Mobility Management in TSCH-Based Industrial Wireless Networks

Marco Pettorali, Francesca Righetti, Carlo Vallati, Sajal K. Das, *Fellow, IEEE*, and Giuseppe Anastasi

***Abstract*—Wireless Sensor and Actuator Networks (WSANs) are an effective technology for improving the efficiency and productivity in many industrial domains, and are also the building blocks for the Industrial Internet of Things (IIoT). To support this trend, the IEEE has defined the 802.5.4 *Time-Slotted Channel Hopping (TSCH)* protocol. Unfortunately, TSCH does not provide any mechanism to manage node mobility, while many current industrial applications involve Mobile Nodes (MNs), e.g., mobile robots or wearable devices carried by workers. In this article, we present a framework to efficiently manage mobility in TSCH networks, by proposing an enhanced version of the Synchronized Single-hop Multiple Gateway (SHMG) architecture. We first define a flexible scheduling algorithm, called *Shared Downstream-Dedicated Upstream (SD-DU)*, that can be configured to adapt to different types of traffic in industrial applications. Then, we develop a mathematical framework to formalize the problem of Border Routers (BRs) placement to guarantee the complete coverage of the deployment area, in the presence of obstacles and unreliable communication. A methodology for network sizing is also proposed to calculate the maximum number of MNs that can be supported by the network without violating the application requirements. Finally, we evaluate the performance of the proposed solutions, both analytically and through simulations. Our results show that the proposed enhancements allow a very effective management of node mobility, by providing mobility transparency without a significant impact on performance.**

***Index Terms*—Industrial wireless networks, TSCH, mobility.**

I. Introduction

The *Industrial Internet of Things (IIoT)* paradigm is expected to radically reshape the way industrial processes are organized, in many domains ranging from manufacturing and production systems, to logistics and transportation [1]. To foster this trend the Internet Engineering Task Force (IETF) has defined the 6TiSCH architecture [2], whose goal is the integration of IoT devices into existing IPv6 networks, ensuring an industrial grade of service.

The 6TiSCH architecture provides IPv6 services over the *Time Slotted Channel Hopping* (TSCH) access protocol defined in the IEEE 802.15.4 standard [3]. TSCH allows short-range wireless communication with guaranteed-bandwidth, bounded latency, and energy-efficiency through a *scheduled* access protocol. Moreover, to increase the network capacity and mitigate the negative effects of interference, that are quite common in industrial environments, it leverages *multi-channel communication* and *frequency hopping*, respectively.

Hence, TSCH is very well-suited to support the requirements of industrial applications, in terms of timeliness and reliability. However, it does not include any specific mechanism to support node mobility in an efficient way. This is in contrast with the nowadays trend of industrial applications to involve mobile nodes, such as mobile robots/objects, sensors/actuators mounted on rotating parts of machines, or wearable devices carried by workers. Hence, the efficient management of mobile nodes in TSCH (and 6TiSCH) networks is an open and relevant research issue [1].

Several existing works [4], [5], [6], [7], [8], [9], [10], [11], [12] have considered node mobility in IoT networks. However, only a few of them deals with TSCH/6TiSCH networks [7], [8], [10]. Moreover, these works, rather than defining an overall framework to handle mobility, only address specific aspects, such as the definition of a Distributed Traffic-aware Scheduling Function for mobile TSCH-based networks [7], the definition of a new MAC protocol [10], or the design of an efficient route management method [8].

Integrating mobile nodes in a TSCH (or 6TiSCH) network requires to address a number of issues strictly related to node mobility. First, according to TSCH specifications, nodes must associate to the network through a complex joining procedure, in order to acquire synchronization information. This association procedure must be repeated frequently in case of node mobility. In addition, once a node has associated with the network, resources must be allocated to it for communication. During this time, the (mobile) node is not able to communicate and, hence, mobility may severely impact the performance of the overall system.

The proposals for mobility support in TSCH networks available in the literature [5], [7], [8], [10] mainly deal

Manuscript received 21 February 2023; revised 21 December 2023; accepted 4 January 2024. Date of publication 16 January 2024; date of current version 6 August 2024. This work was supported in part by the Italian Ministry of Education and Research (MUR) through the CrossLab and FoReLab projects (‘Departments of Excellence’ Program), in part by the PNRR National Centre for HPC, Big Data, and Quantum Computing (Spoke 1, CUP: I53C22000690001), in part by Project “JOULE: Joint Resource Management in Reconfigurable I4.0 Factories” under Grant 2022TMT4WA, in part by Project “CAVIA: enabling the Cloud-to-Autonomous-Vehicles continuum for future Industrial Applications” under Grant 2022JAFATE, and in part by the US NSF under Grants EPCN-2319995, CSSI-2104078, CNS-2008878, SaTC-2030624, and SCC-1952045. Recommended for acceptance by C. Assi. *(Corresponding author: Marco Pettorali.)*

Marco Pettorali, Francesca Righetti, Carlo Vallati, and Giuseppe Anastasi are with the Department of Information Engineering, University of Pisa, 56126 Pisa, Italy (e-mail: marco.pettorali@phd.unipi.it; francesca.righetti@unipi.it; carlo.vallati@iet.unipi.it; giuseppe.anastasi@unipi.it).

Sajal K. Das is with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: sdas@mst.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMC.2024.3354798, provided by the authors.

Digital Object Identifier 10.1109/TMC.2024.3354798

with handover latency, fast association, and time synchronization. The problem of resource allocation, instead, is almost ignored. The most promising solution for efficient mobility support is the *Synchronized Single-Hop Multi-Gateway (SHMG)* architecture proposed in [5]. SHMG adopts a centralized scheme and relies upon a Network Coordinator (NC) and Border Routers (BRs) to provide connectivity to Mobile Nodes (MNs), which communicate only with BRs via single-hop communication. In addition, all the BRs are synchronized and operate as a single BR covering the whole area. Hence, when an MN associates with a BR, it automatically associates with all others in the system. This allows to manage mobility very effectively and, specifically, to meet the stringent requirements of industrial applications.

The SHMG architecture is a promising approach to mobility management in TSCH-based networks, especially for industrial applications that are typically time sensitive and loss intolerant. However, as highlighted in [13], some important components need to be properly defined to make SHMG suitable for a real-world context. In this article, which extends a previous conference work by the same authors [13], we enhance the SHMG architecture in the following directions that are extremely relevant for the effective management of node mobility in industrial settings.

- *BR deployment policy:* We propose a methodology to derive the optimal deployment of BRs, while ensuring that MNs can experience a minimum level of communication reliability at *any* location in the area.
- *Quality of Service (QoS) provisioning:* We design a scheduling algorithm, namely *Shared Downstream-Dedicated Upstream (SD-DU)*, for allocating communication resources to MNs in such a way to guarantee the QoS requirements of the application. We also propose a methodology for deriving the maximum number of MNs that can be supported by the network while satisfying the QoS requirements.

The final result is a general framework that can be exploited in real settings to derive an optimal BR placement, schedule communication resources for meeting the requirements of real-time and loss-sensitive applications, and derive the maximum number of nodes that can be supported by the system, thus covering entirely the network management workflow.

To summarize, the novel contributions of this article are:

- Formalize the problem of optimal BR deployment, under the assumption of lossy channel and presence of obstacles, and compare the performance of different BR deployment policies.
- Define a flexible scheduling, Shared Downstream-Dedicated Upstream (SD-DU), to meet industrial application requirements with different traffic patterns.
- Derive a methodology to determine the maximum number of MNs that can be supported by the system, without violating QoS requirements of the application.
- Evaluate the performance of the proposed solutions, through analysis and simulation, in realistic scenarios characterized by different mobility patterns and presence of obstacles.

The reminder of the article is organized as follows. Section II presents the TSCH access protocol, while Section III overviews the related work. Section IV introduces the SHMG architecture and presents the proposed SD-DU scheduling algorithm. Section V addresses the problem of optimal BR deployment. Section VI describes the analytical model used to assess the performance of SD-DU scheduling, while Section VII uses the same model to derive the maximum number of MNs. Section VIII presents the simulation analysis of the proposed solutions. Finally, Section IX concludes the article.

II. IEEE 802.15.4 TSCH

The *Time Slotted Channel Hopping (TSCH)* access protocol, defined by IEEE in the 802.15.4 standard [3], has been conceived for short-range wireless communication in industrial and critical domains with stringent QoS requirements. Specifically, the TSCH protocol allows guaranteed bandwidth, bounded latency, high reliability, and energy efficiency, by leveraging *time-slotted access*, *multi-channel communication*, and *frequency hopping*.

Time-slotted access is provided by dividing the time into *timeslots* of fixed duration, that are assigned to specific nodes for data transmission or reception. Timeslots are grouped in a *slotframe*, which repeats periodically over time. This allows guaranteed bandwidth and time-bounded latency in communication. In addition, to increase the network capacity, multi-channel communication is used, i.e., more nodes are allowed to transmit simultaneously, during the same timeslot, using a different channel. There are 16 different channels available, each identified by a *channel offset*, i.e., an integer value in the range 0–15. Finally, channel hopping is used to mitigate the negative effects of multi-path fading and interference. Specifically, each node changes its operating frequency at each timeslot, according to a predefined hopping sequence. In order to exploit the available frequencies, the standard imposes that the slotframe length and the number of used frequencies (e.g., 16) must be co-prime.

A communication resource in TSCH corresponds to a *cell* in the two dimensional slotframe and is identified by a couple of information, namely (*timeslot*, *channel offset*). Cells can be either *dedicated* or *shared*. Dedicated cells are allocated to a couple of nodes for contention-free communication. Shared cells, instead, are allocated to all (or groups) of nodes, and are accessed on a contention basis. If more nodes transmit simultaneously on a shared cell, a collision may occur. After a collision, subsequent collisions are prevented by nodes through the TSCH CSMA (Carrier Sense Multiple Access) algorithm defined in [3].

In order to join the TSCH network, a node must perform a preliminary association procedure that consists of the following steps. The node starts scanning all the available channels, until it receives an *Enhanced Beacon (EB)*. EBs are special frames, used to advertise the network configuration, that are broadcast periodically by nodes already in the network, cycling on all the available frequencies. They contains synchronization and configuration information that allows the receiving node to join the network and starting communicating with other nodes.

After the association procedure, to communicate with other nodes, each node needs to allocate and deallocate cells dynamically, depending on the amount of traffic to manage. However, the IEEE 802.15.4 standard does not specify how and when cells must be allocated and, hence, cell scheduling and management is left to upper layers. For instance, when using the 6TiSCH architecture, this task is performed by the (6top) sublayer [2], which is responsible for the negotiation and allocation of cells by nodes. In general, a scheduling algorithm is required to compute dynamically the number of cells required by each node to meet the application requirements. In the scenarios considered in this paper, the cell scheduling algorithm must take into account also the mobility of nodes.

III. Related Work

Only a few works have addressed the problem of mobility management in TSCH networks, and the majority of them propose solutions for fast association [7], [8], [10]. In particular, in [7] the authors propose a Distributed Traffic-aware Scheduling Function (DT-SF) which monitors the network topology and the queue occupancy on each MN, and uses this information to decide whether to apply changes to the topology or not. Instead, in [8], the authors aim at optimizing the selection of the best next hop for multi-hop communication, whenever the topology changes.

Nevertheless, the proposed solutions only mitigate the problem of the handover from one BR to another, giving no guarantees on the performance achieved by MNs when moving. Thus, those solutions may not be suitable for industrial applications, which often have stringent requirements, in terms of communication reliability and timeliness. In addition, cells for communicating with the new BR may not be available, unless allocated proactively, and this may cause additional delay and packet loss. In the following we focus on the efficient management of node mobility.

The work in [10] proposes an enhanced version of the TSCH protocol, namely, *Mobility-aware TSCH (MTSCH)*. MTSCH aims at reducing the time taken by the association procedure (joining time) by using a single frequency for sending EBs. In addition, it exploits ACK messages to advertise synchronization information, so as to allow MNs to keep their time synchronization low. However, as above, no guarantee is provided to MNs, in terms of packet loss or de-synchronization time, and this may not be suitable for industrial applications.

A more promising approach to node mobility management in TSCH networks is the *Synchronized Single-Hop Multi-Gateway (SHMG)* architecture proposed in [5]. In SHMG, all the BRs are synchronized and operate as a single BR covering the whole area. Hence, when an MN associates with a BR, it automatically associates with all the other ones in the system. This allows to manage mobility very effectively. However, in [5] the focus is mainly on effective mobility management, while other important issues are missing (e.g., BR deployment policy), or just mentioned (e.g., cell scheduling and QoS management).

In this work, we extend the original proposal [5] and present a general framework, based on the SHMG architecture, that can be used in a real environment with physical obstacles and unreliable communication. Specifically, we derive the optimal BR placement, schedule communication resources in order to guarantee the requirements of real-time ad loss sensitive (industrial) applications, and derive the maximum number of nodes that can be supported by the system.

