

# LASA: Location-Aware Scheduling Algorithm in Industrial IoT Networks with Mobile Nodes

Marco Pettorali<sup>1</sup>, Francesca Righetti<sup>1</sup>, Carlo Vallati<sup>1</sup>, Sajal K. Das<sup>2</sup>, and Giuseppe Anastasi<sup>1</sup>

<sup>1</sup>Dept. of Information Engineering, University of Pisa, Pisa, Italy, {name.surname}@unipi.it

<sup>2</sup>Dept. of Computer Science, Missouri University Science and Technology, Rolla, Missouri, USA, sdas@mst.edu

**Abstract**—The *Synchronized Single-hop Multiple Gateway* (SHMG) is a framework recently proposed to support mobility into 6TiSCH, the standard network architecture defined for Industrial Internet of Things (IIoT) deployments. SHMG supports industrial applications with stringent requirements by adopting the *Shared-Downstream Dedicated-Upstream* (SD-DU) scheduling policy, which allocates to Mobile Nodes (MNs) a set of dedicated transmission opportunities for uplink data. Such allocation is performed on all the Border Routers (BRs) of the network without considering the location of MNs. Transmission opportunities are reserved also in BRs far from the current location of the MN, resulting in a waste of resources that limits the maximum number of nodes supported by the network. To overcome this problem, we propose a *Location-Aware Scheduling Algorithm* (LASA) that takes into account the position of MNs to build and maintain an efficient communication schedule. Specifically, LASA tries to prevent conflicts arising due to node mobility, in a preventive manner, so as to minimize packet dropping. We evaluate LASA via simulation experiments. Our results show that LASA allows to increase the number of MNs by more than four times, with respect to SD-DU, yet guaranteeing a Packet Delivery Ratio higher than 98%.

**Index Terms**—IIoT, 6TiSCH, Scheduling Function, Mobility, Location aware

## I. INTRODUCTION

The Industrial Internet of Things (IIoT) is expected to interconnect industrial objects in order to enable the IoT paradigm in industrial contexts characterized by stringent requirements in terms of Quality of Service (QoS). To foster this trend, the Internet Engineering Task Force (IETF) has standardized the 6TiSCH architecture [1], an inter-operable IPv6-based standard that allows to interconnect industrial devices to the Internet, via one or more Border Routers (BRs), with an industrial grade of service. 6TiSCH leverages the Time Slotted Channel Hopping (TSCH) mode of the IEEE 802.15.4 standard for short-range wireless communication [2]. TSCH ensures *time-bounded and predictable latency* via slotted access, *increased network capacity* through multi-channel communication, and *improved reliability* thanks to channel hopping. One of the key component of the architecture is the Scheduling Function (SF) used to allocate communication resources to nodes.

Interconnected objects are not limited to stationary devices. Actually, many industrial applications involve mobile objects, such as industrial autonomous vehicles, mobile robots, wearable devices carried by workers, etc. However, the 6TiSCH architecture does not include any mechanism for the efficient management of node mobility and implicitly assumes that nodes are stationary.

In the literature, some previous works [3]–[9] have considered node mobility in the IIoT. However, only a few of them deals with the 6TiSCH architecture [6]–[9], and none of them consider the definition of a whole framework to handle mobility. Instead, the *Synchronized Single-hop Multiple Gateway* (SHMG) framework, originally proposed in [4] and then extended in [10], is a promising approach to handle node mobility in 6TiSCH networks. SHMG relies on a centralized approach. With reference to Fig. 1, the Network Coordinator (NC) is responsible for calculating a communication schedule and allocating communication resources to each Mobile Node (MN) and Border Router (BR), based on the application requirements and network conditions. Then, the communication schedule is diffused to all the BRs and MNs in the network.

While many scheduling algorithms can be used within the SHMG framework, the scheduling algorithms proposed so far [4], [10] are very conservative and allocate timeslots in a permanent and dedicated way in all the BRs, in order to guarantee the stringent requirements of industrial applications and avoid service interruptions, due to node mobility. For instance, the *Shared-Downstream, Dedicated-Upstream* (SD-DU) scheduling [10] uses this approach for allocating timeslots for uplink communication (i.e., from MN to BR), while timeslots for downlink communication are shared by a certain number of MNs, under the assumption that the communication from BR to MN is less frequent.

The above proposals do not take into account the location of MNs when computing the communication schedule. Consequently, the same schedule is installed in all the BRs within the network. This approach results in a redundant communication schedule that guarantees seamless connectivity to MNs, but at the cost of allocating a considerable high amount of communication resources, limiting the number of MNs that can be accommodated in the system.

To overcome these shortcomings, while ensuring the highest possible communication reliability to meet the stringent requirements of industrial applications, in this paper we propose a *Location-Aware Scheduling Algorithm* (LASA) for the SHMG framework. We assume that MNs are capable of tracking their position and direction, and then send them to the NC. Based on this information, the NC calculates an initial schedule by solving an optimization problem to build the optimal communication schedule with the minimum number of conflicts. Conflicts are generated when two or more

MNs in the communication range of the same BR have their communication resources allocated in the same timeslot. Then, LASA applies a conflict management policy to handle conflicts that may arise due to the movements of MNs.

We evaluate LASA by means of an extensive simulation analysis. Our results show that it allows to increase the number of mobile nodes that can be accommodated in the system by more than four times, with respect to SD-DU, still guaranteeing a Packet Delivery Ratio higher than 98%.

The reminder of the paper is organized as follows. Section II introduces the 6TiSCH architecture and the SHMG framework. In Section III we describe our Location-Aware Scheduling Algorithm. In Section IV we present the set up of our simulation experiments, and in Section V we present the results obtained. Finally, in Section VI, we draw our conclusions.

## II. THE 6TiSCH ARCHITECTURE

In this Section, we describe the 6TiSCH architecture [1] defined by IETF to enable the IIoT paradigm, and the SHMG framework proposed for managing MNs in 6TiSCH networks. Fig. 1 shows the 6TiSCH reference network and protocol stack. A 6TiSCH network consists of a number of MNs connected to the Internet through Border Routers (BRs).

MNs and BRs communicate using the Time Slotted Channel Hopping (TSCH) mode of the IEEE 802.15.4 standard [2]. TSCH leverages *time-slotted access*, *multi-channel communication*, and *frequency hopping*. To get time-slotted access, time is divided into (*timeslots*) of fixed duration, grouped to form a *slotframe* that repeats periodically over time. Moreover, to increase network capacity, nodes can exploit the 16 different channels available, identified by a *channel offset*, i.e., an integer value in the range 0-15, to communicate simultaneously in the same timeslot. Finally, to mitigate the negative effects of multi-path fading and interference, the frequency hopping mechanism allows nodes to change their operating frequency at each timeslot, following a predefined hopping sequence.

In the TSCH two-dimensional slotframe, each *cell* corresponds to a communication resource and is identified through a couple of information, namely, timeslot and channel offset. There are two types of cells, namely *dedicated* or *shared*. Dedicated cells guarantee contention-free communication among couple of nodes, while shared cells can be accessed, on a contention basis, by all the nodes, and to prevent possible collisions, the nodes adopt the TSCH CSMA (Carrier Sense Multiple Access) algorithm [2].

