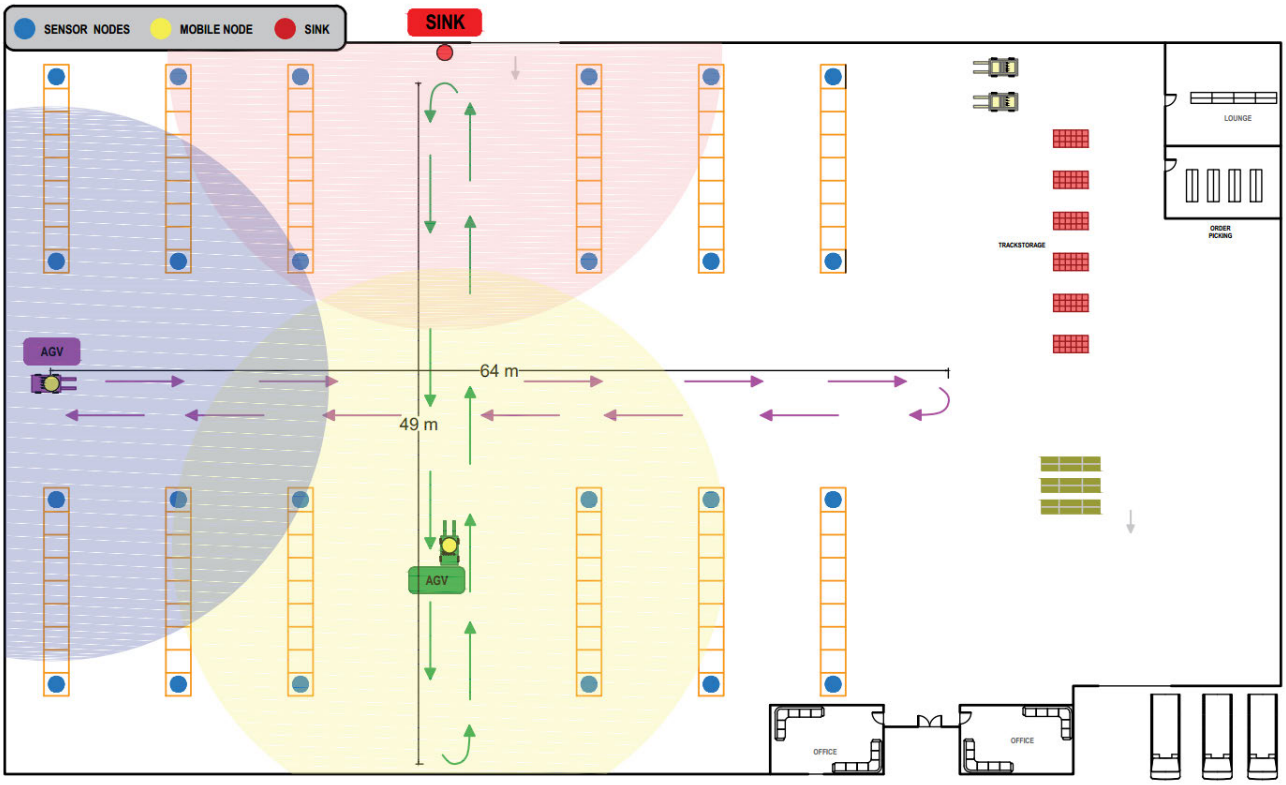
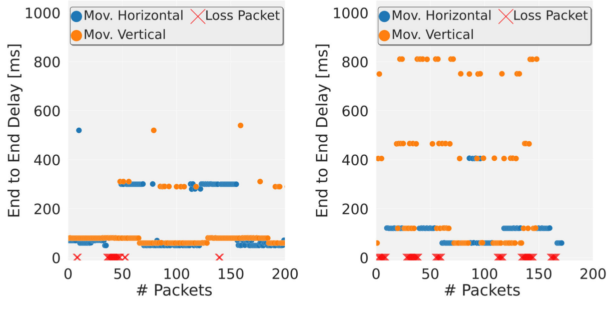


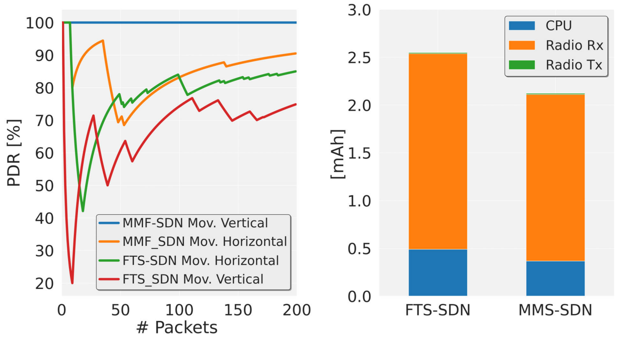
FIGURE 16. Scenario in an industrial environment with 25 fixed nodes and 2 mobile nodes.