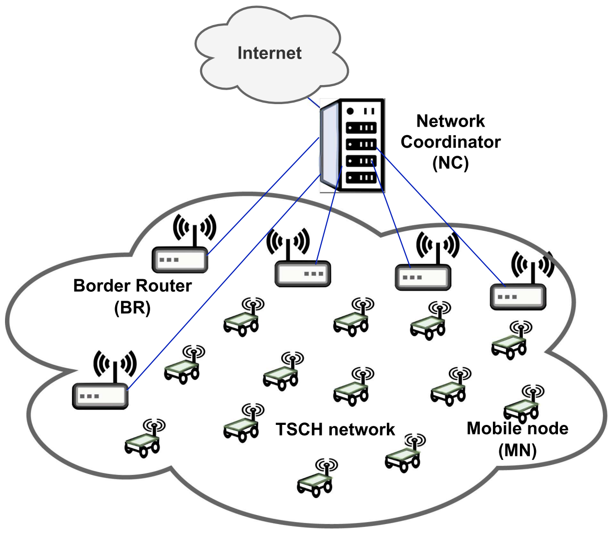


Fig. 1. Reference network architecture.

This article extends a previous conference paper [13], where we started addressing the problem of BR deployment and proposed SD-DU as a flexible scheduling algorithm that can adapt to different traffic patterns, depending on the specific application domain. In [13], the proposed solutions were evaluated under the assumption of ideal channel conditions and open deployment area (i.e., without obstacles). In this article, we consider a more realistic environment, with obstacles and unreliable communication, and propose a methodology for the optimal deployment of BRs that allows a minimum level of communication reliability in any (accessible) location. In addition, on top of the proposed SD-DU scheduling algorithm we define a methodology for deriving the maximum number of MNs that can be supported by the network without violating the application requirements.

IV. Mobility Management in TSCH Networks

This section presents our extension to the *Synchronized Single-Hop Multi-Gateway (SHMG)* architecture, originally proposed in [5]. We first describe the architecture and its components and then present our proposed SD-DU scheduling algorithm to provide QoS in industrial environments.

A. SHMG Architecture

The SHMG architecture relies on a centralized scheme and includes the following entities (see Fig. 1).

- *Network Coordinator (NC)*, the entity in charge of computing the communication schedule for all the nodes in the network. The communication schedule is then replicated on all the BRs. The NC is also the source of synchronization: clock information is sent periodically by the NC to all BRs, and from BRs to MNs.

- *Border Routers (BRs)*, static nodes acting as gateways between the TSCH network and the wired backbone network. Each BR serves as parent node for all MNs in its communication range. BRs are assumed to be connected to the NC through a reliable and high-speed link. All the BRs are assumed to be connected to the power grid and have the same MAC address, so they are seen as a single access point. The precise synchronization of the BRs is of paramount importance for the overall performance of the network. Several research papers in the literature tackle this challenge to ensure a maximum time drift from the reference in the order of microseconds. However, how this can be achieved is not specified in [5] and we consider this aspect out of scope for this paper as well.
- *Mobile Nodes (MNs)*, that send and receive data packets through the closest BR (single-hop communication).

In the SHMG architecture, each BR builds a star topology with the MNs within its transmission range. This reduces the latency experienced by data packets. In addition, to allow a smooth and efficient handover from one BR to another, all the BRs are synchronized and the communication schedule, computed by the NC, is installed on all the BRs. Therefore, MNs do not need to allocate cells for communication, when moving to a new BR.

In addition to time synchronization and schedule management, the NC also determines the best BR to forward downstream traffic to a certain MN. This decision is based on the quality of communication between BRs and MNs. Specifically, each BR collects information about the quality of communication with all MNs in its range, and sends them to the NC. Whenever a data packet must be delivered to a certain MN, if the MN is in the range of two or more BRs, the NC selects the BR with the best communication quality towards the MN.

Also, the NC adopts a *deduplication table* to detect duplicate copies of packets transmitted by an MN and received by multiple BRs. It does so by computing the hash value for incoming packets and comparing them against the values stored in the table. If a matching hash value is found, the incoming packet is duplicated and, hence, discarded.

The key feature of the SHMG architecture is that the different BRs appear as a single BR, from the MN perspective. This minimizes the time required for managing the handover and, hence, the impact of node mobility on the performance. In principle, if the deployment area is completely covered by BRs, an MN can move from a BR to another without experiencing any service discontinuity.

B. SD-DU Scheduling

This Section defines a flexible scheduling algorithm tailored to the SHMG architecture, that can be easily configured to meet the requirements of different applications in various industrial domains. Since the SHMG architecture leverages a centralized scheme, the communication schedule is computed by the NC and, then, installed on all the BRs.

To motivate the definition of our scheduling algorithm, it is worthwhile to emphasize some characteristics of industrial applications. In such applications, data typically flow in the upstream direction (i.e., from MNs to the NC, in our reference scenario), while data sent in the downstream direction (e.g., from the NC to MNs) is more or less sporadic, depending on the specific use case. Hence, the resulting traffic pattern is asymmetric. This is the case, for instance, of monitoring applications. However, some other applications exhibit a symmetric traffic pattern, where upstream and downstream flows are approximately balanced. This happens, for instance, in applications where each data packet must be explicitly acknowledged (e.g., IoT applications using Constrained Application Protocol (CoAP) [14] with Confirmable messages) or in control applications generating symmetric traffic patterns. In addition, many industrial applications have stringent requirements in terms of communication reliability and/or timeliness. They often require a guaranteed bandwidth to send/receive data, with high reliability and low (or bounded) delay. Based on the above remarks, we define below a Shared Downstream - Dedicated Upstream (SD-DU) algorithm that can guarantee the requirements of industrial applications and is flexible enough to adapt to different traffic patterns (e.g., asymmetric versus symmetric).

Before presenting SD-DU in detail, we define the following parameters: (i) S_l, represents the number of timeslots in a slotframe (i.e., the slotframe length); (ii) M, is the number of MNs deployed in the network; (iii) G, denotes the number of MNs using the same timeslot for the reception of downstream traffic. We will see that G is the parameter that regulates the amount of bandwidth allocated to the downstream traffic. Hence, varying its value allows SD-DU to adapt to different traffic scenarios.

With reference to the TSCH slotframe described in Section II, SD-DU allocates the following types of cells.

- *Control Cell:* A single shared cell, at the beginning of the slotframe (timeslot $= 0$, channel offset $= 0$), used by all the nodes to send TSCH control information.
- *Upstream Cell:* A dedicated cell allocated to a specific MN for its upstream communication with the BR.
- *Downstream Cell:* A cell allocated for downstream traffic. This cell may either be shared or dedicated.

SD-DU takes a different approach to allocate upstream and downstream cells. For upstream communication, a dedicated cell per timeslot is allocated to each MN, in order to provide a guaranteed bandwidth. Since BRs cannot receive simultaneously on different frequencies, only a single cell per timeslot will be used. Hence, the total number of cells/timeslots allocated for upstream communication in each slotframe will be $N_u = M$.

For downstream traffic, instead, SD-DU allocates a number of timeslots less than, or equal to, the number of MNs, depending on the amount of downstream traffic to manage. Specifically, the NC allocates one timeslot for a group of G MNs. Thus, the number of downstream timeslots in the slotframe will be $N_d = \lceil M/G \rceil$. Since different BRs can transmit simultaneously during the same timeslot (using different channel offsets) and there are 16 different channels available, up to 16 MNs can be addressed in a single downstream timeslot, using different cells

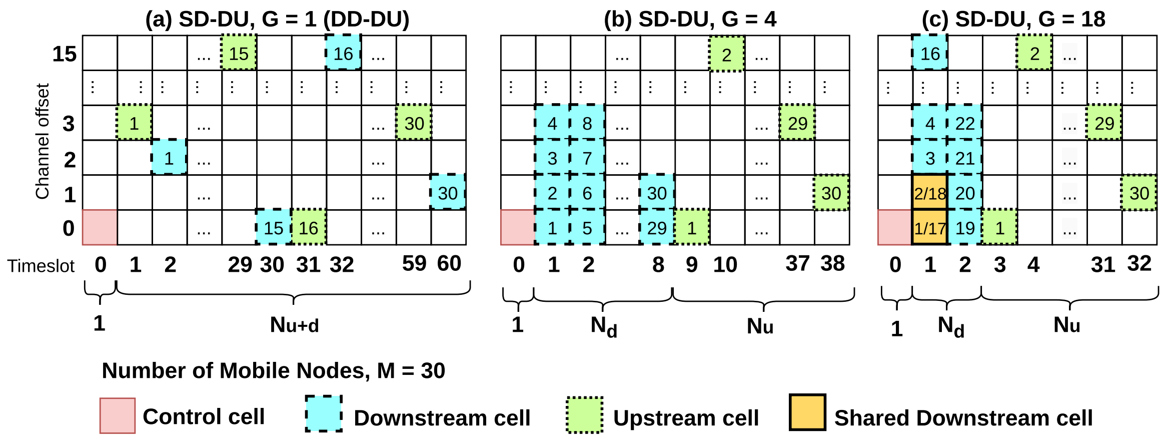


Fig. 2. SD-DU scheduling algorithm.

in that timeslot. Obviously, if $G \leq 16$ each cell will be used by a single MN. If $G > 16$, the cells will be shared by multiple MNs.

The reason for sharing downstream cells is to minimize the usage of resources when downstream traffic is sporadic, by exploiting the fact that different BRs can transmit during the same timeslot using different channel offsets. This means that different BRs can simultaneously send packets to MNs on different channel offsets, but only if the MNs sharing the same timeslot are associated with different BRs. If this is not the case, the same BR is not capable of transmitting simultaneously on different channel offsets. Hence, downstream packets will be buffered and sent to the MNs in the subsequent slotframes, according to a FIFO order.

In order to better illustrate the SD-DU allocation policy, Fig. 2 shows the communication schedule for a network of $M = 30$ MNs, considering different G values, namely, $G = 1, 4, 18$. When $G = 4$ or 18, (Fig. 2(b) and (c), respectively), the downstream cells are allocated in the first N_d timeslots of the slotframe, while the upstream cells in the next N_u ($N_u = M$) timeslots. Hence, considering also the control cell, the slotframe length is $S_l = N_d + N_u + 1$.

Of course, increasing the G value has two contrasting effects. On one hand, it decreases the number of timeslots devoted to downstream traffic (e.g., 8 with $G = 4$ and 2 with $G = 18$) and, hence, the slotframe length S_l. On the other hand, increasing the G value, increases the collision probability for downstream packets. Thus, G must be tuned very accurately, according to the amount of downstream traffic generated by the application. We highlight that, in the case where $G > 16$, some of the MNs will use the same shared cell, as shown in Fig. 2(c), where two shared cells have been allocated in timeslot 1 for MNs 1 and 17 (at channel offset 0), and MNs 2 and 18 (at channel offset 1).

In addition, when $G > 1$, it may happen that more MNs (up to G), with their downstream cell allocated in the same timeslot, are simultaneously in the communication range of the same BR. In this case, the BR will follow a priority-based policy (e.g., longest queuing time first) to decide the MN to send the downstream packet to.

Fig. 2(a) shows the special case when $G = 1$. In this configuration, SD-DU allocates two dedicated cells/timeslots per slotframe to each MN – one each for upstream and downstream communications. For this reason, hereafter, the SD-DU configuration with $G = 1$ will be also referred to as *Dedicated Downstream - Dedicated Upstream (DD-DU)*. In this case, the slotframe length is $S_l = 2M + 1$.

Since DD-DU (i.e., SD-DU with $G = 1$) has dedicated cells for both upstream and downstream flows, it may be convenient to organize the schedule in such a way that the downstream cells of an MN are allocated right after the upstream cells of the same MN, as illustrated in Fig. 2(a) This minimizes the end-to-end delay experienced by an MN, which may be relevant for time-sensitive applications.

Before concluding this Section, we want to point out that the SD-DU algorithm guarantees the amount of bandwidth required by the application. However, just relying on the scheduling algorithm does not allow to satisfy the application requirements. In order to ensure a certain QoS, the following elements are required: (i) a proper BR deployment strategy to ensure the optimal coverage of the sensing area with a minimum level of communication reliability; (ii) a mathematical framework to assess the QoS achieved by each MN, depending on the parameter settings; and (iii) a methodology to derive the maximum number of MNs that can be admitted in the network, still ensuring the application's QoS requirements.

In the following we will address all such issues by presenting a BR deployment strategy in Section V, a model to assess the performance of the SD-DU scheduling algorithm in Section VI and a methodology for network sizing in Section VII. Finally, in Section VIII, we carry out a performance evaluation, based on simulation, in order to assess the performance of our proposed strategies in a realistic environment.