Since the TSCH standard [2] does not specify how cells are allocated to nodes, the 6TiSCH architecture includes the Operation (6top) sublayer, as shown in the protocol stack in Fig. 1. The 6top sublayer is responsible for the management of cells allocation to nodes, by relying on a Scheduling Function (SF) and on the 6top protocol (6P) [11]. The SF is the algorithm that computes the number of cells a node requires to satisfy its traffic requirements, while 6P negotiates the allocation/deallocation of cells among neighbor nodes.

On top of the 6top sublayer, the 6LoWPAN adaptation protocol encapsulates IPv6 datagrams in TSCH frames. Multi-hop data delivery is managed through the IPv6 Routing

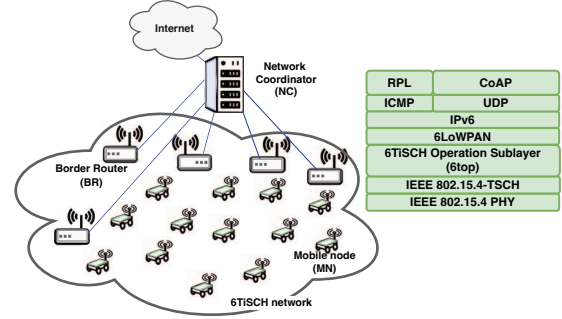


Fig. 1: 6TiSCH reference network and protocol stack.

Protocol for Low-Power and Lossy Networks (RPL) [12]. Finally, the User Datagram Protocol (UDP) protocol manages data messages generated by the application.

### A. The Synchronized Single-hop Multiple Gateway framework

The *Synchronized Single-hop Multiple Gateway* (SHMG) framework implements a centralized scheme to manage node mobility in 6TiSCH networks. It leverages the following entities shown in Fig. 1. The *Network Coordinator (NC)* is a central entity that calculates the communication schedule. *Border Routers (BRs)* are static nodes that serve as gateways between the 6TiSCH network and the Internet, providing connectivity to Mobile Nodes (MNs) in the whole deployment area. Finally, *Mobile Nodes (MNs)* generate data packets to transmit to the closest BR through single hop communication.

In the considered framework, each BR forms a star topology with all the MNs located in its transmission range. To allow a smooth and efficient handover from one BR to another, in the SHMG framework, all the BRs are synchronized and the communication schedule, computed by the SD-DU SF executed in the NC, is installed on all the BRs. Therefore, the MNs do not need to acquire cells after moving from one BR to another. Moreover, assuming that the BRs are deployed as in [10], and consequently the MNs' deployment area is completely covered by BRs, any MN can move without experiencing any service discontinuity.

### B. SD-DU Scheduling Function

The *Shared Downstream - Dedicated Upstream (SD-DU)* SF computes the allocation of communication resource in the two-dimensional TSCH slotframe considering different kinds of cells: (i) *Control Cell*: a single shared cell for all the nodes with (timeslot = 0, channel offset = 0) for control traffic. (ii) *Downstream Cells*: a number of dedicated cells allocated for data traffic from the NC to the MNs, through one of the BRs; *Upstream Cells*: a dedicated cell for each MN for data traffic from the MN to the NC, through one of the BRs.

SD-DU is a flexible SF that takes different approaches for the allocation of cells. For upstream cells, it allocates one dedicated cell for each MN, and since a BR cannot receive packets on different frequencies during the same timeslot, only one channel offset per timeslot is used. For downstream cells, instead, SD-DU allocates a number of cells that depends on the amount of traffic from the NC to the MNs. In this case, since

MNs may be located in the communication range of different BRs, it is possible to exploit multi-channel communication by allocating downstream cells in the same timeslot at different channel offsets.

### III. LOCATION-AWARE SCHEDULING

In this Section, we present the *Location-Aware Scheduling Algorithm (LASA)* for the SHMG framework. Unlike SD-DU, which does not consider the location of MNs and thus builds a conservative schedule wasting more resources than necessary, LASA computes the communication schedule by taking advantage of the MNs' location within the deployment area. We assume that MNs are capable of retrieving their location and direction within the deployment area. The kind of methods and sensors through which MNs retrieve their location and direction are out of scope for this work, however an overview of such methods is given in [13], [14].

We consider two mobility patterns for MNs, namely *linear* and *random* mobility. In both cases, the initial position of each MN could be any point of the deployment area. Then, the MN's position change over time according to the mobility pattern in use. When adopting the *linear* pattern, each MN moves from its initial position on a straight (vertical or horizontal) line with a constant speed, bouncing back when it reaches the border of the deployment area. This pattern is typical of mobile robots/objects moving along a constrained path. For the *random* mobility pattern, instead, an MN moves linearly from the current location to a randomly selected location with a constant speed. Upon reaching a new position, the MN generates another random position and moves there.

As a reference traffic generation pattern, here we consider periodic data traffic, as in monitoring applications [15], generated from the MNs and addressed to the NC (through different BRs). We assume that all MNs have the same packet generation period (i.e., data rate), which defines the length of the slotframe. LASA is designed to guarantee one cell per slotframe to each MN with a probability close to 100%, despite conflicts due to node mobility. Along with an appropriate deployment of BRs, this allows to meet the stringent requirements of industrial applications, even in the presence of node mobility, without service interruption.

LASA runs on the NC, that collects the location information (e.g., position and direction) from the MNs. To this end, and to avoid the overhead of generating and sending ad-hoc packets, MNs include their position and direction information in their data packets, adding a *Position Notification (PN)* field in the payload. We define the period  $P_{PN}$  to represent the sending interval of PN, with respect to the data traffic generation period  $1/r$ , e.g., if  $P_{PN} = 1$ , the PN field is embedded in each data packet, if  $P_{PN} = 5$ , the PN field is embedded every 5 data packets, and so on.

The PN field contains a *Region Identifier* and a *Direction Identifier*, that are integer values that map the MN's position in the deployment area. Specifically, to identify a region, we build a virtual grid on the deployment area composed by  $W \times H$  regions, identified by an integer number between 0 and  $W \cdot H - 1$ , as shown in Fig. 2. To identify the direction, instead,

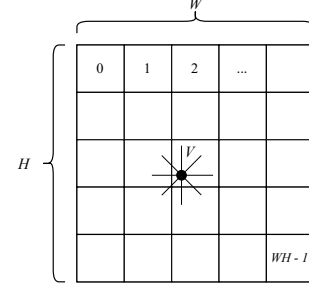


Fig. 2: Mapping of MN's position and direction in LASA.

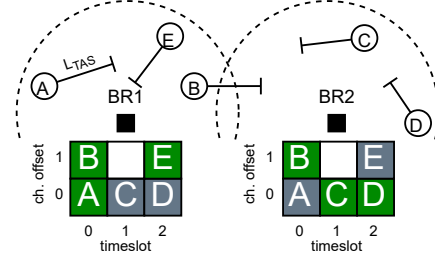


Fig. 3: Cell activation mechanism and preventive allocation.

we divide the  $360^\circ$  angle of each MN in  $V$  angles, identified by an integer between 0 and  $V - 1$ , where 0 represents the north. It follows that the size (in bits) of the PN field is  $\lceil \log_2(W \cdot H) \rceil + \lceil \log_2(V) \rceil$ . It is worth to highlight that the greater the number of regions or angles, the higher the precision of the MN's position, at the cost of more bits needed for the PN field. The values of  $W, H, V$  can be set to satisfy the trade-off between the overhead in bits of the PN, and the precision of the MN location required by the application.

