2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM) | 978-1-6654-0876-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/WoWMoM54355.2022.00046

Mobility Management in Industrial IoT Environments

Marco Pettorali[1], Francesca Righetti[1], Carlo Vallati[1], Sajal K. Das[2], and Giuseppe Anastasi[1]

[1]Dept. of Information Engineering, University of Pisa, Pisa, Italy, {name.surname}@unipi.it
[2]Dept. of Computer Science, Missouri University Science and Technology, Rolla, Missouri, USA, sdas@mst.edu

***Abstract*—The Internet Engineering Task Force (IETF) has defined the 6TiSCH architecture to enable the *Industrial Internet of Things (IIoT)*. Unfortunately, 6TiSCH does not provide mechanisms to manage node mobility, while many industrial applications involve mobile devices (e.g., mobile robots or wearable devices carried by workers). In this paper, we consider the *Synchronized Single-hop Multiple Gateway* framework to manage mobility in 6TiSCH networks. For this framework, we address the problem of positioning Border Routers in a deployment area, which is similar to the "Art Gallery" problem, proposing an efficient deployment policy for Border Routers based on geometrical rules. Moreover, we define a flexible Scheduling Function that can be easily adapted to meet the requirements of various IIoT applications. We analyze the considered Scheduling Function in different scenarios with varying traffic patterns, and define an algorithm for sizing the system in such a way to guarantee the application requirements. Finally, we investigate the impact of mobility on the performance of the system. Our results show that the proposed solutions allow to manage node mobility very effectively, and without significant impact on the performance.**

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

I. Introduction

The *Industrial Internet of Things (IIoT)* extends the traditional IoT paradigm to industrial applications characterized by stringent requirements, in terms of communication reliability and timeliness. It is expected to have a strong impact on many different domains, such as manufacturing and production systems, logistics, transportation, energy, and many others [1].

To support this trend, the Internet Engineering Task Force (IETF) has defined the 6TiSCH architecture (IPv6 over the *Time Slotted Channel Hopping* (TSCH) mode of IEEE 802.15.4e) that allows the integration of IoT devices (i.e., sensors and actuators) into existing IPv6 networks, ensuring an industrial grade of service [2]. The 6TiSCH architecture relies on the TSCH access protocol defined in the IEEE 802.15.4 standard [3]. The TSCH allows short-range wireless communication with time-bounded, guaranteed-bandwidth, and energy-efficient service, through a *scheduled* access protocol. In addition, it leverages *multi-channel communication* to increase the overall network capacity, and *frequency hopping* to mitigate the negative effects of interference that are quite common in the industrial environments.

The 6TiSCH architecture implicitly assumes that industrial IoT devices are stationary, as it does not specify any specific mechanism to support node mobility. On the other hand, many industrial applications involve mobile devices, such as mobile robots/objects, sensors/actuators mounted or rotating parts of machines, or wearable devices carried by workers. Hence, the investigation of efficient solutions for managing mobility in 6TiSCH networks is an open and relevant research issue [1].

Several existing works [4]–[8] have considered IoT mobility. However, only a few of them deals with 6TiSCH architecture [7], [8]. Moreover, these works address specific aspects, such as the definition of a Distributed Traffic-aware Scheduling Function for mobile 6TiSCH networks [7], or the design of an efficient route management method [8]. The authors in [7], [8] consider a limited fraction (say about 10%) of mobile nodes in the network.

Integrating mobile nodes in a 6TiSCH network requires to address a couple of issues strictly related to node mobility. First, 6TiSCH specifications require the node association to the network through a complex joining procedure, during which the node acquires synchronization and routing information. This association procedure must be repeated frequently in case of node mobility. During the association procedure, since the mobile node is not able to communicate, node mobility may severely impact the performance of the overall system. Hence, efficient solutions are required for managing node mobility.

Some existing works have addressed the problem of fast association in 6TiSCH networks [4], [5], [8]. However, some of these solutions are not suitable for industrial applications with stringent requirements. In this paper, we refer to the *Synchronized Single-Hop Multi-Gateway (SHMG)* framework presented in [5]. SHMG takes a centralized approach. The network includes a Network Coordinator (NC), in addition to the Mobile Nodes (MNs) and Border Routers (BRs). MNs can communicate only with BRs through 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 the other ones in the system. This allows to manage mobility very effectively and, specifically, to meet the stringent requirements of industrial applications.

The SHMG framework appears to be a promising approach for managing mobility in 6TiSCH networks, especially when dealing with real-time and/or loss sensitive applications. However, some important components need to be defined in order to make it suitable for a real-world setting. This includes the deployment policy for Border Routers, and a Scheduling

978-1-6654-0876-9/22/$31.00 ©2022 IEEE

DOI 10.1109/WoWMoM54355.2022.00046

Function for allocating communication resources in order to meet the requirements of the specific application. Also, it is important to investigate the scalability of the proposed approach, and the impact of node mobility on its performance.

A. Contributions of This Paper

In this paper, we make the following major contributions, considering the general SHMG framework [5] but extending it in several novel ways.

(1) We propose an efficient deployment policy of BRs that uses geometrical rules to completely cover the considered area with a minimum number of BRs, which is similar to the “Art Gallery” problem [9].

(2) We define a flexible Scheduling Function, referred to as *Shared Downlink - Dedicated Uplink (SD-DU)* scheduling, specifically tailored to the SHMG framework. The proposed Scheduling Function can be easily adapted to meet the requirements of different IIoT applications, to cope with different use cases.

(3) We derive analytical formulas for SD-DU scheduling, and define an algorithm to calculate the maximum number of MNs that can be accommodated to guarantee the application requirements. This allows an appropriate sizing of the network.

(4) We investigate, through simulation experiments, the impact of node mobility on the performance of the overall system. To this end, we developed an OMNeT++ module that implements mobility in 6TiSCH networks, according to the SHMG framework.

The reminder of the paper is organized as follows. Section II presents the 6TiSCH architecture. In Section III we describe the SHMG framework, propose a BR deployment policy and define the SD-DU scheduling. Section IV analyzes the proposed Scheduling Function in terms of maximum packet rate and end-to-end delay. Based on this analysis, in SectionV we define an algorithm for computing the maximum number of MNs that can be accommodated to meet the application requirements. Section VI presents simulation studies to demonstrate the impact of mobility on the performance. Finally, Section VII concludes the paper.