V. Border Routers Deployment

This Section analyzes different strategies for the optimal deployment of BRs in the considered area. To this aim, we first introduce a packet error model for the wireless communication between MNs and BRs. Then, we will use this model to compare different BR deployment policies. Basing on this analysis, we will be able to select the best BR deployment policy, i.e., the one that allows to cover the considered area with the lowest number of BRs, while ensuring a minimum level of communication reliability.

A. Packet Error Model

In the following, we present the packet error model for the communication between MNs and BRs. For brevity, we provide here only a brief presentation of the model. A more detailed description is available in Appendix A, available online.

We define the Packet Error Rate (PER) as the probability that a data packet received by a destination node contains erroneous bits. In addition, we denote by x the distance between the sender and the receiver node, and L_b the packet length in bits. Given those parameters, the Packet Error Rate experienced by a destination node can be expressed as

$$PER(x,s) = 1-(1-BER(SINR(x,s)))^{L_b}, \quad (1)$$

where s is the shadow fading factor, a random variable that models the impact of possible obstacles among nodes, SINR denotes the Signal to Interference plus Noise Ratio, and BER is the Bit Error Ratio. Our model considers the BER function recommended in [15] and expressed as follows.

$$BER(SINR(x,s)) = \frac{1}{30}\sum_{k=2}^{16}(-1)^k\binom{16}{k}e^{20\cdot SINR(x,s)\cdot\left(\frac{1}{k}-1\right)}. \quad (2)$$

It is important to highlight that the SINR depends on

- The power of the received signal, also known as Received Signal Strength Indicator (RSSI) that depends on the transmission power of the signal P_t and the path loss function $g(x,s)$. For the path loss function we use the same parameters adopted in [16] for an industrial indoor environment.
- The level of the current interference.
- The level of white noise of the environment.

Given the PER, the success probability of a packet transmission, $\Pi(x,s)$, can be computed as

$$\Pi(x,s) = 1-PER(x,s) = (1-BER(SINR(x,s)))^{L_b}. \quad (3)$$

To derive the expected value of $\Pi(x,s)$, namely $e(x)$, we assume that the shadow fading is a Gaussian random variable with expected value equal to zero and standard deviation σ_s equal to 3.6, as in [16]. Thus,

$$e(x) = E\left[\Pi(x,s)\right] = \int_{-\infty}^{+\infty}\Pi(x,s)\cdot\varphi\left(\frac{s}{\sigma_s}\right)ds, \quad (4)$$

where $\varphi(\frac{s}{\sigma_s})$ is the Gaussian probability density function with standard deviation equal to σ_s.

For our packet error model, we assume no external interference. The average success probability of a packet transmission, as a function of the sender-receiver distance, is reported in Fig. 3. We can observe that it is equal to 1 up to a distance of approximately 25 m. Then, it degrades very sharply and drops to to 0 at distances larger than 100 m.

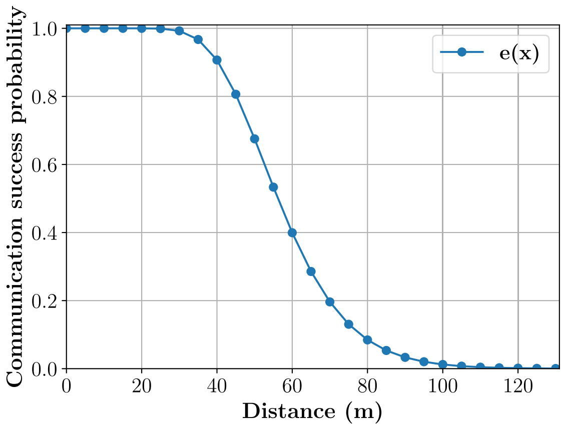


Fig. 3. Average success probability of a packet transmission as a function of the sender-receiver distance.

B. Deployment Policies for Border Routers

The BR deployment policy must guarantee a minimum level of communication reliability, at any point of the considered area, with the minimum number of BRs. We define the minimum level of communication reliability through the *target success probability* Π_t. Specifically, we request that all packets exchanged between an MN and its BR must experience an average success probability at least equal to Π_t. The target success probability depends on the reliability requirements of the application. On the other hand, it has an impact on the communication range of the BR, i.e., the maximum distance between an MN and its BR at which it is possible to guarantee the target success probability, assuming that no obstacle prevents the communication between them. For instance, according to the packet error model in Section V-A, if $\Pi_t = 0.75$, the maximum distance must be lower than 47 m (see Fig. 3). Instead, if an average success probability of 0.50, or even 0.25, can be accepted, then the maximum distance increases to 56 m and 67 m, respectively.

In the following, for the purposes of our analysis, we will assume the coverage of a BR as a disk with a radius equal to the maximum distance corresponding to the target success probability Π_t required by the application. Inside this disk, the success probability experienced by packets is at least equal to Π_t. Hence, our goal is to cover the whole area with the minimum number of disks (i.e., BRs). However, the presence of obstacles changes the shape of the area covered by a BR, as we assume that the BR coverage is limited to the area not obstructed by obstacles.

In literature, several deployment policies have been proposed to trade-off coverage versus number of BRs. In our analysis, we consider four different policies that are briefly described below.

Kershner's theorem [17] provides the lower bound on the number of disks (BRs) required to fully cover the deployment area. The optimal deployment is obtained by geometrically filling the area with equilateral triangles and, then, by placing the BRs on their vertexes. The assumption is that the disk radius is very small, if compared to the size of the deployment area. Since we are considering disks with a maximum radius of 67 m and deployment areas in the order of hundreds of meters, this assumption does not hold in our scenario. Moreover, Kershner's theorem assumes that no obstacles are present. To cope with obstacles, we divide the deployment area in rectangular regions, each one of which does not include any obstacle. On each region, we apply the theorem to compute the minimum number of BRs required to cover that region. Then, we sum the number of BRs required for each region to obtain the total number of BRs.

Below, we will use the Kershner's theorem only as a term of comparison, when evaluating the different deployment policies, namely *Intersecting Flowers*, *Wang-Hu-Tseng* and *GeneticAlBR*.

The *Intersecting Flowers* policy [13] follows the principles of Kershner's theorem, however, it does not make any assumption on the disk size. The rationale is to divide the deployment area in equilateral triangles and place BRs at their vertexes to guarantee the full coverage, while minimizing the regions covered by two or more BRs, to avoid interference. As the original version of Intersecting Flowers does not consider obstacles, we take the same approach used for Kershner's theorem, i.e., we divide the deployment area into rectangular regions without obstacles.

The *Wang-Hu-Tseng* policy [18] assumes that the deployment area is a polygon. In addition, this solution also considers the presence of fixed polygonal obstacles in the area. It divides the deployment area in *small* and *large* regions. In small regions, BRs can be deployed in a row, while in large regions they are placed as in [17], following the equilateral triangle division of the deployment area. After this step, if there are uncovered zones at the border of the deployment area, one more BR will be added for each of them.

The last BR deployment policy considered in our analysis is based on a genetic algorithm and, hence, throughout it will be referred to as *GeneticAlBR*. It also assumes the presence of obstacles, and takes an approach similar to the one presented in [19], which instead addresses the problem of deployment of a fixed number of sensors with different communication radii for maximizing the sensing area, and does not consider the presence of obstacles. Specifically, an initial number of BRs is considered. Then, for each such initial number, possible deployment solutions are derived and evaluated using a fitness function. The latter represents the coverage fraction provided by the candidate solution, with respect to the whole area. Since the deployment area must be fully covered, the algorithm will stop when a 100% coverage is reached, otherwise the number of BRs is increased and the algorithm is re-executed. Hence, the solution with the lowest number of BRs will be selected.

1) Problem Formulation: All the considered deployment policies presented above take, as input parameters, the target success probability Π_t, the deployment area $\mathcal{A} \subset \mathbb{R}^2_+$, and the obstacles set $\Omega = \omega_0 \dots \omega_O$, where ω_i represents the region of $\mathcal{A}$ corresponding to the ith obstacle, and O is the number of obstacles. Hence, a BR deployment C can be denoted as

$$C = f(\Pi_t, \mathcal{A}, \Omega), \tag{5}$$

where $C = \{c_0, \dots, c_{B-1}\}, c_i = (x_i, y_i) \in \mathcal{A}, c_i \notin \omega_j$ for $0 \leq i \leq B-1, 0 \leq j \leq O$ is the set of coordinates of B BRs to be deployed.

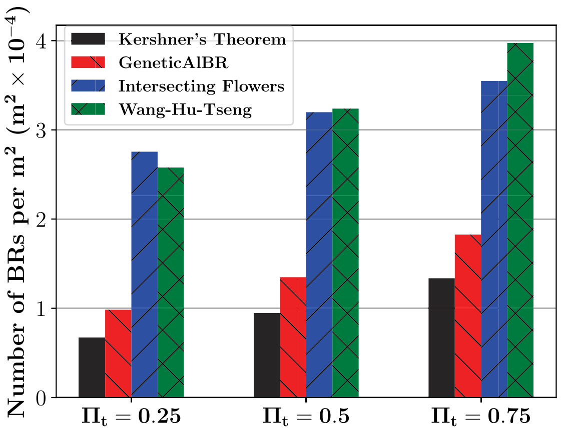


Fig. 4. Density of BRs for the considered policies.

The objective of any deployment policy is to find the minimum number of BRs, while guaranteeing the full coverage of the deployment area, even in the presence of obstacles, with a certain target success probability Π_t in packet transmissions. Let us define $x_{\max}$ as the maximum distance between any point m in the deployment area and the closest element $c \in C$ (i.e., a BR) such that the segment that connects m and c does not intersect any obstacle $\omega_i \in \Omega$.

Hence, the problem of optimal BR deployment can be formulated as follows.

$$\begin{aligned} \min \quad & \|f(\Pi_t, \mathcal{A}, \Omega)\| \\ \text{s.t.} \quad & e(x_{\max}) \geq \Pi_t, \end{aligned} \tag{6}$$

where $e(x_{\max})$ is the equation defined in (4).

To solve this problem, we need to perform the following operations with all the considered deployment policies.

1) Numerically compute the maximum acceptable value for $x_{\max}$ in meters, given Π_t, i.e., $x_{\max} = e^{-1}(\Pi_t)$, using the bisection method (with a tolerance of 0.5 mm).
2) Apply the BR deployment policy to compute a deployment C in area $\mathcal{A}$, considering the target communication range $x_{\max}$, and ensuring that any of the points in the area is covered by one BR distant at most $x_{\max}$.

2) Comparison of BR Deployment Policies: To assess which of the considered policies provides the best trade-off between coverage and number of BRs, we derived the density of BRs in the deployment area, defined as the number of BRs per square meter. For each policy, we considered different values for the target success probability Π_t (i.e., 0.25, 0.5, and 0.75). We considered deployment areas with different shapes and sizes. In particular, we used a set of industrial building maps.[1] For the sake of comparison, we scaled each map appropriately, so as to have one side of the area equal to 400 meters.

Fig. 4 shows the density of BRs, provided by the considered policies for different Π_t values. Specifically, we computed the average value of the density of BRs in all the considered deployment areas. We can observe that GeneticAlBR allows the lowest

[1][Online]. Available: https://thefloorplandesign.com/industrial/

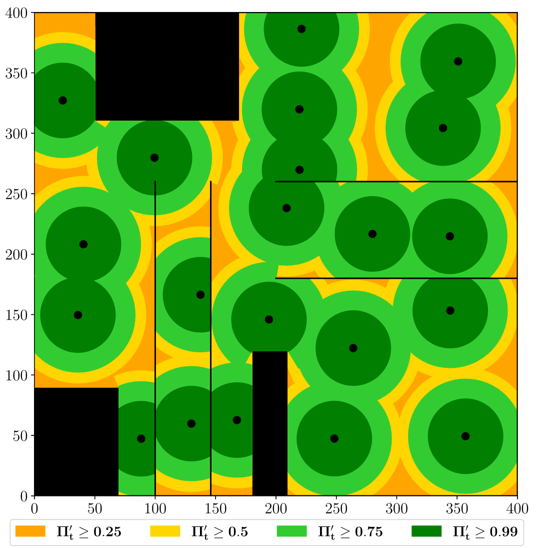


Fig. 5. Communication success probability in the reference deployment area ($\Pi_t = 0.25$, $area$: 400×400 m).

(average) density. Hence, in the following we will only consider GeneticAlBR.

It is worthwhile to highlight that, while the BR deployment provided by GeneticAlBR guarantees the target success probability Π_t over the whole area, there might be locations in which the actual success probability experienced by packets is $\Pi'_t > \Pi_t$. Fig. 5 shows the actual success probability in a specific deployment for an area of 400 m $\times$ 400 m and with $\Pi_t = 0.25$. MNs located in the orange area experience $\Pi'_t \geq \Pi_t = 0.25$ (the minimum), whereas MNs in yellow, green, and dark green zones get $\Pi'_t \geq 0.5, 0.75$ and 0.99, respectively.

