

Distributed Time Slots Assignment Protocol in Dynamic Networks

Hicham Lakhlef, Ghada Jaber, Abdelmadjid Bouabdallah, Fabio D'andreagiovanni, Ahmed Lounis

▶ **To cite this version:**

Hicham Lakhlef, Ghada Jaber, Abdelmadjid Bouabdallah, Fabio D'andreagiovanni, Ahmed Lounis. Distributed Time Slots Assignment Protocol in Dynamic Networks. IEEE Symposium on Computers and Communications (ISCC 2020), Jul 2020, Rennes, France. pp.1-6, 10.1109/ISCC50000.2020.9219697 . hal-03094834

HAL Id: hal-03094834
https://hal.science/hal-03094834

Submitted on 4 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed Time Slots Assignment Protocol in Dynamic Networks

Hicham Lakhlef † Ghada Jaber † Abdelmadjid Bouabdallah † Fabio D'Andreagiovanni*,† Ahmed Lounis †

† Sorbonne Universités, Université de Technologie de Compiègne, CNRS,
Heudiasyc UMR 7253, CS 60319 ; 60203 Compigne Cedex
* French National Centre for Scientific Research (CNRS), France

hlakhlef@utc.fr gjaber@utc.fr bouabdal@utc.fr d.andreagiovanni@hds.utc.fr alounis@utc.fr

***Abstract*—This paper addresses the problem of communication in resource-limited broadcast/receive wireless networks. In large scale and resource-limited wireless networks, as the Internet of Things (IoT), a massive amount of data is becoming increasingly available, and consequently implementing protocols achieving error-free communication channels presents an important challenge. Indeed, in this new kind of network, the prevention of message conflicts and message collisions is a crucial issue. In terms of graph theory, solving this issue amounts to solve the distance-2 coloring problem on the network. This paper presents a first study on dynamic management in distance-2 coloring in resource-limited wireless networks. We propose a distributed distance-2 coloring in a dynamic network where a new node can join the network. Thanks to the proposed protocol, we assign a time slot to the new node without re-running the whole algorithm of time slot assigning. Our protocol is time-efficient and uses only local information with a high probability.**

Keywords: Wireless network, Broadcast/receive, Collision, Conflict, Distance-2 graph coloring, Dynamic Networks, Synchronous system, Time slot assignment

I. Introduction

In recent years, there have been major advances in the development of objects equipped with microcontrollers and radio frequency transceivers for digital communications. This gave birth to a new paradigm of networks called *Internet of Things (IoT)*, which are constituted by a wide variety of communicating things (e.g., sensors, mobile phones, car, domestic appliances) that have the ability to collect and exchange data of the most various nature (e.g., temperature, traffic conditions, pollution) for processing and decision-making. The IoT will soon occupy a place comparable to that of electricity in our everyday life and will play a fundamental role in socio-economic development. However, large IoT scale networks, usually deployed in open environments, are vulnerable to failures resulting from collisions on communication channels or from component failures [1], [2], [3].

Broadcast communication is advantageous when a node needs to find information without knowing exactly what another node can provide, or when a node wants to provide information to a large number of nodes in due time. Broadcasting is the most general and intensive method of communication in the sense that many messages are required and many network devices are involved. Many fundamental problems of distributed computing such as fault tolerance, consensus, the election of leaders, routing, are based on broadcast/receive operations. With the recent advances in miniaturization, networks have become large-scale and with limited and heterogeneous resources which creates several communication problems. Indeed, in large-scale restricted resource networks, interference in the communication channels is one of the most important problems [4], [5].

In a synchronous system, the nodes execute a sequence of rounds, where a round is a bounded time slot where any message sent at the beginning of a slot is received at most by the end of the same time slot by its receiver [6]. A broadcast/receive communication system provides nodes with two operations: a broadcast that allows sending a message to all its (one hop) neighbors, and a receive which allows receiving messages. This paper considers a synchronous system with broadcast/receive communication. In such a system:

- if two neighbor nodes of a node p_i invoke the operation broadcast() during the same time slot (round), a message *collision* occurs.
- if a node p_i and one of its neighbors invoke broadcast() during the same time slot (round), a message *conflict* occurs.

In case of collision or conflict in a channel C, the status of C is $S(C) =$ *COLLISION*. Otherwise, the status is *SINGLE* (only one broadcast on the channel) or *NULL* (no broadcast). As already mentioned, this paper considers this broadcast/receive communication model.