II. The 6TiSCH Architecture

The 6TiSCH architecture [2] has been standardized by the Internet Engineering Task Force (IETF) to enable IIoT paradigm. As depicted in Fig. 1, the 6TiSCH reference system model relies on 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, high reliability, and energy efficiency. To this end, it leverages *time-slotted access*, *multi-channel communication*, and *frequency hopping*.

Time-slotted access is obtained 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. To increase the network capacity, more nodes are allowed to transmit simultaneously, during the same timeslot, using a different channel (multi-channel communication). There are 16 different channels available, each identified by a *channel offset*, i.e., an integer value in the range 0-15. To mitigate the negative effects of multi-path fading and interference, each node changes its operating frequency at each timeslot according to a predefined hopping sequence (channel hopping). 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.

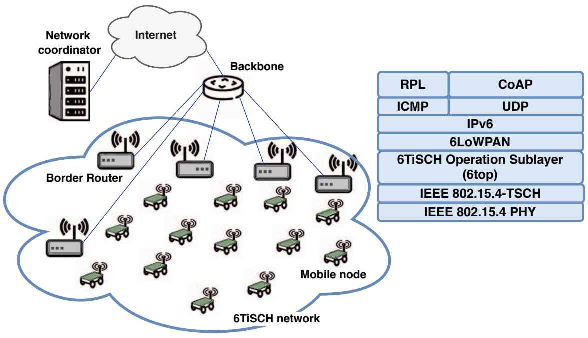


Fig. 1: 6TiSCH Reference Network and Protocol Stack.

In TSCH, each element in the two-dimensional slotframe, namely a *cell*, is identified through timeslot and channel offset. Cells can be either *dedicated* or *shared*. Dedicated cells are allocated to a sender-receiver pair for contention-free communication. In contrast, shared cells are allocated to multiple pairs of nodes and accessed on a contention basis. In case of simultaneous transmission by more than one node on a shared cell, a collision may occur. To prevent subsequent collisions, after the first one, the nodes use the TSCH CSMA (Carrier Sense Multiple Access) algorithm [3].

While TSCH allows nodes to allocate and deallocate cells for communication, depending to their traffic needs, the standard does not specify how nodes can allocate them. To fill in this gap, the 6TiSCH protocol stack includes the Operation (6top) sublayer. This sublayer implements the abstraction of IP link over TSCH (see Fig. 1) and, specifically, it is responsible for the allocation of cells to nodes. To this end, it relies on a Scheduling Function (SF) and the 6top protocol (6P) [10]. The Scheduling Function is used to compute the number of cells required by a node to manage the current traffic needs and meet the application requirements, while the 6P protocol is used to negotiate the allocation/deallocation of cells with neighbor nodes.

On top of the 6top sublayer (Fig. 1), the 6LoWPAN adaptation protocol is in charge of encapsulating IPv6 datagrams into TSCH frames. Multi-hop delivery of IPv6 datagrams is managed through the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [11]. RPL organizes nodes in a *Destination Oriented Directed Acyclic Graph (DODAG)* rooted at a BR. Then, each node selects a preferred parent in the DODAG, and uses it to forward data packets towards the BR (and, through the BR, to the final destination). Finally, data messages generated by the application are managed by the User Datagram Protocol (UDP) protocol.

Before joining the 6TiSCH network, a node must perform a complex association phase that consists of several steps. Specifically, each node has first to scan all the available channels until it receives a TSCH *Enhanced Beacon (EB)*. EBs are special frames used to advertise the network configuration and are broadcast by nodes (already in the network) cycling on all the available channels. Then, after receiving an EB, each node waits for a *DODAG Information Object (DIO)* message from the RPL protocol to select its preferred parent and conclude the association phase.

III. Mobility Management in 6TiSCH networks

As noted in Section II, 6TiSCH does not include any specific mechanism for managing node mobility. Instead, each node must associate with the network before communicating, and the association procedure takes a significant amount of time. In case of a mobile node, the association procedure must be repeated every time the node changes its location; while it is in progress, the node is not able to communicate. In addition, after the selection of the new preferred parent, the node must acquire the cells necessary for communication, which takes additional time. Hence, the 6TiSCH standard procedures are not adequate for supporting node mobility.

Although solutions for fast association have been proposed in the literature [7], [8], they only mitigate the problem related to the handover of mobile nodes and may not be suitable for industrial applications with stringent requirements in terms of communication reliability and timeliness. In addition, cells for communicating with neighbors in the new location may not be available, unless allocated proactively.

This paper refers to the *Synchronized Single-Hop Multi-Gateway (SHMG)* framework [5] that allows to manage the handover very quickly, and is thus suitable for industrial applications with stringent requirements. This incurs limited flexibility and huge consumption of network resources. Below we first describe the general SHMG architecture, and then extend the proposed architecture with an efficient deployment policy of BRs and a flexible Scheduling Function.

A. Synchronized Single-Hop Multi-Gateway Architecture

The SHMG architecture leverages a centralized scheme and is composed of the following entities, as depicted in Fig. 1:

- **Network Coordinator (NC)**: The NC is responsible for allocating and deallocating cells for each node, according to the SF. The communication schedule is replicated on all the BRs. The NC is also the source of synchronization: clock information is sent by the NC to all BRs, and from BRs to MNs.
- **Border Routers (BRs)**: The BRs are static nodes that serve as gateways between the 6TiSCH network and the Internet. They are the parent node for all the MNs in their communication range.

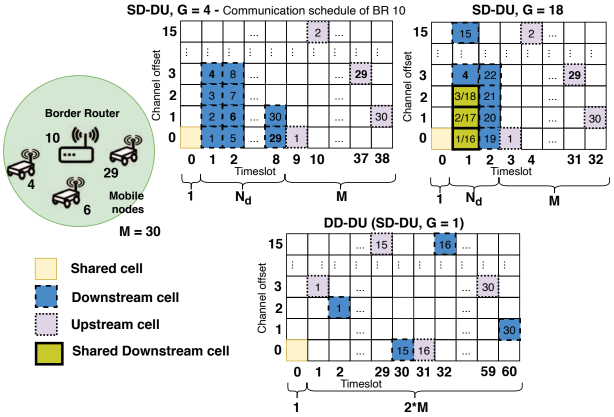


Fig. 2: An Example of Scheduling Function.