VI. Analytical Model for QoS Analysis

This Section presents an analytical model to evaluate the QoS provided to MNs by the SD-DU algorithm. We consider a *worst case scenario* where *all* MNs are located inside the communication range of a single BR at a (fixed) distance $x_{\max}$ from the BR, i.e., the maximum distance that guarantees the target success probability Π_t. We considered this worst case scenario as it provides the lower bound for the QoS achieved by an MN in a real environment.

We assume that MNs and BRs are part of a wireless network, based on the 802.15.4 TSCH standard, while BRs and NC communicate through a wired network (e.g., a LAN) characterized by high data rate and high reliability. Since the wired part of the communication does not affect significantly the performance experienced by MNs, in our analysis we will focus on the wireless part only.

To make the analysis as general as possible, we consider two scenarios characterized by different traffic patterns, namely, a *Convergecast* scenario with asymmetric traffic and a *Request/Response* scenario with symmetric traffic.

In the *Convergecast* scenario, which is typical of monitoring applications, MNs send messages periodically to a collection point (*upstream* flow). Without losing in generality, we can assume that the collection point is the NC. Data may also be sent sporadically in the reverse direction, i.e., from the NC to one or more MNs (*downstream* flow). Downstream messages are used, for instance, to instruct or reconfigure MNs, and the overall downstream data rate depends on the specific application. Hence, the resulting traffic pattern is more or less asymmetric, depending on the specific use case.

In the *Request/Response* scenario, MNs still send (request) messages periodically to a certain destination node (again, we assume that the destination coincides with the NC). Upon receiving a request message, the destination node replies with a response message. This happens when using CoAP Confirmable messages [14], and in all cases where the MNs need to send and receive data continuously to/from a decision point. Hence, in this case, the upstream and downstream flows have similar requirements, i.e., the traffic pattern is symmetric.

In the following Section we analyze the performance provided by the SD-DU algorithm in the two considered scenarios, based on the *packet transmission rate* r_t, defined as the number of packets per second transmitted by a node. To this end, we define the following performance indexes: (i) *packet reception rate* r_r, defined as the number of packets per second correctly received by the destination node; (ii) *end-to-end delay*, defined as the time interval between the generation of a packet at the source node and its correct reception at the destination node.

A. Convergecast Scenario

We start our analysis with the Convergecast scenario and derive the packet transmission rate both for upstream ($r_t^{c\_up}$) and downstream ($r_t^{c\_down}$) flows. For the upstream direction, since the SD-DU algorithm allocates one cell per MN in each slotframe, the transmission rate of each MN must be less than, or equal to, the maximum transmission rate allowed by the system. Hence, given the number of MNs (M) and the value of G, the upstream transmission rate of an MN can be expressed as follows.

$$r_t^{c\_up} \leq r_{t,MAX}^{c\_up}(M,G) = \frac{1}{S(M,G)}, \tag{7}$$

where $S(M,G)$ is the slotframe length, expressed in seconds, and can be written as

$$S(M,G) = \left(1 + \left\lceil \frac{M}{G} \right\rceil + M + o\right) \cdot T_S, \tag{8}$$

where T_S is the timeslot duration (15 ms), and $o \in \mathbb{N}$ is an integer value that represents the number of timeslots that need to be added to the slotframe to guarantee that the number of available channels and the slotframe length are co-prime (see Section II).

Following the same line of reasoning for the downstream flow, we can derive the downstream transmission rate. To this end, we recall that, overall, $\lceil M/G \rceil$ timeslots are allocated for downstream communication, and each of such timeslot is shared by a group of, at most, G MNs. Hence, the (maximum) downstream data rate of the BR towards any MN in its range can be expressed as

$$r_t^{c\_down} \leq r_{t,MAX}^{c\_down}(M,G) = \frac{1}{G \cdot S(M,G)}. \tag{9}$$

In the Convergecast scenario, according to our assumptions, a packet is successfully received when it is correctly delivered from the MN to the BR (upstream flow), or from the BR to the MN (downstream flow). The BR deployment policy ensures that the (average) success probability, i.e., the probability that a packet is correctly received by the destination, is at least Π_t in any (accessible) point of the deployment area. Since all MNs are assumed to be located at the maximum distance from the BR, the packet reception rate for the upstream and downstream flows can be expressed as follows.

$$r_r^{c\_up} \leq r_{r,MAX}^{c\_up}(M,G) = \frac{1}{S(M,G)} \cdot \Pi_t \tag{10}$$

$$r_r^{c\_down} \leq r_{r,MAX}^{c\_down}(M,G) = \frac{1}{G \cdot S(M,G)} \cdot \Pi_t. \tag{11}$$

Let us now derive the end-to-end delay. Since we do not consider the retransmission of corrupted packets, below we only focus on packets correctly delivered to the destination (i.e., BR or MN). For upstream direction, the end-to-end delay, d^{c_up}, is the time interval between the generation of a message at an MN and its correct reception by the BR. Observe that the worst condition occurs when a packet is generated by the MN, in a given slotframe, immediately after its allocated upstream cell has passed. Since only one cell per slotframe is allocated to each MN, the packet will be transmitted in the next slotframe. Hence, the upstream end-to-end delay is given by

$$d^{c\_up} \leq d_{\text{MAX}}^{c\_up}(M,G) = S(M,G). \tag{12}$$

The downstream end-to-end delay, d^{c_down}, corresponds to the time interval between the arrival of a message at the BR (from the NC) and its reception by the MN. Since each downstream timeslot is shared by G MNs, to calculate the downstream end-to-end delay, we need to consider (i) the waiting time for a downstream cell (at most one slotframe, i.e., $S(M,G)$), and (ii) the waiting time in the local queue of the BR to send the $G-1$ packets destined to the other $G-1$ MNs sharing the same timeslot. In the worst case, this queuing delay is equal to $(G-1)$ slotframes. Hence, the downstream end-to-end delay can be expressed as

$$d^{c\_down} \leq d_{\text{MAX}}^{c\_down} = G \cdot S(M,G). \tag{13}$$

B. Request/Response Scenario

In the Request/Response scenario, upstream and downstream traffic flows are tightly coupled. MNs periodically send request messages to the NC through the BRs, which in turn sends back response messages. Hence, the maximum transmission rate $r_{t,MAX}^{req/res}(M,G)$ experienced by an MN is given by the minimum among the upstream and the downstream transmission rates. Following the same line of reasoning used in the Convergecast scenario, the maximum transmission rate in the upstream and downstream directions can be expressed as

$$r_{t,MAX}^{req/res-up}(M,G) = \frac{1}{S(M,G)} \tag{14}$$

$$r_{t,MAX}^{req/res-down}(M,G) = \frac{1}{G \cdot S(M,G)}. \tag{15}$$

Since the downstream component is always lower than, or equal to, the upstream one, the maximum transmission rate of an MN in the Request/Response scenario is given by

$$r_{t,MAX}^{req/res}(M,G) = \frac{1}{G \cdot S(M,G)}. \tag{16}$$

In the Request/Response scenario, a message transmitted by an MN is assumed to be successfully delivered if (i) the request message is received correctly by the BR, and (ii) the corresponding response message is received correctly by the MN. Since the BR deployment ensures a minimum transmission success probability equal to Π_t, the maximum packet reception rate $r_{r,MAX}^{req/res}$ is given by

$$r_{r,MAX}^{req/res}(M,G) = \frac{1}{G \cdot S(M,G)} \cdot \Pi_t^2. \tag{17}$$

Finally, we derive the end-to-end delay. In this scenario, it corresponds to the time interval between the generation of the request message by an MN and the reception of the corresponding response message by the same MN. Hence, the end-to-end delay is the sum of (i) the upstream delay taken by the request message to reach the BR (at most, one slotframe $S(M,G)$); (ii) the waiting time at the BR for the downstream cell in which to send the response message (at most, $(M+1) \cdot T_S$); and (iii) the queuing time at the BR in case there are response messages destined to the other $G-1$ MNs sharing the same downstream timeslot (at most, $(G-1) \cdot S(M,G)$). Hence, the (maximum) end-to-end delay can be expressed as follows.

$$d_{\text{MAX}}^{req/res} = G \cdot S(M,G) + (M+1) \cdot T_S. \tag{18}$$

We observe that, if G=1 (i.e., when SD-DU degenerates to DD-DU), the previous equation reduces to

$$d_{\text{MAX}}^{req/res} = S(M,G) + (M+1) \cdot T_S. \tag{19}$$

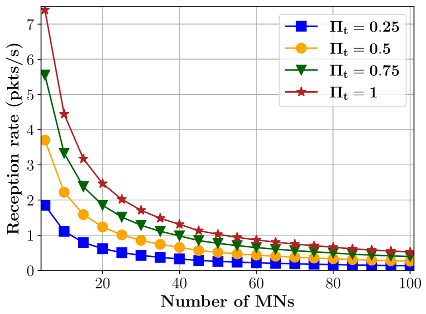
In this special configuration (i.e., DD-DU), if each downstream cell is allocated just after the corresponding upstream cell, as shown in Fig. 2, the maximum end-to-end delay reduces further, and can be expressed as follows.

$$d_{DD\text{-}DU}^{req/res} = S(M,G) + T_S. \tag{20}$$

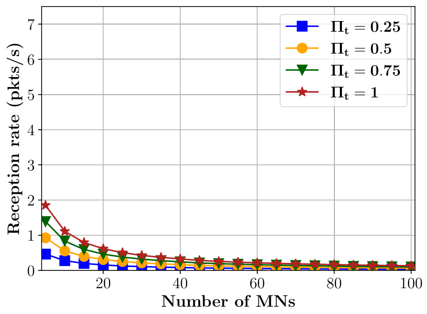
This optimization can be very useful for time-sensitive applications, where the end-to-end delay should be as low as possible.

C. Analytical Results

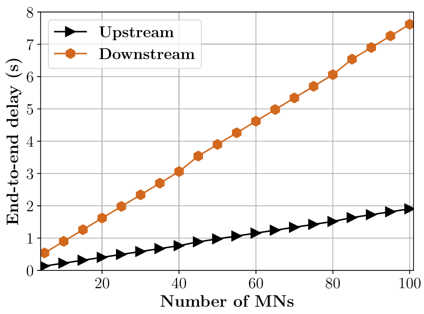
In the following, we exploit the analytical formulas derived in the previous Sections to assess the performance provided by SD-DU with different G and Π_t values, in the two considered scenarios, for an increasing number of MNs. In our analysis, we assume that the packet transmission rate r_t of each MN is equal



(a) Upstream reception rate

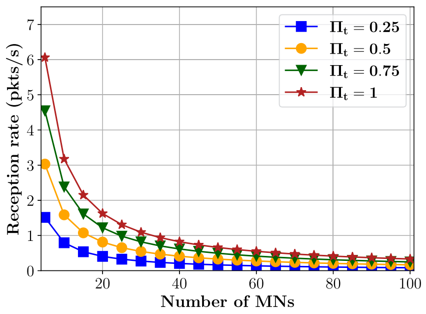


(b) Downstream reception rate

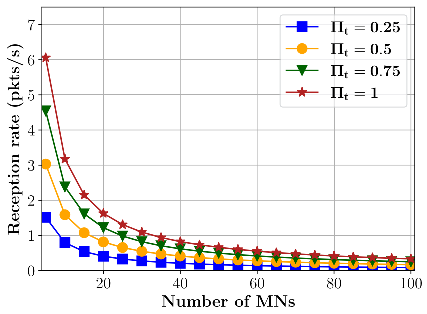


(c) Upstream and downstream delays

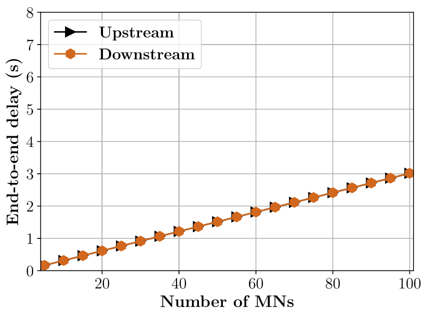
Fig. 6. Packet reception rate and end-to-end delay. Convergecast scenario, G = 4.



(a) Upstream reception rate



(b) Downstream reception rate



(c) Upstream and downstream delays

Fig. 7. Packet reception rate and end-to-end delay. Convergecast scenario, G = 1.

to the maximum packet transmission rate, $r_{t,MAX}(M, G)$. Similarly, we consider the maximum end-to-end delay $d_{\text{MAX}}(M, G)$.

Here we only report the results obtained with $G = 4$ (SD-DU-4) and $G = 1$ (SD-DU-1, or DD-DU). The results obtained with other values larger than 1 show a trend similar to that observed with $G = 4$ and are, thus, omitted for brevity. The case with $G = 1$, instead, is worth of a separate discussion, since the behavior of the SD-DU algorithm significantly changes when dedicated cells are allocated for both downstream and upstream traffic.

