Distributed colorings for collision-free routing in sink-centric sensor networks

Alfredo Navarra, Cristina M. Pinotti, Andrea Formisano

1 Dipartimento di Matematica e Informatica, Università degli Studi di Perugia, Via Vanvitelli 1, I-06123 Perugia, Italy

Abstract

When the environment does not allow direct access to disseminated data, a sensor network could be one of the most appropriate solutions to retrieve the map of interesting areas. Based on existing approaches, we start our study from the standard random deployment of a sensor network and then we consider a coarse-grain localization algorithm that associates sensors with coordinates related to a central node, called the sink. Once each sensor is associated with an estimated position, it starts to send data to the sink according to a designed schedule of communications that minimizes energy consumption and time by means of collisions avoidance. The outcome is a challenging combinatorial coloring problem for a specific graph class. We propose a schedule of communications based on distributed and fast coloring algorithms. The proposed solutions solve the underlying problems for the graphs of interest by means of an optimal, and in some cases near-optimal, number of colors. Finally, as the localization provides coarse-grain coordinates, different sensors might be associated with the same coordinates. Hence, in order to avoid that all such sensors perform the same actions (i.e., waste energy), a leader-election mechanism is considered.

1. Introduction

A duty-cycle wireless sensor and sink network (DC-WSN) consists of many randomly deployed tiny low-cost sensors that follow a duty-cycle (that is, a sleep-awake cycle), and a few powerful entities, called the sinks. At the best of our knowledge, the duty-cycle behavior was first introduced to save energy in sensor applications for wildlife monitoring [18]. Clearly, DC-WSNs are an extension of wireless sensor networks (WSNs) as we address uncertainty about the existence of a wireless link originating from the random sleep-awake schedules (see [17] for a complete review of WSNs and DC-WSNs).

Specifically, we consider a dense DC-WSN where each sink is mobile and, upon reaching a specific location, remains there to collect data from the sensors in the surrounding area, called the sink-region. Sensors are randomly deployed and are employed in applications where they remain unattended in a vast, possibly hostile, geographical area for long periods of time (e.g., environment monitoring and intruder tracking) [2,4]. Sensors perceive the physical world in their proximity, while sinks, equipped with much better processing capabilities, higher transmission power, and longer battery life, move around the area to collect, aggregate, and transmit to the external world the sensed data collected by the sensors [1,8].

© 2011 Published by Elsevier B.V.
When a sink reaches an area of interest in the network, the sensors in its vicinity must be organized into a short-lived and mission-oriented subnetwork called the sink-centric network.

In the rest of this paper, we will focus on the sink-centric network. We describe a new virtual infrastructure surrounding one sink that will be used for routing purposes, that is a variant of another virtual infrastructure previously proposed in [2,16]. Such new infrastructure, that consists of a discrete coordinate system on the sink-region, is imposed by a localization protocol (also referred to as a training protocol). Sensors that acquire identical coordinates form a cluster of indistinguishable nodes. This means that the information sent from a cluster to the sink will be always the same, regardless of the sending sensor. This suggests the usage of a leader election mechanisms inside each cluster in order to avoid that each sensor of a same cluster performs the same action, wasting precious energy. Once sensors in the sink-region are localized, sensory data are relayed to the sink based on a geographical routing protocol. Latency, energy efficiency, and collision avoidance are addressed in the design of the routing protocol. We assume that a collision occurs when a sensor receives more than one message at the same time. Therefore, to avoid message collisions, communication schedules have to be designed. The main contribution of the paper is the design of a communication schedule based on fast and distributed coloring algorithms, that are applied in order to accomplish collision-free leader election and routing tasks.

1.1. Outline

The remaining of this paper is organized as follows. After summarizing our model in Section 2, Section 2.1 describes the first contribution of the paper. In particular, the virtual infrastructure used in the literature is modified in favor of a uniform usage of the involved sensors. Section 3 introduces and optimally solves some coloring problems arising from the requirement of scheduling the communications from the sensors toward the sink without collisions with respect to some possible virtual infrastructures. Section 4 proposes a general framework that provides near-optimal solutions for all the considered virtual infrastructures. Section 5 describes how the proposed coloring can be used for both leader election and routing purposes. Finally, Section 6 provides concluding remarks, and points out possible directions for further investigations.

2. The model

This section revises the model assumptions and the virtual infrastructure proposed in [2,16,12] to organize DC-WSNs with respect to a central sink. Time is assumed to be divided into slots. All the sensors and the sink use equally long, in-phase slots, but they do not necessarily start counting time from the same slot. All the sensors possess three basic capabilities: sensing, computation, and wireless communication; and operate subject to the following constraints:

a. Each sensor alternates between sleep periods and awake periods – a sleep-awake cycle has a total length of \( L \) time-slots, out of which the sensor is in awake mode for \( d \) slots and in sleep mode for the remaining \( L - d \); The awake period is always made of \( d \) consecutive slots regardless of the starting time-slot (hence it may span over two consecutive cycles);

b. Each sensor is asynchronous – it wakes up for the first time according to its internal clock and it is not engaged in an explicit synchronization protocol, neither with the sink nor with other sensors. Sensors that wake up simultaneously at time-slot \( x \) are said of type \( x \) or equivalently, they belong to time-zone \( x \);

c. Individual sensors are unattended – once deployed it is neither feasible nor practical to devote attention to individual sensors;

d. No sensor has global information about the network topology, but each one can receive transmissions from the sink;

e. The sensors are anonymous – they are not associated with unique IDs;

f. Each sensor has a modest non-renewable energy budget and a limited transmission range \( r \) (the same for all the sensors);

g. Sensors can transmit and receive on multiple frequency channels. Moreover, the number of channels and frequencies are the same for all the sensors.

Concerning the training protocol that will be further discussed later, it imposes a virtual coordinate system (as in [16,12]) onto the sensor network by establishing:

1. Coronas: The sink-region area is divided into \( k \) coronas \( C_0, C_1, \ldots, C_{k-1} \) each of fixed width \( \rho > 0 \). The coronas are centered at the sink and determined by \( k \) concentric circles whose radii are \( \rho, 2\rho, \ldots, k\rho \), respectively;

2. Sectors: The sink-region is divided into \( h \) equiangular sectors \( S_0, S_1, \ldots, S_{h-1} \), originated at the sink, each having a width of \( \frac{2\pi}{h} \) radians.

If it is clear by the context, we may sometimes refer to clusters and sectors simply by specifying their cardinal number, i.e., \( C_i \) or \( S_j \) might be both denoted simply by \( i \). A cluster is the intersection between a corona \( c \) and a sector \( s \) where all sensors acquire the same coordinates, and it is denoted by \((c,s)\). Once the training protocol has terminated, we assume a data logging application, where the sensors are required to send their sensory data to the sink. When sensors transmit, if an awake sensor receives more than one message concurrently on the same frequency channel, we assume that it hears noise, i.e., a collision occurs and the messages get lost.
2.1. Localization

Many research papers have provided different approaches to make anonymous sensors aware of their coarse-grain positions [2,4,16,12,3,11,13,14]. In order to perform the training, two main procedures are usually executed. In the first procedure, the sink makes use of its uniform omnidirectional antenna for training sensors about the relying coronas. In the second one, the sink makes use of its directional antenna for training sensors about the relying sectors. Our interest is in the final virtual infrastructure implied by the coordinates acquired by the sensors during a training protocol. Differently from previous approaches, we maintain the area of each cluster roughly the same among the whole network. In this way, we better guarantee a uniform usage of the disseminated sensors in favor of better performances, and of an extended network lifespan. In order to obtain the desired configuration, let $\ell$ be the number of sectors imposed in corona 1.\(^2\) Considering $\rho = 1$, the number of sectors will be doubled at each corona $c = 2^p$, $0 < p \leq \lfloor \log_2(k-1) \rfloor$. In fact, corona $c = 2^p$ has area $\pi(2^{p+1} + 1)$ that is less than the double of the area of corona $c = 2^{p-1}$.