- **Mobile Nodes (MNs)**: The MNs generate data packets and transmit them to the closest BR through a single hop communication.

In the considered framework, each BR forms a star topology with all the MNs located in its transmission range, and hence with single-hop communication. To allow a smooth and efficient handover from one BR to another, all the BRs are synchronized and the communication schedule, defined by the NC, is installed on all the BRs. Therefore, the MNs do not need to acquire cells after moving from one BR to another.

In addition to time synchronization and schedule management, the NC also manages the selection of the best path towards an MN for downstream traffic. To this end, the BRs continuously collect information about the quality of communication with MNs in their range, and send the collected information to the NC. Whenever a data packet must be delivered to an 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, based on the available information.

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

B. Scheduling Function

In a 6TiSCH network, the communication schedule is derived through a Scheduling Function (SF) that is used to dynamically compute the number of cells required by each node. Since the considered SHMG framework leverages a centralized scheme, the SF is derived and updated by the NC, and then installed on all the BRs.

In this Section we define a flexible SF, tailored to the SHMG architecture, that can be easily configured to meet the requirements of different applications. As a preliminary step, let us recall some characteristics of IIoT applications that motivate the definition of our Scheduling Function (SF).

In the IIoT use cases, data packets typically flow upstream (from the nodes to the BRs), while downstream traffic (from the BRs to the nodes) is sporadic. Hence, the resulting traffic pattern is typically asymmetric. However, for certain applications, for example, using Confirmable messages in the Constrained Application Protocol (CoAP) [12], a symmetric traffic pattern is also possible for upstream and downstream traffic flows. Moreover, many industrial applications have stringent requirements of 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 a new SF, referred to as the *Shared Downstream - Dedicated Upstream (SD-DU)* that can guarantee the requirements of industrial applications and is flexible enough to adapt to different traffic patterns (e.g., asymmetric vs. symmetric).

With reference to the two-dimensional TSCH slotframe, SD-DU allocates different kinds of cells as follows.

- **Control Cell**: A single shared cell with (timeslot = 0, channel offset = 0) used by all the nodes for TSCH/RPL control information.
- **Downstream Cells**: A number of cells allocated to the downstream data traffic, depending on the number of MNs and the amount of downstream traffic to manage.
- **Upstream Cells**: A dedicated cell for each MN in the network used for upstream data traffic.

Before describing the SF in detail, we introduce the following parameters: (i) S_l, the number of timeslots in a slotframe (slotframe length); (ii) M, the number of MNs in the network; (iii) G, the number of MNs using the same timeslot for receiving downstream data.

The SD-DU takes a different approach for the allocation of cells in the upstream and downstream directions. For upstream traffic, the NC allocates one dedicated cell for each MN. Since the BR cannot receive simultaneously on different frequencies during the same timeslot, a single cell per timeslot will be used. Hence, the number of cells/timeslots allocated to the upstream traffic will be $N_u = M$.

On the other hand, for downstream traffic, the NC 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 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.

Fig. 2 shows an example of a network with $M = 30$ MNs and the communication schedule of a specific BR, with different values of $G = 1, 4,$ and 18. We recall that all the BRs share the same communication schedule. The only difference is that they use different cells to transmit/receive data packets, depending on the MNs currently in their transmission range. When $G = 4$ or 18, (the two top schedules), the downstream cells are allocated to the first N_d timeslots of the slotframe, while the upstream cells to the next M timeslots. The slotframe length is $S_l = N_d + N_u + 1 = \lceil M/G \rceil + M + 1$.

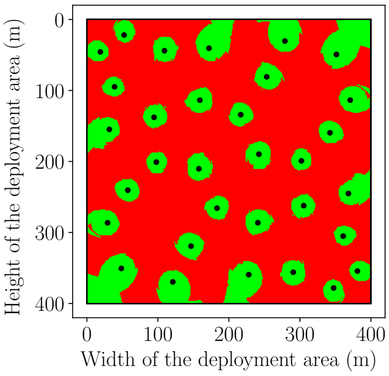


Fig. 3: Covered (green) and interference (red) areas with the Kazazakis-Argyros method [13].

When considering SD-DU with $G = 4$ (top left communication schedule), we can see that the first 8 timeslots (after the control timeslot) are used for downstream traffic, and that only 4 channels, out of the 16 available ones, are exploited.

The SD-DU configuration with $G = 18$ (top right) exhibits a shorter slotframe length, since there are only 2 timeslots used for downstream traffic. In addition, in the first timeslot, there are 3 cells (i.e., [1,0], [1,1], [1,2]), each shared by a couple of nodes. Using a large value for G reduces the slotframe length, at the cost of increasing the collision probability for downstream packets.

The bottom schedule in Fig. 2 represents the configuration of SD-DU with $G = 1$. In this configuration, the NC allocates one dedicated downstream cell/timeslot, in addition to one dedicated upstream cell/timeslot, for each MN. Throughout the paper, the SD-DU configuration with $G = 1$ will be referred to as *Dedicated Downstream - Dedicated Upstream (DD-DU)*. In this case, the slotframe length is $S_l = N_d + N_u + 1 = 2M + 1$.

Since DD-DU (i.e., SD-DU with $G = 1$) has dedicated cells for both upstream and downstream flows, it may be more convenient to organize the schedule allocation such that the NC allocates the downstream cells of a node just after the upstream cells of the same node, as illustrated in Fig. 2.

C. Efficient Deployment of Border Routers (BRs)

In this section, we address the problem of positioning BRs in the deployment area, in such a way to guarantee the continuity of service to MNs. Our goal is to achieve almost complete coverage (beyond 99%) of the deployment area, while minimizing the number of BRs. At the same time, we want to minimize the total area covered by two or more BRs (interference area), in order to avoid collisions at the MNs. This is because an MN that happens to be in the interference area might receive a packet simultaneously from different BRs, thus experiencing a collision.