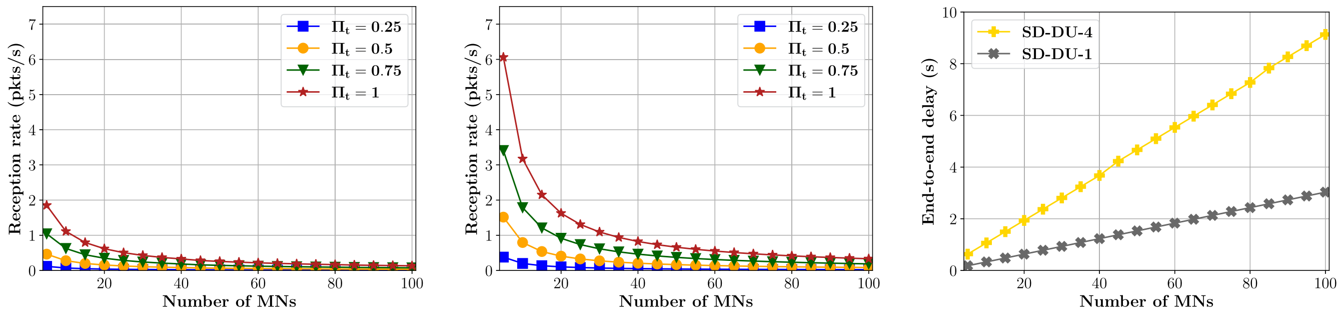
Fig. 6 shows the performance in the Convergecast scenario with $G = 4$ (SD-DU-4). We can observe that the upstream reception rate decreases rapidly as the number of MNs increases (Fig. 6(a)). This is because the slotframe length increases with the number of MNs, as we need to allocate more cells, and a longer slotframe reduces the number of transmission opportunities in the same amount of time. Also, as expected, the upstream reception rate increases with Π_t: the higher the target success probability, the higher the (upstream) reception rate.

The downstream packet reception rate, shown in Fig. 6(b), exhibits a trend similar to the upstream rate. However, the former is significantly lower, even when a small number of MNs is considered. This is due to the fact that only $\lceil M/G \rceil$ downstream timeslots are allocated for each slotframe, which results in the possibility to receive downstream packets every G slotframes for each MN.

Fig. 6(c) shows both the upstream and downstream component of the end-to-end delay. Only a single curve for each direction is shown, since the target success probability does not have an impact on the delay, as shown in (12) and (13), because corrupted packets are not retransmitted. It is worth to highlight that both the components of the end-to-end delay are influenced only by the number of MNs in the network and the G parameter, that also determine the slotframe size (see (8)). In fact, as expected, both directions of the delay increase with the number of MNs. In addition, the downstream component is significantly larger than the upstream one, as each downstream timeslot is shared by G MNs.

Fig. 7 shows the results obtained with $G = 1$. In this configuration, the upstream reception rate (Fig. 7(a)) exhibits a trend similar to the one observed with $G = 4$ (Fig. 6(a)), however, the corresponding rates are now lower. This is because, a lower G value results in a longer slotframe. Instead, the downstream reception rate is now higher than before since each downstream timeslot is dedicated to a single MN (Fig. 7(b)). In terms of delay (Fig. 7(c)), in this configuration, the upstream and downstream components are completely overlapped, as expected.

Based on the results presented above, we can conclude that in a Convergecast scenario, where the traffic pattern is asymmetric and the downstream flow is significantly lower than the upstream one, sharing downstream timeslots among MNs (i.e., using a G value larger than 1) allows a more efficient bandwidth utilization, resulting in higher rates and lower delays for upstream communication. Obviously, the selection of the



(a) Reception rate, G = 4 (b) Reception rate, G = 1 (c) End-to-end delay

Fig. 8. Packet reception rate and end-to-end delay. Request/Response scenario.

appropriate G value depends on the specific use case under consideration.

Let us analyze the performance in the Request/Response scenario that are reported in Fig. 8. Specifically, the first two plots allow to compare the reception rate experienced by MNs in this scenario with $G = 4$ (Fig. 8(a)) and $G = 1$ (Fig. 8(b)), for different values of Π_t. Instead, the third plot compares the end-to-end delay, again, for $G = 4$ and $G = 1$.

We can observe that, in this scenario, sharing downstream timeslots (i.e., setting G larger than 1) does not provide any advantage. Instead, it results in a performance degradation, both in terms of packet reception rate and end-to-end delay. This is because the data traffic is bidirectional and symmetric and, hence, the same allocation policy must be used for both the upstream and downstream direction.

In conclusion, the analytical results demonstrate that the SD-DU algorithm is very flexible and can be adapted to various scenarios with different traffic patterns. Specifically, G can be tuned appropriately, depending on the amount of downstream traffic to be managed. For scenarios with sporadic downstream traffic a large G is recommended as it improves the performance in the upstream direction. Instead, when traffic is symmetric, $G = 1$ is the appropriate option, as it guarantees the same level of performance in both directions. Based on this conclusions, in the following we will use $G = 4$ in the Convergecast scenario and $G = 1$ in the Request/Response scenario.

VII. A Methodology for Network Sizing

In Section VI we defined an analytical model to assess the performance experienced by an MN, in terms of packet reception rate and end-to-end delay, depending on various parameters. In this Section, we exploit this model to define a methodology that allows to determine the maximum number of MNs that can be supported by the network, while satisfying the QoS requirements of the application.

To this end, we assume that all the MNs have the same QoS requirements that can be expressed through the following parameters: (i) minimum packet reception ratio (R^*), (ii) minimum transmission rate r_t^* and (iii) maximum end-to-end delay d^*. The packet reception ratio is defined as the fraction of packets that are correctly received by the destination and, hence, define the minimum level of reliability that can be tolerated by the application. In addition to the QoS parameters, our methodology also requires, as input parameters, the G value used by the SD-DU scheduling algorithm and the target success probability Π_t used by BR deployment algorithm (GeneticAIBR).

In order to guarantee the QoS requirements, the following conditions must hold

$$\begin{cases} R(M, G, \Pi_t) = \frac{r_r}{r_t^*} & \geq & R^* \\ r_{t,MAX}(M, G) & \geq & r_t^* \\ d(M, G) & \leq & d^* \end{cases}, \tag{21}$$

where $R(M, G, \Pi_t)$ denotes the packet reception ratio, $r_{t,\text{MAX}}$ (M, G) the maximum transmission rate, and $d(M, G)$ the end-to-end delay experienced by a generic MN, given the input parameters M, G, and Π_t.

The conditions in (21) represent a general formulation of the network sizing problem. Specific formulas for the Convergecast and Request/Response scenarios will be detailed in Sections VII-A and VII-B, respectively. Finally, in Section VII-C, we will use our methodology to calculate the maximum number of MNs, depending on the required QoS parameters, in the two considered scenarios.

A. *Convergecast Scenario*

For the Convergecast scenario we have different QoS requirements for the upstream and downstream traffic components. Hence, our methodology first calculates the maximum number of MNs that can be accommodated, namely $M_{\text{MAX}}^{c\_up}$ and $M_{\text{MAX}}^{c\_down}$, while meeting the QoS requirements in the upstream and downstream directions, respectively. Then, the maximum number of MNs that can be supported by the systems will be given by $\min(M_{\text{MAX}}^{c\_down}, M_{\text{MAX}}^{c\_up})$.

By using the formulas derived in Section VI-A, (21) can be customized to the Convergecast scenario. Specifically, for the upstream traffic, we use (7), (10) and (12), while for the downstream traffic we refer to (9), (11) and (13). Hence, the general conditions in (21) can be expressed as in (22) and (23), for upstream and downstream traffic, respectively.

$$\begin{cases} \Pi_t & \geq & R^* \\ \frac{1}{S(M_{c\_up},G)} & \geq & r_t^{c\_up*} \\ S(M_{c\_up},G) & \leq & d^{c\_up*} \end{cases} \tag{22}$$

$$\begin{cases} \Pi_t & \geq & R^* \\ \frac{1}{G\cdot S(M_{c\_down},G)} & \geq & r_t^{c\_down*} \\ S(M_{c\_down},G) & \leq & d^{c\_down*} \end{cases}. \tag{23}$$

It is important to highlight that the first condition in both (22) and (23) only depends on the Π_t parameter. Hence, the reliability requirement can be guaranteed only if the BR deployment allows an (average) success probability Π_t larger than, or equal to, the minimum packet reception ratio R^* required by the application.

Under the assumption that the the first condition in both (22) and (23) is satisfied by the BR deployment, it is possible to calculate the maximum number of MNs, $M^x_{\text{MAX}}|x \in \{c\_up, c\_down\}$, that can be supported, according to the upstream and downstream conditions. Specifically, by inverting (8) and after some algebraic manipulations, M^x_{MAX} can be expressed as follows

$$M^x_{\text{MAX}} = \left\lfloor \frac{\lfloor \frac{S^x_{\text{MAX}}}{T_S} \rfloor - 1}{\frac{1}{G} + 1} \right\rfloor - o, \tag{24}$$

where S^x_{MAX} represents the maximum slotframe length (expressed in seconds) that satisfies the application requirements, and can be expressed as follows for the upstream and downstream directions

$$S^{c\_up}_{\text{MAX}} = \min(d^{c\_up*}, 1/r_t^{c\_up*}) \tag{25}$$

$$S^{c\_down}_{\text{MAX}} = \min(d^{c\_down*}, 1/(G \cdot r_t^{c\_down*})). \tag{26}$$

Once $M^{c\_up}_{\text{MAX}}$ and $M^{c\_down}_{\text{MAX}}$ are known, the maximum number of MNs that can be supported by the system can be calculated as follows

$$M_{\text{MAX}} = \min(M^{c\_up}_{\text{MAX}}, M^{c\_down}_{\text{MAX}}). \tag{27}$$

B. Request/Response Scenario

In the Request/Response scenario the data traffic is balanced between the upstream and downstream components. Using formulas derived in Section VI-B, in this specific scenario (21) can be written as follows

$$\begin{cases} \Pi_t^2 & \geq & R^* \\ \frac{1}{S(M_{req\_res},G)} & \geq & r_t^* \\ S(M_{req\_res},G) + T_s & \leq & d^* \end{cases}. \tag{28}$$

As above, the first condition represents the reliability constraint, and it only depends on the success probability Π_t considered in the BR deployment (in this scenario it depends on Π_t^2 since a message transmission is successful only if both the request and response messages are correctly received). Following the same approach used for the Convergecast scenario, the maximum number of MNs $M^{req\_res}_{\text{MAX}}$ that can be supported by the system can be computed as

$$M^{req\_res}_{\text{MAX}} = \left\lfloor \frac{1}{2}\left(\frac{S^{req/res}_{\text{MAX}}}{T_s} - 1\right)\right\rfloor - o, \tag{29}$$

where

$$S^{req/res}_{\text{MAX}} = \min(d^* - T_s, 1/r_t^*). \tag{30}$$

TABLE I
Maximum Number of MNs Supported by the Network

Maximum delay(s)	Minimum transmission rate (pkt/s)						
	$r_t^{c\_up*}$					$r_t^{c\_down*}$	
	2	**1**	**1/2**	**1/4**	**1/8**	**1/4**	**1/8**
0.5	25	25	25	25	25	25	25
1	25	52	52	52	52	52	52
1.5	25	52	79	79	79	52	79
2	25	52	105	105	105	52	105
2.5	25	52	105	132	132	52	105

Convergecast scenario: G = 4, πt = 0.75.

C. Network Sizing

This Section uses the equations derived in Sections VII-A and VII-B to calculate the maximum number of MNs that can be accommodated in the system, depending on the QoS requirements, in the Convergecast and Request/Response scenarios.

In the Convergecast scenario, we assume that SD-DU is used with G $=$ 4 (SD-DU-4), and that the BR deployment allows a target success probability $\Pi_t = 0.75$. Hence, a maximum packet reception ratio of 75% can be guaranteed, irrespective of the number of MNs. Leveraging the system of inequalities derived in Section VII-A, Table I shows the maximum number of MNs that can be supported by the system, while satisfying the minimum transmission rate and the maximum delay required by the application. For the transmission rate, both the upstream and downstream rates need to be taken into account. In Table I, we consider lower transmission rates for the downstream flow, with respect to the upstream flow, since in the Convergecast scenario the downstream traffic is typically limited or sporadic.

The results in Table I show that, when the required (upstream) transmission rate $r_t^{c\_up*}$ is high (e.g., 1-2 pkt/s), M_{MAX} is not influenced, or is only marginally influenced, by the maximum tolerated delay, since the requirement on the transmission rate is predominant. Instead, when $r_t^{c\_up*}$ is lower, the delay requirement becomes relevant. In general, a less stringent requirement in terms of maximum delay, allows to accommodate more MNs.