In doing so, we obtain that the proposed subdivision guarantees the following result:

**Lemma 1.** The ratio given by the area spanned by two generic clusters is at most 2.

**Proof.** Let $(c,s)$, $c > 1$, be a generic cluster of the imposed virtual infrastructure, and let $p = \lfloor \log_2 c \rfloor$ that implies $2^p \leq c < 2^{p+1}$. The area spanned by corona 1 is $3\pi$ and it is divided into $\ell$ sectors. The area spanned by the generic corona $c$ is $((c+1)^2 - c^2)\pi$ and it is divided by construction into $2^p\ell$ sectors. Hence, the area of one cluster in corona 1 is equal to $A_1 = \frac{3\pi}{\ell}$, while the area of one cluster in corona $c$ is equal to $A_c = \frac{(2c+1)\pi}{2^p\ell}$. The ratio gives:

$$\frac{A_1}{A_c} = \frac{3\pi}{\ell} \times \frac{2^p\ell}{(2c+1)\pi} \leq \frac{3 \cdot 2^p}{2 \cdot 2^p} = \frac{3}{2},$$

and

$$\frac{A_1}{A_c} = \frac{3\pi}{\ell} \times \frac{2^p\ell}{(2c+1)\pi} \geq \frac{3 \cdot 2^p}{2((2^{p+1} - 1) + 1)} \geq \frac{3 \cdot 2^p}{4 \cdot 2^p} = \frac{3}{4}.$$

Hence, the biggest ratio between the area of two generic clusters of the imposed virtual infrastructure gives:

$$\frac{3}{2}A_1 \times \frac{4}{3}A_1 = 2. \quad \Box$$

Fig. 1 illustrates the virtual infrastructure when $\ell = 3$. The sectors in corona $c$ are numbered from 0 to $h_c - 1$ starting to count from the sector above the x-axis. Noting that the outmost corona $c = k - 1$ will be divided into $h = \ell \cdot 2^{\lfloor \log_2(k-1) \rfloor}$ sectors, the virtual infrastructure can be obtained as an ordinary coordinate system with $k$ coronas and $h$ sectors, in which sensors in the inner coronas just ignore further subdivisions into more than the required sectors.

\(^2\) As it will be better clarified later, corona 0 is not considered in our arguments.

3. Coloring

Once that sensors are deployed and localized (cf. Section 2.1), we need to schedule their communications toward the sink in order to deliver the sensory data. Such communications should be scheduled in such a way that collisions are avoided, and the transmission delays as well as the energy consumption are minimized. To this aim, in the following, we introduce a frequency channel assignment (in terms of a coloring algorithm) on the adjacency graph associated with the virtual infrastructure imposed by the localization algorithm.

Namely, recalling that $\ell$ is the number of clusters in corona 1 of the virtual infrastructure, the adjacency graph $G_\ell$ has one node for each cluster in corona $c \geq 1$ and one edge for each pair of nodes corresponding to adjacent clusters. Formally:

**Definition 1.** The adjacency graph $G_\ell$ has one node $(c, s)$, with $1 \leq c \leq k - 1$ and $0 \leq s \leq h_c$, for each cluster in corona $c \geq 1$ of the virtual infrastructure. Two nodes $(c, s)$ and $(c', s')$, with $c > c'$, are adjacent if

1. $c = c'$ and $|s - s'| = 1$, or
2. $c = c' + 1$ and for some $x \in \mathbb{N}^+$, $2^{x-1} \leq c' < c < 2^x$ and $s = s'$, or
3. for some $x \in \mathbb{N}^+$, $c = c' + 1 = 2^x$ and $s' = \left\lfloor \frac{s}{2} \right\rfloor$.

Fig. 1 shows the virtual infrastructure when $\ell = 3$ and the corresponding adjacency graph $G_\ell$. For the rest of our discussion, we do not take into consideration corona 0, as the schedule of communications in there (included forwarding communications from outer coronas) is not necessary, due to the proximity of the sensors with the sink that can retrieve the information by itself. It is like assuming that if a transmission reaches corona 0 then it has reached the sink.

In the rest of this section we focus on colorings of graph $G_\ell$. In particular, assuming that the transmission range assigned to the sensors implies the adjacency graph $G_\ell$ among clusters, we require the following coloring:

**Definition 2.** A distance-two coloring (or frequency channel assignment), is a function that assigns to each node of $G_\ell$ a color in such a way that any two nodes at distance smaller than or equal to 2 are not assigned to the same color.

By providing the defined coloring, we obtain a collision-free schedule of the transmissions that must be performed by the sensors. In fact, such a coloring implies that two sensors residing in two different clusters with a common neighbor never transmit on the same channel, as different colors specify different communication frequency channels. Hence, adjacent clusters can perform in parallel their communications without causing collisions. Clearly, the minimization of the used colors implies the minimization of the frequencies that the schedule requires for a round of transmissions. In the following, we will refer to distance-two coloring simply as coloring algorithm. We will postpone to Section 5 how such a coloring (or, scheduling) can be used for leader election and/or for routing purposes.

By construction, it follows that the largest subset of pairwise nodes at distance at most 2 of $G_\ell$, $\ell > 0$, has size 6 (see for instance the shadowed nodes in Figs. 1 and 2 for the cases $\ell = 3$ and $\ell = 4$, resp.). Thus:

**Lemma 2.** Any coloring of $G_\ell$, $\ell > 0$, that satisfies the distance-two constraint requires at least 6 colors.

**Proof.** A generic node $x \in G_\ell$ corresponds to a cluster $(c, s)$ in the imposed coordinate system. The neighbors of $x$ corresponding to other clusters in $c$ are at most 2. By construction, $x$ admits only one neighbor corresponding to corona $c - 1$ while at most two neighbors corresponding to corona $c + 1$. In fact, clusters are doubled at every corona labeled by a power of two. Moreover, the clusters established at some corona are maintained along all the coronas with bigger labels. Thus, $(c, s)$ admits only one neighbor $(c - 1, s)$ in corona $c - 1$ and at most two neighbors in corona $c + 1$, when $c + 1$ is labeled by a power of two. All these neighbors along with $x$ form then the biggest set of nodes in $G_\ell$ at pairwise distance at most 2. Hence 6 colors are required by any coloring satisfying the distance-two constraint.

Motivated by the well-known Brooks theorem [5], which proves that $\Delta$ colors are sufficient to color any graph with a maximum degree of $\Delta$, it is possible to do a coloring within distance 2 of $G_\ell$, that is, a coloring of the graph $G_\ell^2$ (i.e., $G_\ell$ augmented by those edges between any two nodes at distance 2 in $G_\ell$), using at most 16 colors. In fact, $G_\ell^2$ has maximum degree 16 (see for instance Fig. 1). Moreover, since $G_\ell$ is planar, related bounds concerning distance-two colorings for the square of planar graphs can be found in [15,7], and references therein. However, in our context, all such results do not imply any better bound than the cited 16.

In the following, on the contrary, by exploiting the structure of $G_\ell$, we provide optimal coloring algorithms using exactly 6 colors, and near-optimal coloring algorithms using at most 9 colors for the considered graphs.

Let $|i|_j$, with $i$ and $j \in \mathbb{N}^+$, denote the modulo operation, that is the non-negative remainder of the integer division of $i$ by $j$, then the next property can be stated:

**Property 1.** For any $\ell > 2$ and $c = 2^p - 1$ with $h_c = \ell \cdot 2^p + 1$, the set $S_{(c, x)} = [(c - 1, x), (c, x), (c, |x - 1|_{h_c}), (c, |x + 1|_{h_c}), (c + 1, |2x|_{2h_c}), (c + 1, |2x + 1|_{2h_c})]$ consists of 6 nodes at pairwise distance at most 2.
On the left, the virtual infrastructure divided into clusters uniquely identified when \( \ell = 4 \). On the right, the corresponding adjacency graph \( G_4 \). The shadowed nodes represent a maximal subset of nodes at pairwise distance at most 2 in the graph, i.e. each pair of nodes in the subset is at distance at most 2.

Table 1
Resume of the proposed coloring algorithms with their performances.

<table>
<thead>
<tr>
<th>( \ell )</th>
<th>Algorithm</th>
<th># of colors</th>
<th>Lower bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \ell = 3 \cdot 2^i, i \geq 0 )</td>
<td>OPT3</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>( \ell = 4 ) or ( \ell = 5 )</td>
<td>OPT4</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>( \ell = 0, \ell &gt; 4 )</td>
<td>Col4</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>( \ell \geq 7 )</td>
<td>ColG</td>
<td>9</td>
<td>6</td>
</tr>
</tbody>
</table>

Proof. It suffices to note that all the clusters of each set defined in the claim are at mutual distance less than or equal to 2. □

From now on, let cluster \((c, x)\) be termed the root of set \( S_{(c,x)} \), for any \( c \) and \( x \). In the following, we propose some distance-two coloring algorithms that differ in the number of used colors and in the complexity. In particular, as shown in Table 1, algorithm OPT3 for coloring \( G_3 \) is shown to be optimal since it makes use of exactly 6 colors. We then show that such an algorithm can be easily extended to any \( G_\ell \) with \( \ell = 3 \cdot 2^i, i \geq 0 \). For \( G_4 \), we provide algorithm OPT4 that makes use of 7 colors. Moreover, we prove that OPT4 is optimal by providing an impossibility result about the colorability of \( G_4 \) by means of only 6 colors. The optimal algorithm OPT4 also provides a method for obtaining an optimal coloring of \( G_5 \) by means of 7 colors. Moreover, we propose a sub-optimal algorithm Col4 for coloring any \( G_\ell \), with \( |\ell|_4 = 0 \), that uses 8 colors, and hence, for \( \ell > 4 \), at most 2 colors more than the optimum. All the algorithms presented in what follows exhibit a very useful property, that is, each cluster can be colored within a constant number of steps. Hence, the sensors can apply the designed algorithms once they know the coordinates of the clusters where they reside and the coloring of the first 2 (sometimes 4) coronas. Finally, Col4 is extended for coloring any \( G_\ell \) with \( \ell \geq 7 \) using at most 9 colors.