In terms of graph theory, solving this issue of communication amounts to solve the distance-2 coloring problem on the network. A large number of works [11-20] have proposed distributed vertex coloring algorithms applicable to wireless networks. Unfortunately, so far these proposed coloring protocols are not robust to both conflicts and collisions on the level of communication channels. Additionally, some of them use centralized algorithms. Furthermore, they are not all suited for resource-limited wireless nodes.

Therefore, our goal and original contribution of this work is to provide a scalable communication for large scale and resource-limited communicating things. That is, a collision and conflict-free communication whatever the number of nodes in the network and whatever the number of neighbors a node can have. Towards this objective, we presented several research results. In [7], [8], we proposed a distance-2 coloring algorithm that allows a safe communication for tree networks. In [10], we proposed a parallel multi-coloring algorithm for tree networks and in [9], we proposed a distributed distance-2 coloring algorithm that allows a safe communication for general networks. However, this last is not optimal in terms of communication. All these papers do not deal with the problem of dynamic (nodes joining) in limited resources wireless using distributed protocols. Our paper is the first to provide a solution to such a problem.

The remainder of this paper is organized as follows. In Section II, we present the system model and some definitions. In Section III, we present the problem formulation and some mathematical tools used to prove and analyze the proposed solution. This section is followed by Section IV, where we present the proposed protocol. In section V we present the protocol analysis and its computational. Finally, Section VII concludes our paper.

II. Synchronous Broadcast/Receive Model

In this section, we present the timing model and the communications operations.

A. System model and definitions

a) Nodes, initial knowledge, and the communication graph: The system model consists of n sequential nodes denoted p_1, ..., p_n, connected by an arbitrary communication graph.

Each node p_i has an identity id_i, which is known only by itself and its neighbors (nodes at distance 1 from it). The constant $neighbors_i$ is a local set, known only by p_i, including the identities of its neighbors (and only them). As noticed in the Introduction, so that a node p_i not to confuse its neighbors, it is assumed that each two nodes at distance less than or equal to 2 have distinct identities. Hence, any two nodes at distance greater than 2 may have the same identity. When computing complexities, $\log x$ refers to the logarithm in base 2 and $\ln x$ refers to the natural logarithm.

Let Δ_i denote the degree of a node p_i (i.e. $|neighbors_i|$) and Δ denote the maximal degree of the graph $(\max\{\Delta_1, \cdots, \Delta_n\})$.

When considering a node p_i, $1 \leq i \leq n$, the integer i is called its index. Indexes are not known by the nodes. They are only a notation convenience used as a subscript to distinguish nodes and their local variables.

b) Timing model: We assume that processing duration is equal to 0. This is justified by the following observations: (a) the duration of the local computations of a node are negligible concerning message transfer delays and (b) the processing duration of a message maybe considered as a part of its transfer delay.

Communication is synchronous in the sense that there is an upper bound D on message transfer delays and this bound is known by all the nodes (global knowledge). From an algorithm design point of view, we consider that there is a global clock, denoted $CLOCK$, which is increased by 1, after each period of D physical time units. Each value of $CLOCK$ defines what is usually called a *time slot* or a *round*.

c) Communication operations: The nodes are provided with two operations denoted broadcast() and receive(). A node p_i invokes broadcast TAG(m) to send the message m, whose type is TAG, to all its neighbors. It is assumed that a node invokes broadcast() only at the beginning of a time slot. When a message TAG(m) arrives at a node p_i, this node is immediately warned of it, which triggers the execution of operation receive() to obtain the message. Hence, a message is always received and processed during the time slot –round– in which it was broadcast.

From a linguistic point of view, we use the two following **when** notations when writing algorithms, where predicate is a predicate involving $CLOCK$ and possibly local variables of the concerned node:

when TAG(m) **is received do** processing of the message.
when predicate **do** code entailing one broadcast() invocation.

