



**HAL**  
open science

## About The Power of Anonymous Radio Networks

Jérémie Chalopin, Yves Métivier, Thomas Morsellino

► **To cite this version:**

Jérémie Chalopin, Yves Métivier, Thomas Morsellino. About The Power of Anonymous Radio Networks. 2010. hal-00540222v1

**HAL Id: hal-00540222**

**<https://hal.science/hal-00540222v1>**

Preprint submitted on 28 Nov 2010 (v1), last revised 16 May 2011 (v3)

**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.

# About The Power of Anonymous Radio Networks

J. Chalopin<sup>1</sup>, Y. Métivier<sup>2</sup>, and T. Morsellino<sup>2</sup>

<sup>1</sup> LIF, CNRS, Université Aix-Marseille  
39, rue Joliot Curie  
13453 Marseille Cedex 13, France  
*jeremie.chalopin@lif.univ-mrs.fr*

<sup>2</sup> Université de Bordeaux, LaBRI UMR 5800, CNRS  
351, cours de la Libération  
33405 Talence, France  
*{metivier, morsellino}@labri.fr*

**Abstract.** We address the enumeration and the leader election problems over multi-hop radio networks. We consider an asynchronous radio communication model where each process emits a message and all its neighbours hear this message after arbitrary and unpredictable time. This amount of time depends on the processing time taken by the network layer to ensure message delivery. In this paper, we characterize graphs in which we can solve these problems and we present an enumeration algorithm and a leader election algorithm. For both problems, we highlight the importance of the initial knowledge. For the enumeration problem, each process only knows the size of the graph and, contrary to related works, the number of its neighbouring processes is unknown. Whereas for the election problem, we show that this knowledge is not sufficient; our algorithm supposes that each process initially knows a map of the network. Given a graph of maximum degree  $\Delta$  with  $n$  nodes and  $m$  edges, our algorithms both yield  $O(mn^2)$  emitted messages of size  $O(\Delta \log n)$  bits.

## 1 Introduction

An ad-hoc radio network is a collection of processes which communicate through radio medium without using preexisting infrastructure such as access points. If all processes can communicate directly, the network is said to be single-hop, otherwise, it is multi-hop. This paper deals with well-known distributed computing problems: enumeration and leader election.

### 1.1 Enumeration and Election

The aim of a naming algorithm is to give unique identities to all processes. The enumeration problem is a variant of the naming problem and aims to give to each node a unique number between 1 and the size of the graph. In ad-hoc radio networks such as radio sensor networks, existence of identified nodes allows better routing of information, resource management and performance. We consider *anonymous* radio networks where initially processes do not have identifiers and execute the same algorithm. In this setting, it is not always possible to give unique names to all vertices.

Naming and enumeration algorithms constitute a building block of many other distributed algorithms such as election, introduced by LeLann [LeL77]. A distributed algorithm solves the election problem if it always terminates and in the final configuration exactly one process is marked as *elected* and all the other processes are marked as *non-elected*. Using enumeration/naming algorithm, one can promote the process with the highest (resp. lowest) identifier as *elected*. However, enumeration and election problems are not necessarily equivalent (see [BCG<sup>+</sup>96, CMZ06]). We are here interested in characterizing graphs in which there exists an algorithm that solves the enumeration problem or that solves the election problem.

### 1.2 An Example of Application: Monitoring with Wireless Sensors

In [BB06], Balasundaram *et al.* provide a survey of existing models and algorithms for important problems which happen in different areas of radio communication. We extend their work by presenting a common

application in which enumeration and leader election problems take place. Consider a large spacial area in which there exist several natural phenomena such as earthquakes. In order to monitor and prevent serious disasters, one deploys a large number of radio sensors. From the spontaneous aspect of this network, the manner of how to identify sensors with an address becomes a serious issue. Since each sensor cannot rely on a preexisting infrastructure such as *DHCP*<sup>3</sup> servers, it is not practical to manually assign a unique identity to each sensor. In this situation, we have to develop a distributed way to solve this problem. However, the only available knowledge of this network is the amount of sensors that have to be deployed. Let us recall that communications are symmetric, i.e., once a sensor can directly communicate with another one, the latter can also communicate with the first one. But, each sensor is not aware of the number of other sensors belonging to its neighbourhood. From these assumptions, we put to each sensor the same algorithm which allows it to get a unique identity between 1 and the number of sensors. This assignation is given by the computation of an enumeration algorithm over an anonymous network.

At the end of the computation, each node has a unique identity. Thus, we can more easily deploy various algorithms in order to monitor sensible areas. For instance, one can decide to collect monitored data or field information from every sensors. Nonetheless, considering different aspects inherent in radio communications and sensors such as energy consumption, power of computation or geographical position, we can dynamically choose a sensor which acts temporarily as a leader. In order to avoid energy wasting or network flooding, this node can gather data from the entire area and may be viewed as an interface between other sensors and scientists. For this problem, we develop a leader election algorithm. Since sensors have the same characteristic, we can choose a leader according to its identity in the network. For instance, we choose the sensors with the lowest identity. But, we have to take into account that the network sits over asynchronous radio communications (see [San06] (Chap. 3)). Thus, our leader election algorithm has to ensure that there is no conflict before declaring a sensor as elected.

### 1.3 The Model

We consider an asynchronous radio communication model. In this model, a network of processes is represented as a simple labelled graph  $G = (V, E)$  where  $V$  is the set of processes and  $E$  is the set of communication links.

**The Network.** Our work is based on the distributed, multi-hop radio network paradigm that functions without using any infrastructure such as a base station or access points for communications. Emitted messages are only heard by reachable nodes. In a given step, if more than one node try to emit a message to another one a collision occurs and no message is heard.