Knowing the MNs' position and direction allows LASA to exploit multi-channel communication, by allocating cells in the same timeslot at different channel offsets. This is possible thanks to the fact that, on each BR, only the cells of the MNs within its communication range will be activated. This is shown in Fig. 3, where the green cells on the communication schedule are the active ones for the considered BR, while the grey ones are deactivated. Here, the MNs E and D have their cells allocated on the same timeslot, but at two different channel offsets, and since they are connected to two different BRs, they can send their data without interfering.

LASA allocates one cell per slotframe to each MN. The resulting allocation, however, may include *conflicts*. An MN has a conflict when it has assigned a cell in a timeslot that is also used by another MN, at a different channel offset, which is in the communication range of the same BR. This generates a conflict since a BR can receive data only on one channel in a given timeslot; thus it is forced to choose which cell to activate its radio to receive data. This allows only one MN per timeslot to correctly deliver data, leaving the other without a corresponding receiving cell. Fig. 3 shows an example of conflict: A and B are in the range of BR1 and have a cell in the same timeslot, although with different channel offsets.

Handling conflicts is extremely important for the network performance. Hence, in addition to presenting how the initial

communication schedule is built (Section III-A) and how mobility is managed (Section III-B), in the following we also define some conflict management policies (Section III-C).

#### A. Initial Communication Schedule

LASA calculates the initial communication schedule at network bootstrap. To this aim, it is assumed that the traffic requirements of all the MNs and their initial exact location are known by the NC. Firstly, LASA computes the slotframe length  $S$ , in number of timeslots, calculated as the maximum length that satisfies the application requirements in terms of transmission rate  $r$ . To satisfy a traffic generation rate of  $r$ , LASA must allocate 1 cell every  $1/r$  seconds to each MN, thus  $S$  is computed as  $S = \lfloor 1/(T_s \cdot r) \rfloor$ , where  $T_s$  is the timeslot duration in seconds ( $T_s = 0.015$  s).

Subsequently, knowing the slotframe length  $S$ , and the initial exact location of all the MNs, LASA can compute an initial communication schedule that minimizes the number of conflicts. To this end, we define an optimization problem, namely the *Minimum-Conflict Schedule (MCS)* problem, to compute the optimal communication schedule, i.e., the schedule that minimizes the number of conflicts in the network.

The MCS problem calculates the allocation matrix  $A \in \{0, 1\}^{M \times S}$ , where  $A_{m,s}$  is set to 1 when the MN  $m$  is allocated to the timeslot  $s$ . The allocation matrix represents the communication schedule for the whole network. For each timeslot  $s$ , the channel offset, an integer value in the range  $[0, N_c - 1]$ , is chosen randomly for each MN  $m$  and timeslot  $s$ . The inputs taken by the MCS problem are the following inputs: (i) the number  $B$  of BRs deployed in the area; (ii) the number  $S$  of timeslots in the slotframe; (iii) the number  $M$  of the deployed MNs; (iv) the location matrix  $R \in \{0, 1\}^{B \times M}$ , where  $R_{b,m}$  is set to 1 when the BR  $b$  has the MN  $m$  in its communication range, and 0 otherwise. This matrix represents the initial location of MNs in the deployment area.

The objective function of the MCS problem, defined in Eq. 1.1, aims at minimizing the overall number of conflicts network wide in the allocation. In the equation, the number of MNs allocated in the timeslot  $s$  in the communication range of BR  $b$  is calculated as  $\sum_{m=0}^{M-1} R_{b,m} \cdot A_{m,s}$ , consequently, the number of conflicts is calculated as  $\max \left\{ \sum_{m=0}^{M-1} R_{b,m} \cdot A_{m,s} - 1, 0 \right\}$ .

Given the inputs and the objective function, the MCS problem can be defined as follows:

$$\min \sum_{b=0}^{B-1} \sum_{s=0}^{S-1} \max \left\{ \sum_{m=0}^{M-1} R_{b,m} \cdot A_{m,s} - 1, 0 \right\} \quad (1.1)$$

$$\text{subject to } \sum_{s=0}^{S-1} A_{m,s} = 1, \quad \forall m \in [0, M-1] \quad (1.2)$$

$$\sum_{m=0}^{M-1} A_{m,s} \leq N_c, \quad \forall s \in [0, S-1] \quad (1.3)$$

$$\sum_{m=0}^{M-1} A_{m,s} \leq \lceil M/S \rceil, \quad \forall s \in [0, S-1] \quad (1.4)$$

Eq. 1.2 to Eq. 1.4 define the problem constraints. The first constraint (Eq. 1.2) guarantees that only one cell per slotframe is allocated to each MN. The second constraint (Eq. 1.3) limits the number of cells that can be allocated in each timeslot to the number of available channels  $N_c$ . The third constraint (Eq. 1.4) guarantees that the cells are evenly allocated to all the timeslots.

The initial optimal schedule, i.e., with the minimum number of conflicts, that the NC has derived from the execution of the MCS, is installed by the NC on all the MNs, hence all the MNs have the same schedule. Then, the NC installs the communication schedule on the BRs, by *only* activating, on each BR, the cells allocated to the MNs that are within its communication range, as shown in Fig. 3.

#### B. Mobility management

In the previous Section we presented how LASA computes the communication schedule at network bootstrap. Here we define how LASA manages node mobility during the network operational phase, when MNs change position and BR, with the possibility to generate new conflicts. Our approach allows MNs to maintain the same schedule, and leaves to the NC the whole complexity of activating/deactivating cells on BRs.

To manage mobility, the NC exploits the location information that MNs embed in their data packets on the PN field. Having the position and direction of an MN, the NC can derive its trajectory and activate MN's cells on the BRs that are located along the path, thus ensuring, in a preventive manner, seamless connectivity to the MN. The preventive activation of cells on BRs along the path of the MN results in a trade-off between activating more cells on different BRs, which can help to ensure seamless connectivity, and the generation of new conflicts.