(a) MMF-SDN

(b)FTS-SDN

FIGURE 17. End-to-End delay for horizontal and vertical movement of Scenario 3.



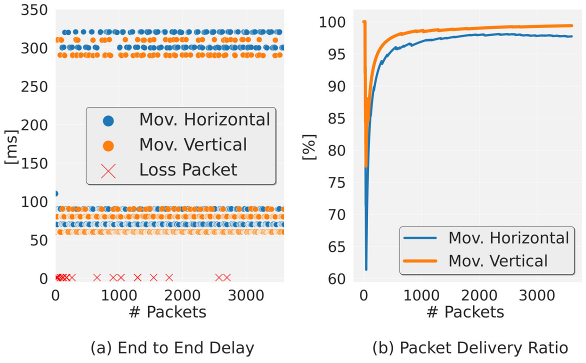
(a) Packet Delivery Ratio

(b) Power consumption

FIGURE 18. MMF-SDN and FTS-SDN comparison Scenario 3.

the mobile node. It is possible to limit the amount of duplicate packets with RSSI conditions on multicast packets to limit the reception area, using sequence numbers for a node to discard the repeated packets it receives or implementing applications in the controller that allow debugging on the routes on which duplicate packets are received. Even so, the increase in energy consumption produced by these packets is not enough to overcome the consumption produced in FTS-SDN.

To validate the effects of the stability obtained in MMF-SDN, a test was carried out for one hour, where a total of 3600 packets were sent. Fig. 19 (a) shows the end-to-end delay for the path types of scenario 3. Two groups are clearly observed, one below 100 ms and the other below 330 ms with a lower density. These are packets that were queued because the paths of the two mobile nodes coincided, therefore, the delay corresponds to one more slotframe. In each of the groupings there are 4 bands of dots separated by one timeslot (10 ms). These correspond to each of the nodes with direct connection to the sink. Fig. 19 (b) shows how in both cases there are initial packet losses. Afterwards, a stability is observed that allows a PDR higher than 95% to be obtained. The vertical movement has a lower amount of losses due to



(a) End to End Delay (b) Packet Delivery Ratio

FIGURE 19. Stability of MMF-SDN in Scenario 3.

the fact that in its displacement it can have two independent routes to the sink. By sending in multicast, the traffic arrives duplicated to the sink, which gives a PDR close to 100%.

V. CONCLUSION

The dynamism of industrial environments requires WSNs to be able to adapt quickly to physical and logical changes, while maintaining determinism and quality of service. For this reason, MMF-SDN, a software-defined WSN management protocol, is proposed to efficiently incorporate nodes in motion. It eliminates the difficulties caused by the lack of convergence in the routing protocol when the parent node changes, while maintaining strict QoS parameters. The lower dependency on control traffic that MMF-SDN has proven to have, allows this type of traffic to be reduced by 50% with respect to FTS-SDN. This and the optimized use of shared slots, reduces energy consumption in reception by up to 50%. The allocation of accumulated multicast resources used by MMF-SDN manages to stabilize the network with mobile nodes, unlike convergence-based approaches such as FTS-SDN, which must have a response from the controller to each change in the topology. The use of multicast transmission in the mobile node eliminates the difficulty of switching parents that occurs in FTS-SDN, such that switching parents does not produce an additional delay. This allows a 70% reduction in the end-to-end delay. In addition, it reduces the slotframe occupancy and length by more than 30%, since for each mobile node the same transmit and receive cells are used in all parent nodes. The collisions that occur in the shared slots increase the packet loss and the total amount of control traffic. However, they do not have an impact on MMF-SDN performance, because MMF-SDN does not need to quickly update the mobile position, because the resources remain allocated in each of the possible entries. The allocation of accumulated multicast resources requires an initial discovery phase. In industrial environments where the number of nodes and movement paths are usually operate in clear limits, there is high determinism after this phase, as resources remain allocated and there is an immediate and transparent change of parent node.