*Asynchronous Radio Communications.* We assume that the media access and data link problems such as collisions, interferences or additions are solved by the *Data Link* sublayer *Mac Access Control* [CM99] by using channel allocation protocols derivated from CSMA, CSMA/CA [Mec07] or CSMA/CD [Tan02] (Chap. 4). Thus, during a computation step, the model ensures that a node always hears an incoming message. Therefore, we consider the asynchronous radio communication message passing model: processes cannot access a global clock and execute computation steps (atomic emit, hear and internal computation) at arbitrary speed. A message emitted from a process to neighbours arrives within some finite but unpredictable time depending on algorithms used to ensure collision-free and interference-free transmissions.

*Network and Processes Knowledge.* In this paper, radio networks are anonymous with arbitrary topology that can be represented by simple connected graphs. A process cannot distinguish its neighbours (there is no port-numbering function). For the enumeration algorithm, we assume that every node knows the size of the network whereas for the election algorithm the whole graph is known. The state of each process  $v$  is represented by the label  $\lambda(v)$  of the corresponding vertex  $v$ .

*Remark 1.* Our model may be viewed as an extension of the classical asynchronous point-to-point message passing model without port-numbering (see [Tel00] (Chap. 12)).

<sup>3</sup> Dynamic Host Configuration Protocol

## 1.4 Overview of our Contribution

In the sequel, we consider the asynchronous radio communications depicted above and we give complete characterizations of radio networks where there exists an enumeration algorithm or a leader election algorithm (Theorem 1 and Theorem 2). In our model, enumeration and election problems are not equivalent.

The structure of this paper is as follows. First, we present in Section 2 the notion of fibration that is a generalization of coverings for directed graphs. We present the fundamental lemma (Lemma 1) which connects fibrations and radio communications.

Then, for both problems, we prove impossibility results (Sections 3.1 and 4.1) and state they are sufficient by giving two algorithms (Algorithms 1 and 2).

About the complexity, we prove in Proposition 3 that our algorithms yield a polynomial number of messages of polynomial size. In the case of Yamashita and Kameda's and Boldi *et al.*'s, algorithms need messages of exponential size.

We highlight the importance of the initial knowledge. Our algorithms do not require each process to know its degree. For the enumeration problem, processes only know the size of the network. However, we show that in our model, this initial knowledge is not sufficient when one considers the election problem. Thus, we present a leader election algorithm where we assume that each process knows the whole network but is not aware of its position in this network.

## 1.5 Related Works: Comparison and Comments

**Distributed Algorithms on Anonymous Point-to-Point Networks.** Graphs where election or naming are possible were already studied for different basic models. Solutions depend on the type of basic computation steps, the type of network topology or the initial knowledge.

Considering the classical message passing model, Yamashita and Kameda [YK96a] characterize graphs for which there exists an election algorithm and underline the importance of the port-numbering function that allows each vertex to distinguish its neighbours. Their work introduces in [YK99] the *broadcast-to-mailbox* communication model that is closely related to our work in the sense that there is no port-numbering function, but they assume that each node knows its degree. They state that there exists an election algorithm for a graph  $G$  if and only if the symmetricity of  $G$  is equal to 1 by counting the number of vertices having the same view. The view from a vertex  $v$  of  $G$  without port-numbering function is the infinite labelled tree rooted in  $v$  corresponding to all walks in  $G$  from  $v$ . In this model, the election problem and the enumeration problem are not equivalent.

Boldi *et al.* [BCG<sup>+</sup>96] consider a model where the network is a directed multigraph. They distinguish models with or without port-numbering, and synchronous or interleaved activation models. When a process is activated, it changes its state depending on its previous one and the states of its ingoing neighbours. They use particular homomorphisms which we also use in our work. If there exists a port-numbering function, election and enumeration are equivalent and the asynchronous and synchronous models can be compared with the model of Yamashita and Kameda described above. From the complexity point of view, in both models, their algorithms need to apply  $O(n^2)$  computations steps and vertices need to exchange an exponential amount of information. The required memory in each process is exponential with  $2^{O(n)}$  bits. When the vertices initially know their degree, the model of Boldi *et al.* is equivalent to the model of Yamashita and Kameda. However, this is not longer the case, when vertices have no initial information about their degree.

Mazurkiewicz [Maz97] considers a more powerful model where in one step, labels are modified on a star, according to rules depending on this star. He presents an efficient enumeration algorithm for the graphs that are minimal for the covering relation, i.e., they only cover graphs that are isomorphic. In his algorithm, each process exchanges an amount of information that is polynomial depending on the size of the graph.

In [Cha05,CMZ06,CM04,Maz04], several models are studied in which a computation step is the synchronization between two adjacent vertices. In these models, election and naming are not always equivalent.

**Distributed Algorithms on Radio Networks.** The algorithmic research on anonymous radio networks is principally oriented to gossiping and broadcasting problems. Much of them are concentrated on randomized

algorithms [Chl01]. In [CGÖR00,CGG+00], authors present deterministic solutions for broadcasting but make the assumption that nodes only know their own label and that the topology is unknown.

Cidon and Mokryn present in [CM99] a leader election algorithm in multi-hop radio networks. This algorithm partitions the network into fragments that are collections of processes where one process is identified as a candidate and marked initially as active. Therefore, they consider networks that are not anonymous: each vertex has a unique identity. During the computation, a candidate can become inactive and joins another candidate's fragment. If there exists only one fragment containing all the processes with one process marked as active, the algorithm terminates. The model used in their work is similar to the broadcast-to-mailbox model of [YK96a,YK99].

In [Pel07], Pelc depicts a deterministic algorithm in radio network with no collision detection in order to decide whether a given node is accessible. He assumes that there exists a global clock and considers anonymous and synchronous networks in which processes are not aware of the global topology. One computation step is possible by taking into account messages previously heard by nodes. He defines a computation step according to histories of transmitted messages. This algorithm has a polynomial time complexity of  $O(n^4)$  (where  $n$  is the size of the network). He studies the existence of a deterministic universal activation algorithm. It depends on the synchronicity of communication and he proves that there exists a universal activation algorithm for synchronous communications but not for asynchronous communications. Note that this universal algorithm does not terminate.