Our problem is somewhat similar to the well known “Art Gallery” problem, i.e., how to surveil an art gallery with the minimum number of guards. The Art Gallery problem was originally stated by Klee (see [9]) and investigated by Chvatal [14]. It originates from a real-world problem and aims at finding the smallest number of guards necessary to have a clear vision of every point of an art gallery. It was proved that providing an optimal solution to the Art Gallery problem for simple polygonal areas is NP-hard [15].

There exist works in the literature that provide an approximate solution to the above problem. For example, in [13] is presented a method for positioning the minimum number of observation points (i.e., guards), to visually inspect a two dimensional area, assuming that the guards have a limited visibility. We implemented the Kazazakis-Argyros method [13], assuming a square deployment area and a circular communication range for BRs.

Fig. 3 shows the results obtained, when the deployment area is $400 \times 400\ m^2$ and the communication range of BRs is set to 76 m. We can see that the used method allows a complete coverage of the considered area with 32 BRs. We also observe that this method generates a large interference area (red zones in Fig. 3). This means that the solution provided by the Kazazakis-Argyros approach is redundant in the sense that it provides a complete coverage using a large number of BRs and generating many interference zones. Hence, it is not suitable for our purpose.

Below, we define a new heuristic deployment policy inspired by [16], namely *Intersecting Flowers*, to minimize the number of deployed BRs and the total interference area, yet guaranteeing a complete coverage. To this end, we assume a rectangular area of width W and height H, which is typical in industrial settings [17].

From a geometrical point of view, the trade-off between maximizing the coverage and simultaneously minimizing the total interference area, can be found by positioning BRs at the vertices of an equilateral triangle (see Fig. 4). Given those positions for three BRs, we need to find out how many BRs can be deployed in the considered rectangular area. For this, we compute the side of the triangle L and its height h.

Let r denote the radius of the communication range of each BR. Since the intersection of the communication ranges of three BRs is the centroid of the triangle, $L = \sqrt{3} \cdot r$, and $h = \sqrt{3}/2 \cdot L = 3/2 \cdot r$. Once we know the triangle dimensions, we can compute the number of BRs to be deployed in the area $W \times H$, by deriving the number of rows $row_n = \lceil H/h \rceil$ and the number of BRs per row $BR_n = \lceil W/L \rceil$.

Starting from the first row, given the number of BRs per row, BR_n, after placing the first and the last BRs at the edges of the deployment area, it is possible to compute the distance among BRs in a row as $BR_d = W/(BR_n - 1)$. Furthermore, to maximize the coverage, we add more BR to the row if $BR_d > L(1 + 0.075)$. The deployment of BRs on the other rows follows the same approach. It is worth highlighting that, to maintain each BR at the vertices of an isosceles triangle, the even and odd rows will have a different number of BRs.

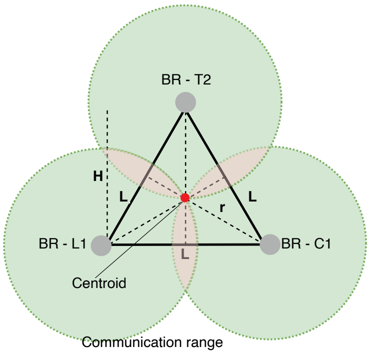


Fig. 4: BR deployment for complete coverage with minimum number of BRs and interfering area.

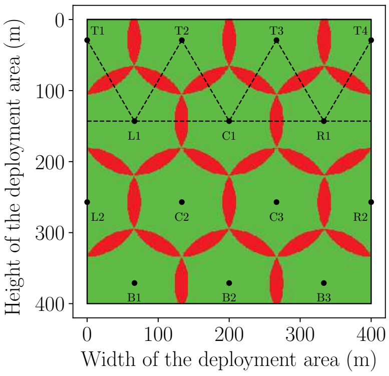


Fig. 5: Covered (green) and interference (red) areas with our proposed Intersecting Flowers policy.

An example of deployment based on the the Intersecting Flowers policy is shown in Fig. 5. We can visually observe that the total interference area (red zones) is dramatically reduced, with respect to the deployment derived from the Kazazakis-Argyros policy (Fig. 3). In addition, the number of deployed BRs is much lower (14 vs. 32), while the area is still completely covered.

Table I compares our Intersecting Flowers policy with the Kazazakis-Argyros policy, in terms of coverage computed using the Simpson’s Rule Integrator [18], and the number of BRs deployed per 10,000 m^2. The total interfering area is always much lower, when using Intersecting Flowers, and is thus omitted for brevity. From these results, we conclude that, with respect to Kazazakis-Argyros method, the Intersecting Flowers policy reduces drastically the number of BRs (and the total interference area), while providing always a coverage close to 100%. Therefore, our performance evaluation in Section VI will use the Intersecting Flowers policy.

IV. Scheduling Analysis

In Section III-B we have defined a flexible SF for the considered SHMG framework, namely SD-DU, that can be configured through a parameter G to adapt to different use cases with different traffic patterns. In this section, we analyze

TABLE I: Coverage and # of BRs for various policies.

Deployment policy	**Deployment area (m^2)**		
	200 × 200	300 × 300	400 × 400
	Coverage (percentage)		
Kazazakis-Argyros	100	100	100
Intersecting Flowers	100	100	99.995
	Number of BRs per 10,000 m^2		
Kazazakis-Argyros	2.0	1.778	2.0
Intersecting Flowers	1.25	1.222	0.875

this SF, in terms of maximum packet rate provided and the delay introduced. In the following, we will refer to two different scenarios, characterized by different traffic patterns, namely a *Convergecast* scenario and a *Request/Response* scenario.

In the Convergecast scenario [19], which is typical in monitoring applications, the MNs report data periodically to a collection point (upstream flow). Without loss of generality, we will assume that the collection point is the NC. Data may also be sent in the reverse direction, i.e., from the NC to one or more MNs (downstream flow). Downstream data traffic is used, for instance, to instruct or reconfigure MNs, and its total amount 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, the MNs send request messages to a decision point (again, we assume that the decision point coincides with the NC), and the latter replies with a Response Message. This happens, for instance, when using the CoAP Confirmable messages [12]. In addition, this scenario arises 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 characteristics and requirements.