In the the Request/Response scenario, we assume that SD-DU is used with G $=$ 1 (SD-DU-1, or DD-DU) and, again, we consider a BR deployment with $\Pi_t = 0.75$. Hence, according to (28), the highest packet reception ratio guaranteed by the system is approximately 56%. Table II shows the maximum number of MNs that can be accommodated, in this scenario, depending on the QoS requirements. We can observe that, given certain requirements (i.e., minimum transmission rate and maximum end-to-end delay), a lower number of MNs can be supported, in comparison with the Convergecast scenario. This is because the data traffic is balanced between the upstream and downstream

TABLE II
MAXIMUM NUMBER OF MNS SUPPORTED BY THE NETWORK

	Minimum transmission rate (pkt/s)				
Maximum delay(s)	**2**	**1**	**1/2**	**1/4**	**1/8**
0.5	15	15	15	15	15
1	16	32	32	32	32
1.5	16	32	49	49	49
2	16	32	65	65	65
2.5	16	32	66	82	82

Request/response scenario: G = 1, πt = 0.75

TABLE III
SIMULATION PARAMETER SETTINGS

Duration of each replica (s)	1000
Initial portion discarded (s)	100
Number of independent replicas	35
Number of MNs	10 to 150, step = 10
Packet generation rate (pkt/s)	0.5
MN speed (m/s)	0.5, 2, 5
Π_t	0.25, 0.75
Number of BRs	21, 37
Communication range of BRs (m)	66.9, 47.2

direction and, hence, the SD-DU must allocate the same number of cells for both flows, thus consuming more bandwidth.

VIII. PERFORMANCE EVALUATION

The analytical results presented in the previous Sections were based on the worst-case scenario introduced in Section VI assuming that all the MNs are placed within the communication range of a single BR, at the maximum distance x_{max}. In this section, we carry out a simulation analysis of the proposed framework in order to validate the analytical results presented above and, in addition, investigate its performance compared to existing scheduling algorithms and in more realistic conditions.

A. Simulation Setup

For our simulation analysis, we developed an OMNeT++[2] based network simulator, called *Mobile6TiSCH*[3] [20]. Mobile6TiSCH implements the full protocol stack defined for the 6TiSCH architecture [2]. In addition, it implements the SHMG architecture, and the SD-DU scheduling algorithm.

To compare the performance of the proposed SD-DU scheduling with existing scheduling algorithms, we also implemented in Mobile6TiSCH Orchestra [21], ALICE [22], and AMUS [23]. To mimic a realistic communication environment, we introduced in the simulator the packet error model described in Section V-A.

Finally, to assess the impact of mobility on the performance achieved by MNs, we considered three different mobility patterns, namely, *static*, *linear*, and *random*. In all the mobility patterns, each MN starts at an initial position, randomly selected within the deployment area. Then, its position changes over time according to the considered mobility pattern. In the *static* pattern, the position remains unchanged over time. In the *linear* pattern, the MN moves from its initial position, at a constant speed, following an horizontal or vertical line. When it reaches the border of the deployment area or an obstacle, it inverts its direction and moves back towards the opposite border. Finally, in the *random* pattern, starting from the current position, the MN selects a random speed and target point and, then, moves towards the target point following a straight line.

The parameter settings used in our simulation experiments are summarized in Table III. Specifically, we assumed that the deployment area is the one represented in Fig. 5, and we considered different BR deployments, generated by the GeneticAlBR algorithm with different values of the target transmission probability Π_t. For brevity, in the following we only show the results obtained with $\Pi_t = 0.25$ and $\Pi_t = 0.75$ (the trend is the same when considering other Π_t values). The number of BRs deployed (and their communication range) is 21 (66.9 m) when $\Pi_t = 0.25$, and 37 (47.2 m) when $\Pi_t = 0.75$.

To assess the performance of the proposed approach under different scenarios, we conducted experiments using various deployment areas, i.e., deployment areas featuring different shapes, sizes, and obstacle configurations. Moreover, we also considered scenarios involving the misplacement of BRs from their intended positions, i.e., randomly relocating them within a 10-meter range, to assess how deployment errors can affect the overall performance. Such experiments, omitted due to space limitations, reported results in-line with the ones presented in this paper, showing the solidity of the proposed approach. Specifically, the former experiments yielded uniform results, while the latter displayed consistent outcomes with only a slight degradation in performance when BRs are mispositioned.

MNs are programmed to generate data packets periodically, with a period of 2 seconds, resulting in a data rate of 0.5 pkt/s. We also considered different data rates (i.e., 0.25 pkt/s and 1 pkt/s). However, we observed similar trends in the results and, for the sake of space, we only present the results for 0.5 pkt/s.

Furthermore, we evaluated the performance of the network under a bursty traffic model, where MNs generate packets in batches. To ensure a fair comparison, we considered the same average packet generation rate used in scenarios with periodic traffic (i.e., 0.5 pkt/s). Specifically, we configured MNs to produce 5 packets every 10 seconds and 10 packets every 20 seconds.

Finally, MNs move at a constant speed, according to the considered mobility pattern. In our experiments, we considered three different speed, namely, 0.5 m/s, 2 m/s, and 5 m/s.

In our simulation experiments, we measured the same performance indices considered in the previous Sections, namely the *packet reception ratio (PRR)* and *end-to-end delay* experienced by each MN. For the end-to-end delay, we plot the 95th percentile. To obtain statistically sound results, for each simulation experiment we run 35 independent replicas, each with a duration of 1000 s. We derived confidence intervals with 95% confidence level.

[2]https://omnetpp.org/
[3]https://github.com/marcopettorali/Mobile6TiSCH

B. Simulation Results

This section presents the simulation results obtained in the Convergecast and Request/Response scenarios. In particular, we first compare the proposed SD-DU algorithm with existing scheduling algorithms, namely, Orchestra, ALICE and AMUS (Section VIII-B1). Then, we assess the impact of the mobility pattern and speed on the performance achieved by MNs (Section VIII-B2). Finally, we investigate the impact of bursty traffic (Section VIII-B3).

1) Comparison of Scheduling Algorithms: In this set of experiments, we configured the SD-DU scheduling algorithm to use G $= 4$ in the Convergecast scenario, and G $= 1$ in the Request/Response scenario, as above. Hence, in the Convergecast scenario, each MN has one dedicated timeslot for upstream traffic and one timeslot shared with other G-1 MNs for downstream traffic. For the sake of brevity, in the Convergecast scenario, we only considered the upstream traffic component, since downstream traffic is sporadic and does not show any particular trend worth to be analyzed in detail. Instead, in the Request/Response scenario each MN has two dedicated timeslots, one for upstream and one for downstream traffic. Below, we describe the allocation performed by the other considered scheduling algorithms. They were initially conceived for multi-hop stationary networks. Since we refer to the SHMG architecture, we adapted them to this specific environment to ensure a proper comparison. For each algorithm, we assume that the cell allocation is generated by the NC and is then replicated on all the BRs.

In Orchestra [21], cells are allocated by the NC using a hash function with the addresses of the MNs and the BRs as input parameters. We considered the sender-based approach, as it is more efficient than the receiver-based one [21]. Accordingly, the algorithm allocates one dedicated cell per MN for upstream traffic, which timeslot and channel offset are determined by computing the hash function of the MN's MAC address. Moreover, a single shared cell for all the MNs is scheduled for downstream traffic, which is determined by exploiting the BRs' MAC address. It may be worthwhile emphasizing that, dedicated cells may be allocated to different MNs on the same timeslot. Hence, if two or more MNs with a dedicated cell in the same timeslot move under the same BR, the latter can receive data from only one MN, thus resulting in packets being discarded even under ideal communication conditions.

Also in ALICE [22], cells are allocated by the NC using a hash function with the MN and BR addresses as input parameters. Unlike Orchestra, ALICE allocates two shared cells for each MN, one for upstream traffic and one for downstream traffic, respectively. As above, in case two or more MNs with their upstream cells allocated on the same timeslot happen to move under the same BR, traffic originated only from one MN is correctly received by the BR.

Finally, with AMUS [23] – which was already designed as a centralized scheduling algorithm – the NC allocates one dedicated upstream cell and one dedicated downstream cell, both on different timeslots, for each MN. It is worth to highlight that, in the considered single-hop scenario, AMUS behaves in the same manner as SD-DU with G $= 1$ (i.e., DD-DU).

As a final remark, since in our approach data packets are not re-transmitted when dropped, as a periodic generation of packets is considered, initially we disabled re-transmissions also for Orchestra, ALICE, and AMUS, to ensure a fair comparison. However, we also carried out additional experiments with re-transmissions enabled for Orchestra, ALICE, and AMUS (see below).

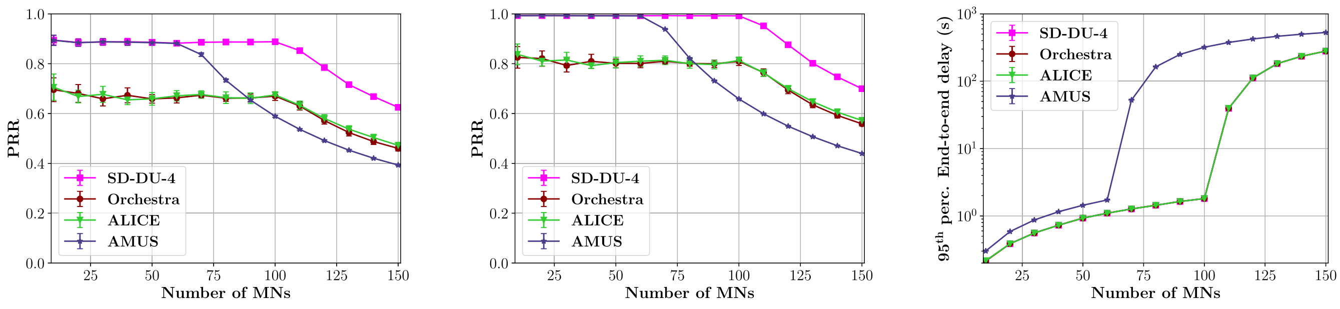
Fig. 9 shows the PRR and end-to-end delay in the Convergecast scenario experienced by MNs, for the considered scheduling algorithms, when the mobility pattern is linear and MNs move at 2 m/s. As it can be seen, the end-to-end delay does not depend on the target success probability Π_t, for all the considered algorithms, as corrupted packets are not re-transmitted (this is because we show a single plot). Instead, the PRR depends significantly on Π_t, as expected.

If we compare the results obtained with the different scheduling algorithms, we can notice that SD-DU provides the best performance, in terms of both PRR and delay. AMUS exhibits a behavior similar to SD-DU, however, its performance starts degrading before than SD-DU. This is because, AMUS basically behaves exactly as SD-DU with G $= 1$. Hence, it generates a slotframe longer than that generated by SD-DU with G $= 4$. This results in a slightly larger delay, even when the number of MN is limited. Also, in the Convergecast scenario, allocating one dedicated downstream slot per MN results in an inefficient resource allocation, which is responsible for the lower PRR, in comparison with SD-DU, when the number of MNs increases over a certain threshold.

Orchestra and ALICE exhibit the same end-to-end delay as SD-DU, while the PRR is significantly lower. This is because the dedicated upstream cells allocated to different MNs, by both Orchestra and ALICE, in some cases may be scheduled on the same timeslot, as cannot occur in SD-DU or AMUS. Hence, when two or more MNs have a cell allocated in the same timeslot, if they move under the same BR, only a single packet can be received. This justifies the same performance exhibited by Orchestra and ALICE and the lower PRR, in comparison with SD-DU and AMUS.

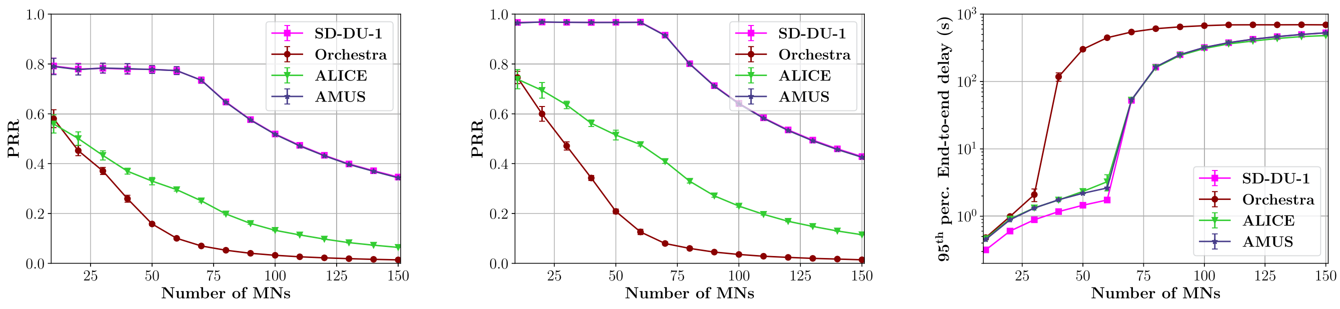
Focusing on SD-DU, we can observe that even when $\Pi_t = 0.25$, the PRR is approximately 90%, and grows up to approximately 100% when $\Pi_t = 0.75$. Regardless of the Π_t value, the PRR remains constant until the number of MNs exceeds the value of 105. Then, it decreases drastically. Similarly, the end-to-end delay gradually grows up until the number of MNs stays below 105 and, then, suddenly spikes (note that the scale in Fig. 9(c) is logarithmic). Please note that 105 is the maximum number of MNs that can be accommodated in the system, for this parameter settings, according to our sizing methodology (see Table I). Hence, the simulation results are fully compliant with our analytical predictions. When exceeding this maximum number, the SD-DU algorithm (with G=4) is not able to guarantee a transmission rate of 0.5 pkt/s to all the MNs.