To derive the right portion of the MN's trajectory in which BRs should activate its cells, we define the *Target Allocation Segment (TAS)* as the segment on the MN's trajectory that starts from the actual position of the MN (known by the NC), and ends at a distance  $L_{TAS}$ . The proper value of  $L_{TAS}$  can be calculated knowing the minimum communication success probability in the whole area, defined as  $\Pi_t$ . This value can be measured experimentally on a given BR deployment. The distance  $L_{TAS}$  is calculated in order to ensure that it is sufficient to allow the BRs along the TAS to receive at least one PN with probability  $\Pi_s$ . To compute the minimum number of PNs  $N_{PN}$  that needs to be sent to guarantee that at least one is received, the following inequality must hold:

$$1 - (1 - \Pi_t)^{N_{PN}} \geq \Pi_s. \quad (2)$$

where  $1 - (1 - \Pi_t)^{N_{PN}}$  is the probability that at least one PN is received over  $N_{PN}$  transmissions. It follows that  $N_{PN}$  can be derived from Eq. 2 applying different transformations, omitted here for the sake of brevity:

$$N_{PN} = \left\lceil \frac{\ln(1 - \Pi_s)}{\ln(1 - \Pi_t)} \right\rceil. \quad (3)$$

Knowing  $N_{PN}$ , the value of  $L_{TAS}$  can be computed as

$$L_{TAS} = N_{PN} \cdot D_{PN} \quad (4)$$

where  $D_{PN}$  is the distance, in meters, traveled by an MN at a constant speed  $v$ , between the transmission of two PNs, and can be computed as:

$$D_{PN} = v \cdot T_{PN} \quad (5)$$

where  $T_{PN} = P_{PN} \cdot 1/r$  is the inter-arrival time of PNs in seconds,  $r$  is the MNs' traffic rate in packets/second, and  $P_{PN}$  is the sending interval of PN.

Fig. 3 reports an example of preventive allocation: here we can see a conflict on BR1, that has both the cells nodes A and B active. Moreover the figure shows the TAS for each MN, highlighting that, since the TAS of B enters the communication range of BR2, the cell for B has been activated also on BR2.

As seen in Eq. 4,  $L_{TAS}$  could be computed as  $N_{PN} \cdot D_{PN}$ . However, to avoid unnecessary preventive allocations, we need to take into consideration also the mobility pattern to compute a more accurate value of  $L_{TAS}$  as discussed in the following for the two considered mobility patterns.

1)  $L_{TAS}$  with linear mobility pattern: when moving following the linear mobility pattern, MNs always move on horizontal or vertical lines in the deployment area. Hence, MNs do not change direction along the TAS, unless they reach the border of the area reversing their direction. In this case, however, MNs are still in the communication range of the same BR, and the schedule of the BR does not change.

We define  $D_{BR}$  as the average distance traveled by an MN within the communication range (assumed to be circular) of a BR. Following the approach presented in [16],  $D_{BR}$  can be computed as follows:

$$D_{BR} = \frac{4}{\pi} \cdot R \quad (6)$$

where  $R$  is the radius of the BR's communication range.

Consequently, if the distance traveled by an MN among two PNs,  $D_{PN} \leq D_{BR}/N_{PN}$ , where  $D_{BR}/N_{PN}$  is the maximum distance that can be tolerated among two PNs, then the MN can send all  $N_{PN}$  packets when within the current BR, without the need to perform preventive allocation on other BRs. In this case, the optimal value of  $L_{TAS}$  is 0. Instead, if  $D_{PN} > D_{BR}/N_{PN}$  preventive allocation on BRs must be performed. Summarizing,  $L_{TAS}$  can be computed as:

$$L_{TAS} = \begin{cases} 0 & \text{if } D_{PN} \leq D_{BR}/N_{PN} \\ N_{PN} \cdot D_{PN} & \text{otherwise} \end{cases} \quad (7)$$

2)  $L_{TAS}$  with random mobility pattern: if the mobility pattern is random, MNs may change direction at any point during the TAS, causing each time a new evaluation of the cells to activate/deactivate on each BR. We assume that the direction change is uniformly distributed and that, on average, MNs change direction after travelling  $D_{BR}/2$  meters in the communication area of a BR. Thus, if PNs are received by

the current BR, the following inequality must hold:  $D_{PN} \leq (D_{BR}/2)/N_{PN}$ , and it is not necessary to perform preventive allocation on other BRs, i.e.,  $L_{TAS}$  is 0, otherwise:

$$L_{TAS} = \begin{cases} 0 & \text{if } D_{PN} \leq (D_{BR}/2)/N_{PN} \\ N_{PN} \cdot D_{PN} & \text{otherwise} \end{cases} \quad (8)$$

The last feature of LASA in managing mobility, defined as *BackUp Allocation Strategy (BUAS)*, is the possibility to exploit timeslots with temporarily no active cells on BRs, to allocate cells to possibly receive packets from the closest MN available with a matching cell allocated.

The NC periodically sends the coordinates of all the MNs,  $p_0 \dots p_{M-1}$ ,  $p_i = (x_i, y_i)$ , to the BRs. When a BR  $b$  has a timeslot  $s$  with no active cells, it locally computes the set  $I_s$  of the MNs that have a cell allocated in the current timeslot  $s$ , and computes the id  $C_{b,s}$  of its closest MN as:

$$C_{b,s} = \arg \min_{i \in I_s} \|p_i - c_b\| \quad (9)$$

where  $c_b$  represents the coordinates of the BR  $b$ . Then, for the current timeslot, it turns its radio on in receiving mode to possibly receive data from the  $C_{b,s}$  MN.

### C. Conflict management

So far, we have described how LASA can exploit the information on MNs' location to build an efficient and location-aware communication schedule that minimizes the number of conflicts in each BR. However, when MNs move from one BR to the other, there is always the possibility that they generate conflicts, e.g., MNs that are in the communication range of the same BR and have their cells allocated in the same timeslots at different channel offsets. When a conflict occurs, each BR must decide the channel offset on which to activate to receive the packets from the MN allocated in that channel offset. To this aim, we introduce some conflict management policies.

Before describing each policy in details, let us define  $M_s$  as the set of MNs that has a cell allocated in the same timeslot  $s$ , and  $M_{s,b}$  as the subset of  $M_s$  that contains the MNs that are in the communication range of the BR  $b$ . Hence, a conflict on the BR  $b$  in the timeslot  $s$  occurs if  $|M_{s,b}| \geq 2$ .

We define the following four conflict management policies for LASA.

- *Random*: each BR  $b$ , at each slotframe, picks at random an MN from the  $M_{s,b}$  set with uniform distribution, and activates its radio to receive data from that MN.
- *Round-Robin*: each BR  $b$ , at each slotframe, activates in receiving mode, in turn, for each of the MN in  $M_{s,b}$ , starting from the first MN, through the last one, and then starting again.
- *Closest first*: the NC provides the BR  $b$  with the information on the position of the MNs in  $M_{s,b}$ . When a conflict occurs, the BR can determine the closest MN in  $M_{s,b}$  and

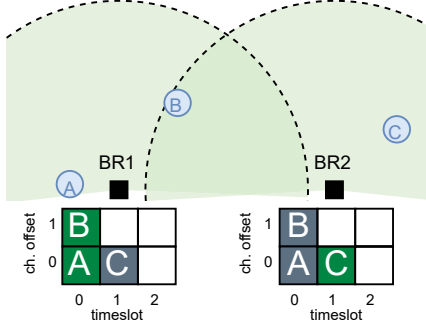


Fig. 4: Avoidable conflict.

activate on its channel offset to receive its data traffic. The closest MN is computed at each slotframe.

- **Oldest-first:** for each MN in  $M_{s,b}$ , the BR stores the timestamp at which the MN entered its communication range. When a conflict occurs, the BR set up its radio to receive packets from the MN with the oldest timestamp.