III. The Distance-2 Coloring Problem

This problem is a well-known graph coloring problem called *distance-2* coloring. The aim is to design distributed algorithms associating colors with each node (which will define the time-slots during which it will be allowed to broadcast) such that the following properties are satisfied.

a) Communication operations : Traditional wired round-based synchronous systems assume a dedicated communication medium for each pair of nodes (i.e., this medium is not accessible to the other nodes). Hence, in these systems, a node p_i obeys the following sequential pattern during each round: (a) first p_i sends a message to all or a subset of its neighbors, (b) then p_i receives the messages sent to it by its neighbors during the current round, and (c) finally executes a local computation which depends on its local state at the beginning of the round and the messages it has received during the current round.

b) Using the colors to define the time slots: The colors obtained by the nodes are used as follows, where $colors_i$ are the colors obtained by node p_i. The time slots (rounds) during which p_i is allowed to broadcast a message to its neighbors correspond to the values of $CLOCK$ such that $\exists c \in colors_i$ with $\big((CLOCK \bmod (K+1)\big) = c)$. As we will see, these time slots are different from the time slots used during the distributed distance-2 algorithms which are presented below. It follows that these algorithms must provide each node with the (initially unknown) value of K.

IV. Problem formulation and tools

We have a distance-2 colored network where each node knows the colors of its neighbors [6-7], the degree of the networks is Δ. The colors already used in the network are $\{0, 1, 2, ..., \Delta - 1\}$. For each node i in the network, let $C_i = \{c_1, c_2, ..., c_m\}$ be the set of channels that it uses to communicate with its m neighbors.
If a new node N wants to join the network, how we assign using a distributed solution a correct color to N under the following constraints:

- The nodes do not know at which time a new node will join the network
- If collision or conflict situations arise the message is lost
- The node N will have a color which is not used by its future neighbors and the neighbors of its future neighbors

Due to space constraints, we assume in this paper that the new node knows Δ and when the new node arrives its future neighbors do not broadcast any message (except the messages of our protocol in Section 4) until the new node informs them about the obtainment of its color. These assumptions will be lifted in the journal version.

The following tools are used in the protocol analysis:
The De Morgan law for a set of events $E_1, E_2, ..., E_m$ provides that:

$$\overline{\bigcap_{i=1}^{m} E_i} = \bigcup_{i=1}^{m} \overline{E_i} \tag{1}$$

To analyze the tail of a nonnegative random variables X, we use the following bounds of Chernoff [20]:

$$\Pr\left[X > (1+\delta)c \cdot E\left[X\right]\right] < \left(\frac{e^{\delta}}{(1+\delta)^{(1+\delta)}}\right) \tag{2}$$

$$\Pr[X > (1+\epsilon)E[X]] < e^{-\frac{\epsilon^2}{3}E(x)} \tag{3}$$

And the Markov's inequality, if X is a nonnegative random variable and $a > 0$, then

$$\Pr\left[X \geq a \cdot E[x] \leq \frac{1}{a}\right] \tag{4}$$

It is well known that, for every constant c

$$\lim_{x\to\infty} f(1 + \frac{c}{x})^x = e^c \tag{5}$$

V. Solution

Our solution consists of two phases. In the first phase, the new informs its future neighbors about its presence. In the second phase, the future neighbors exchange information with the new node to allow it choosing a correct color.

A. Phase 1

During this phase the new node N attracts the attention of its future neighbors about its presence in the network. Let $neighbors_N$ be the set of future neighbors of the new node. The detail of this phase are spelled out in following protocol:

```
(01) t ← 0
(02) repeat
(03)        broadcast HELLO (id_i, t)
(04)        t ← t + 1
(05) while ((∃C ∈ C_N) ∧ (S(C) = COLLISION ) ∧(t < Δ))
```

Algorithm 1: Hello protocol (code for N)

The node N broadcasts a *HELLO* message on all channels in the list C_N. It repeats this operation while there was a collision in the previous broadcast. This message contains its *ID* and the current time slot t. The parameter t will be used by nodes in $neighbors_N$ to start the execution of phase 2 at the same time. Then N waits for the message responses of its future neighbors where they broadcast their colors already used. To correctly color N, these colors must not be in $color_N$.