Fig. 10 shows the PRR and end-to-end delay for the Request/Response scenario. In this scenario SD-DU is configured with G $= 1$. We can observe that the general trend is similar to the one in the Convergecast scenario. However, in the same conditions, in the Request/Response scenario MNs experience a



(a) Packet Reception Ratio, $\Pi_t = 0.25$ (b) Packet Reception Ratio, $\Pi_t = 0.75$ (c) End-to-end delay

Fig. 9. Impact of the Scheduling Function in the Convergecast scenario (Linear mobility, 2 m/s, $r_t = 0.5$ pkt/s).



(a) Packet Reception Ratio, $\Pi_t = 0.25$ (b) Packet Reception Ratio, $\Pi_t = 0.75$ (c) End-to-end delay

Fig. 10. Impact of of the Scheduling Function in the Request/Response scenario (Linear mobility, 2 m/s, $r_t = 0.5$ pkt/s).

lower performance, in terms of both PRR and end-to-end delay. This is because, now, we need to consider both the request and the response messages.

If we compare the results of the different algorithms, we can notice that SD-DU and AMUS now exhibits the same performance. This is because, as emphasized above, AMUS behaves exactly as SD-DU with G = 1. The slight difference in terms of end-to-delay is due to the optimization in the allocation of downstream cells performed in SD-DU, as also discussed in Section IV-B.

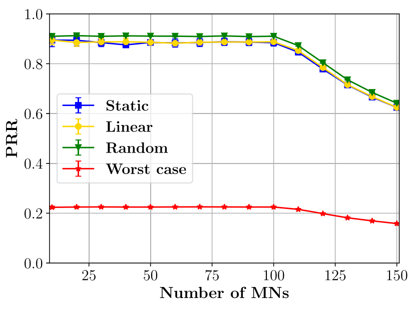
If we analyze the results obtained with Orchestra, we can notice a rapid drop in the PRR. This is because only a single shared cell is allocated to all the MNs in the downstream direction. For the same reason, the end-to-end delay is significantly higher than the one obtained with SD-DU and AMUS, as downstream packets are queued at the BRs, waiting for transmission.

With ALICE the performance is slightly better w.r.t. Orchestra. The PRR, however, is still significantly lower than the one obtained with SD-DU and AMUS. It may be worthwhile recalling that ALICE allocates one dedicate downstream cell per MN, however, the same timeslot may be allocated to different MNs (as cannot occur in SD-DU and AMUS). Hence, conflicts may arise when two, or more, MNs happen to be under the same BR, thus resulting in packet dropping.

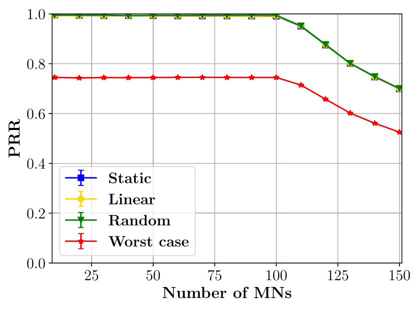
Focusing on SD-DU, we can observe that, when $\Pi_t = 0.75$, the PRR is very close to 100% and the end-to-end delay is in the order of few seconds. The maximum number of MNs that can be supported by the system, while guaranteeing the required transmission rate (0.5 pkt/s), reduces to 66, because now $G = 1$ and, thus, a dedicated slot per MN is allocated, both for upstream and downstream. Again, this value is exactly the same provided by our sizing methodology (see Table II).

As anticipated, we also conducted additional simulation experiments in which Orchestra, ALICE, and AMUS are configured with retransmissions enabled. Such results, omitted here due to space constraints, revealed only minor differences with respect to the previous results without retransmissions. Specifically, in both the Convergecast and Request/Response scenarios, Orchestra and ALICE show a PRR slightly better than SD-DU, when the number of MNs is limited (e.g., ≤ 40), albeit with a significant increase in the end-to-end delay. As the number of MNs increases, the performance deteriorates rapidly. This decline is due to the retransmission of packets that accumulate in the buffer of MNs, especially with a higher number of MNs. AMUS's performance remains unchanged when retransmissions are introduced. This is because the allocation of *tentative cells* can not be enabled due to a lack of available free cells in the schedule.

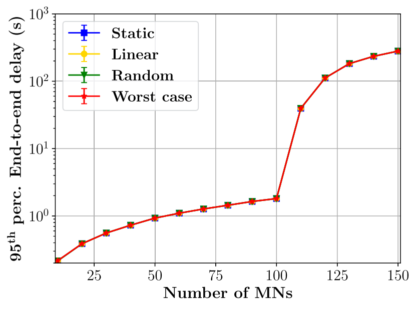
In conclusion, the simulation experiments presented above confirm the analytical results discussed in Section VI-C. In addition, our results show that SD-DU provides the best performance, both in terms of PRR and end-to-end delay, in all the considered scenarios. Our results also confirm that Orchestra and ALICE are not suitable for mobile industrial applications,



(a) Packet Reception Ratio, $\Pi_t = 0.25$

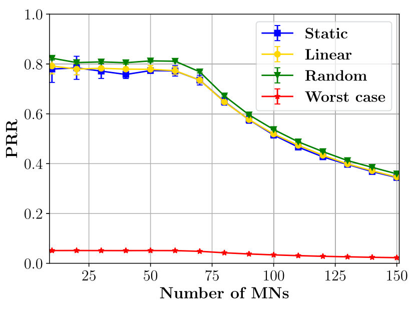


(b) Packet Reception Ratio, $\Pi_t = 0.75$

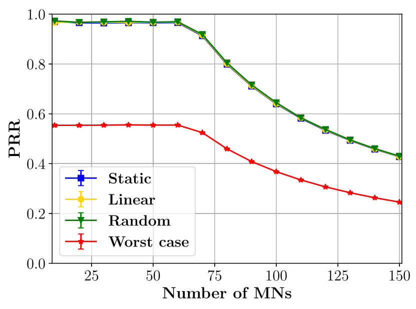


(c) End-to-end delay

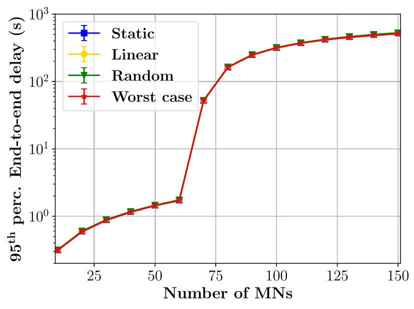
Fig. 11. Impact of Mobility in the Convergecast scenario (G = 4, speed = 2 m/s, $r_t = 0.5$ pkt/s).



(a) Packet Reception Ratio, $\Pi_t = 0.25$

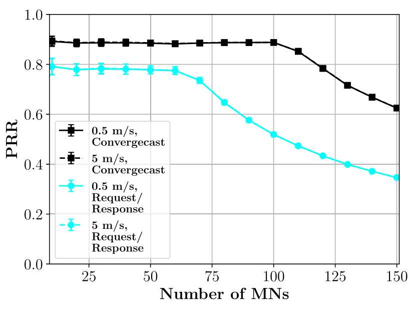


(b) Packet Reception Ratio, $\Pi_t = 0.75$

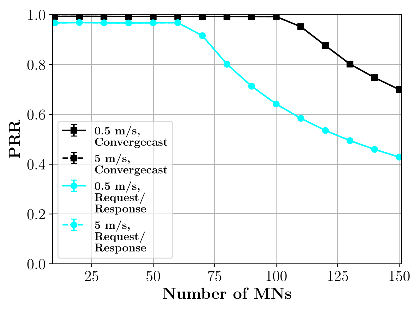


(c) End-to-end delay

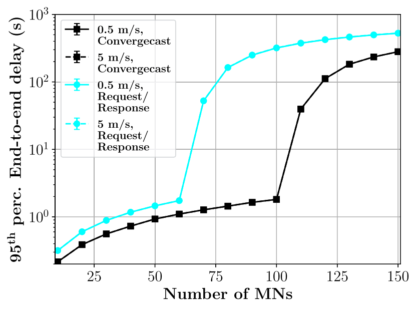
Fig. 12. Impact of Mobility in the Request/Response scenario (G = 1, speed = 2 m/s, $r_t = 0.5$ pkt/s).



(a) Packet Reception Ratio, $\Pi_t = 0.25$



(b) Packet Reception Ratio, $\Pi_t = 0.75$



(c) End-to-end delay

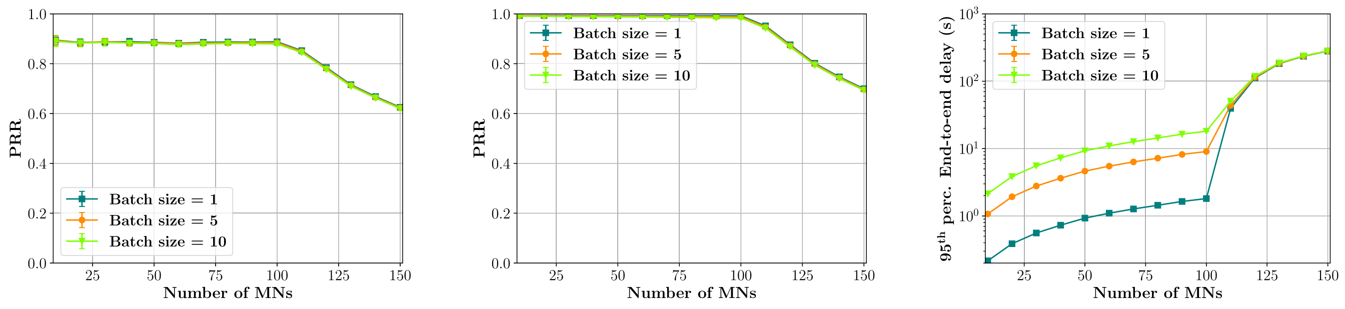
Fig. 13. Impact of MN speed (Linear mobility, $r_t = 0.5$ pkt/s).

especially when the traffic pattern is symmetric. In the considered scenarios, AMUS behaves similarly to SD-DU, however, the latter offers higher performance when Convergecast traffic is considered, thus proving to be both efficient and flexible in the considered scenarios. Considering that SD-DU offers the best performance, in the reminder of the analysis, we focus only on SD-DU.

2) Impact of Mobility: In this Section, we investigate the impact of mobility on the performance of MNs, assuming SD-DU scheduling. We consider the three mobility patterns described in Section VIII-A, namely static, linear, and random. In addition, we analyze the impact of MN speed, which varies from 0.5 m/s to 5 m/s. As above, we consider both the Convergecast and Request/Response scenarios, with $\Pi_t = 0.25$ and $\Pi_t = 0.75$.

Fig. 11 shows the PRR and end-to-end delay experienced by MNs, for the three considered mobility patterns, in the Convergecast scenario. For comparison, we also simulated the worst-case scenario considered in our analysis.

As above, we only show a single plot for the end-to-end delay, as it does not depend on the target success probability. The results in Fig. 11 also show that, in the Convergecast scenario, the mobility pattern of MNs has no significant impact on the performance.



(a) Packet Reception Ratio, $\Pi_t = 0.25$ (b) Packet Reception Ratio, $\Pi_t = 0.75$ (c) End-to-end delay

Fig. 14. Impact of batch size in the Convergecast scenario with burst traffic (G = 4, Linear mobility, speed = 2 m/s).

When $\Pi_t = 0.25$, it is possible to observe that the static and linear mobility patterns result in a slightly lower PRR than that obtained with random mobility. This is due to the presence of obstacles in the deployment area that can lead to areas where the quality of communication is poor. With static mobility a MN that is placed initially in one of such areas cannot move to an area with better coverage, with linear mobility, instead, the likelihood of leaving the area is very low. With random mobility, instead, a MN likely moves in a region with better coverage after a certain period in the simulation. It is worth to highlight that the impact of mobility is reduced as shown by our results thanks to the optimal BR deployment, which significantly reduces the areas with poor coverage.

Fig. 12 shows the impact of the mobility pattern in the Request/Response scenario. The same conclusions drawn for the Convergecast scenario apply. The only difference is that the performance values are lower, in comparison with the Convergecast scenario, due to the symmetric traffic flow.