#### IV. SIMULATION SETUP

In this Section we describe the simulation setup used to carry out the performance evaluation of LASA.

To perform our simulation analysis, we exploit the *Mobile-6TiSCH* simulator [17]<sup>1</sup>, based on OMNeT++<sup>2</sup>, that implements the SHMG architecture. Mobile-6TiSCH supports different mobility patterns, and in our analysis we evaluate the network performance with two mobility patterns, namely *linear* and *random*.

The initial communication schedule, that is applied to MNs by the NC, is computed exploiting the optimization problem defined in Section III-A, that is a Constrained Optimization Problem (COP) [18], and we exploit the Google OR-TOOLS library<sup>3</sup> to implement and solve it on the NC.

The deployment area we consider is a square of side 400 m, in which the BRs deployment has been carried out following the policy defined in [10]. The BR radius is set to 44.8 m, which results in a deployment of 40 BRs in the area, and that guarantees a minimum communication success probability  $\Pi_t = 0.75$ , as it is measured in our experiment.

To allow MNs to send their location to the NC through the Region and Direction identifiers, the 400 m  $\times$  400 m area is divided in a virtual grid of  $W = 64 \times H = 64$  squares of side 6.25 m. Moreover, to map the MNs' direction, the 360° angle is divided in a number of directions  $V$  equal to 16. This map of the deployment area results in a size of the PN field that is  $\lceil \log_2(64 \cdot 64) \rceil + \lceil \log_2(16) \rceil = 16$  bits, resulting in an overhead to send MN's location information that is considerably low, i.e., 2 bytes for each data packet that includes the PN field. We set the MNs' packet periodic generation rate  $r$  to be 2 pkts/s, i.e., MNs generate one data packet every 0.5 seconds. LASA computes the number of timeslots in a slotframe  $S$  depending on the rate  $r$  (as described in Section III-A), hence in our simulation  $S = 33$ . For each

simulation experiment, to obtain statistically sound results, we run 10 independent replicas with a duration of 1000 seconds. The results we present are shown with confidence intervals obtained with a 95% confidence level.

As metric, we considered the **Packet Delivery Ratio (PDR)**, defined as the ratio between the number of packets correctly received by the NC, and the total number of packets generated by MNs. Moreover, to get a deeper insight on it, we also analyzed the different **components of the packet loss**, identifying four causes of packet loss: (i) *avoidable conflicts*, conflicts that are generated when an MN is in the communication range of two BRs, and the BR that has its cell active, is also the one that has a conflict on that cell, while the other BR would not have any conflict in the MN's cell (as shown in Fig. 4); (ii) *unavoidable conflicts*, conflicts that are generated when an MN has its allocated cell active on one BR that has a conflict on that cell; (iii) *transmission error*, packets lost due to the unreliability of the wireless medium; (iv) *out of range*, packets not received by any BR, as a consequence of an inaccurate location estimation of MNs or lack of preventive allocation.

Fig. 4 reports an example of avoidable conflict. Node A and node B have their cells allocated both in timeslot 0 and channel offset 0 and 1, respectively. This generates a conflict on BR1. However, considering that A is in the communication range of BR1, while B is in the communication range of both BR1 and BR2, if the cell of B is activated only on BR2, no more conflict will be present on BR1.

#### V. PERFORMANCE EVALUATION

In this Section, we present and discuss the results of the performance evaluation we carried out to analyze LASA. We first evaluate and compare the performance of the considered conflict management policies (Section V-A). Then, we investigate the impact of the Position Notification (PN) frequency (Section V-B). Finally, to have a deeper insight on the individual performance of MNs, we analyze the PDR of each single MN (Section V-C).

##### A. Comparison of Conflict Management policies

Figs. 5 and 6 compare the performance of the considered conflict management policies, namely *Random*, *Round-Robin*, *Closest First* and *Oldest-First*. In those graphs, we analyze the Packet Delivery Ratio (PDR) obtained with the different policies, considering a varying number of MNs moving at a speed of 5 m/s and, with  $P_{PN} = 1$ , so as each MN sends its location information in each data packet.

Fig. 5 shows the PDR when the mobility pattern is random. It is worth to highlight that, since the number of timeslots in a slotframe is  $s = 33$ , if the number of MNs is lower than or equal to 33, no conflict can be generated. This is due to the fact that, in this case, only one channel offset per timeslot can be exploited to satisfy the traffic requirements of all the MNs. Indeed, with a number of MNs up to 33 the PDR is the maximum, i.e., above 99.7%. It is worth to highlight that the 0.03% of packet loss is due to the unreliability of the wireless

<sup>1</sup><https://github.com/marcopettorali/Mobile6TiSCH>

<sup>2</sup><https://omnetpp.org/>

<sup>3</sup><https://developers.google.com/optimization>

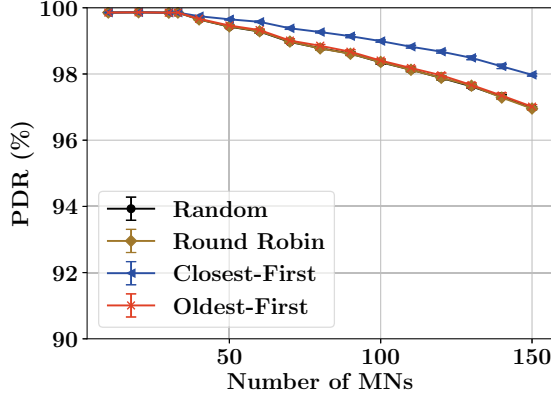


Fig. 5: PDR with Random Mobility,  $P_{PN} = 1$ ,  $v = 5$  m/s.

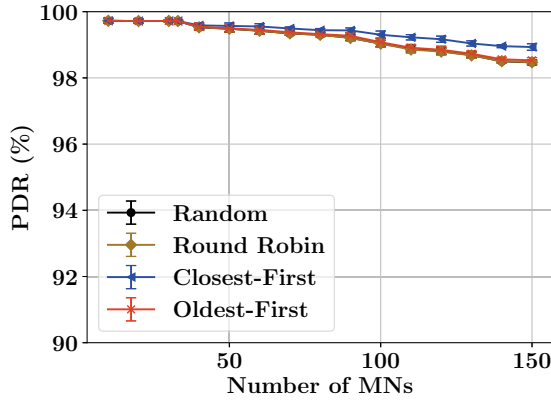


Fig. 6: PDR with Linear Mobility,  $P_{PN} = 1$ ,  $v = 5$  m/s.

medium. If we compare the 4 management policies, we can observe that, up to 150 MNs, the PDR is always above 97%. More in detail, if we consider the Random, Round-Robin and Oldest-First policies, they exhibit comparable performance, whereas Closest First is the one that guarantees the higher PDR, i.e., 98%. The same trend can be observed when linear mobility is considered, as shown in Fig. 6. However, we obtain values of PDR that are slightly higher with respect to the ones obtained in the scenario with random mobility. This can be explained considering that MNs, with linear mobility, always move on the same line, thus the information on their position and direction is more accurate.