B. Phase 2

This phase consists of two stages. In the first stage, the future neighbors of N try concurrently to send to N the colors already used by them. This set of given by:

$$\bigcup_{h \in neighbors_N} color_h \tag{6}$$

In the second stage, after the reception of all used colors from its future neighbors, N chooses a new color that allows it to is to broadcast correctly in future communication. The details of these stages explained in the following:

1) Stage 1: At the reception of the message $HELLO$, the node do not broadcast any message expect the message in protocol *Sending used colors* until it receives a message adding_completed from N. In this phase, the future neighbors of N try to send to N theirs list of colors using the protocol Sending_used_colors. Using the parameter t and Δ the nodes in $neighbors_N$ call this protocol at the same time. More precisely, let θ be the number of time slots since node i received t in the first $HELLO$ message received. Each node checks if $t + \theta = \Delta$ to start the protocol Sending_used_colors. The details of this protocol are described in Algorithm 2

```
Procedure color(t) for node i

(01) if (CLOCK mod Δ = color_i)
(02)    then with a probability equal to 1/2^t broadcast COLOR (id_i)
(03) end if
(04) return(S(c))
```

Procedure color(t) for node i

```
Protocol for node i
(01)  L ← 1
(02)  while (L ≤ Δ)
(03)     j ← 0
(04)     repeat
(05)              r ← COLOR (2^j)
(06)              ← j + 1
(07)     until (r = NULL)
(08)     φ ← Δ
(09)     l ← 0
(10)     u ← 2^i
(11)     while (l + 1 < uφ) do
(12)              m ← ⌈(l+u)/2⌉
(13)              r ← COLOR (m)
(14)              if (r = SIGNLE)
(15)                 then end of the protocol for node p
(16)              endif
(17)              if (r = NULL)
(18)                 then u ← m
(19)                 else
(20)                      l ← m
(21)              endif
(22)     endwhile
(23)     while (r ≠ SINGLE)
(24)              r ← COLOR (m)
(25)              if (r = NULL)
(26)                 then u ← u^2
(27)                 else
(28)                      u ← ⌈√u⌉
(29)     endwhile
(30)     L ← L + 1
(31)  endwhile
```

Algorithm 2: Sending_used_colors (code for nodes in $neighbors_N$)

2) Stage 2: After the reception of all messages from the nodes in $neighbors_N$, the node N can choose a color that is not used by its neighbors at distance 1 and its neighbors at distance 2. More precisely, it chooses a color c where $\nexists\, n \in neighbors_N$ with $c \in colors_n$. Once this color is chosen, N uses the time slot corresponding to its color to broadcast a message adding_completed to its neighbors.

VI. Protocol Analysis

Lemma 1. Phase 1 takes at most Δ times slots. And at the end of Phase 1 all nodes in $neighbors_N$ have received at least one HELLO message correctly.

Proof

Given that all nodes in $neighbors_N$ can have different colors. Therefore, in this case, there is no pair of nodes that can broadcast or receive any message at the same time slot. As every node in $neighbors_N$ does not broadcast any message to the node N (as assumed in Section 3), a message $\mathsf{HELLO}(\mathsf{id_N}, \mathsf{t})$ that was sent in the time slot t corresponding to the color of a node $neighbors_N$ is received correctly. Therefore in the worst case (all nodes in $neighbors_N$ have different colors), in Δ rounds there were Δ messages that were received correctly.

$\square_{Lemma\ 1}$

Let us now take, without loss of generality (w.l.o.g), one execution of the protocol Sending_used_colors (i.e. $L = 1$). We obtain for this case the following results:

Lemma 2. The second loop (**repeat** loop) of the protocol Sending_used_colors terminates in $\log\log(\Delta^3) + \log\log\log n$ time slots with a probability of at least $1 - \frac{1}{\Delta^2 \log n}$.

Proof

Let us assume w.l.o.g, that there is a time slot R where the return $r = NULL$ in line 13 of Algorithm 2. This means all calls to the procedure color(t), $0 \leq t < R$ returned $SINGLE$ or $COLLISION$.

Let us assume:

$$R = \log\log(\Delta^3 \log n) \tag{7}$$

Let X be the random variable that denotes the number of nodes that transmitted in the call color(R) in line 02 of Algorithm 2 of the procedure color(t). It is clear that in this round a node broadcasts with a probability equal to $\frac{1}{2^{2^R}}$. As all nodes broadcast with the same probability and we have at most Δ nodes trying to broadcast, X is with Binomial distribution $X \sim Bin(\Delta, \frac{1}{2^{2^R}})$. Therefore the expected value if X is:

$$E[X] = \Delta \frac{1}{2^{2^R}} < \frac{\Delta}{2^{2^R}} \leq \frac{\Delta}{\Delta^3 \log n} = \frac{1}{\Delta^2 \log n} \tag{8}$$