3.1. Optimal coloring for \( G_3 \)

An optimal distance-two-coloring OPT3 can be found for the virtual infrastructure that partitions the first corona in 3 sectors (Fig. 1) and whose adjacency graph is denoted as \( G_3 \). Algorithm OPT3 is based on two subsets of colors: \{0; 1; 2\}, \{3; 4; 5\}. The first set is used for odd coronas, the second one for even coronas. This realizes the property for which two clusters at two adjacent different coronas cannot get the same color. Moreover, for each corona a sequence of the colors is properly selected and repeated for coloring all the sectors in anti-clockwise order. Thus, two clusters associated with the same color at the same corona are at a distance that is a multiple of 3.

Starting from corona 1, we color the clusters using the sequence of colors \{0; 1; 2\} in an anti-clockwise order. Then corona 2 will be colored in the same way but using the sequence \{3; 4; 5\} twice. Any other cluster \((c, s)\) is colored according to the following actions:

- **Shifting**: Given a sequence of colors \{0, 1, 2\}, a shifting operation consists in summing \(|-1|_3\) to each element of the sequence, hence obtaining \{2, 0, 1\}.
Swapping: Given a sequence of colors \( \{0, 1, 2\} \), a swapping operation consists in exchanging the first element of the sequence with the third one, hence obtaining the sequence \([2, 1, 0]\).

Any cluster \((c, s), c > 2\), is colored in the following way (see Fig. 4): if the number of clusters in corona \(c\) is the same as in corona \(c - 2\), then corona \(c\) is colored with the sequence obtained from the sequence used in corona \(c - 2\) by applying a shifting operation. If the number of clusters in corona \(c\) is doubled with respect to corona \(c - 2\), then corona \(c\) is colored with the sequence obtained from the sequence used in corona \(c - 2\) by applying a swapping operation.

**Lemma 3.** Algorithm OPT3 assigns colors to clusters satisfying the distance-two constraint.

**Proof.** It has been already pointed out how different colors are assigned to clusters at distance 1. Moreover, if two clusters of the same color belong to the same corona, then they are at a distance that is a multiple of 3. Therefore, the proof only needs to show the correctness of the coloring for clusters at distance 2 in different coronas. Let \((c, s)\) and \((c', s')\) be two clusters at distance 2, with \(c > c'\).

**Fig. 3** shows the possible configurations. If the number of sectors in \(c\) is the same as in \(c'\), then \(s\) must be equal to \(s'\). In this case, the sequence of colors used to color \(c\) is obtained form the sequence used in \(c'\) after a shifting. This implies that colors assigned to \((c, s)\) and \((c', s')\) are different. Another configuration occurs when the number of sectors in \(c\) is doubled with respect to \(c'\), then \(s\) is equal either to \(2s'\) or to \(2s' + 1\). Since in this case the sequence of colors used to color \(c\) has been obtained from the one used in \(c'\) after a swapping operation, by construction \((c, s)\) may assume any color in the sequence but the one associated to \((c', s')\). In fact, let \([0, 1, 2]\) be the sequence of colors used in \(c'\), then the sequence \([2, 1, 0]\) is used in \(c\). Hence, if the sequence of colors used at corona \(c'\) is \([0, 1, 2]\) and OPT3 assigns color 0 (or 1, or 2, resp.) to \((c', s')\), then it assigns color \(2\) (0, 1, resp.) to \((c, 2s')\), and 1 (2, 0, resp.) to \((c, 2s' + 1)\). □

We now show that each color assigned by OPT3 to a generic cluster \((c, s)\) can be evaluated in a constant number of steps with the only assumption of knowing the sequences of colors used in coronas 1 and 2. For ease of analysis, from now on we focus only on one set of three colors for coloring all the coronas instead of presenting two specular arguments for odd and even coronas, respectively.

Assuming we know the sequence of colors used at a generic corona \(c'\), the sequence used to color any \(c > c'\) that has the same number of clusters of \(c'\) can be easily evaluated. In fact, it is sufficient to apply the shifting operation \(c - c'\) times. As such operation is associative, the result does not change if we decrease by \(|(c - c')|_3\) each single element of the sequence. However, when the number of sectors is doubled with respect to \(c'\) then we need a more careful computation. Actually, we evaluate the first and the third colors of the sequence independently. The second then comes as a consequence. The next technical lemma provides a first contribution to the evaluation of the required sequence of colors.

**Lemma 4.** Let \([0, 1, 2]\) be the sequence of colors used for corona 1, \(c = 2^p\) for some \(p > 0\), and \([X', Y', Z']\) be the sequence of colors used for corona \(c' = 2^{p-1}\), then the sequence of colors \([X, Y, Z]\) used for corona \(c\) can be evaluated as follows:

(a) if \(|p|_2 = 0\) then \(X = X', Z = |Z' + 1|_3\) and \(Y = \{X', Y', Z'\} \setminus \{X, Z\}\);
(b) if \(|p|_2 = 1\) then \(X = |X' - 1|_3, Z = |Z' + 1|_3\) and \(Y = \{X', Y', Z'\} \setminus \{X, Z\}\).

**Proof.** We prove the lemma by induction on \(p\). The base of the induction is given for the two cases \(p = 1\), and \(p = 2\). In the first case, corona \(c = 2^p = 2\) is colored by using the sequence \([2, 1, 0]\) obtained from the one of corona 1 by applying a swapping operation, hence obtaining \(X = 2 = |X' - 1|_3, Z = 0 = |Z' + 1|_3\) and \(Y = 1\). In the second case, corona \(c = 2^p = 4\) is colored by using the sequence \([2, 0, 1]\) obtained from the one of corona 2 by first applying a shifting operation and then a swapping one, hence obtaining \(X = 2 = X', Z = 1 = |Z' + 1|_3\) and \(Y = 0\). We assume the claim as true for any \(p - 1 \leq 2\) and we prove it for \(p = 2\). Corona \(c = 2^p\) is colored by using the sequence \([X, Y, Z]\) obtained from the sequence \([X', Y', Z']\) used...
in corona $c' = 2^{p-1}$ after applying $2^{p-1} - 1$ shifting operations and one swapping operation. Hence, $X = |Z' - (2^{p-1} - 1)|_3$, $Y = |Y' - (2^{p-1} - 1)|_3$, and $Z = |X' - (2^{p-1} - 1)|_3$. This leads to $X = |Z' - (2^{p-1} - 1)|_3 = |Z' - (2^{p-1} - 1)|_3 = |Z' - 2^{p-1} + 1|_3 = |Z' - 2^{p-1} + 1|_3$. By induction, the sequence $X$, $Y$, and $Z$ is known, the corona will be colored in anti-clockwise order from sector 0. Precisely:

$$X' = X - \left\lfloor \frac{p}{2} \right\rfloor _3 - |c - c'|_3, \quad Z' = Z + |p|_3 - |c - c'|_3, \quad X' \neq Y' \neq Z' \quad \text{and} \quad Y' \in \{0, 1, 2\}.$$ 

Once the sequence of colors $\{X', Y', Z'\}$ used to color corona $c$ is known, the corona will be colored in anti-clockwise order from sector 0. Precisely:

$$OPT3(c, s) = \begin{cases} 
X' & \text{if } |s|_3 = 0, \\
Y' & \text{if } |s|_3 = 1, \\
Z' & \text{if } |s|_3 = 2.
\end{cases}$$

Hence, evaluating $OPT3(c, s)$ takes constant number of steps. \hfill \Box

Fig. 4 shows the correct coloring obtained for both the odd and the even coronas by applying the described $OPT3$ algorithm. The initial step is constituted by starting with the coloring of corona 1 with the sequences $\{0, 1, 2\}$ and $\{5, 4, 3\}$ for odd and even coronas, respectively. It is interesting to notice that $OPT3$ can be easily extended to any graph $G_\ell$, where $\ell = 3 \cdot 2^x$ for any $x \geq 1$.