If we compare those results, with the performance obtained by the SD-DU SF (discussed in [10]), we can see that LASA, with a packet generation rate  $r$  of 2 pkts/s, allows the deployment of up to 150 MNs while guaranteeing a PDR higher than 98%. Instead, SD-DU allows only 33 MNs to be deployed, to satisfy the same packet generation rate  $r$ . Hence, with respect to SD-DU, LASA can accommodate a number of MNs that is more than four times larger.

Comparing the different conflict management policies, it emerges that Closest First is the one that guarantees the higher PDR. The reason for such behaviour can be found in the way it manages the so called *avoidable* conflicts, defined in Section V and shown in Fig. 4. As regarding avoidable conflicts, the

Closest First policy ensures that the BRs only activate the cells of their closest MN. This solves the conflict on BR1 reported as an example in Fig. 4, since BR1 will activate the cell for A (its closest for timeslot 0), and BR 2 will activate the cell for the MN B.

Fig. 7 shows the various components of packet loss for the different conflict management policies (Round Robin, Closest First and Oldest First), considering the random mobility pattern, a speed of  $v = 5$  m/s and  $P_{PN} = 1$ . For the sake of space, we do not show the graph for the Random policy, that exhibits results similar to Round Robin. We can observe that, with respect to the other policies, when using Closest First the number of avoidable conflicts reduces significantly, thus decreasing the overall packet loss. This is due to the fact that the probability that an MN is the closest to more than one BR is very low. The other policies are less effective in avoiding such conflicts. Specifically, if we compare the packet loss components for the other policies, i.e., Round Robin (Fig. 7a) and Oldest First (Fig. 7c), we can see that the fraction due to avoidable conflicts for both the policies is around 2% with 150 MNs, whereas it is around 1.3% with Closest First. As expected, the other loss components (i.e., due to transmission errors and out of range) do not depend on the conflict management policy, as they are comparable for all the policies.

From the previous results, it clearly emerges that the Closest First policy is the one that guarantees the better performance. For this reason, in the following experiments we will consider only the Closest First policy.

#### B. Impact of the Position Notification frequency

Fig. 8 shows the impact on performance of the  $P_{PN}$  parameter, i.e., the frequency at which Position Notification are sent, for random mobility (Figs. 8a and 8b), and linear mobility (8c and 8d). We considered two different speeds of MNs (namely, 2 m/s and 5 m/s) and different values for  $P_{PN}$ , (i.e., 1, 5, 10).

We first analyze the impact of the  $P_{PN}$  on the PDR, for different mobility patterns, by comparing together Figs. 8a and 8c. It can be seen that, in general, the different values used for the  $P_{PN}$  has a negligible impact on the PDR, considering an MNs' speed of 2 m/s. Especially for the linear mobility scenario (Fig. 8c), the difference among the three values of  $P_{PN}$  is very close to the 0. For the random mobility scenario (Fig. 8a), instead, the PDR is around 98% when MNs embed the PN field in their packets with  $P_{PN} = 1$ , or  $P_{PN} = 5$ , and slightly decreases to 97.5% with  $P_{PN} = 10$ . It follows that, when MNs move at low speed (i.e., 2 m/s), it is not necessary for the MNs to collect and send their position and direction very frequently, as also one update every 5 seconds results in a PDR which is higher than the 97.5%. Instead, when considering a speed of 5 m/s, Figs. 8b and 8d, the impact of the value chosen for  $P_{PN}$  is more evident, as the PDR decreases up to 6%, in the random mobility scenario (Fig. 8b). As a remark, the performance of LASA in terms of PDR are higher when considering the linear mobility pattern, with respect to

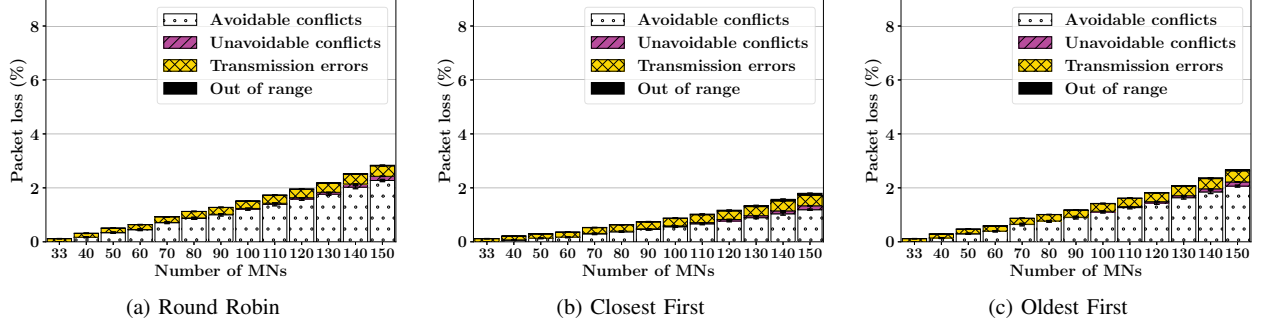


Fig. 7: Packet loss components, with different conflict management policies, Random mobility,  $v = 5$  m/s,  $P_{PN} = 1$