The probability that the return is $NULL$ in round R is

$$\Pr[X = 0] = 1 - \Pr[X \geq 1] \tag{9}$$

We bound now this probability using the expected value and Markov inequality presented in (4):

$$\begin{aligned} \Pr[X = 0] &= 1 - \Pr[X \geq 1] \\ &< 1 - \Pr[X \geq E[X]\Delta^2 \log n] \\ &< 1 - \frac{1}{\Delta^2 \log n} \end{aligned} \tag{10}$$

These completes the proof. $\square_{Lemma\ 2}$

Lemma 3. With a probability equal to $1 - \frac{1}{\Delta^2 \log n}$, the third loop terminates in $\Delta \log\log(\Delta^3 \log n)$ time slots.

Proof

Let us assume, w.l.o.g, that $\phi = 1$ in line 8. Then, the reader can see that this loop is a variant of a binary search tree. Let us consider a binary search tree with $\log m$ divisions if in each division we make a broadcast then the number of time slots is $\log\log(\Delta^3 \log n)$. Let us observe that the stop condition in line 11 is set to $\Delta \cdot u$, consequently, there is Δ execution of the binary search tree. Therefore, the number of time slots is $\Delta \log\log(\Delta^3 \log n)$. These and the results of lemma 2 complete the proof.

$\square_{Lemma\ 3}$

Lemma 4. With probability exceeding $\frac{5}{4\Delta^2 \log n}$ when the third loop terminates, the last value of u satisfies the double inequality

$$\frac{\Delta^4}{(\ln(R16\Delta^2 \log n))^2} \leq 2^{2u} \leq 16R^2\Delta^4(\log n)^2 \tag{11}$$

Proof

Let $E1$ be the event that when color(t) is called in line 12, then $\Delta \leq \frac{2^m}{4R\Delta^2 \log n}$ and yet the status of the channel is *NULL*. And let $E2$ be the event that when color(t) is called in line 12 $\Delta \geq 2^m \ln(4R\Delta^2 \log n)$ and yet the status of the channel is collision.

Let Z be the random variable that denotes the number nodes broadcasting in the call color(m). If $\Delta \leq \frac{2^m}{4R\Delta^2 \log n}$ then $E[Z] = \frac{\Delta}{2^m} \leq \frac{1}{4\Delta^2 log}$. By using Markov inequality, we obtain:

$$\Pr[Z > 1] \leq \Pr[Y > E[Z]4R\Delta \log n] < \frac{1}{4R\Delta^2 \log n} \tag{12}$$

If $\Delta \geq 2^m \ln(4R\Delta^2 \log n)$, the status of the channel is *NULL* with a probability at most:

$$\begin{aligned} \Pr[Z = 0] &= (1 - \frac{1}{2^m})^{\Delta} \\ &< e^{-\frac{\Delta}{2^m}} \\ &< e^{-\ln(4R\Delta^2 \log n)} = \frac{1}{4R\Delta^2 \log n} \end{aligned} \tag{13}$$

As we make R calls at most with a probability equal to $1 - \frac{1}{\Delta^2 \log n}$, from equations (12) and (13) it is clear that with a probability $1 - \frac{5}{4\Delta^2 \log n}$ at the end of the second loop the following double inequality is satisfied:

$$\frac{\Delta^2}{(\ln(16R\Delta^2 \log n))} \leq 2^m \leq 4R\Delta^2(\log n) \tag{14}$$

These completes the proof. $\square_{Lemma\ 4}$

Lemma 5. If (14) is satisfied at the end of the third loop in *Sending_used_colors* algorithm, then with a probability of at least $(1 - \frac{1}{\Delta^2 \log n})$, the fourth loop in *Sending_used_colors* algorithm terminates in at most $(e^{\log \Delta} \ln(\Delta^2 \log n) + \log \Delta \ln n)$ time slots.