**Corollary 1.** For any positive integer $x$, let $\ell = 3 \cdot 2^x$. Then, $G_\ell$ can be colored with 6 colors by means of the same rules defined by algorithm $OPT3(c, s)$.

**Proof.** It is sufficient to notice that the coloring for $G_\ell$ with $\ell = 3 \cdot 2^x$ is almost the same than that obtained on $G_3$ starting from corona $c = 2^1$. The only difference resides in the number of coronas dividing two consecutive powers of 2 (for instance, if $x = 1$, $G_6$ has 12 clusters in corona 2 and 24 clusters in corona 4, while $G_3$ has 12 clusters in corona 4, and 24 clusters in corona 8). However, this does not affect the validity of the proof of Lemma 3, hence ensuring the possibility to obtain a feasible distance-two coloring for the general case of $\ell = 3 \cdot 2^x$. \hfill \Box

### 3.2. Optimal coloring for $G_4$

**Lemma 5.** Any distance-two coloring for the first two coronas of $G_4$ requires 6 colors. Moreover, in order to perform the distance-two coloring for the first two coronas of $G_4$ by means of 6 colors, each color must be used exactly twice.

---

Note that the sequence of colors used for corona 1 may refer, without distinction, to the set of three colors used for odd coronas or even coronas.
Fig. 4. The coloring obtained by applying algorithm OPT3 on the virtual infrastructure with three sectors at corona 1, and considering the sequences \((0, 1, 2)\) and \((5, 4, 3)\) for corona 1 for coloring the odd and the even coronas, respectively.

**Proof.** Consider corona 1, it requires 4 different colors in order to accomplish a distance-two coloring. Without loss of generality, we can assign color RED\(^2\) to cluster \((1, 0)\), BLUE to \((1, 1)\), GREEN to \((1, 2)\), and YELLOW to cluster \((1, 3)\). Clusters \((2, 0)\) and \((2, 1)\) can be colored either with two new colors or with the GREEN and a new color. No other possibilities are allowed.

In the former case, assume that colors PINK and BROWN are used for clusters \((2, 0)\) and \((2, 1)\), respectively. Then, clusters \((2, 2)\) and \((2, 3)\) must take colors YELLOW and PINK, respectively, unless more than 6 colors are used. Similarly, clusters \((2, 6)\) and \((2, 7)\) must take colors BROWN and BLUE, respectively. It follows that clusters \((2, 4)\) and \((2, 5)\) require two new colors to accomplish the distance-two constraint.

In the latter case, without loss of generality, clusters \((2, 0)\) and \((2, 1)\) take colors GREEN and PINK, respectively. It follows that clusters \((2, 6)\) and \((2, 7)\) require the BLUE color and a new one, say BROWN. Hence, six colors are necessary. In order to show that six colors are enough, we can complete the above coloring by assigning colors RED and PINK to clusters \((2, 4)\) and \((2, 5)\), and colors YELLOW and BROWN to clusters \((2, 2)\) and \((2, 3)\), hence necessarily using each color twice. □

**Lemma 6.** Assuming that 6 colors are enough for a distance-two coloring of \(G_4\), then any color used in corona 1 must be used at least 5 times in the first 4 coronas of \(G_4\).

**Proof.** Let RED be the color used for cluster \((1, 0)\) in \(G_4\). Without loss of generality, by Lemma 5, such a color is also used in \((2, 4)\) (the other possibility would be \((2, 5)\)). Now, consider all the sets of six nodes defined by Property 1 in \(G_4\). If six colors are enough for a distance-two coloring of \(G_4\), then we show that in order to satisfy the property, color RED must be used 5 times in the first 4 coronas. The set, \(S_{(2,4)}\), is the only one already containing the RED color. By the distance-two constraint, the set \(S_{(2,5)}\), must have the RED color at cluster \((3, 6)\) or \((4, 10)\) or \((4, 11)\). The set \(S_{(2,6)}\) must have the RED color at cluster \((3, 6)\) or \((4, 12)\) or \((4, 13)\). Finally, the set \(S_{(2,7)}\) must have the RED color at cluster \((3, 6)\) or \((4, 14)\) or \((4, 15)\). In order to satisfy all those three sets, the only solution is to assign color RED to cluster \((3, 6)\). In fact, assuming that the last set gets the RED color at either \((4, 14)\) or \((4, 15)\), then: In the first case, there is no cluster in the second set that can be colored by RED. In the second case, the only cluster from the second set that can be colored by RED is \((4, 12)\) but then there is no cluster from the first set that can be colored by RED. This implies that only cluster \((3, 6)\) from those sets can be colored by RED since it is shared by all such sets. There remain other four sets that must accomplish Property 1. They are, (i) \(S_{(2,0)}\), (ii) \(S_{(2,1)}\), (iii) \(S_{(2,2)}\), and (iv) \(S_{(2,3)}\). The cluster that must be colored by RED from (i) must be chosen among \((3, 0)\), \((4, 0)\) or \((4, 1)\). From (ii) we have \((3, 1)\), \((3, 2)\), \((4, 2)\) or \((4, 3)\), from (iii) \((3, 2)\), \((3, 1)\), \((4, 5)\) or \((4, 6)\), and from (iv) \((3, 2)\), \((4, 6)\) or \((4, 7)\). If either \((4, 0)\) or \((4, 1)\) is chosen from (i) to be colored in RED, then only cluster \((3, 2)\) shared by all the remaining sets can be colored by RED. Symmetrically, if cluster \((3, 0)\) is chosen from (i) than it remains only to chose one cluster from (iv) between \((4, 6)\) and \((4, 7)\). In total, the RED color must be used 5 times in the first 4 coronas of \(G_4\), and this clearly holds for any other color used in corona 1. □

\(^4\) In the proofs, we call explicitly colors by name, i.e. GREEN, RED, BLUE, YELLOW, PINK and BROWN for clarity.
Proof. Assume by contradiction that six colors are enough for a distance-two coloring of $G_4$. Consider the first 4 coronas of $G_4$ composed of 36 nodes. From Lemma 6, we have that each color used in corona 1 must be used exactly 5 times in the first 4 coronas of $G_4$. It follows that only 20 nodes of the first 4 coronas are colored by the four colors used in corona 1. Then, one of the other two available colors must be used at least 8 times for coloring the remaining 16 nodes. In particular, since from Lemma 5 such a color has been used only two times in the first two coronas of $G_4$, it must be used at least other six times in coronas 3 and 4. By construction of $G_4$ and from the distance-two constraint, a color can be used at most five times in corona 4 without allowing any occurrence in corona 3, or one time in corona 3 and other four times in corona 4. In total, the same color can be used at most five times in coronas 3 and 4, hence contradicting the hypothesis. □

We are now ready to present a distance-two coloring algorithm for $G_4$ that makes use of exactly 7 colors, hence, from Theorem 2 it is optimal. The new algorithm, called OPT4, cannot exploit the property of using two disjoint subsets of colors for the even and the odd coronas since there is no way to use only 3 colors on one corona. However, we show that by applying two operations similar to the aforementioned Shifting and Swapping, we can obtain a feasible distance-two coloring by means of 7 colors. Actually, we still make use of an operation similar to the Swapping, while we introduce a new one in place of the Swapping. Let $c > 2$ and recalling that $h_c$ is the number of clusters in corona $c$, the new operations are:

- **Rotating**: Given a cluster $(c, s)$, if $h_c = h_{c-2}$, then $(c, s)$ is colored with the same color of cluster $(c-2, s-1)_{h_{c-2}}$.
- **Doubling**: Given a cluster $(c, s)$, if $h_c = 2h_{c-2}$, then $(c, s)$ is colored with the same color of cluster $(c-2, |\lfloor \frac{s}{2} \rfloor + 1)_{h_{c-2}}$ if $s$ is even, and $(c-2, |\lfloor \frac{s}{2} \rfloor + 1)_{h_{c-2}}$ if $s$ is odd.

We now show by induction that the defined operations lead to a feasible distance-two coloring for $G_4$ if the first 4 coronas are suitably initialized. The obtained coloring is optimal and it makes use of only 7 colors. Moreover, it verifies the further property:

**Property 2.** Given two coronas $c \geq 4$ and $c' = c + 1$, with the same number of clusters, the colors associated with any pair of clusters $(c, s)$ and $(c', s + 2h_c)$ or $(c', s)$ and $(c, s + 2h_c)$ are always different.