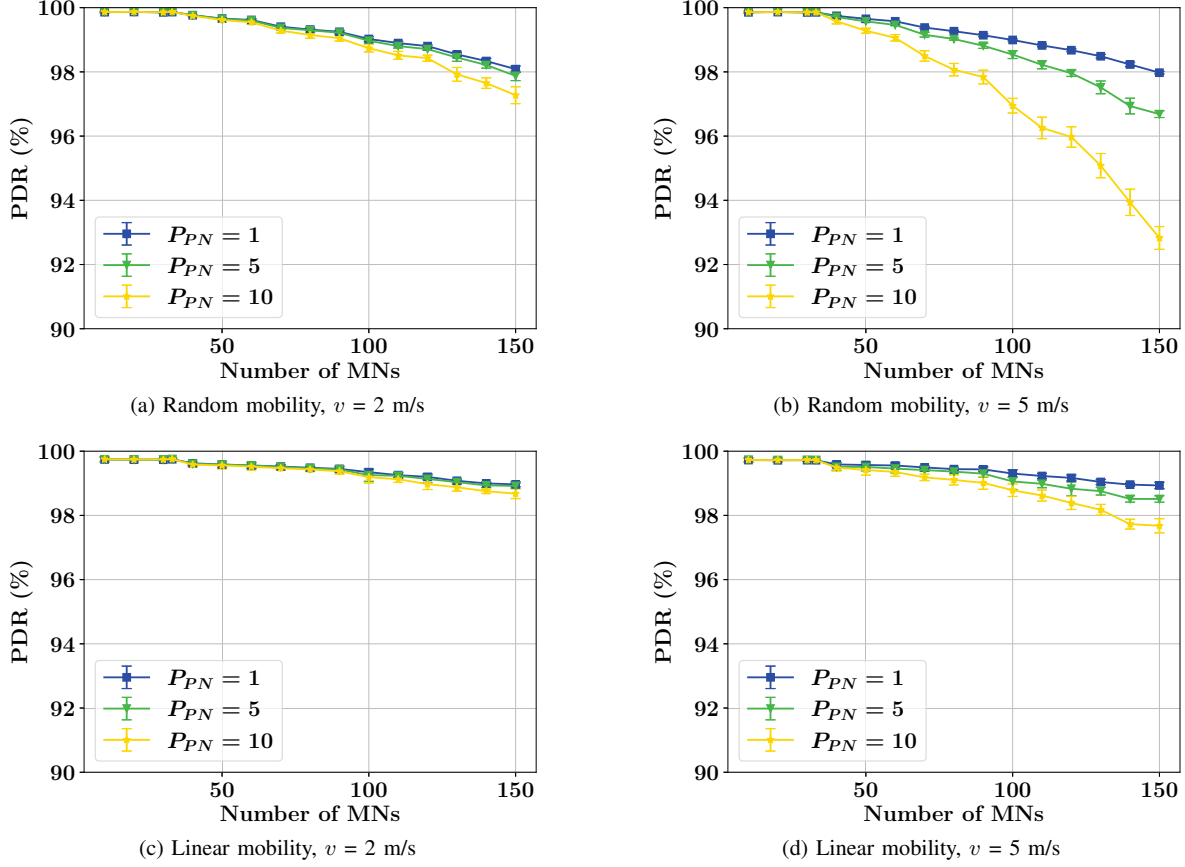


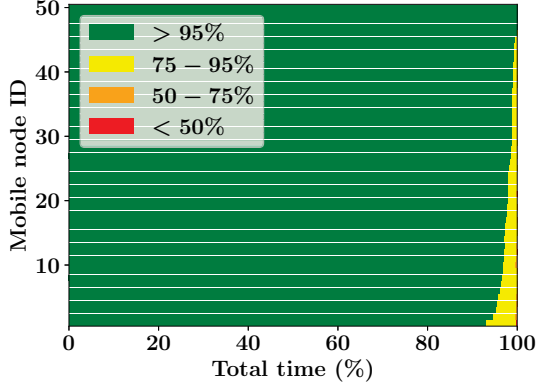
Fig. 8: Impact of  $P_{PN}$  on the PDR.

the random mobility pattern. This can be explained by the fact that when MNs change their position and direction frequently and randomly, it is more difficult to have their updated position and also to perform an effective preventive allocation.

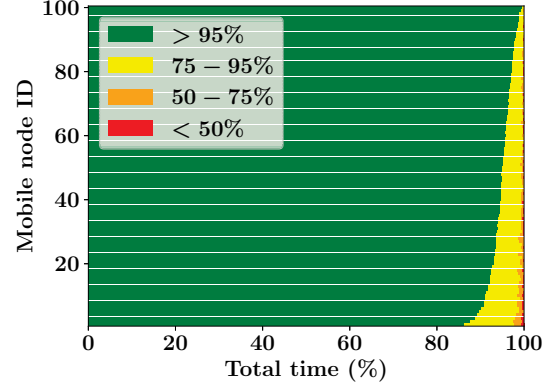
Clearly, sending location information periodically has a cost, in terms of bandwidth consumed for transmitting the related information in data packets and energy consumed by sensors to acquire the current position and direction of the MN. It may be worthwhile emphasizing that energy consumption is typically not so relevant in case of mobile objects, as most of the energy is consumed for mobility. Also,

location information is piggybacked in data packets, whose size is typically lower than the maximum packet size. Hence, location information does not consume additional bandwidth in practice, and the cost due to tracking and sending location information, in terms of additional bandwidth and energy consumption, is typically negligible. However, they can be quantified, as shown below.

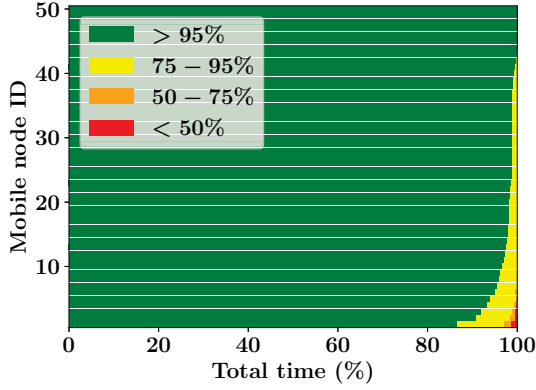
The number  $B_{PN}$  of additional bytes to send periodically for notifying location information, denoted by  $L_{PN}$  depends on: (i) the required level of accuracy; (ii) the notification period  $P_{PN}$ ; and (iii) the packet generation rate  $r$  of the MNs. Hence,



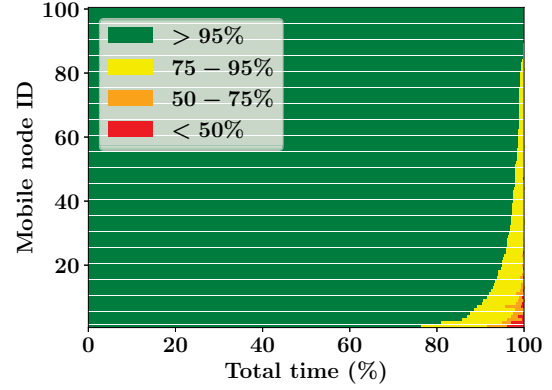
(a) Random mobility, 50 MNs.



(b) Random mobility, 100 MNs.



(c) Linear mobility, 50 MNs.



(d) Linear mobility, 100 MNs.

Fig. 9: PDR for each MN, with the Closest first conflict management policy. Speed = 5 m/s,  $P_{PN} = 1$ .

$B_{PN}$  can be expressed as

$$B_{PN} = L_{PN} \cdot r \cdot \frac{1}{P_{PN}} \quad (10)$$

Assuming the parameter settings used in our simulation experiments (i.e.,  $L_{PN} = 2$  Bytes,  $r = 2$  pkt/s) even sending the location information at each packet (i.e.,  $P_{PN} = 1$ ), results in an overhead of 4 Bytes/s, which is negligible for most applications.

To quantify the energy cost, we need to emphasize that this strictly depends on the specific method used by MNs to retrieve their position and direction. However a general formula can be given, as follows:

$$E_{PN} = (E_{pos} + E_{dir}) \cdot r \cdot \frac{1}{P_{PN}} \quad (11)$$

where  $E_{pos}$  and  $E_{dir}$  correspond to the energy consumed for computing the position and direction, respectively. Of course, the total energy consumption depends on the acquisition rate ( $r \cdot 1/P_{PN}$ ).

### C. Individual Performance of MNs

To conclude our performance evaluation, we want to provide a deeper insight on how the PDR is distributed among MNs. Specifically, by considering the PDR over time for each MN,

we want to determine the time it experiences a PDR higher than 95%, between 75% and 95%, between 50% and 75% or lower than 50%. Such time is expressed as a percentage of the total experiment time. Moreover, we want to assess if the PDR is evenly distributed among all the MNs.

In Fig. 9 we report the time the PDR of each MN is in each of the defined PDR intervals, considering the random mobility pattern with 50 and 100 MNs (see Figs. 9a and 9b), and the linear mobility pattern (see Figs. 9c and 9d), and  $P_{PN} = 1$ .