In our analysis, we consider the worst case arising when all the MNs are located inside the communication range of the same BR. Since this is the most critical situation to manage, we will calculate the performance achieved by an MN in such a worst-case condition.

1) Convergecast Scenario: For this scenario, let us derive the *maximum rate* and *end-to-end delay*, as functions of the number of MNs. The maximum rate is defined as the maximum number of packets per second that can be transmitted by a node, while the end-to-end delay is the time interval between the generation of a packet at the source node and its reception at the destination node. Specifically, we will distinguish between the upstream and downstream maximum rates and end-to-end delay. Both the upstream and downstream end-to-end delay include a wired component (between the NC and BR) and a wireless component (between the BR and MN). Assuming that the BRs and NC are connected through a high-speed network (e.g., a LAN), the wired component can be neglected, with respect to the wireless one. Thus, in the following, we will focus on the wireless component only.

To derive the maximum upstream rate, recall that each MN has one dedicated cell per slotframe for upstream communication and, consequently, it can transmit at most one data packet per slotframe. Hence, the maximum upstream rate (in packets per second) of each MN can be computed as

$$r^{c\_up}_{SD\text{-}DU}(M) = \frac{1}{S_{l,SD\text{-}DU}(M) \cdot T_S} \tag{1}$$

where M is the number of MNs, T_S is the timeslot duration (10 ms), and $S_{l,SD\text{-}DU}$ is the slotframe length given by

$$S_{l,SD\text{-}DU}(M) = 1 + \left\lceil \frac{M}{G} \right\rceil + M + OM \tag{2}$$

where $O \in \mathbb{N}$ is an integer such that the resulting slotframe length, $S_{l,SD\text{-}DU}(M)$, is the smallest value that is co-prime with the number of available frequencies (see Section II).

Observe that, with DD-DU scheduling (i.e., $G = 1$), the slotframe length becomes $S_{l,DD\text{-}DU}(M) = 1 + 2 \cdot M + O$. Hence, the maximum upstream rate can be expressed as

$$r^{c\_up}_{DD-DU}(M) = \frac{1}{(1 + 2 \cdot M + O) \cdot T_S}. \tag{3}$$

Let us now derive the end-to-end upstream delay. Note that the worst condition occurs when the packet is generated in a given slotframe, just after the slot is allocated to the source MN, and transmitted in the next one. Under this assumption, the end-to-end upstream delay can be expressed as

$$d^{c\_up}_{SD-DU}(M) = S_{l,SD\text{-}DU}(M) \cdot T_S. \tag{4}$$

In the Convergecast scenario, it may occasionally happen that the NC has packets to send to the MNs through BRs. In this case, the traffic flows in the downstream direction. To derive the maximum downstream rate, $r^{c\_down}_{SD\text{-}DU}(M)$, we need to consider that (i) in the worst-case, all MNs are located in the communication range of the same BR; (ii) the BR can transmit at most one packet per timeslot; and (iii) the number of downstream timeslots in a slotframe is $\lceil \frac{M}{G} \rceil$. Hence, the maximum downstream rate (in the worst-case) can be expressed as

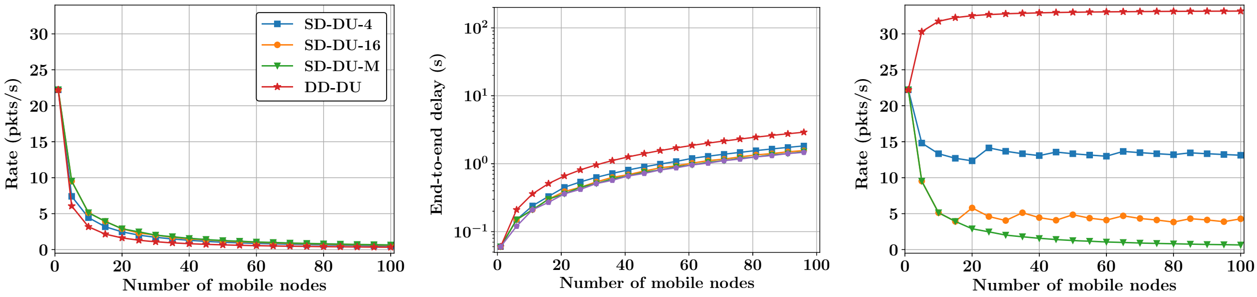
$$r^{c\_down}_{SD\text{-}DU}(M) = \frac{\lceil \frac{M}{G} \rceil}{S_{l,SD\text{-}DU}(M) \cdot T_S}. \tag{5}$$

The above equations also apply to DD-DU scheduling (i.e., $G = 1$), for which the slotframe length is $S_{l,DD\text{-}DU}(M)$.

2) Request/Response Scenario: Let us now repeat the previous analysis for the Request/Response scenario, where the MNs send periodic Request messages to the NC, which in turn replies with Response messages. We assume that Requests and Responses are tightly coupled and have the same requirements, in terms of the quality of service. Hence, the upstream and downstream flows have the same characteristics.

In this scenario, the *end-to-end delay* is related to the entire (upstream + downstream) path, and hence defined as the time interval between the generation of a Request message at the MN and the reception of the corresponding Response message by the same node. As above, we assume that the wired delay component is negligible as compared to the wireless component. The definition of maximum upstream/downstream rate is the same as above.