As future work, the use of Machine Learning applications in the controller will be considered, which will allow data flows to be analysed, in order to improve routing and quality of service. This application removes cells that are not used in the nodes or those that produce duplicate data flows. Furthermore, due to the multicast nature and synchronism obtained with TSCH, it is possible to implement applications that exploit the constructive interference that occurs when transmitting the same data packet from different sources. This allows the limitations of the initial discovery phase to be reduced due to a faster information exchange between the controller and the mobile nodes.

REFERENCES

[1] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the Internet of Things and industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017, doi: 10.1109/MIE.2017.2649104.

[2] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018, doi: 10.1109/TII.2018.2852491.

[3] B. Safaei, A. Mohammadsalehi, K. T. Khoosani, S. Zarbaf, A. M. H. Monazzah, F. Samie, L. Bauer, J. Henkel, and A. Ejlali, "Impacts of mobility models on RPL-based mobile IoT infrastructures: An evaluative comparison and survey," *IEEE Access*, vol. 8, pp. 167779–167829, 2020, doi: 10.1109/ACCESS.2020.3022793.

[4] 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, doi: 10.1109/ACCESS.2017.2666200.

[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, doi: 10.3390/s21041075.

[6] *IEEE Standard for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, Standard 802.15.4-2011, 2016. Accessed: Jan. 4, 2021. [Online]. Available: https://standards.ieee.org/standard/802_15_4-2020

[7] G. Alderisi, G. Patti, O. Mirabella, and L. L. Bello, "Simulative assessments of the IEEE 802.15.4e DSME and TSCH in realistic process automation scenarios," in *Proc. IEEE 13th Int. Conf. Ind. Informat. (INDIN)*, Jul. 2015, pp. 948–955, doi: 10.1109/INDIN.2015.7281863.

[8] H. Kurunathan, R. Severino, A. Koubaa, and E. Tovar, "IEEE 802.15.4e in a nutshell: Survey and performance evaluation," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 1989–2010, 3rd Quart., 2018, doi: 10.1109/COMST.2018.2800898.

[9] 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, doi: 10.3390/s18103556.

[10] 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 Networks (WoWMoM)*, Jun. 2013, pp. 1–6, doi: 10.1109/WoWMoM.2013.6583485.

[11] R. Soua, P. Minet, and E. Livolant, "Wave: A distributed scheduling algorithm for convergecast in IEEE 802.15.4e TSCH networks," *Trans. Emerg. Telecommun. Technol.*, vol. 27, no. 4, pp. 557–575, Apr. 2016, doi: 10.1002/ett.2991.

[12] 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, doi: 10.1109/jsen.2013.2266417.

[13] M. 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.*, Doha, Qatar, Apr. 2016, pp. 1–6, doi: 10.1109/WCNC.2016.7565002.

[14] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust mesh networks through autonomously scheduled TSCH," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2015, pp. 1–4, doi: 10.1145/2809695.2809714.

[15] S. Oh, D. Hwang, K.-H. Kim, and K. Kim, "Escalator: An autonomous scheduling scheme for convergecast in TSCH," *Sensors*, vol. 18, no. 4, p. 1209, 2018, doi: 10.3390/s18041209.

[16] S. Kim, H.-S. Kim, and C. Kim, "ALICE: Autonomous link-based cell scheduling for TSCH," in *Proc. 18th Int. Conf. Inf. Process. Sensor Netw.*, Apr. 2019, pp. 15–18.

[17] D. Kreutz, F. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015, doi: 10.1109/JPROC.2014.2371999.

[18] 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, doi: 10.1016/j.jnca.2016.12.019.

[19] S.-C. Lin, I. F. Akyildiz, P. Wang, and M. Luo, "QoS-aware adaptive routing in multi-layer hierarchical software defined networks: A reinforcement learning approach," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun. 2016, pp. 25–33, doi: 10.1109/SCC.2016.12.

[20] 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, doi: 10.1016/j.comnet.2014.06.002.

[21] CISCO. (2020). *2020 Global Networking Trends Report*. CISCO. Accessed: May 1, 2021. [Online]. Available: https://www.cisco.com/c/dam/m/enus/solutions/enterprise-networks/networking-report/files/GLBL-ENGNB-060NARPTPDFMOFU-no-NetworkingTrendsReport-NBrpten0186125.pdf

[22] 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, doi: 10.1016/j.comnet.2019.04.029.