Proof

Let p be the smaller integer satisfying:

$$2^{p-1} \leq \Delta \leq 2^p \tag{15}$$

Let E_3 be the event that when $color(m)$ is called in line (23) the following inequality is satisfied:

$$p - log(\Delta) \leq m \leq p + log(\Delta) \tag{16}$$

Then the return r of $color(m)$ is $SINGLE$ with a probability equal to:

$$\begin{aligned} P[r = SINGLE|E_3] &= \binom{\Delta}{1} \frac{1}{2^u}(1 - \frac{1}{2^u})^{\Delta - 1} > \frac{\Delta}{2^u} e^{\frac{-\Delta}{2^u}} \\ &> \frac{2^{u + \log \Delta}}{2^u} e^{\frac{-2^{u+\log \Delta}}{2^u}} \\ &> \frac{\Delta}{e^{\log \Delta}} \end{aligned} \tag{17}$$

Therefore if the event E_3 is satisfied $\frac{e^{\log \Delta}}{\Delta} \ln(\Delta^2 \log n)$ times the message is broadcast correctly with a probability:

$$\begin{aligned} Pr &= 1 - (1 - \frac{\Delta}{e^{\log \Delta}})^{\frac{e^{\log \Delta}}{\Delta} \ln(\Delta^2 \log n)} \\ &= (1 - \frac{\ln(\Delta^2 \log n)}{\frac{e^{\log \Delta}}{\Delta} \ln(\Delta^2 \log n)})^{\frac{e^{\log \Delta}}{\Delta} \ln(\Delta^2 \log n)} \\ &> 1 - e^{-\ln(\Delta^2 \log n)} \\ &> 1 - \frac{1}{\Delta^2 \log n} \end{aligned} \tag{18}$$

In what follows, we give a born on the number broadcasts to satisfy E_3 $\frac{e^{\log \Delta}}{\Delta} \ln(\Delta^2 \log n)$ times with a high probability.

Let E_4 be the event that when color(m) is called $m \geq p + \log \Delta$ and the status of the channel is collision. Let E_5 be the event that when color(m) is called $m \geq p + \log \Delta$ and the status of the channel is null. Let E_6 be the event that when color (m) is called $m \leq p - \log \Delta$ and the status of the channel is null. Let E_7 be the event that when color(m) is called $m \leq p - \log \Delta$ and the status of the channel is collision.

Let $V(E_x)$, $3 \leq x \leq 7$ be the random variable that counts the number of times the event E_x is satisfied. Then, it is clear that:

$$\Pr[E_3] = \overline{\Pr[E_4] \cap \Pr[E_5] \cap \Pr[E_6] \cap \Pr[E_7]} \tag{19}$$

We have

$$\begin{aligned} \Pr[E_4] &= \Pr[Z > 1 | m \geq p + \log \Delta] \\ &< \Pr[Z > \frac{2^m}{2^p} E[Z]] \\ &< \Pr[Z > \Delta E[Z]] \\ &< \frac{1}{\Delta + 1} \end{aligned} \tag{20}$$

And we have

$$\begin{aligned} \Pr[E_6] &= (1 - \frac{1}{2^m})^{\Delta} \\ &< e^{-\frac{\Delta}{2^m}} \\ &< e^{-\frac{2^{p-1}}{2^m}} \\ &< e^{-\frac{\Delta}{2}} \\ &< \frac{1}{\Delta + 1} \end{aligned} \tag{21}$$

Let suppose that color(m) in the fourth loop is executed $t = (e^{\log \Delta} \ln(\Delta^2 \log n) + \log \Delta \ln n)$ times.

From (11) we have:

$$\frac{\Delta^2}{(\ln(16R\Delta\log n))} \leq 2^m \leq 4R\Delta^2(\log n) \tag{22}$$

$$\begin{aligned} m &\leq \log(4R\Delta^2(\log n)) \\ &\leq p + \log\Delta + 2 + \log\log\log(\Delta^3\log n) + \log\log n \end{aligned} \tag{23}$$

$$\begin{aligned} m &\geq \log\frac{\Delta^2}{(\ln(16R\Delta\log n))} \\ &\geq p + \log\Delta - \log(\log(16\Delta\log n)) - 1 \\ &\geq p + \log\Delta - \log\log\log n - \log\log\Delta - \\ &\log\log\log\log(\Delta^3\log n) - 3 \end{aligned} \tag{24}$$

From (23) and (24) it is clear that $|m-p| < \log\Delta + \log\log n + 2\log\log\log n + 3$. Then, we have:

$$V(E_5)+V(E_7) < V(E_4)+V(E_6)+\log\log n+2\log\log\log n+3 \tag{25}$$

Since color(m) is executed t times:

$$E[V(E_4)+V(E_6)] = \frac{(e^{\log\Delta}\ln(\Delta^2\log n)+\log\Delta\ln n)}{\Delta+1} \tag{26}$$