When 50 MNs are deployed, both for the random and for the linear mobility pattern (Figs. 9a and 9c), it can be seen that, with random mobility, all the MNs experience a high level of PDR, i.e., greater than 95%, (green color), for at least the 93% of time, while with linear mobility the PDR is higher than 95% for the 87% of the time. The majority of the MNs, also experience a PDR between 75% than 95% (yellow color), for a very short amount of time, i.e., from the 1% to the 7% of time with random mobility, and from the 1% to the 13% of time with linear mobility. It can be noticed, however, that only when considering linear mobility (Fig. 9c), a few number of MNs show a PDR lower than 50% (orange and red colors) for at most the 4% of time. When considering 100 MNs, (Figs. 9b and 9d), we observe a similar trend to the one described

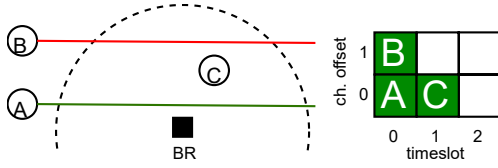


Fig. 10: MNs' trajectories with linear mobility pattern.

before, with a slight decrease in the time in which the PDR is higher than 95%, and consequently an increase of time in which its values are in the other lower levels.

In the comparison of the random and linear mobility patterns, it can be noticed that even though the PDR values for the whole network up to 100 MNs, shown in Fig. 8, for the two mobility patterns considered are comparable, especially when  $P_{PN} = 1$ , from Fig. 9 it emerges that with the linear mobility, a small group of MNs experience low level of PDR for a small percentage of time, i.e., 4%. This can be explained considering the trajectories that MNs follow with linear mobility and the conflict management policy adopted, i.e., Closest First. In fact, as shown in Fig. 10, it may happen that two MNs, e.g., A and B, that have their cells allocated in the same timeslot, move on two lines, one of which is always closer to the BR. In this case, the BR will activate only the receiving cell for A, preventing the node in the farthest trajectory, B, to correctly deliver its data. Consequently, given that the linear trajectories are fixed, MN B will always suffer from packet loss within the communication range of BR. With random mobility pattern this behaviour does not occur since MNs frequently change trajectories. This explains why a group of MNs experiences less PDR while following a linear mobility pattern.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed the Location-Aware Scheduling Algorithm (LASA) for the SHMG framework that considers the location of MNs in the computation of the communication schedule. LASA aims at optimizing the usage of resources and avoiding the allocation of redundant transmission opportunities, thus improving the overall network capacity while guaranteeing high level of communication reliability. The proposed approach includes the computation of the initial communication schedule, performed exploiting the Minimum-Conflict Schedule (MCS) optimization problem, the mobility management during the network operational phase by activating/deactivating cells on BRs based on the MNs location, also in a preventive manner, and a policy for the management of conflicts arising due to node mobility. The proposed approach was evaluated via simulations. Our results showed that it allows to deploy a number of MNs four times higher with respect to the number of MNs that can be deployed with the SD-DU SF, while guaranteeing a Packet Delivery Ratio over the 98%.

As future work, we plan to improve the effectiveness of the algorithm by introducing conflict avoidance policies to compute a communication schedule that can ensure a higher communication reliability.

## ACKNOWLEDGMENTS

This work was partially supported by the Italian Ministry of Education and Research (MUR) in the framework of the (i) CrossLab and FoReLab projects ("Departments of Excellence" program) and (ii) PNRR National Centre for HPC, Big Data and Quantum Computing (Spoke 1, CUP: I53C22000690001), and by the US NSF grants CSSI-2104078, SCC-1952045, and OAC-1725755, and by the US NSF grant under award no. 2008878 ("FLINT: Robust Federated Learning for Internet of Things").

## REFERENCES

- [1] P. Thubert, "An Architecture for IPv6 over the Time-Slotted Channel Hopping Mode of IEEE 802.15.4 (6TiSCH)," RFC 9030, May 2021.
- [2] IEEE, "Ieee standard for low-rate wireless networks," *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, 2020.
- [3] Z. Ming and M. Xu, "Nba: A name-based approach to device mobility in industrial iot networks," *Computer Networks*, vol. 191, 2021.
- [4] J. Haxhibeqiri, A. Karaagac, I. Moerman, and J. Hoebke, "Seamless roaming and guaranteed communication using a synchronized single-hop multi-gateway 802.15.4e tsch network," *Ad Hoc Networks*, vol. 86, 2019.
- [5] H. Farag, P. Österberg, M. Gidlund, and S. Han, "Rma-rp: A reliable mobility-aware routing protocol for industrial iot networks," in *2019 IEEE Global Conference on Internet of Things (GCIoT)*, 2019.
- [6] O. Tavallaie, J. Taheri, and A. Y. Zomaya, "Design and optimization of traffic-aware tsch scheduling for mobile 6tisch networks," in *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, 2021.
- [7] M.-J. Kim and S.-H. Chung, "Efficient route management method for mobile nodes in 6tisch network," *Sensors*, vol. 21, no. 9, 2021.
- [8] W. Jerbi, O. Cheickhrouhou, A. Guermazi, and H. Trabelsi, "Msu-tsch: A mobile scheduling updated algorithm for tsch in the internet of things," *IEEE Transactions on Industrial Informatics*, 2022.
- [9] O. Nielsen, L. K. Schnügger, C. Orfanidis, and X. Fafoutis, "Mobility-focused joining in tsch networks," in *2022 IEEE 47th Conference on Local Computer Networks (LCN)*, 2022.
- [10] M. Pettorali, F. Righetti, C. Vallati, S. K. Das, and G. Anastasi, "Mobility management in industrial iot environments," in *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2022.
- [11] Q. Wang, X. Vilajosana, and T. Watteyne, "6TiSCH Operation Sublayer (6top) Protocol (6P)," RFC 8480, Nov. 2018.
- [12] R. Alexander, A. Brandt, J. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey, and T. Winter, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, Mar. 2012.
- [13] Y. Li, Y. Zhuang, X. Hu, Z. Gao, J. Hu, L. Chen, Z. He, L. Pei, K. Chen, M. Wang, X. Niu, R. Chen, J. Thompson, F. M. Ghannouchi, and N. El-Sheimy, "Toward location-enabled iot (le-iot): Iot positioning techniques, error sources, and error mitigation," *IEEE Internet of Things Journal*, vol. 8, no. 6, 2021.
- [14] P. S. Farahsari, A. Farahzadi, J. Rezaadeh, and A. Bagheri, "A survey on indoor positioning systems for iot-based applications," *IEEE Internet of Things Journal*, vol. 9, no. 10, 2022.
- [15] O. Durmaz Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast data collection in tree-based wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, 2012.
- [16] H. C. Li, "Average length of chords drawn from a point to a circle," *Pi Mu Epsilon Journal*, vol. 8, no. 3, 1985.
- [17] M. Pettorali, F. Righetti, and C. Vallati, "Mobile6tisch: a simulator for 6tisch-based industrial iot networks with mobile nodes," in *2022 18th International Conference on Mobility, Sensing and Networking (MSN)*.
- [18] B. Mayoh, E. Tyugu, and J. Penjam, *Constraint programming*. Springer Science & Business Media, 2013, vol. 131.