## 2 Preliminaries

In order to describe our characterization, one needs to consider directed graphs (digraphs for short) that can have multiple arcs and self-loops. In this section, we present various definitions about digraphs and labelled digraphs. We also present fibrations which are a particular type of homomorphism. From these definitions, we give a fundamental lemma that establishes a link between fibrations and asynchronous radio communications.

### 2.1 Labelled Simple Graphs and Directed Graphs

Undirected graphs without multiple edges or loop are also called simple graphs. Each such a graph is written as  $G = (V(G), E(G))$  where  $V(G)$  is the set of vertices of  $G$  and where the set of edges  $E(G)$  is a set of pairs of distinct vertices of  $G$ . For each edge  $\{u, v\} \in E(G)$ ,  $u$  and  $v$  are the *ends* of  $\{u, v\}$  and  $u$  and  $v$  are said to be *adjacent* or *neighbours*. We denote by  $N_G(u)$  the set of all vertices of  $G$  adjacent to  $u$  and  $\deg_G(u)$  is the degree of  $u$  in  $G$ , i.e., the size of  $N_G(u)$ .

A simple graph  $G$  is *connected* if for all vertices  $u, v \in V(G)$ , there exists a path from  $u$  to  $v$ . Otherwise, it is *disconnected*. In the following, we will only consider strongly connected simple graphs.

A digraph  $D$  is defined by a set  $V(D)$  of vertices, a set  $A(D)$  of arcs and by two maps  $s_D$  and  $t_D$  (in general, the subscripts will be omitted) from  $A(D)$  to  $V(D)$ . For each arc  $a \in A(D)$ ,  $s(a)$  is the *source* of  $a$  and  $t(a)$  is its *target*. We say that  $a$  is going out of  $s(a)$  and coming into  $t(a)$ . A self-loop is an arc with the same source and target.

A digraph  $D$  is *strongly connected* if for all vertices  $u, v \in V(D)$ , there exists a sequence of arcs  $a_1, a_2, \dots, a_p$  such that  $s(a_1) = u, \forall i \in [1, p-1], t(a_i) = s(a_{i+1})$  and  $t(a_p) = v$ . In the following, we will only consider strongly connected digraphs.

A *symmetric* digraph  $D$  is a digraph endowed with a symmetry, that is, an involution  $Sym : A \rightarrow A$  such that for every  $a \in A : s(a) = t(Sym(a))$ .

**Definition 1.** A homomorphism  $\varphi$  from the digraph  $D$  to the digraph  $D'$  is given by a pair of functions  $\varphi_V : V(D) \rightarrow V(D')$  and  $\varphi_A : A(D) \rightarrow A(D')$  commuting with the source and target maps, i.e.,  $s_{D'} \circ \varphi_A = \varphi_V \circ s_D$  and  $t_{D'} \circ \varphi_A = \varphi_V \circ t_D$ .

In this paper, we consider digraphs where the vertices are labelled with labels from a recursive set  $L$ . A digraph  $D$  labelled over  $L$  will be denoted by  $(D, \lambda)$ , where  $\lambda : V(D) \rightarrow L$  is the labelling function. The

digraph  $D$  is called the underlying digraph and the mapping  $\lambda$  is a labelling of  $D$ . A mapping  $\varphi: V(D) \rightarrow V(D')$  is a homomorphism from  $(D, \lambda)$  to  $(D', \lambda')$  if  $\varphi$  is a digraph homomorphism from  $D$  to  $D'$  which preserves the labelling, i.e., such that  $\lambda'(\varphi(x)) = \lambda(x)$  for every  $x \in V(D)$ . Labelled digraphs will be designated by bold letters like  $\mathbf{D}, \mathbf{D}', \dots$ . If  $\mathbf{D}$  is a labelled digraph, then  $D$  denotes the underlying digraph.

Given a labelled connected undirected simple graph  $\mathbf{G} = (G, \lambda)$ , one associates a labelled symmetric strongly connected digraph denoted by  $Dir(\mathbf{G}) = (Dir(G), \lambda)$  and defined as follows. The set of vertices of  $Dir(\mathbf{G})$  is the set of vertices of  $G$ , i.e.,  $V(Dir(\mathbf{G})) = V(G)$  and each vertex in  $\mathbf{G}$  has the same label in  $Dir(\mathbf{G})$ . For each edge  $\{u, v\} \in E(G)$ , there exist two arcs  $a_{(u,v)}, a_{(v,u)} \in A(Dir(\mathbf{G}))$  such that  $s(a_{(u,v)}) = t(a_{(v,u)}) = u$ ,  $t(a_{(u,v)}) = s(a_{(v,u)}) = v$  and  $Sym(a_{(u,v)}) = a_{(v,u)}$ . Note that this digraph does not contain multiple arcs or self-loop.

Our characterizations use the notion of view. Informally, the view of a vertex  $v$  in a digraph  $\mathbf{D}$  is obtained by considering all labelled walks in  $\mathbf{D}$  ending in  $v$ .

**Definition 2.** *The view  $T_{\mathbf{D}}(v_0)$  of a vertex  $v_0$  is an infinite rooted labelled tree that can be defined recursively. The root of the tree is a vertex  $x_0$  that corresponds to  $v_0$  and is labelled by  $\lambda(v_0)$ . For each incoming neighbour  $v_i$  of  $v_0$  in  $\mathbf{D}$ , there is an arc between  $x_0$  and the root  $x_i$  of the tree  $T_{\mathbf{D}}(v_i)$ .*

We can also define the  $d$ -view of a vertex  $v$ , i.e., its view truncated at depth  $d$ :

**Definition 3.** *Given a labelled digraph  $\mathbf{D}$  and an integer  $d$ , the  $d$ -view  $T_{\mathbf{D}}^d(v_0)$  of a vertex  $v_0 \in V(D)$  is a tree of depth  $d$  that can be defined recursively as follows:*

- $T_{\mathbf{D}}^0(v_0)$  is a single-vertex graph whose node is denoted  $x_0$ ,
- $T_{\mathbf{D}}^{d+1}(v_0)$  is obtained by taking a copy of  $T_{\mathbf{D}}^d(v_i)$  for each incoming neighbour  $v_i$  of  $v_0$  in  $\mathbf{D}$ .