Thus, the probability that there are $e^{\log\Delta}\log n + \log\Delta\ln n$ satisfied events of E_4 and E_6 is:

$$\begin{aligned} PS &= \Pr[V(E_4)+V(E_6) > \frac{e^{\log\Delta}}{\Delta}\ln(\Delta^2\log n)+\log\Delta\ln n] \\ &< \Pr[V(E_4)+V(E_6) > (1+\frac{1}{\Delta})E[V(E_4)+V(E_5)]] \\ &< e^{-\frac{1}{\Delta^2\cdot 3}E[V(E_4)+V(E_5)]} \\ &< e^{-\frac{1}{\Delta^2\cdot 3}\frac{(e^{\log\Delta}\ln(\Delta^2\log n)+\log\Delta\ln n)}{\Delta+1}} \\ &< \frac{1}{\Delta^2\log n} \end{aligned} \tag{27}$$

Let us suppose that $V(E_4)+V(E_5) < \frac{e^{\log\Delta}}{\Delta}\ln(\Delta^2\log n)+\log\Delta\ln n$ is satisfied. Then the flowing result completes the proof:

$$\begin{aligned} V(E_3) &= t - (V(E_4)+V(E_5)+V(E_6)+V(E_7)) \\ V(E_3) &> t - (2V(E_4)+2V(E_6)) \\ &> (e^{\log\Delta}\ln(\Delta^2\log n)+\log\Delta\ln n) - \\ &(2(V(E_4)+V(E_6))+\log\log n+2\log\log\log n+3) \\ &> \frac{e^{\log\Delta}}{\Delta}\ln(\Delta^2\log n) \end{aligned} \tag{28}$$

These complete the proof. $\square_{Lemma\ 5}$

Theorem 1. From lemma 1, lemma 2, lemma 5, for $L = 1$ with a probability of at least $(1-\frac{9}{4\Delta^2\log n})$, the protocol terminates in $\Delta + \log\log(\Delta^3\log n) + (e^{\log\Delta}\ln(\Delta^2\log n) + \log\Delta\ln n) = \Delta + O(\log\Delta\ln n)$ times slot.

Theorem 2. The protocol Sending_used_colors is executed Δ times. And the new node will be added correctly if all nodes in $neighbors_N$ succeed to broadcast the message color($\mathsf{id_i}$, color$_\mathsf{i}$) of line 2 of Algorithm 2 in procedure color(t) to the new node. Therefore, as $|neighbors_N| \leq \Delta$, with a probability of more than $(1-\frac{9}{4\Delta\log n})$, our protocol termites in $\Delta + \Delta O(\log\Delta\ln n)$ times slot.

VII. Conclusion

In this paper, we presented a first study on dynamic management in distributed time slots assignment based on distance-2 coloring. Given the fact that in a dynamic network the new nodes may arrive randomly and unpredictably in each time slot, during these time slots future neighbors of new nodes can be busy receiving or sending messages, creating a situation of interference in communication channels. Therefore, it is impossible to define deterministic protocols to manage the dynamic in distance-2 colored networks to color new nodes. In this work, we proposed a new randomized based protocol to allow a new node to join the network and obtain a communication slot correctly and efficiently. In the journal version, we will lift the assumptions of Section 3 stating that when the new node arrives its future neighbors do not broadcast any message until the new node obtains its color. We will also make experimental studies to evaluate the performance of our solution. This filed of research will be followed by several research to deal with all aspects of dynamic management in distance-2 coloring and muti-coloring protocols to allow a scalable communication whatever the network characteristics.

Acknowledgment

This work is supported by the French CNRS STFOC project (Security and Fault Tolerance in IoT) and the Labex MS2T.