Finally, we investigate the impact of MN speed. Fig. 13 shows the PRR and end-to-end delay for two different speeds, both in the Convergecast and Request/Response scenarios, for $\Pi_t = 0.25$ and $\Pi_t = 0.75$. For the sake of space, we only show the results obtained with linear mobility. However, we observed the same behavior with the random mobility pattern. The trend, for both PRR and end-to-end delay, is the same as that in the previous experiments. The curves related to different speeds are completely overlapped in all the considered scenarios, meaning that the MN speed has no significant impact on the performance. In fact, with linear mobility, even with very low and very high speeds, MNs are not able to escape areas with low quality of communication. Instead when using random mobility, even if the MNs move slowly, they have a chance of leaving such areas and moving towards areas with better coverage.

The simulation results presented so far highlight that, when using our proposed framework, mobility has no significant impact on the performance achieved by MNs, at least in the considered scenarios. This is due to a number of reasons. First, the GeneticAlBR allows a complete coverage of the deployment area, even in the presence of obstacles, and ensures a minimum level of communication reliability, regardless of the MN position in the area. In addition, all BRs are synchronized and, hence the handover is simplified. Finally, the SD-DU scheduling algorithm allocates TSCH cells in such a way to guarantee the transmission rate of each MN (up to a maximum number of MNs) and the schedule is installed on all the BRs. Hence, communication resources are immediately available when the MN moves into the area covered by a new BR.

3) Impact of Traffic Model: In this section, we assess the performance achieved by MNs when packets are generated in batches, rather than individually. We keep an average packet generation rate of 0.5 pkt/s and consider three scenarios with different batch sizes: (i) a single packet generated every 2 seconds; (ii) 5 packets every 10 seconds; (iii) 10 packets every 20 seconds.

In Fig. 14, we present the results for the Convergecast scenario with linear mobility and MNs moving at a speed of 2 m/s. These results highlight that the PRR is not affected by the batch size. In fact, when the number of MNs is below 110, all packets are eventually transmitted, thanks to the capability of SD-DU to guarantee the average transmission rate of 0.5 pkt/s. However, when the batch size increases, packets tend to accumulate in the transmission queue of MNs and the delay increases accordingly, as shown in Fig. 14(c). As above, we present only one plot for the end-to-end delay, as it remains unaffected by the target success probability. We also conducted experiments with the Request/Response scenario, which yielded coherent results (omitted for space limitations).

IX. Conclusion and Future Work

In this paper, we have proposed a general framework for the efficient management of node mobility in industrial wireless networks based on the TSCH protocol. Our framework is based on the SHMG architecture, but extends it to be applicable to real contexts. Specifically, we have presented a methodology to derive the optimal deployment of BRs in a sensing area with obstacles, while ensuring a minimum level of communication reliability at any (accessible) location within the area. Furthermore, we have defined a flexible scheduling algorithm to allocate communication resources to MNs in such a way to ensure the QoS requirements of the application, in different application domains. Finally, we have proposed a methodology to derive the maximum number of MNs that can be supported by the network while satisfying the QoS requirements. The proposed solutions have been evaluated through both analysis and simulation. In particular, simulation experiments have been used to compare the proposed scheduling algorithm with existing algorithms and to investigate the impact of mobility on the QoS achieved by

MNs in a realistic environment with obstacles and unreliable communication.

Our results have shown that the proposed scheduling algorithm, coupled with the methodology for network sizing, allows to provide the QoS required by the application. In addition, the proposed framework allows to manage the handover of MNs very quickly, and mobility has no significant impact on the QoS experienced by mobile node. These achievements are very important, especially in industrial environments, where applications have stringent QoS requirements and do not tolerate service interruptions. On the other hand, they are obtained by replicating the communication schedule on all the BRs, so that communication resources are immediately available when the MN moves to a new BR. Hence, the cost to pay is the high consumption of communication resources. This cost may not be justified for common applications, but can be be acceptable for industrial applications with stringent QoS constraints.

To reduce this cost, as a future work we intend to investigate scheduling algorithms that can reduce high overhead of our proposed solution. Possible research directions include (i) to relax the deterministic approach used by the SD-DU algorithm, possibly leveraging the high capacity of TSCH networks due to multi-channel communication, and (ii) to exploit information about the mobility of nodes in order to allocate communication resources at run-time. In addition, we plan to investigate the security aspects of the SHMG architecture and how to ensure time synchronization among BRs, which are not considered in this paper. In conclusion, we intend to carry out the implementation and performance evaluation of the architecture in an actual testbed. This will allow us to closely examine the network's behavior in presence of real-world factors, such as external interference and multipath fading.

REFERENCES

[1] A. Hazra, M. Adhikari, T. Amgoth, and S. N. Srirama, “A comprehensive survey on interoperability for IIoT: Taxonomy, standards, and future directions,” *ACM Comput. Surv.*, vol. 55, no. 1, Nov. 2021, Art. no. 9.

[2] P. Thubert, “An architecture for IPv6 over the time-slotted channel hopping mode of IEEE 802.15.4 (6TiSCH),” RFC 9030, p. 57, May 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9030

[3] “IEEE standard for low-rate wireless networks,” IEEE Std 802.15.4–2020 (Revision of IEEE Std 802.15.4-2015), pp. 1–800, 2020, doi: 10.1109/IEEESTD.2020.9144691.

[4] Z. Ming and M. Xu, “NBA: A name-based approach to device mobility in Industrial IoT networks,” *Comput. Netw.*, vol. 191, 2021, Art. no. 107973.

[5] J. Haxhibeqiri, A. Karaağaç, I. Moerman, and J. Hoebeke, “Seamless roaming and guaranteed communication using a synchronized single-hop multi-gateway 802.15.4e TSCH network,” *Ad Hoc Netw.*, vol. 86, pp. 1–14, 2019.

[6] H. Farag, P. Österberg, M. Gidlund, and S. Han, “RMA-RP: A reliable mobility-aware routing protocol for Industrial IoT networks,” in *Proc. IEEE Glob. Conf. Internet Things*, 2019, pp. 1–6.

[7] 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 Des. Implementation*, New York, NY, USA, 2021, pp. 234–246.

[8] M.-J. Kim and S.-H. Chung, “Efficient route management method for mobile nodes in 6TiSCH network,” *Sensors*, vol. 21, no. 9, pp. 1–16, 2021, doi: 10.3390/s21093074.

[9] K. Manikannan and V. Nagarajan, “Optimized mobility management for RPL/6LoWPAN based IoT network architecture using the firefly algorithm,” *Microprocessors Microsystems*, vol. 77, 2020, Art. no. 103193.

[10] 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.

[11] A. Mohammadsalehi, B. Safaei, A. M. H. Monazzah, L. Bauer, J. Henkel, and A. Ejlali, “ARMOR: A reliable and mobility-aware RPL for mobile Internet of Things infrastructures,” *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1503–1516, Jan. 2022.

[12] H. Kim, H.-S. Kim, and S. Bahk, “MobiRPL: Adaptive, robust, and RSSI-based mobile routing in low power and lossy networks,” *J. Commun. Netw.*, vol. 24, no. 3, pp. 365–383, Jun. 2022.

[13] M. Pettorali, F. Righetti, C. Vallati, S. K. Das, and G. Anastasi, “Mobility management in Industrial IoT environments,” in *Proc. IEEE 23rd Int. Symp. World Wireless Mobile Multimedia Netw.*, 2022, pp. 271–280.

[14] Z. Shelby, K. Hartke, and C. Bormann, “The constrained application protocol (CoAP),” RFC 7252, p. 112, Jun. 2014. [Online]. Available: https://www.rfc-editor.org/info/rfc7252

[15] IEEE standard for information technology– local and metropolitan area networks– specific requirements– part 15.4: Wireless medium access control (MAC) and physical layer (PHY) specifications for low rate wireless personal area networks (WPANs), IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003), pp. 1–320, 2006, doi: 10.1109/IEEESTD.2006.232110.

[16] G. Janssen, P. Stigter, and R. Prasad, “Wideband indoor channel measurements and BER analysis of frequency selective multipath channels at 2.4, 4.75, and 11.5 GHz,” *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1272–1288, Oct. 1996.

[17] R. Kershner, “The number of circles covering a set,” *Amer. J. Math.*, vol. 61, no. 3, pp. 665–671, 1939.

[18] Y.-C. Wang, C.-C. Hu, and Y.-C. Tseng, “Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks,” in *Proc. 1st Int. Conf Wireless Internet*, 2005, pp. 114–121.

[19] Y. Yoon and Y.-H. Kim, “An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks,” *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1473–1483, Oct. 2013.

[20] M. Pettorali, F. Righetti, and C. Vallati, “Mobile6TiSCH: A simulator for -based Industrial IoT networks with mobile nodes,” in *Proc. 18th Int. Conf. Mobility Sens. Netw.*, 2022, pp. 614–618.

[21] 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.*, 2015, pp. 337–350.

[22] 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.*, 2019, pp. 121–132.

[23] 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.*, 2016, pp. 1–6.

Marco Pettorali received the master's degree (cum laude) in computer engineering from the University of Pisa, in 2021. He is currently working toward the PhD degree with the Department of Information Engineering, University of Pisa. His research interests include Industrial Internet of Things (IIoT) and WSNs for industrial applications. He has served as publicity chair for the 7th Workshop on Smart Service Systems (SmartSys) co-located with IEEE SMARTCOMP 2022.

Francesca Righetti received the bachelor's and master's degrees in computer engineering from the University of Pisa, Italy, in 2014 and 2017, respectively, and the PhD degree in information engineering from the University of Pisa, in 2021. She is an assistant professor with the Department of Information Engineering, University of Pisa. Her research interests include WSNs, Internet of Things (IoT), cloud/fog/edge computing. She took part in several national and international projects. She organized the IEEE International Workshop in Smart Service Systems (SmartSys 2022), co-located with IEEE SMARTCOMP. In addition, she has served in the TPC of international conference and workshops, including IEEE SMARTCOMP, IEEE CCNC, IEEE MSN, and IEEE MELECON.

Carlo Vallati is an associate professor with the Department of Information Engineering, University of Pisa. He is also the director of the Crosslab on Cloud Computing, Big Data and Cybersecurity funded by the Italian Ministry of Education and Research (MIUR) in the framework of the "Departments of Excellence" program. He is co-author of more than 100 peer-reviewed papers in international journals and conference proceedings. He has been involved in multiple national and international research projects. He has served as a program committee member for more than 40 international conferences/workshops and as Workshop Chair for IEEE IoT-SoS and IEEE SmartSys. He has served as TPC co-chair of IEEE SMARTCOMP 2020 and general vice chair of PERCOM 2022. He is a member of the EB of *Ad Hoc Networks*, *Journal of Reliable Intelligent Environments*, and *Applied Sciences*.

Sajal K. Das (Fellow, IEEE) is a Curators' Distinguished professor of computer science and the Daniel St. Clair Endowed chair with the Missouri University of Science and Technology. His research interests include cyber–physical systems, IoT, wireless and sensor networks, mobile and pervasive computing, smart environments, applied machine learning, data science and cyber security. He has made fundamental contributions to these areas and published extensively in high quality journals and refereed conference proceedings. He holds 5 US patents and coauthored 4 books. His h-index is 99 with more than 41,000 citations according to Google Scholar. He is a recipient of 14 Best Paper Awards in prestigious conferences like ACM MobiCom and IEEE PerCom, and numerous awards for teaching, mentoring and research including the IEEE Computer Society's Technical Achievement Award for pioneering contributions to sensor networks and the University of Missouri System President's Award for Sustained Career Excellence. He is the founding editor-in-chief of *Pervasive and Mobile Computing Journal* (Elsevier), and associate editor of *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Dependable and Secure Computing*, *IEEE/ACM Transactions on Networking*, and *ACM Transactions on Sensor Networks*. He is a distinguished Alumnus of the Indian Institute of Science, Bangalore, and also a fellow of the National Academy of Inventors.

Giuseppe Anastasi is a professor of computer engineering with the University of Pisa, Italy, where he was the head of the Department of Information Engineering (DII) during 2016–2020. He is also the director of the CrossLab for the Digital Transformation, funded by the Italian Ministry of Education and Research (MIUR) in the framework of the "Departments of Excellence" program. His current scientific interests include Internet of Things, cloud/fog/edge computing, cyber-physical systems, cybersecurity, and quantum Internet. He has co-edited two books and published about 170 research papers in the area of computer networking and distributed systems. His publications have received more than 11,000 citations, according to Google Scholar (h-index = 45). He is currently serving as a Steering Committee member of the IEEE SMARTCOMP conference. Previously, he served as the area editor of *Pervasive and Mobile Computing* (PMC), *Computer Communications* (ComCom), and *Sustainable Computing* (SUSCOM). He has also served as general/program chair of many international conferences.