From this definition, we can state that the set of  $d$ -views of a digraph  $\mathbf{D}$  is finite. Thus, we can define a partial order  $\succeq$  on this set as follows:

**Definition 4.** *For every vertex  $v, w \in V(D)$ , if  $T = T_{\mathbf{D}}^d(w)$  is a subtree of  $T' = T_{\mathbf{D}}^d(v)$  then  $T' \succeq T$ . Note that if there exists an isomorphism between  $T$  to  $T'$ , they are said to be similar, denoted  $T \approx T'$ .*

*Remark 2.* As corollary of Definition 4, let  $\mathbf{H}$  be a sub-digraph of  $Dir(\mathbf{G})$ , for every vertex  $v \in Dir(\mathbf{G})$ ,  $T_{Dir(\mathbf{G})}^d(v) \succeq T_{\mathbf{H}}^d(v)$ .

Moreover, let  $\mathbf{D}$  and  $\mathbf{D}'$  be two digraphs. If  $\mathbf{D}$  is fibred over  $\mathbf{D}'$  via  $\varphi$ , then  $T_{\mathbf{D}}(v) \approx T_{\mathbf{D}'}(\varphi(v))$ , i.e, the view of  $v$  in  $\mathbf{D}$  is isomorphic to view  $\varphi(v)$  in  $\mathbf{D}'$ .

## 2.2 Homomorphisms and Fibrations

Fibrations are an important tool for this work (see [Bod89,BV02] for definitions and properties).

A fibration is a homomorphism that induces an isomorphism between the incoming arcs of each vertex and the incoming arcs of its image.

**Definition 5.** *A digraph  $D$  is fibred over a digraph  $D'$  via a homomorphism  $\varphi$  if  $\varphi$  is a homomorphism from  $D$  to  $D'$  such that for each arc  $a' \in A(D')$  and for each vertex  $v \in \varphi^{-1}(t(a'))$ , there exists a unique arc  $a \in A(D)$  such that  $t(a) = v$  and  $\varphi(a) = a'$ ; this arc  $a$  is called the lifting of  $a'$  at  $v$ .*

*We say that the homomorphism  $\varphi$  is a fibration from  $D$  to  $D'$ , the digraph  $D$  is the total digraph of  $\varphi$  and the digraph  $D'$  is the base of  $\varphi$ .*

*The fibre over a vertex  $v'$  (resp. an arc  $a'$ ) of  $D'$  is defined as the set  $\varphi^{-1}(v')$  of vertices of  $D$  (resp. the set  $\varphi^{-1}(a')$  of arcs of  $D$ ).*

*The digraph  $\mathbf{D}$  is minimal for the fibration relation if for every digraph  $\mathbf{D}'$  such that  $\mathbf{D}$  is fibred over  $\mathbf{D}'$ ,  $\mathbf{D}$  and  $\mathbf{D}'$  are isomorphic.*

*A simple graph  $\mathbf{G}$  is minimal for the fibration relation if  $Dir(\mathbf{G})$  is minimal for the fibration relation.*

From [BV02], we know that there exists a unique digraph  $\mathbf{B}_{\mathbf{G}}$  such that  $Dir(\mathbf{G})$  is fibred over  $\mathbf{B}_{\mathbf{G}}$ , and for each  $\mathbf{D}$  such that  $Dir(\mathbf{G})$  is fibred over  $\mathbf{D}$ ,  $\mathbf{D}$  is fibred over  $\mathbf{B}_{\mathbf{G}}$ . This digraph is called the *minimal base* of  $\mathbf{G}$ .

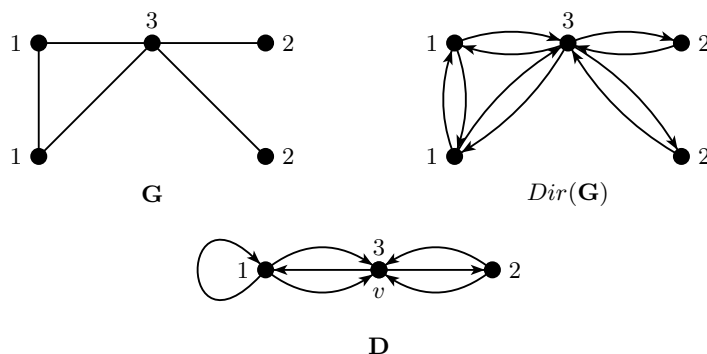
If a digraph  $\mathbf{D}$  is fibred on a digraph  $\mathbf{D}'$  via a homomorphism  $\varphi$ , and if  $\mathbf{D}$  and  $\mathbf{D}'$  are not isomorphic, we say that  $\mathbf{D}$  is *properly* fibred over  $\mathbf{D}'$  and that  $\varphi$  is a *proper* fibration.

In this work, we need to define non-trivial fibrations. Non-trivial fibrations are fibrations in which the size of the fibre of at least one vertex is not a singleton:

**Definition 6.** *The fibre of a vertex  $v$  is trivial if  $|\varphi^{-1}(v)| = 1$ , otherwise, it is non-trivial. A fibration  $\varphi$  is non-trivial if at least one fibre is non-trivial; it is proper if all fibres are non-trivial.*

*A simple graph  $\mathbf{G}$  is minimal for the non-trivial fibration relation if  $Dir(\mathbf{G})$  is minimal for the non-trivial fibration relation.*

An example of (non-trivial) fibration is given in Figure 1.