**Theorem 3.** Algorithm OPT4 assigns colors to clusters satisfying the distance-two constraint, it is optimal and satisfies Property 2.

Proof. We prove the theorem by induction on the number $c > 4$ of coronas. The first 4 coronas are colored as shown in Fig. 5. It is easy to check that the coloring of corona 5 obtained by applying a doubling operation is a feasible distance-two coloring that satisfies Property 2.

We then assume by induction that $G_c$ can be colored up to any given corona $c > 4$ by starting from the 4 coronas shown in Fig. 5 and by applying OPT4. We show that we can extend the coloring to corona $c + 1$ by means of the same algorithm while maintaining all the required properties. We have to consider three different cases: (i) $c + 1$ contains the same number of clusters than $c - 1$; (ii) $c + 1$ contains a number of clusters doubled with respect to $c - 1$, and $c + 1$ is even; (iii) $c + 1$ contains a number of clusters doubled with respect to $c - 1$, and $c + 1$ is odd.

In case (i), the rotating operation has been applied and hence, all the colors assigned to clusters of corona $c + 1$ satisfy the distance-two constraint since, by induction, the same sequence of colors was feasible at corona $c - 1$. We then need
to show that any cluster \((c + 1, s)\) gets a color different than that of its neighbors \((c, |s - 1|_{h_c}), (c, s), (c, |s + 1|_{h_c})\), and \((c - 1, s)\) at distance less than or equal to 2 in coronas \(c\) and \(c - 1\). By the rotating operation, \((c + 1, s)\) gets the same color of cluster \((c - 1, |s - 1|_{h_c})\). By induction, such a color is different from the ones assigned to clusters \((c - 1, s), (c, |s - 1|_{h_c})\), and \((c, s)\) since those clusters are also at distance less than or equal to 2 from \((c - 1, |s - 1|_{h_c})\). Moreover, by Property 2, the color assigned to \((c - 1, |s - 1|_{h_c})\) also differs from the one assigned to \((c, |s + 1|_{h_c})\). Thus the color assigned to \((c + 1, s)\) differs from those of \((c, |s - 1|_{h_c}), (c, s), (c, |s + 1|_{h_c})\), and \((c - 1, s)\). To conclude the case, we need to show that the applied operation also satisfies Property 2. Clearly, the color assigned to \((c, |s - 2|_{h_c})\) is different than that of \((c + 1, s)\), because cluster \((c, |s - 2|_{h_c})\) is at distance 2 from \((c - 1, |s - 1|_{h_c})\). Thus, it remains to show that the color assigned to cluster \((c, |s + 2|_{h_c})\) is different from that of \((c + 1, s)\). To this aim, we need to distinguish two cases: (a) \(c - 1\) has as many clusters as corona \(c - 2\); (b) the number of clusters in corona \(c - 1\) is doubled with respect to corona \(c - 2\) (i.e., \(c - 1\) is a power of two).

Case (a): Clusters \((c + 1, s)\) and \((c, |s + 2|_{h_c})\) get their colors from \((c - 1, |s - 1|_{h_c})\) and \((c - 2, |s + 1|_{h_c})\), respectively, that are different by induction on Property 2.

Case (b): if \(s\) is odd, cluster \((c + 1, s)\) gets its color from \((c - 3, |s + 1|_2 - 1|_{h_{c - 2}})\) by applying first a doubling operation and then a rotation; while cluster \((c, |s + 2|_{h_c})\) gets the same color as cluster \((c - 4, |s + 1|_2 - 1|_{h_{c - 2}})\) by the same operations. Since by induction on Property 2, clusters \((c - 3, |s + 1|_2 - 1|_{h_{c - 2}})\) and \((c - 4, |s + 1|_2 - 1|_{h_{c - 2}})\) have associated different colors, the same holds for clusters \((c + 1, s)\) and \((c, |s + 2|_{h_c})\). If \(s\) is even, cluster \((c + 1, s)\) gets its color from \((c - 3, |s + 1|_2 - 1|_{h_{c - 2}})\) while cluster \((c, |s + 2|_{h_c})\) gets its color from \((c - 2, |s + 1|_2 - 1|_{h_{c - 2}})\). By induction, \((c - 3, |s + 1|_2 - 1|_{h_{c - 2}})\) and \((c - 2, |s + 1|_2 - 1|_{h_{c - 2}})\) are differently colored because they are adjacent.

In case (ii), the doubling operation has been applied. Coronas \(c - 1\) and \(c\) have \(h_c = h_{c - 1}\) clusters, while corona \(c + 1\) has \(2h_c\) clusters.

Without loss of generality, we assume \(s\) to be even. Observe that clusters \((c + 1, s)\) and \((c - 1, |s - 1|_{h_c})\) get the same color. Regarding clusters \((c + 1, |s - 1|_{h_c})\) and \((c + 1, |s - 2|_{h_c})\), they get their colors from \((c - 1, |s + 1|_{h_c})\) and \((c - 1, |s + 2|_{h_c})\), respectively. Since such clusters are at distance 1 from cluster \((c - 1, |s + 1|_{h_c})\), they differ in color from cluster \((c + 1, s)\). Regarding the neighbors \((c, |s + 1|_{h_c})\), \((c, s)\), \((c, s - 1)\) of cluster \((c + 1, s)\), they are at distance not greater than 2 from cluster \((c - 1, |s - 1|_{h_c})\) and thus they must be colored differently. Moreover, by Property 2, clusters \((c, |s + 1|_{h_c})\) and \((c - 1, |s - 1|_{h_c})\) have different colors. Thus, cluster \((c + 1)\) and all its neighbors at distance less than or equal to 2 in corona \(c\) or \(c - 1\) get different colors. The case of odd \(s\) is symmetric.

Note that Property 2 does not apply here since \(c\) and \(c + 1\) do not have the same number of clusters.

In case (iii), let corona \(c - 1\) have \(h_{c - 1} = h_c/2\) clusters, while coronas \(c\) and \(c + 1\) have \(h_c = h_{c - 1}\) clusters. The rotating operation has been applied from corona \(c\) to corona \(c + 1\) and hence, all the colors assigned to clusters of corona \(c + 1\) satisfy the distance-two constraint since, by induction, the same sequence of colors was feasible at corona \(c - 1\). Regarding the clusters at distance not greater than 2 from \((c + 1, s)\) in coronas \(c - 1\) and \(c\), observe that the doubling operation has been applied. Assume \(s\) to be even. Cluster \((c + 1, s)\) gets the same color than \((c - 1, |s - 1|_{h_{c - 1}})\). The neighbors of cluster \((c + 1, s)\) at distance less than or equal to 2 belonging to the preceding coronas, \((c, s), (c, |s - 1|_{h_c}), (c, |s + 1|_{h_c})\) and \((c - 1, |s - 1|_{h_{c - 1}})\), are also at distance less than or equal to 2 from \((c - 1, |s - 1|_{h_{c - 1}})\), hence they are colored, by induction, differently from \((c + 1, s)\). We also need to prove that Property 2 holds for \((c + 1, s)\) and the two clusters \((c, |s - 2|_{h_c})\) and \((c, |s + 2|_{h_c})\). Since \((c, |s - 2|_{h_c})\) is at distance 1 from \((c - 1, |s - 1|_{h_{c - 1}})\), it gets a different color than \((c + 1, s)\). For the color of cluster \((c, |s + 2|_{h_c})\), we know by induction that it comes from a doubling operation applied to cluster \((c - 2, |s + 1|_{h_{c - 1}})\) that is at distance 2 from \((c - 1, |s + 1|_{h_{c - 1}})\), and by induction the claim holds. The case of odd \(s\) is symmetric. □

The above theorem provides a useful method for obtaining feasible distance-two colorings that make use of 7 colors for a generic graph \(G_c\). In fact, it suffices to provide a suitable coloring of the first 4 coronas of \(G_c\) in such a way that by applying the first doubling operation for coloring corona 5, the distance-two constraint and Property 2 are verified. Fig. 5, for instance, shows a suitable coloring for the first 4 coronas of \(G_5\).

It is worth to point out that, once the suitable coloring for the first 4 coronas of \(G_c\) is provided, the algorithm described above color \(G_c\) in time linear in the number of its nodes. Moreover, note that each single cluster \((c, s)\) can be colored in time \(O(\log c)\) once observed that: (i) each cluster derives its color from the color assigned to one of the clusters in the first 4 coronas (ii) if the coronas \(c - j, c - j + 1, \ldots, c - 1, c\) have the same number of clusters and \(j\) is even, the color of cluster \((c, s)\) is the same as the color of cluster \((c - j, |s - 1|_{h_{c - 1}})\).