References

[1] L. Atzori, A. Iera, G.o Morabito: "The Internet of Things: A survey". *Computer Networks*, Volume: 54, Issue: 15, pp. 2787-2805, 2010

[2] M. Vecchio, R. Giareda, F. Marcelloni: "Adaptive Lossless Entropy Compressors for Tiny IoT Devices". *IEEE Transactions on Wireless Communications*, Volume: 13, Issue: 2 pp: 1088-1100, 1000 2014

[3] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert, M. Mazura, M. Harrison, M. Eisenhauer, P. Doody: "Internet of Things Strategic Research Agenda", Chapter 2 in *Internet of Things Global Technological and Societal Trends*, River Publishers, ISBN 978-87-92329-67-7, 2011

[4] R. Talak, S. Karaman, E. Modiano, Capacity and delay scaling for broadcast transmission in highly mobile wireless networks, *IEEE Transactions on Mobile Computing*, 10.1109/TMC.2019.2923733, 2019

[5] K. Nakano, S. Olariu: Randomized leader election protocols in radio networks with no collision detection. In: *International Symposium on Algorithms and Computation*. pp. 362373, 2000

[6] M. Raynal, " Fault-Tolerant Message-passing Distributed Systems": An Algorithmic Approach. Springer, 459 pages, 2018 (ISBN: 978-3-319-94140-0).

[7] D. Frey, H. Lakhlef, and M. Raynal, Optimal collision/conflict-free distance-2 coloring in wireless broadcast/receive synchronous tree networks. 45*th Annual Conference on Parallel Processing* (ICPP16), pp. 350-359, 2016.

[8] H. Lakhlef, J. Bourgeois, S. Harous, J. Myoupo: "Collision-Free Routing Protocol in Multi-hop Wireless Sensor Networks". In CIT 2015, The 15th IEEE Int. *Conf. on Computer and Information Technology*, Liverpool, UK, pages 9299, October 2015

[9] A. Bouabdallah, H. Lakhlef, M. Raynal, Franois Taani: Providing Collision-Free and Conflict-Free Communication in General Synchronous Broadcast/Receive Networks. IEEE AINA, *The 31st IEEE International Conference on. Advanced Information Networking and Applications* : 399-406, 2017

[10] H. Lakhlef, M. Raynal, F. Taani: Vertex Coloring with Communication Constraints in Synchronous Broadcast Networks, *IEEE Transactions on Parallel and Distributed Systems*, 10.1109/TPDS.2018.2889688

[11] L. Barenboim, M. Elkin, and F. Kuhn, Distributed (Delta+1)-coloring in linear (in Delta) time. *SIAM Journal of Computing*, 43(1):72-95 (2014)

[12] J. Blair and F. Manne, An efficient self-stabilizing distance-2 coloring algorithm. *Proc. 16th Colloquium on Structural Information and Communication Complexity* (SIROCCO10), pp. 237-251 (2009)

[13] O. Chipara, C. Lu, J. Stankovic, and G.-C. Roman, Dynamic conflict-free transmission scheduling for sensor network queries. *IEEE Transactions on Mobile Computing*, 10(5):734-748 (2011)

[14] M. Gairing, W. Goddard, S. T. Hedetniemi, P. Kristiansen, and A. A. McRae, Distance-two information in self-stabilizing algorithms. P*arallel Processing Letters*, 14(03-04):387398, (2004).

[15] T. Herman, S. Tixeuil, A distributed TDMA slot assignment algorithm for wireless sensor networks. Proc. *Intl Workshop on Algorithmic Aspects of Wireless Sensor Networks*, pp. 45-58 (2004)

[16] I. Jemili, D. Ghrab, A. Belghith, B. Derbel, and A. Dhraief, Collision aware coloring algorithm for wireless sensor networks. In Proc. *9th Intl Wireless Communication and Mobile Computing Conference* (IWCMC13), pp 1546-1553 (2013).

[17] N.A. Lynch, "Distributed algorithms". M*organ Kaufmann*, 872 pages (1996)

[18] S. Mahfoudh, G. Chalhoub, P. Minet, M. Misson, and I. Amdouni, "Node coloring and color conflict detection in wireless sensor networks". *Future Internet*, 2(4):469, 2010.

[19] D. Peleg, Distributed computing, a locally sensitive approach. *SIAM Monographs on Discrete Mathematics and Applications*, 343 pages, ISBN 0-89871-464-8 (2000)

[20] I. Rhee, A.Warrier, J. Min, and L. Xu, Drand: Distributed randomized TDMA scheduling for wireless ad-hoc networks. *IEEE Transactions on Mobile Computing*, V.: 8, pp:1384-1396, 2009

[21] R. Motwani and P. Raghavan, Randomized Algorithms. Cambridge Univ. Press, 1995.