**Fig. 1.** The labelled digraph  $Dir(\mathbf{G})$  is fibred over the digraph  $\mathbf{D}$ . This fibration is non-trivial and not proper since  $\mathbf{D}$  contains a vertex  $v$  such that its fibre is trivial;  $\mathbf{D}$  is minimal for the non-trivial fibration relation.

We define the notion of candidate for a digraph  $\mathbf{D}$  such that  $Dir(\mathbf{G})$  is fibred over  $\mathbf{D}$ .

**Definition 7.** *Let  $\mathbf{D}$  be a digraph such that there exists a simple graph  $\mathbf{G}$  such that  $Dir(\mathbf{G})$  is fibred over  $\mathbf{D}$  via  $\varphi$ , a vertex  $v$  is a candidate of  $\mathbf{D}$  if its fibre is trivial, i.e.  $|\varphi^{-1}(v)| = 1$ . We denote  $C_{\mathbf{D}}$  the set of candidates of  $\mathbf{D}$ .*

### 2.3 Fibrations and Radio Communications

In order to extend the lifting lemma of Angluin [Ang80] to asynchronous radio communications, we need to define associated computations on directed graphs.

Following Section 1.3, we define a computation step associated to asynchronous radio communications. Informally, when a process emits a message, it modifies its state according to only its previous state, while its neighbouring processes that hear the message modify their states following their previous states and the state of the emitting process. Consequently, associated computations only need directed communications between emitting processes and hearing processes. The following lemma connects fibrations and asynchronous radio communication steps. A *maximal* execution  $\rho$  of an algorithm is either an infinite execution, or a finite execution such that in the final configuration, there is no message in transit and no process wants to emit a message.

**Lemma 1 (Asynchronous Lifting Lemma).** *Consider a digraph  $\mathbf{D}_1$  fibred over a digraph  $\mathbf{D}_2$  via  $\varphi$  and let  $\mathcal{A}$  be an algorithm based on the asynchronous radio communication model. If there exists a maximal execution  $\rho_2$  of  $\mathcal{A}$  on  $\mathbf{D}_2$  which yields  $\mathbf{D}'_2$  then there exists a maximal execution  $\rho_1$  of  $\mathcal{A}$  on  $\mathbf{D}_1$  which yields  $\mathbf{D}'_1$  such that  $\mathbf{D}'_1$  is fibred over  $\mathbf{D}'_2$  via  $\varphi$ .*

### 3 An Enumeration Algorithm for Radio Networks

In this section, we give a necessary condition based on an impossibility result which states that there exists no enumeration algorithm for a graph  $\mathbf{G}$  such that  $Dir(\mathbf{G})$  is not minimal for the fibration relation. Then, we prove that this condition is sufficient by presenting an enumeration algorithm  $\mathcal{M}$  (Algorithm 1) which relies on asynchronous radio communications and inspired by the work of Mazurkiewicz [Maz97].

#### 3.1 Impossibility Result

Given a network represented by a graph  $\mathbf{G}$ , we present a necessary condition that must be satisfied by  $\mathbf{G}$  to admit an enumeration algorithm. This is an impossibility result that relies on the notion of fibrations for asynchronous computations. Following the proof of Lemma 1 presented above, we obtain a contradiction.

Informally, as explained above, two neighbouring processes may have the same state at the end of the computation. In this sense, computation steps associated to asynchronous radio communications allow fibration relation with self-loop.

**Proposition 1.** *Let  $\mathbf{G}$  be a labelled graph such that  $Dir(\mathbf{G})$  is not minimal for the fibration relation, there is no enumeration algorithm for  $\mathbf{G}$  in the asynchronous radio communication model.*

*Proof.* Consider a simple graph  $\mathbf{G} = (G, \lambda)$  and a strongly connected digraph  $\mathbf{D} = (D, \eta)$  such that  $Dir(\mathbf{G})$  is properly fibred over  $\mathbf{D}$  via a fibration  $\varphi$ . Given an algorithm  $\mathcal{A}$  using asynchronous radio communications, consider an execution of  $\mathcal{A}$  on  $\mathbf{D}$  as described in Lemma 1. Note that if there exists an infinite execution of  $\mathcal{A}$  on  $\mathbf{D}$ , then following Lemma 1 there exists an infinite execution of  $\mathcal{A}$  on  $\mathbf{G}$ . Finally,  $\mathcal{A}$  is not an enumeration algorithm for  $\mathbf{G}$ .

Suppose that there exists a finite execution of  $\mathcal{A}$  on  $\mathbf{D}$  which yields a configuration  $\mathbf{D}'$ . In the final configuration every message has been arrived and no process has to emit a message. Thus, each vertex has its final label. Following Lemma 1, there exists a lifted execution of  $\mathcal{A}$  on  $Dir(\mathbf{G})$  that yields a configuration  $\mathbf{G}'$  such that  $\mathbf{G}'$  is properly fibred over  $\mathbf{D}'$  via  $\varphi$ . Since  $\mathbf{G}'$  is properly fibred over  $\mathbf{D}'$  it implies that there exist at least two vertices that have the same label in  $\mathbf{G}'$  and therefore in  $Dir(\mathbf{G})$ . Hence, the algorithm  $\mathcal{A}$  does not give a distinct label to each vertex. Therefore,  $\mathcal{A}$  is not an enumeration algorithm for  $\mathbf{G}$ .