3.3. Optimal coloring for \(G_5\)

As outlined in the previous section, by providing a suitable coloring for the first 4 coronas of \(G_5\) (see Fig. 5, on the right) we can apply the same rotating and doubling operations of OPT4 to obtain a distance-two coloring for \(G_5\) by means of 7 colors. We now show that such an algorithm is also optimal. First, any distance-two coloring of corona 1 of \(G_5\) requires exactly 5 colors. Then:

Lemma 7. If six colors are enough for a distance-two coloring of \(G_5\), any color used in corona 1 of \(G_5\) can be used at most 7 times in the first 4 coronas of \(G_5\).
Proof. Let RED be the color used for cluster \((1, x)\) in \(G_5\), with \(0 \leq x \leq 4\). By the distance-two constraint, color RED cannot be reused at the roots of the 6 sets \([S(2,2|x-2|10), S(2,2|x-1|10), S(2,2|x+1|10)], S(2,2|x+2|10), S(2,2|x+3|10)]\). However, since each set must have exactly one occurrence of color RED if six colors are enough, color RED must be necessarily reused in some clusters of coronas 3 and 4 of the sets \([S(3,2|x-2|10), S(3,2|x-1|10), \ldots, S(3,2|x+3|10)]\). Precisely, color RED must color 2 clusters in corona 3 of \([S(2,2|x-2|10), S(2,2|x+3|10)]\). Indeed, if color RED is never used for the clusters in corona 3 of such sets, it can be used at most 4 times in the clusters of corona 4, and at least 2 sets among \([S(2,2|x-2|10), S(2,2|x+3|10)]\) will have no occurrence of color RED, and cannot be completely colored. Moreover, if color RED is used only once in corona 3, say to color cluster \((3, x)\), color RED already occurs in the 3 sets \([S(3,2|x-1|10), S(3,2|x), S(3,2|x+1)]\). Then, to cover the remaining 3 sets \([S(3,2|x+2|10), S(3,2|x+3|10), S(3,2|x+4|10)]\), 3 clusters in corona 4 should be colored RED, but this is impossible. In conclusion, to guarantee one occurrence of color RED in each set \([S(2,2|x-2|10), S(2,2|x-1|10), \ldots, S(2,2|x+3|10)]\), we must use color RED 3 times: once in corona 1, and twice in corona 3.

To complete the coloring of the first 4 coronas of \(G_5\), it remains to color the 4 sets \([S(3,2|x+4|10), S(3,2|x+5|10), S(3,2|x+6|10)]\). Since color RED must be used exactly once in each set, color RED can be overall used at most 7 times.

We can now state that at least 7 colors are needed for \(G_5\), thus proving that the proposed coloring is optimal.

**Theorem 4.** Any distance-two coloring for \(G_5\) requires more than 6 colors.

**Proof.** Assume by contradiction that 6 colors are enough for a distance-two coloring of \(G_5\). By Lemma 7, the first 5 colors used in corona 1 can be used at most 7 times. Thus, at most 35 clusters in the first 4 coronas of \(G_5\) can be colored with the colors used in corona 1. Since overall there are 45 clusters to be colored in the first 4 coronas of \(G_5\), the remaining color, say color BLUE, must be used at least 10 times in coronas 2, 3, and 4 of \(G_5\). This can be achieved only assigning color BLUE to exactly one cluster in coronas 2 or 4 of each set rooted at the clusters of corona 2. However to satisfy the distance-two constraint, color BLUE can be used at most \(|\frac{20}{3}| = 6\) and \(|\frac{10}{3}| = 3\) times in coronas 4 and 2, respectively. Therefore, color BLUE can be used at most 9 times, and no coloring of \(G_5\) with 6 colors is possible.

**4. Coloring for any \(G_\ell\)**

So far, we have shown how to optimally color \(G_3, G_4, G_5, G_6\) and some other cases. Now, we show how to color any \(G_\ell\), with \(\ell \geq 7\), using at most 9 colors. We start proposing a coloring that works for any \(\ell\) multiple of 4 that requires 8 colors. Then, we extend it for coloring any \(G_\ell\) just using one extra color. Since, by Lemma 2, at least 6 colors are required, our coloring uses at most 3 extra colors.

**4.1. Coloring for \(G_\ell\), with \(|\ell|_4 = 0\)**

The coloring algorithm, called Col4, is extremely simple. Each sensor copies the color of the cluster where it resides by using the matrix \(M_4\) depicted in Table 2. More precisely, Col4 assigns to cluster \((c, s)\), \(0 < c < k\) and \(0 \leq s < h_c\), the entry of \(M_4\) specified as follows:

\[
\text{Col4}(c, s) = \begin{cases} 
M_4[0, |s|_4] & \text{if } c = 1 \text{ and } 0 \leq s \leq h_c, \\
M_4[|c - 2|_4 + 1, |s|_4] & \text{if } c \geq 2 \text{ and } 0 \leq s \leq h_c.
\end{cases}
\]

Note that, corona 1 is colored with row 0 of \(M_4\), while all the remaining coronas of \(G_4\) are colored by using the rows 1–4 of \(M_4\), cyclically.

We have to show that such a coloring satisfies the imposed distance-two constraint. First of all we point out that two clusters belonging to two different adjacent coronas necessarily acquire two different colors. In fact, \(M_4\) has two different subsets of colors used for even and odd rows, respectively. Another simple observation is that if two clusters of the same color belong to the same corona, then they are at distance at least 4. The next lemma shows the remaining cases that must be addressed to prove the correctness of the coloring (see Fig. 3 for a visualization).

**Lemma 8.** Consider two clusters \((c, s)\) and \((c', s')\). If \(c = c' + 2\) and (a) \(s = s'\), or (b) \(s = 2s'\) or (c) \(s = 2s' + 1\), then Col4\((c, s) \neq \text{Col4}(c', s')\).
Proof. Case (a) can be simply derived by observing Table 2, since in each column no colors are repeated. For the other cases (b) and (c), that occur when the number of clusters in \( c \) doubles that in \( c' \), we distinguish two possibilities: (i) \( c = 2^p + 1 \), and (ii) \( c = 2^p \).

Case (i). When \( c = 2^p + 1 \) and \( p \geq 2 \), according to Eq. (1), cluster \( (c, s) = (2^p + 1, s) \) copies its color from \( M4[2^p - 1][4 + 1 = 4, s][4] \), while cluster \( (c' = 2^p - 1, s') \) copies its color from \( M4[2^p - 3][4 + 1 = 2, s][4] \). Thus, to check that the colors assigned to the clusters \( (c, s) \) and \( (c', s') \) are different, it is sufficient to verify that \( M4[2, i] \neq M4[4, 2i][4] \) and \( M4[2, i] \neq M4[4, 2i + 1][4] \), for \( 0 \leq i \leq 3 \).

When \( c = 3 \), it is sufficient to verify that \( M4[0, i] \neq M4[2, 2i][4] \) and \( M4[0, i] \neq M4[2, 2i + 1][4] \), for \( 0 \leq i \leq 3 \).

Case (ii). When \( c = 2^p \), conflicts may arise only if \( p \geq 2 \). Again, according to Eq. (1), cluster \( (c, s) = (2^p, s) \) copies its color from \( M4[2^p - 2][4 + 1 = 3, s][4] \), while cluster \( (c', s') \) copies its color from \( M4[2^p - 4][4 + 1 = 1, s][4] \).

Thus, the colors assigned to the clusters \( (c, s) \) and \( (c', s') \) are different because \( M4[1, i] \neq M4[3, 2i][4] \) and \( M4[1, i] \neq M4[3, 2i + 1][4] \), for \( 0 \leq i \leq 3 \). \( \blacksquare \)

As a consequence:

Corollary 2. Algorithm Col4 assigns colors to clusters satisfying the distance-two constraint.

Similarly to Corollary 1, the following result holds.

Corollary 3. For any positive integer \( x \), let \( \ell \geq 4x \). Then, \( G_\ell \) can be colored with \( 8 \) colors by means of the same rules defined by algorithm Col4(c, s).

Note that algorithm Col4(c, s) colors each cluster in a constant number of steps and it makes use of 8 colors.

4.2. Coloring for \( G_\ell \), with \( \ell \geq 7 \)

We now address the general case of a graph \( G_\ell \). First we provide a coloring algorithm ColG for the case of \( \ell > 7 \), then we tackle the particular case of \( G_7 \). In all cases the proposed solution makes use of at most 9 colors and exploits the matrix \( M4 \). From now on let \( C(c, s) \) denote the color assigned to cluster \( (c, s) \).