[23] T. Luo, H.-P. Tan, and T. Q. S. Quek, "Sensor OpenFlow: Enabling software-defined wireless sensor networks," *Commun. Lett.*, vol. 16, no. 11, pp. 1896–1899, Nov. 2012, doi: 10.1109/lcomm.2012.092812.121712.

[24] C. Cobarzan, J. Montavont, and T. Noel, "Analysis and performance evaluation of RPL under mobility," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2014, pp. 1–6, doi: 10.1109/ISCC.2014.6912471.

[25] S. Raza, T. V. D. Lee, G. Exarchakos, and M. Gunes, "A reliability analysis of TSCH protocol in a mobile scenario," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–6, doi: 10.1109/CCNC.2019.8651793.

[26] Y. Al-Nidawi and A. H. Kemp, "Mobility aware framework for timeslotted channel hopping IEEE 802.15.4e sensor networks," *IEEE Sensors J.*, vol. 15, no. 12, pp. 7112–7125, Dec. 2015, doi: 10.1109/JSEN.2015.2472276.

[27] O. Gaddour, A. Koubaa, R. Rangarajan, O. Cheikhrouhou, E. Tovar, and M. Abid, "Co-RPL: RPL routing for mobile low power wireless sensor networks using corona mechanism," in *Proc. 9th IEEE Int. Symp. Ind. Embedded Syst. (SIES)*, Jun. 2014, pp. 200–209, doi: 10.1109/SIES.2014.6871205.

[28] M. Barcelo, A. Correa, J. L. Vicario, A. Morell, and X. Vilajosana, "Addressing mobility in RPL with position assisted metrics," *IEEE Sensors J.*, vol. 16, no. 7, pp. 2151–2161, Apr. 2016, doi: 10.1109/JSEN.2015.2500916.

[29] O. Tavallaie, J. Taheri, and A. Y. Zomaya, "Design and optimization of traffic-aware TSCH scheduling for mobile 6TiSCH networks," in *Proc. Int. Conf. Internet Things Design Implement.*, May 2021, pp. 234–246, doi: 10.1145/3450268.3453523.

[30] A. K. 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, no. 7, pp. 4535–4550, Oct. 2021, doi: 10.1007/s11276-021-02722-y.

[31] L. L. Bello, A. Lombardo, S. Milardo, G. Patti, and M. Reno, "Experimental assessments and analysis of an SDN framework to integrate mobility management in industrial wireless sensor networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5586–5595, Aug. 2020, doi: 10.1109/TII.2020.2963846.

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 the "Advanced Communications and Industrial Informatics" Group, Instituto Tecnológico de Informática (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) graduated in industrial electronics and computer engineering from the Universitat Politècnica de València (UPV), Valencia, Spain, in 1987 and 1993, respectively. He received the Ph.D. degree in telecommunications engineering from UPV, in 1998. He is currently an Associate Professor with the Department of Communications (UPV), where he teaches industrial communications and public access networks. He is also the Director of the "Advanced Communications and Industrial Informatics" Group, Instituto Tecnológico de Informática (ITI). He has managed more than 50 research and technological projects. Since 1996, he has authored or coauthored more than 60 technical papers in journals and international conferences. 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.

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 Universitat Politècnica of València as a part-time Associate Lecturer with the Department of Computer Architecture, having been promoted to an Adjunct Professor, in 2001. He was promoted to an Assistant Lecturer, in 2004, and has been working as an Associate Professor, since 2010. 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 review 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, with the University of Aveiro, Portugal, and at Anglia Ruskin University, Cambridge, England. 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 participated as a member of the organizing committee or as a member of the program committee of numerous international conferences of the IES-IEEE.

TERESA ALBERO-ALBERO received the Ph.D. degree in telecommunications engineering from the Universitat Politècnica de València (UPV), Valencia, Spain, in 2002 and 2011, respectively. In 2002, she joined the UPV as a Scholarship Holder and later as a Research Staff. In 2015, she joined as a part-time Adjunct Professor with the Department of Communications and currently belongs to the Department of Computer Architecture. She has teaching experience in the areas of computer architecture and technology, as well as industrial communications. She also belongs to the Instituto Tecnológico de Informática (ITI). Her current research interests include sensor networks, communication networks, image processing, and industrial networks.