#### 3.2 An Enumeration Algorithm.

During the execution of the enumeration algorithm, each vertex  $v$  attempts to get its unique identity label : a number between 1 and  $|V(G)|$ . Once a vertex  $v$  has chosen a number  $n(v)$ , it emits it to its neighbourhood. Since, there does not exist a port-numbering function, a vertex cannot distinguish its neighbours. When a vertex  $v$  hears a message from another vertex  $u$ , it stores the number  $n(u)$ . From all information it has gathered from its neighbours, each vertex  $v$  is able to create its *local view*. Schematically,  $\forall u \in N_G(v)$ , a local view is a set composed with stars centered in  $u$ . Then, a vertex broadcasts its number with its *local view*. If a vertex  $u$  discovers that there exists another vertex  $v$  with the same number then it should decide if it changes its identity : it compares its local view with the local view of  $v$ . If the label of  $u$  or the local view of  $u$  is *weaker* (for an order we define later), then  $u$  chooses another identity and emits it again with its local view. At the end of the computation, if the digraph is minimal for the fibration relation, then every vertex will have a unique number.

**Labels.** We consider a network  $\mathbf{G}$  where  $\mathbf{G} = (G, \lambda)$  is a simple labelled graph. The function  $\lambda : V(G) \rightarrow L$  is the initial labelling and is kept during the computation. We suppose that there exists a total order  $<_L$  on  $L$ . During the execution, the label of each vertex  $v$  is a tuple  $(\lambda(v), n(v), N(v), M(v))$  corresponding to the following information:

- $\lambda(v) \in L$  is the initial label of  $v$  and is not modified by the algorithm.
- $n(v) \in \mathbb{N}$  is the current *number* of the vertex  $v$  computed by the algorithm.



- $N(v) \in \mathcal{P}_{\text{fin}}(\mathbb{N}^2)^4$  is the *local view* of  $v$ . Intuitively, once  $v$  has updated its local view,  $(n, p)$  belongs to  $N(v)$  if  $v$  has  $p$  neighbours that have  $n$  as an identity number.
- $M(v) \in \mathbb{N} \times L \times \mathcal{P}_{\text{fin}}(\mathbb{N}^2)$  is the *mailbox* of  $v$ . The mailbox of  $v$  contains all information heard by  $v$  during the execution of the algorithm. If  $(m, \ell, \mathcal{N}) \in M(v)$ , it means that at some previous step of the execution, there was a vertex  $u$  such that  $n(u) = m$ ,  $\lambda(u) = \ell$  and  $N(u) = \mathcal{N}$ .

---

**Algorithm 1:** Algorithm  $\mathcal{M}$  in the asynchronous radio communication model.

---

```

var:   to_emit : bool init false;
        n_old : int init 0 ;
I : { $n(v_0) = 0$  and no message has arrived at  $v_0$ }
begin
  |  $M_{old} := \emptyset$ ;
  |  $n(v_0) := 1$  ;
  |  $M(v_0) := \{(n(v_0), \lambda(v_0), \emptyset)\}$ ;
  |  $to\_emit := true$ ;
end
S : { $to\_emit = true$ }
begin
  | emit  $\langle (n(v_0), n_{old}, M(v_0)) \rangle$ ;
  |  $n_{old} := n(v_0)$ ;
  |  $to\_emit := false$ ;
end
R : {A message  $\langle (n', n'_{old}, M') \rangle$  has arrived at  $v_0$ }
begin
  |  $M_{old} := M(v_0)$ ;
  |  $M(v_0) := M(v_0) \cup M'$ ;
  | if  $n(v_0) = 0$  or  $\exists (n(v_0), \ell, \mathcal{N}) \in M(v_0)$  such that  $(\lambda(v_0), N(v_0)) \prec (\ell, \mathcal{N})$  then
  |   |  $n(v_0) := 1 + \max\{n \mid \exists (n, \ell, \mathcal{N}) \in M(v_0)\}$ ;
  |   |  $N(v_0) := \text{replace}(n', n_{old})$ ;
  |   |  $M(v_0) := M(v_0) \cup \{(n(v_0), \lambda(v_0), N(v_0))\}$ ;
  |   | if  $M(v_0) \neq M_{old}$  then
  |     |  $to\_emit := true$ ;
end

```

---

Initially, each vertex  $v$  has a label of the form  $(\lambda(v), 0, \emptyset, \emptyset)$  indicating that it has not chosen any number, that it has no information about its neighbours or about the other vertices of the graph.

In order to update the local view of a process  $v_0 \in V(G)$ , we define a function  $\text{replace}(n', n_{old})$  the operation on set of couples of integers defined as follows. First, if  $(n_{old}, p) \in N(v_0)$  then we first remove  $(n_{old}, p)$  from  $N(v_0)$  and we add  $(n_{old}, p - 1)$  to  $N(v_0)$  if  $p > 1$ . Thereafter, if  $(n', p) \in N(v_0)$ , we replace  $(n', p)$  in  $N(v_0)$  by  $(n', p + 1)$ , and otherwise,  $N(v_0) = N(v_0) \cup (n', 1)$ .

**Messages.** In our algorithm, processes exchange messages of the form  $\langle (m, n_{old}, M) \rangle$ . If a vertex  $u$  emits a message  $\langle (m, n_{old}, M) \rangle$  to one of its neighbour  $v$ , then  $m$  is the current number  $n(u)$  of  $u$ ,  $n_{old}$  is the previous number of  $u$ , i.e., the number  $u$  emits to  $v$  in its previous message; if in the meanwhile,  $u$  has not modified its number, then  $n_{old} = m$ . And  $M$  is the mailbox of  $u$ .

---

<sup>4</sup> For any set  $S$ ,  $\mathcal{P}_{\text{fin}}(S)$  denotes the set of finite subsets of  $S$ .

**An Order on Local Views.** As in Mazurkiewicz's algorithm [Maz97], the nice properties of the algorithm rely on a total order on local views, i.e., on finite subsets of  $\mathcal{P}_{\text{fin}}(\mathbb{N}^2)$ . The algorithm described above is such that the local view of any vertex cannot decrease during the computation.