Algorithm ColG properly generalizes Col4 and is applicable to color \( G_\ell \) for each \( \ell > 7 \). It is composed of five phases, namely, \( A, B, C, D, \) and \( E \), as emphasized by the comments in the pseudo-code. Each of these phases colors a different portion of \( G_\ell \). Precisely, Phases \( A, B, C, \) and \( D \) color the first 4 coronas of \( G_\ell \), while Phase \( E \) all the remaining coronas \( c, \) with \( 5 \leq c \leq k \). Note that, depending on the value of \( c \) or \( \ell \), some phases might be skipped.

If \( r \) is only Phases \( C \) and \( E \) are executed to color \( G_\ell \). Precisely, when \( r = 0 \), algorithm ColG acts exactly as Col4 does. Indeed, as one can easily check in Algorithm 1, since \( r = 0 \), cluster \( (c, s) \) always copies its color from \( M4[4][c - 2][4 + 1, s][4] \).

In order to describe the algorithm when \( r > 0 \), let us introduce some notations. Let \( \sigma(c, s) \) denote the subgraph of 9 clusters \( \sigma(c, s) \) rooted at cluster \( (c, s) \) with \( 0 \leq c \leq \ell - 1 \) and extended up to corona 4. Namely, \( \sigma(c, s) = \{(1, s), (2, 2s)[2], (2, 2s + 1)[2], (3, 2s|2], (3, 2s + 1)[2], (4, 4s|4], (4, 4s + 1)[4]), (4, 4s + 2|4], (4, 4s + 3|4] \} \). When \( r > 0 \), the subgraphs \( \sigma(1, 3, j, 1) \) with \( 1 \leq j \leq r \) are colored in Phase \( D \) using up to corona 9. The remaining \( \ell - r \) subgraphs that form a graph \( G_\ell \), with \( \ell = \ell - r \), can be colored according to Col4. Precisely, we act as the subgraphs \( \sigma(1, 3, j, 1) \), with \( 1 \leq j \leq r \), were logically removed from \( G_\ell \), and thus we apply Col4 to the so remaining \( G_\ell ' \). This is done in Phases \( A, B, \) and \( C \) that rely on the matrix \( M4 \). These subgraphs reproduce the coloring generated by Col4 by simply shifting the coloring to the left of as many positions as the number of clusters logically removed along that corona. In more detail, Phase \( A \) colors \( \sigma(1, 10) \) and \( \sigma(1, 1) \), Phase \( B \) colors the subgraphs \( \sigma(1, 3, j, 1)) \), with \( 1 \leq j \leq \ell - 1 \) as they belonged to \( G_\ell \), that is as they were shifted on the left of \( r \) subgraphs.

Finally, since the coronas \( 5 \leq c \leq k \) in \( G_\ell \) have a number of clusters multiple of 4, Col4 is directly applied.

In Fig. 6, we depict with the frameboxes \( A, B, \) and \( D \) the subgraphs colored, respectively, by Phases \( A, B, \) and \( D \) in the initial portion of an arbitrary \( G_\ell \), with \( r \geq 2 \). We can state the following result:

Theorem 5. Algorithm ColG provides a distance-two coloring for any graph \( G_\ell \), \( \ell > 7 \), by means of at most 9 colors.

Proof. The above description emphasizes that Phases \( A, B, C, \) and \( E \) proceed by mimicking the behavior of Col4 on a graph \( G_\ell \), \( \ell = \ell - r \), where the subgraphs \( \sigma(1, 3, j, 1) \), \( 1 \leq j \leq r \), have been logically removed. (This can also be observed by considering the situation visualized in Fig. 6.) Hence, by Corollary 2, the distance-two constraint is satisfied by any pair of clusters colored in Phases \( A, B, C, \) and \( E \). It remains to show that Phase \( D \) preserves the constraint.

First of all, observe that Phase \( D \) colors \( r \) subgraphs \( \sigma \) that occur in \( G_\ell \) at distance greater than 2, because \( \ell > 7 \). Consequently, in verifying the distance-two constraint for color 9, it suffices to consider each subgraph \( \sigma(1, 3, j, 1) \), \( 1 \leq j \leq r \). As one can easily check in Phase \( D \) of Algorithm 1, any two clusters colored 9 are at distance greater than 2. Moreover, any two clusters at distance less that 2 in \( \sigma \) copy their colors from two clusters colored differently in \( G_\ell \) by algorithm Col4.
Algorithm 1 Algorithm ColG for the coloring of $G_\ell$ with $\ell > 7$ and $k$ coronas.

1: procedure ColG($\ell$)
2: \( r \leftarrow |A| \)
3: if $r > 0$ then $\triangleright$ Phase \( A \): lines 3–16
4: \( C(1, 0) \leftarrow M4[0, 0] \)
5: \( C(1, 1) \leftarrow M4[0, 1] \)
6: for $k = 2$ to $3$ do
7: \( C(k, d) \leftarrow M4[k - 1, d] \)
8: end for
9: end for
10: for $s = 0$ to $1$ do
11: \( C(4, 4s + d) \leftarrow M4[3, d] \)
12: end for
13: end if
14: for $j = 1$ to $r - 1$ do $\triangleright$ Phase \( B \): lines 17–28
15: \( C(1, s) \leftarrow M4[0, |s - r|] \)
16: \( C(2, 2s) \leftarrow M4[1, |2s - 2|] \)
17: \( C(3, 2s + 1) \leftarrow M4[2, |2s - 2 + 1|] \)
18: \( C(4, 4s + d) \leftarrow M4[3, d] \)
19: end for
20: for $s = 3r$ to $\ell - 1$ do $\triangleright$ Phase \( C \): lines 29–38
21: \( C(1, s) \leftarrow M4[0, |s - r|] \)
22: \( C(2, 2s) \leftarrow M4[1, |s - 2r|] \)
23: \( C(3, 3s + 1) \leftarrow M4[2, |s - 2r|] \)
24: end for
25: for $s = 6r$ to $2\ell - 1$ do $\triangleright$ Phase \( D \): lines 39–50
26: \( C(2, s) \leftarrow M4[1, |s - 2r|] \)
27: \( C(3, s) \leftarrow M4[2, |s - 2r|] \)
28: end for
29: for $s = 12r$ to $4\ell - 1$ do $\triangleright$ Phase \( E \): lines 51–55
30: \( C(4, s) \leftarrow M4[3, |s|] \)
31: end for
32: for $j = 1$ to $r$ do
33: \( s \leftarrow 3j - 1 \)
34: \( C(1, s) \leftarrow 9 \)
35: \( C(2, 2s) \leftarrow C(1, s + 2) \)
36: \( C(2, 2s + 1) \leftarrow C(1, s - 2) \)
37: \( C(3, 2s) \leftarrow C(2, 2s - 3) \)
38: \( C(3, 2s + 1) \leftarrow C(2, 2s - 2) \)
39: \( C(4, 4s) \leftarrow 9 \)
40: \( C(4, 4s + 1) \leftarrow C(4, 4s - 3) \)
41: \( C(4, 4s + 2) \leftarrow C(4, 4s - 2) \)
42: \( C(4, 4s + 3) \leftarrow 9 \)
43: end for
44: for $c = 5$ to $k$ do $\triangleright$ Phase \( A \): lines 3–16
45: \( C(c, 5) \leftarrow M4[|c - 2|a + 1, |s|] \)
46: end for
47: end for
48: end for
49: end procedure

Fig. 6. A visualization of a portion of the coloring obtained by means of algorithm ColG, relevant to the proof of Theorem 5. The dashed boxes emphasize the clusters colored by Phases $A$, $B$, and $D$ of ColG.
Thus, if the colors in $\sigma(1,3j-1)$, $1 \leq j \leq r$, violate the distance-two constraint, then the same would happen for some clusters in $G_t$. Such a violation would imply that Phases $A$, $B$, and $C$, and consequently algorithm Col4, violate the distance-two constraint. But this would contradict Corollary 2.

For example, let us focus on the first group $\sigma(1,2)$, colored by Phase $D$, i.e. the leftmost one in Fig. 6. As regards the clusters $(2,4)$ and $(2,5)$, their colors are the same than those of clusters $(1,0)$ and $(1,4)$, respectively, that are copied from $M4[1,0]$ and $M4[1,3]$, respectively. Thus, it holds $C(1,0) \neq C(1,4)$. A similar argument can be formulated for the clusters $(3,4)$, $(3,5)$, $(4,9)$, and $(4,10)$, in order to show that all six clusters get different colors and do not violate the distance-two constraint. □

Algorithm 1 does not work when $\ell = 7$ because there are not $|7/4 = 3$ clusters in corona 1 at reciprocal distance 2. However, the following lemma states the colorability of $G_7$ with 9 colors.