To derive the maximum rate achievable by an MN, we need to consider that the upstream and downstream flows are tightly



(a) Maximum upstream rate

(b) End-to-end delay

(c) Maximum downstream rate

Fig. 6: Maximum upstream/downstream rate and end-to-end delay in the Convergecast scenario.

coupled in this scenario. Hence, the maximum rate of an MN is bounded by the maximum downstream rate, if less resources are allocated to the downstream communication. Considering that the number of timeslots per slotframe allocated to each MN for downstream traffic is $\left\lceil \frac{M}{G} \right\rceil / M$, the maximum rate achievable by an MN is given by

$$r^{req/res}_{SD\text{-}DU}(M) = \frac{\left\lceil \frac{M}{G} \right\rceil / M}{S_{l,SD\text{-}DU}(M) \cdot T_S} \tag{6}$$

If $G = 1$ (for DD-DU scheduling), the equation reduces to

$$r^{req/res}_{DD\text{-}DU}(M) = \frac{1}{S_{l,DD\text{-}DU}(M) \cdot T_S}. \tag{7}$$

The end-to-end delay under the worst-case assumption is made up of the following three components: (i) the maximum upstream delay $S_{l,SD\text{-}DU}(M) \cdot T_S$; (ii) the maximum time to wait for the downstream timeslot in which the BR can send a packet, $(M+1) \cdot T_S$; and (iii) the number of slotframes that a downstream packet has to wait, due to the $G-1$ packets enqueued at the BR and destined to other MNs sharing the same downstream timeslot, $(G-1) \cdot S_{l,SD\text{-}DU}(M) \cdot T_S$. Hence, the equation for the end-to-end delay is as follows

$$\begin{aligned} d^{req/res}_{e2e,SD\text{-}DU}(M) = (S_{l,SD\text{-}DU}(M) + (M+1) + \\ + (G-1) \cdot S_{l,SD\text{-}DU}(M)) \cdot T_S \end{aligned} \tag{8}$$

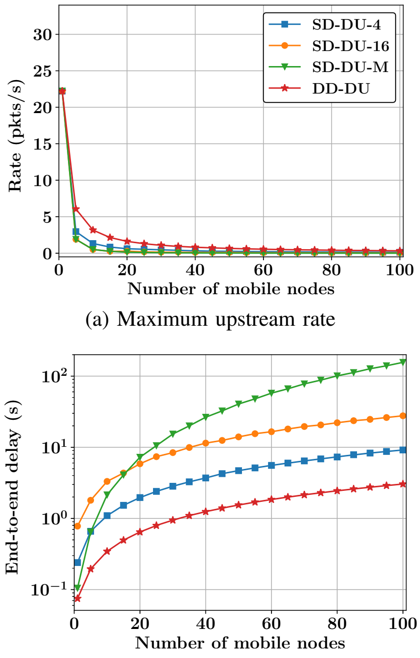
In the special case $G = 1$ (i.e., with DD-DU scheduling), a downstream packet must wait at most an entire slotframe plus the downstream cell in which it is transmitted. Hence the end-to-end delay can be expressed as

$$d^{req/res}_{e2e,DD\text{-}DU}(M) = (S_{l,DD\text{-}DU}(M) + 1) \cdot T_S \tag{9}$$

A. Analytical Results

Applying the above analytical formulas, we now analyze the maximum rate and end-to-end delay experienced by an MN in the worst-case conditions of the above two scenarios with an increasing number of MNs.

Fig. 6 presents the results related to the Convergecast scenario. We analyzed different configurations of SD-DU, by considering different values of the G parameter that regulates the amount of resources allocated for downstream communication. Specifically, we consider $G = 4, 16,$ and M and, in addition, the special case of $G = 1$ (labelled as DD-DU).



(a) Maximum upstream rate

(b) End-to-end delay

Fig. 7: Maximum packet rate and end-to-end delay in the Request/Response scenario.

Fig. 6a shows the maximum upstream rate achievable by an MN, with different configurations of the Scheduling Function, for an increasing number of MNs. We can observe that, when the number of MNs is large (e.g., above 50), there is no significant difference between the various configurations. When the number of MNs is not so high, it is more convenient to use a large G value. This is because, increasing G, reduces the number of timeslots ($\left\lceil \frac{M}{G} \right\rceil$) allocated to the downstream traffic and hence the slotframe size. Therefore, the MNs can transmit more packets per time unit.

The same trend also holds for the end-to-end delay, shown in Fig. 6b. The situation is exactly the opposite for the downstream rate (Fig. 6c). However, the motivation is exactly the same, i.e., increasing the G value increases the number of downstream timeslots and thus the downstream rate.

From these results we conclude that, in a Convergecast scenario with asymmetric traffic, the SD-DU scheduling can be tuned by selecting the most appropriate G value, depending on the amount of downstream traffic to manage.

Let us now turn our attention to the Request/Response scenario, whose results are presented in Fig. 7. In this scenario, the maximum packet rate is achieved when using $G = 1$ (i.e., DD-DU scheduling), and the rate decreases when G increases. This is because the maximum packet rate is bounded by the downstream rate. Increasing the G value decreases the number of downstream slots and consequently, the downstream rate. Similar considerations also apply to the end-to-end delay.

Thus, in the Request/Response scenario characterized by the upstream and downstream flows with similar requirements, the DD-DU (i.e., SD-DU with $G = 1$) is the best option.

V. Network Sizing

In this section, we use the analytical formulas derived in the previous section to compute the maximum number of MNs that can be accommodated in the system, while guaranteeing a certain quality of service (QoS), expressed in terms of the minimum packet rate and/or maximum end-to-end delay that can be tolerated by the application (we assume that all MNs have the same QoS requirements). To this end, we developed Algorithm 1 that can be executed at the NC. The same algorithm can be implemented in the form of a lookup table, as shown in Table II.

The sizing algorithm takes as input: (i) the number of MNs to deploy (N); (ii) the C value in the SD-DU Scheduling Function; (ii) the minimum packet rate (R_r) required by the application; and (iv) the maximum end-to-end delay (D_m) that can be tolerated by the application. The algorithm provides, as output, the maximum number of MNs that can be actually deployed (M) in order to guarantee the applications requirements, in terms of packet rate and end-to-end delay.

The sizing algorithm takes an iterative approach. At each step, it increases the number of MNs and checks whether it is still possible to guarantee the required packet rate and end-to-end delay. For this purpose, the algorithm relies on the equations derived in Section IV (using the appropriate G value). The algorithm terminates either when all the N nodes have been accommodated, or it is not possible to guarantee the application requirements with an additional MN.

We show below an execution of the sizing algorithm. For the sake of space, we only consider the Request/Response scenario, however the algorithm can also be used in the Convergecast scenario. Based on the conclusions drawn in Section IV-A, for the Request/Response scenario we will use $G = 1$ (DD-DU scheduling).