In order to compare two elements of  $\mathbb{N}^2$ , we use the usual lexicographic order on  $\mathbb{N}^2$ :  $(n, p) < (n', p')$  if  $n < n'$ , or if  $n = n'$  and  $p < p'$ .

Let  $N_1, N_2 \in \mathcal{P}_{\text{fin}}(\mathbb{N}^2)$ ,  $N_1 \neq N_2$ . Then  $N_1 \prec N_2$  if the maximal element of the symmetric difference  $N_1 \triangle N_2 = (N_1 \setminus N_2) \cup (N_2 \setminus N_1)$  belongs to  $N_2$ . Note that in particular the empty set is minimal for  $\prec$ .

If  $N(u) \prec N(v)$  then we say that the local view  $N(v)$  of  $v$  is *stronger* than the one of  $u$  (and  $N(u)$  is *weaker* than  $N(v)$ ). We assume for the rest of the paper that the set of initial labels  $L$  is totally ordered by  $<_L$ . We extend  $\prec$  to a total order on  $L \times \mathcal{P}_{\text{fin}}(L \times \mathbb{N})$ :  $(\ell, N) \prec (\ell', N')$  if either  $\ell <_L \ell'$ , or  $\ell = \ell'$  and  $N \prec N'$ . We denote by  $\preceq$  the reflexive closure of  $\prec$ .

### 3.3 Correctness of $\mathcal{M}$

Let  $\mathbf{G}$  be a simple labelled graph. In the following,  $i$  is an integer denoting a computation step. Let  $(\lambda(v), (n_i(v), N_i(v), M_i(v)))$  be the label of the vertex  $v$  after the  $i$ th step of the computation of the algorithm  $\mathcal{M}$  given above. We present some properties satisfied by each execution of the algorithm in the asynchronous radio communication model.

The following lemma, which can be proved easily by induction on the number of steps, recapitulates basic labelling properties.

**Lemma 2.** *For each vertex  $v$  and each step  $i$ ,*

1.  $n_i(v) \neq 0 \implies (n_i(v), \lambda(v), N_i(v)) \in M_i(v)$ ,
2.  $\forall n' \in N_i(v), n' > 0$  and  $\exists \ell' \in L, \exists N' \in \mathcal{P}_{\text{fin}}(\mathbb{N}^2), (n', \ell', N') \in M_i(v)$ .

The algorithm has some remarkable monotonicity properties that are described in the following lemma.

**Lemma 3.** *For each step  $i$  and each vertex  $v$ ,  $M_i(v) \subseteq M_{i+1}(v)$ ,  $n_i(v) \leq n_{i+1}(v)$ , and  $N_i(v) \preceq N_{i+1}(v)$ . Moreover, if  $v$  applies the action  $\mathbf{S}$  at step  $i$  and  $j$ , then  $M_i(v) \neq M_j(v)$ .*

The local knowledge of a vertex  $v$  reflects to some extent some real properties of the current configuration. The two following lemmas enable us to prove that if a vertex  $v$  knows a number  $m$  (i.e., there exist  $\ell, N$  such that  $(m, \ell, N) \in M_i(v)$ ), then for each  $m' \leq m$ , there exists a vertex  $v'$  in the graph such that  $n_i(v') = m'$ . We first show that if  $v$  knows  $m$  there exists  $v'$  such that  $n_i(v') = m$ .

**Lemma 4.** *For every  $v \in V(G)$  and  $(m, \ell, N) \in M_i(v)$ , there exists a vertex  $w \in V(G)$  such that  $n_i(w) = m$ .*

In the following lemma, we show that if a vertex  $v$  knows an identity number  $m$ , then it knows all the numbers smaller than  $m$ .

**Lemma 5.** *For every vertex  $v \in V(G)$  and every step  $i$  such that  $n_i(v) \neq 0$ , given  $(m', \ell', N') \in M_i(v)$ , for every  $1 \leq m \leq m'$ , there exists  $(m, \ell, N) \in M_i(v)$ .*

From Lemmas 4 and 5, we deduce that for each step, the identity numbers of all the vertices form either a set  $[1, k]$  or a set  $[0, k]$  with  $k \leq V(G)$ .

For each step  $i$  and each vertex  $v$ , if there exists  $n' \in N_i(v)$ , from Lemma 2, there exists  $v'$  such that  $n_i(v') = n'$  and therefore  $N(v)$  can only have a finite number of values and the same holds for  $M(v)$ . During the algorithm, the consecutive labelling of each vertex  $v$  form an increasing sequence,  $(n_i(v), N_i(v), M_i(v))$ ,  $i = 1, 2, \dots$  and, each vertex can emit a message only if it modifies its mailbox. Since the number of possible accessible labels is finite (but dependent on the size of the graph), the algorithm always terminates.

From this remark, for each run of  $\mathcal{M}$  on a minimal graph  $\mathbf{G}$ , each vertex has a unique number and the graph associated to the final labelling is isomorphic to  $\mathbf{G}$  and therefore the set of numbers of the vertices is exactly  $1, \dots, |V(\mathbf{G})|$ .

Moreover, we make the assumption that every node knows the size of the network. Hence, once a process get the number  $|V(G)|$ , from Lemmas 4 and 5, it knows that all the vertices have different identity numbers that will not change any more and it can detect locally the termination of the algorithm.

Since we have proven that  $\mathcal{M}$  always terminates, we can give some properties about the final labelling:

**Lemma 6.** Any execution  $\rho$  of  $\mathcal{M}$  on a connected labelled graph  $\mathbf{G} = (G, \lambda)$  terminates and yields to a final labelling  $(\lambda, n_p, N_p, M_p)$  satisfying the following conditions:

1. there exists an integer  $k \leq |V(G)|$  such that  $\{n_p(v) \mid v \in V(G)\} = [1, k]$ ,

and for all vertices  $v, v'$ :

2.  $M_p(v) = M_p(v')$ ,
3.  $(n_p(v), \lambda(v), N_p(v)) \in M_p(v')$ ,
4.  $n_p(v) = n_p(v')$  implies that  $\lambda(v) = \lambda(v')$  and  $N_p(v) = N_p(v')$ ,
5.  $(n, p) \in N_p(v)$  if and only if there exists  $w_1, \dots, w_p \in N_G(v)$  such that for each  $i$   $n_p(w_i) = n$ ; in this case, there exists  $(n_p(v), p') \in N_p(w_i)$  with  $p' \geq 1$ .