**Lemma 9.** There exists a coloring of the graph $G_7$ with 9 colors and satisfying the distance-two constraint.

**Proof.** We proceed by providing an explicit coloring of $G_7$. This is obtained by exploiting the matrix $M4$ to assign one among 8 given colors to each of the clusters of $G_7$, except three of them, that will get the extra color 9.

The color for a cluster $(c,s)$ is determined as follows:

(i) if $c \geq 5$, then $(c,s)$ gets the color $M4[c-2|4+1,|s|4]$;
(ii) if $c < 5$ and $0 \leq s < 2^{|\log_2 c|+1}$, then $(c,s)$ gets the color $M4[c-2|4+1,|s|4]$;
(iii) if $c < 5$ and $3 \cdot 2^{|\log_2 c|} \leq s$, then $(c,s)$ gets the color $M4[c-2|4+1,|s+2^{|\log_2 c|}|4]$;
(iv) the clusters $(1,2)$, $(4,8)$, and $(4,11)$ get the color 9;
(v) the clusters $(2,4)$, $(2,5)$, $(3,4)$, $(3,5)$, $(4,9)$, and $(4,10)$, get the colors 3, 4, 6, 7, 8, and 5, respectively.

The fact that such a coloring satisfies the distance-two constraint can be easily verified by observing that cases (i)-(iii) mimic the coloring produced by Col4. Then, the proof is obtained by proceeding in analogy to what was done in proving Theorem 5. Similarly, the colors assigned in case (v) cannot violate the distance-two constraint, otherwise the same violation would be present in clusters colored by mimicking Col4. □

From all the results of the section, the following corollary establishes an upper bound to the number of colors needed to any $G_t$.

**Corollary 4.** For any integer $\ell > 2$, $G_\ell$ can be colored with at most 9 colors, satisfying the distance-two constraint.

Once the suitable coloring has been performed among sensors, it is used to schedule communications. As mentioned in Section 3, different colors specify different communication frequency channels. This implies that adjacent clusters can perform in parallel their communications without causing collisions. However, before showing how the routing of sensory data can be performed over the virtual infrastructure, we provide a further step in the set-up of the network by electing inside each cluster one leader for each type (time-zone). In this way, we avoid redundant communications among sensors belonging to the same cluster (hence saving energy) while we ensure at least one active sensor at any time. Actually, we could schedule the repetition of the leader election procedure in order to rotate among sensors, hence prolonging the network lifespan.

5. **Leader election and routing**

In this section, we describe how the routing and the leader election can be performed in the sensor network without collisions by means of the coloring algorithms presented in the previous section.

Our routing algorithm requires that, in any cluster, there is a sensor ready to forward the message going toward the sink at any time-slot $t$. Such a sensor will be the leader of the sensors that wake up at time $t$. From now on, we assume that, at any time in any cluster, there is at least one leader awake and ready to forward the message. Specifically, during the routing process, we assume that sensors transmit during the second time-slot of their awake period, while they are listening during their first one. A message that originates at time $t$ in cluster $(c,s)$, will be transmitted by the leader of time-zone $t$ in cluster $c$ at time $t+1|t|$. Such a message will be then received and handled by the awake leader of time-zone $t+1|t|$ in the cluster destination that receives the message at time $t+1|t|$ and forward it toward the sink at time $t+2|t|$. Note that, the destination cluster is $(c-1,s)$ if $c$ is not a power of two, and cluster $(c-1,\lfloor \frac{t+1}{2}\rfloor)$ otherwise. In this way, a message originated in corona $c$ can be potentially routed in $c$ hops to the sink. To this aim, observe that a leader transmission reaches the cluster destination as well as the other adjacent clusters because, during the routing protocol, sensors broadcast with a radius equal to the corona width. Therefore, to avoid that a cluster is simultaneously reached by two different leader transmissions on the same frequency channel, two leaders that use the same frequency channel must reside in two clusters.
that are at least at distance 3. Thus, any frequency channel assignment (or, coloring) suitable for routing without collisions must satisfy the distance-two constraint discussed earlier.

It is worthy to note that a weaker constraint on the distance of leaders transmitting on the same frequency channel is sufficient for the leader election protocol. Indeed, as it will be explained below, during such a protocol, a message that originates in cluster \( c, s \) has for destination the cluster itself. Thus, to avoid collisions, it is sufficient that two leaders that transmit on the same frequency channel reside in two clusters at distance 2. Hence, any coloring suitable for our routing algorithm is also suitable for the leader election.

A brief description of the routing and leader election protocols follows. Once the coloring of the virtual infrastructure has been performed, each sensor residing in a specific cluster is aware of its color. We consider one different frequency channel for each used color. Hence, each sensor will be aware of the frequency channel it has to use for transmission tasks. Our first goal is to elect, inside each cluster, one leader for each time-zone \( 0 \leq x \leq L - 1 \). To this aim, we make use of the well-known uniform leader election for radio networks protocol presented in [9]. In particular, we can consider the so-called Scenario 2 in which an upper bound to the number of sensors competing for the leader election inside each cluster and for each time-zone is known. In fact, by exploiting the arguments presented in [6,11] such an upper bound is \( u \leq \frac{2}{3}A_1 \Lambda = \frac{4r}{L} \Lambda \), where \( \Lambda \) is an estimation of the density of sensors related to one specific time-zone. From [9], the sensors require on average \( \ln \ln u + o (\ln \ln u) \) transmissions. In practice, the protocol works by assigning a probability of transmission to each sensor. A sensor is elected as leader when it is the only one transmitting during the time-slot. If more than one sensor transmit or no one transmits, then the probability to transmit at the subsequent appropriate time-slot decreases or increases, respectively.

In our setting, we perform \( L \) leader elections, one for each time-zone, distributed over \( O (\ln \ln u) \) subsequent sleep-awake periods. Only the sensors with the same time-zone are involved in one election. For each time-zone, each sensor performs one step of its leader election during every period. At the \( i \)-th time-slot of the \( j \)-th awake period, the sensors of time-zone \( i \) perform the \( j \)-th step of their leader election. Each sensor transmits only during the first time-slot of its awake period. In doing so, we obtain the required leader election for all time-zones and all clusters. In fact, the protocol is performed in parallel in all clusters, each cluster transmitting on the frequency channel assigned to it by the coloring protocol. Finally, the routing earlier described can start.

Since in each cluster we have elected one leader for each time-zone, there will always be one leader, in the destination cluster, awake and ready to forward the message. Moreover, since each communication is performed according to the frequency channels that satisfy the distance-two constraint, in each time-slot the message will decrease by one hop its distance from the sink. Thus, using a multi-hop technique, a message originated in corona \( c \) reaches the sink in \( c \) time-slots.

6. Conclusions

We investigated a virtual organization of a sink-centric subnetwork in a dense DC-WSN, that imposes a generalized coordinate system. Such a system provides a coarse-grained location to the sensors and allows a naive geographic routing algorithm. All the sensors that acquire the same coordinates form a cluster. For routing purposes, we assume that the sensors can transmit using different frequency channels. Following a multi-hop approach along the cluster–sink path, sensors in the outer coronas of the virtual infrastructure transmit their messages to the sink through intermediate coronas. The message stream can continuously proceed if there is, at any time, a relaying sensor awake and ready to transmit and no collisions arise on the frequency channels. To avoid collisions, a frequency channel assignment (or, coloring) that satisfies a distance-two constraint is provided for the graph \( G_r \) that represents the virtual infrastructure that has \( \ell \) clusters in corona 1. Optimal coloring algorithms for \( G_r \) with \( \ell = 3 - 2^x \), \( x \geq 0 \), \( \ell = 4 \), and \( \ell = 5 \) have been provided. In particular, \( OPT_3 \) is fully distributed and requires constant number of steps. Moreover, a generic coloring framework for the remaining values of \( \ell > 6 \) has been designed. Although it makes use of at most 9 colors, which is not optimal in general, it requires only constant number of steps to color a cluster, and this is a desirable property in distributed environments. Furthermore, to avoid redundant messages during the routing protocol, we elect leaders in each cluster that act as relaying sensors. To this aim, we adapt a known uniform leader election protocol to our scenario. In the future, we intend to implement our routing algorithm in both simulated and real settings. Moreover, the study of optimal colorings for adjacency graphs \( G_\ell \) with an arbitrary number \( \ell \) of clusters in corona 1 is still an interesting open problem for some values of \( \ell \).

References