Algorithm 1: Algorithm for network sizing.

```
Input: $N$ = Number of mobile nodes to deploy
       $G$ = The parameter in SD-DD scheduling
       $R_r$ = Required rate
       $D_m$ = Max end-to-end delay

Output: $M$ = Number of MNs that can be actually
        deployed

for $i$ = 1 to $N$:
    R = rate($i$, $C$), D = max_delay($i$, $C$)
    if R $\geq R_r$ and D $\leq D_m$:
        then append $i$ to list_candidates_M
$M$ = max(list_candidates_Ns)
```

TABLE II: Maximum number of MNs to meet the assigned requirements in the Request/Response scenario.

End-to-end delay (s)	Rate (pkt/s) 0.1	0.2	0.3	0.4	0.5	1
1	31	31	31	31	31	31
1.5	48	48	48	48	48	32
2	65	65	65	65	65	32
2.5	81	81	81	81	66	32
3	98	98	98	82	66	32
3.5	115	115	110	82	66	32
4	131	131	110	82	66	32

Table II reports the outcomes of the sizing algorithm, for different combinations of minimum packet rate and maximum end-to-end delay tolerated by the application. We observe that, when the requested packet rate is high (say 1 packet per second), only a few MNs can be accommodated, and the end-to-end delay exhibits a significant impact because the packet rate is the major constraint. If the application requires a lower packet rate, it is possible to admit a larger number of MNs (up to 131 for a maximum end-to-end delay of 4 secs).

VI. Performance Evaluation

We evaluate the performance of the SHMG architecture and the proposed Scheduling Function trough simulation experiments. Specifically, we validate the analytical results derived in Section IV for the worst-case scenario. We also evaluate the performance in more general conditions, analyzing the impact of mobility and the continuity of service provided to the MNs.

A. Simulation Setup

For simulation analysis, we developed a module called *Mobile-6TiSCH*, for the OMNeT++ simulation tool[1]. Mobile-6TiSCH implements the full 6TiSCH protocol stack, including TSCH and the 6top sublayer. In addition, it implements the SHMG framework, the Intersecting Flowers deployment policy, and the SD-DU scheduling. Mobile-6TiSCH also implements functionalities to manage node mobility supporting three mobility patterns, namely *static*, *linear*, and *random*.

In all the mobility patterns, each MN is assigned an initial position, such as a random point within the deployment area. Then, the position of the MN may change over time, according to the considered mobility pattern. In the *static* pattern, the initial position remains unchanged over time. In the *linear* pattern, the MNs move either horizontally or vertically with

[1] https://omnetpp.org/

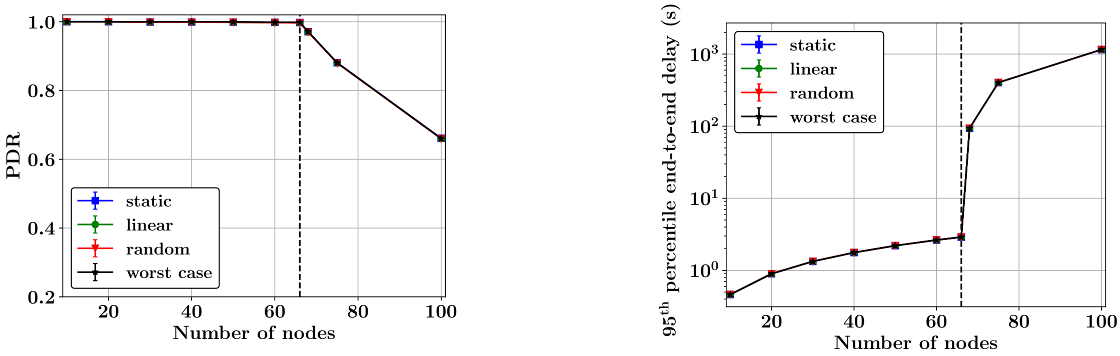


Fig. 8: Impact of mobility on performance in the Request/Response scenario.

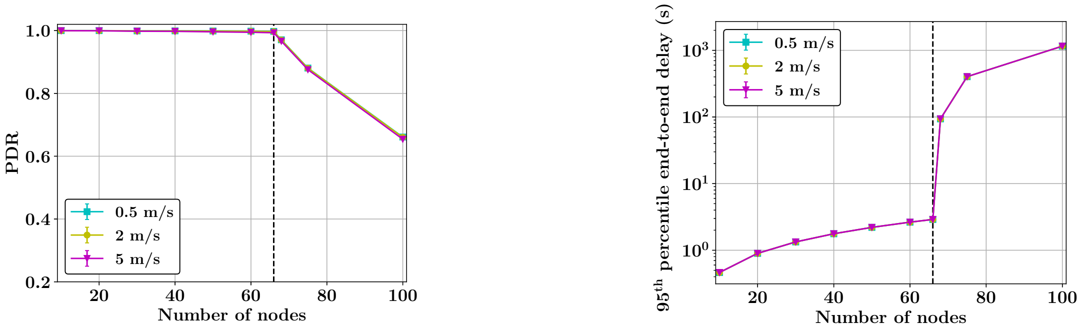


Fig. 9: Impact of the MN speed on the performance in the Request/Response scenario with linear mobility

a constant speed, starting from their initial location. When an MN reaches the border of the deployment area, it inverts the direction and moves back towards the opposite border. This pattern is typical of mobile robots/objects moving along a constrained path. Finally, in the *random* mobility pattern, an MN moves from the current location to a randomly selected new location with a constant speed. Upon arriving at the new position, the MN generates another new random position and moves there.

In our simulation analysis, we considered the following performance metrics:

- **Packet Delivery Ratio (PDR)**, defined as the percentage of data packets received correctly at the final destination, with respect to the total number of data packets generated by the network nodes.
- **End-to-end delay**, defined as the time interval between the instant at which a data packet is generated at the source node and instant at which it is correctly received by the destination node.

It is worth recalling that, for upstream data packets, the source node is an MN, while the destination is the NC. The opposite happens for downstream data packets.

Our simulation experiments considered a deployment area of 400×400 m^2 that can be covered with 14 BRs using the Intersecting Flowers deployment (see Fig. 5). Since we are investigating the impact of mobility on the performance, we assumed that the communication channel is ideal (i.e., no packet loss). Under this assumption, the packets are lost due to mobility only. Finally, unless indicated differently, the MNs generate data packets periodically, with a period of 2s (i.e., 0.5 packets/s) and move with a speed of 2 metre per second (m/s).

To obtain statistically sound results, for each simulation experiment we performed 35 independent replicas of each run, each of 1 hour duration. The results presented below are averaged over all the replicas. We also show confidence intervals, obtained with a 95% confidence level.

B. Simulation Results

In our simulation analysis we considered both the Convergecast and the Request/Response scenarios. However, for the sake of space, we only present the results related to the Request/Response scenario. Following Section IV, we will use SD-DU with $G = 1$, i.e., DD-DU scheduling.

Fig. 8 shows the PDR and 95^{th} percentile of end-to-end delay for the three considered mobility patterns. We also simulated the worst-case scenario considered in Section IV. We observe that in the worst case scenario, the simulation results match the analytical predictions. The vertical dashed line corresponds to the maximum number of MNs that can be accommodated for the considered packet rate (0.5 packets/s). Both the PDR and the end-to-end delay degrade sharply when the number of MNs exceeds the maximum value. This is because, when the number of MNs exceeds the maximum value corresponding to the considered packet generation rate, even if each MN has still a dedicated cell per slotframe, the resulting transmission rate in downstream is lower than the upstream generation rate. Hence, the packets tend to accumulate in the local buffer of the BRs, resulting in the increased delay and packet dropping

Fig. 8 also shows that the mobility pattern of MNs has no impact on the performance metrics, since the curves are overlapped. This is because, when using DD-DU scheduling, each MN is assigned a dedicated cell to both upstream and downstream directions in all BRs. In addition, the deployment area is completely covered by BRs with some overlapping (as shown in Fig. 5). Hence, the MNs can communicate at any time and no packet loss is experienced due to mobility.

To explore if a different speed could have a more significant impact on the performance, we ran a set of experiments by varying the speed of the MNs. Fig. 9 shows the results obtained when the mobility pattern is linear (we found similar results also for random mobility). We observed no impact on the performance even with a speed of 5 m/s.

VII. Conclusions and Future Work

In this paper, we have addressed the problem of mobility for Industrial IoT networks based on the 6TiSCH architecture defined by the IETF. Since 6TiSCH does not include specific mechanisms for mobility management, we have considered the Synchronized Single-hop Multiple-Gateway framework proposed in [5]. We have extended it with an efficient deployment policy for Border Routers and a flexible Scheduling Function that can be easily configured for meeting the requirements of different industrial applications leveraging mobile nodes. We have also analyzed the proposed Scheduling Function and defined an algorithm for calculating the maximum number of (mobile) nodes that can be accommodated, while guaranteeing the QoS required by the application. Finally, through simulation study, we have explored the impact of mobility on the performance of the system. Our results have shown that the proposed approach is able to manage very quickly the handover of mobile nodes, and that the node mobility has very limited effects on the performance.

The overhead for such a quick handover is high consumption of communication resources, since the schedule is replicated on all the BRs. As a future work, we intend to investigate more efficient solutions that can reduce the cost, in terms of allocated resources. Additionally, in this paper we have assumed that all the nodes have the same requirements in terms of QoS. In future, we plan to extend the proposed Scheduling Function to cope with heterogeneous QoS requirements. We will also analyze how to guarantee the required QoS in the presence of packet loss.

Acknowledgments

This work was partially supported by the Italian Ministry of Education and Research (MIUR) in the framework of the Crosslab project (departments of excellence) and the NSF grants under award numbers CSSI-2104078, SCC-1952045, and OAC-1725755 in the United States.

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.

[2] P. Thubert, "An Architecture for IPv6 over the Time-Slotted Channel Hopping Mode of IEEE 802.15.4 (6TiSCH)," RFC 9030, May 2021.

[3] IEEE, "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.

[4] Z. Ming and M. Xu, "Nba: A name-based approach to device mobility in industrial iot networks," *Computer Networks*, vol. 191, p. 107973, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128621001055

[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 Networks*, 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 *2019 IEEE Global Conference on Internet of Things (GCIoT)*, 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 *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, ser. IoTDI '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 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, 2021.

[9] D. Lee and A. Lin, "Computational complexity of art gallery problems," *IEEE Transactions on Information Theory*, vol. 32, no. 2, pp. 276–282, 1986.

[10] Q. Wang, X. Vilajosana, and T. Watteyne, "6TiSCH Operation Sublayer (6top) Protocol (6P)," RFC 8480, Nov. 2018.

[11] R. Alexander, A. Brandt, J. Vasseur, J. Hui, K. Pister, P. Thubert, P. Levis, R. Struik, R. Kelsey, and T. Winter, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, Mar. 2012.

[12] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," RFC 7252, Jun. 2014.

[13] G. Kazazakis and A. Argyros, "Fast positioning of limited-visibility guards for the inspection of 2d workspaces," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, 2002, pp. 2843–2848 vol.3.

[14] V. Chvatal, *A Combinatorial Theorem in Plane Geometry*. Journal of Combinatorial Theory (B), 1975.

[15] J. Culberson and R. Reckhow, "Covering polygons is hard," in *[Proceedings 1988] 29th Annual Symposium on Foundations of Computer Science*, 1988, pp. 601–611.

[16] R. Kershner, "The number of circles covering a set," *American Journal of Mathematics*, vol. 61, no. 3, pp. 665–671, 1939. [Online]. Available: http://www.jstor.org/stable/2371320

[17] W. Li, C. Zhu, V. C. M. Leung, L. T. Yang, and Y. Ma, "Performance comparison of cognitive radio sensor networks for industrial iot with different deployment patterns," *IEEE Systems Journal*, vol. 11, no. 3, pp. 1456–1466, 2017.

[18] G. F. Kuncir, "Algorithm 103: Simpson's rule integrator," *Commun. ACM*, vol. 5, no. 6, p. 347, jun 1962.

[19] O. Durmaz Incel, A. Ghosh, B. Krishnamachari, and K. Chintalapudi, "Fast data collection in tree-based wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 86–99, 2012.