We can therefore prove that there exists a digraph  $\mathbf{D}$  associated to the final labelling of  $\mathbf{G}$  such that  $Dir(\mathbf{G})$  is a fibration of  $\mathbf{D}$ .

**Proposition 2.** Given a graph  $\mathbf{G}$ , we can associate with the final labelling of any execution  $\rho$  of the enumeration algorithm on  $\mathbf{G}$ , a digraph  $\mathbf{D}$  such that  $Dir(\mathbf{G})$  is fibred over  $\mathbf{D}$ .

If  $\mathbf{G}$  is minimal for the fibration relation, the digraph  $\mathbf{D}$  of the previous proposition is isomorphic to  $Dir(\mathbf{G})$ . Consequently, each vertex has a unique number between 1 and  $|V(G)|$  in the final labelling. Moreover, we make the assumption that every node knows the size of the network. Hence, once a process has  $|V(G)|$  different numbers in its mailbox, from Lemmas 4 and 5, it knows that all the vertices have different identity numbers that will not change anymore and it can detect locally the termination of the algorithm.

Thus we have proven the following theorem :

**Theorem 1.** For every graph  $\mathbf{G}$ , there exists a(n) enumeration/naming algorithm on  $\mathbf{G}$  using asynchronous radio communications if and only if the digraph  $Dir(\mathbf{G})$  is minimal for the fibration relation.

### 3.4 Complexity Analysis

Considering radio networks, complexity analysis of distributed algorithms constitutes a building block of many properties such as energy consumption. In this part, we deal with the complexity of Algorithm 1. We are interested in the number of messages exchanged by the processes and their size. We also look at the memory needed by each vertex.

We consider that each vertex does not transmit its whole mailbox when it sends a message, but only the elements  $(n, \ell, N)$  that it has added to its mailbox. Moreover, for sake of simplicity in the analysis of the complexity, we assume that it sends these elements one by one, i.e, each message has the form  $\langle (n, n_{old}, (n', \ell', N')) \rangle$ . Furthermore, each vertex does not need to keep more than one element  $(n, \ell, N)$  for each  $n$  in its mailbox. Indeed, if there are two elements  $(n, \ell, N), (n, \ell', N') \in M(v)$ , and if  $(\ell, N) \prec (\ell', N')$ , we can remove  $(\ell, N)$  from the mailbox. Moreover, we assume that the initial labelling of  $G$  is such that each initial label  $l$  is computed over  $O(\log |V(G)|)$  bits.

**Proposition 3.** Let  $\mathbf{G}$  be a labelled graph of size  $n$  with  $m$  edges and a maximum degree  $\Delta$ . Any run of  $\mathcal{M}$  yields  $O(mn^2)$  emissions of messages of size  $O(\Delta \log n)$  bits. Moreover, it requires  $O(\Delta n \log n)$  bits of memory at any vertex.

Algorithms of Yamashita and Kameda and of Boldi *et al.* presented in Section 1.5 yields  $O(mn)$  messages of size  $2^{O(n)}$  bits. Moreover, each process requires  $2^{O(n)}$  memory bits. Thus, considering different aspects of the complexity,  $\mathcal{M}$  fits particularly well to radio networks.

## 4 A Leader Election Algorithm for Radio Networks

As stated in the introduction, if we can solve the enumeration problem on a graph  $\mathbf{G}$  then we can solve the election problem on this graph by declaring the vertex with the identity number  $|V(\mathbf{G})|$  as elected. Nonetheless, in our model, the enumeration and the election problems are not equivalent. Consider the graph  $\mathbf{G}$  and the digraph  $Dir(\mathbf{G})$  of Figure 1. Since  $Dir(\mathbf{G})$  is fibred over  $\mathbf{D}$ , from Theorem 1, the enumeration problem cannot be solved on  $\mathbf{G}$ . Nonetheless, if every vertex initially knows  $\mathbf{G}$ , at the end of the execution of  $\mathcal{M}$  on  $Dir(\mathbf{G})$ , the vertex labelled 3 in  $Dir(\mathbf{G})$  can declare itself as elected since it knows it is unique in its fibre.

In this section, we also present an impossibility result which states that there exists no leader election algorithm for a graph  $\mathbf{G}$  if  $Dir(\mathbf{G})$  is not minimal for the non-trivial fibration relation, i.e., there exists no vertex such that its fibre is trivial. This condition is sufficient and we give an extension of  $\mathcal{M}$  (Algorithm 2) which effectively solve the election problem.

### 4.1 Impossibility Result

Given a network represented by a simple graph  $\mathbf{G}$ , we present a necessary condition based on non-trivial fibrations that must be satisfied by  $\mathbf{G}$  to admit a leader election algorithm.

**Proposition 4.** *Let  $\mathbf{G}$  be a labelled graph such  $Dir(\mathbf{G})$  is not minimal for the non-trivial fibration relation, there is no leader election algorithm for  $\mathbf{G}$  in the asynchronous radio communication model.*

*Proof.* Consider a simple graph  $\mathbf{G} = (G, \lambda)$  and a strongly connected digraph  $\mathbf{D} = (D, \eta)$  such that  $Dir(\mathbf{G})$  is non-trivially fibred over  $\mathbf{D}$  via a fibration  $\varphi$ . Given an algorithm  $\mathcal{A}$  using asynchronous radio communications, consider an execution of  $\mathcal{A}$  on  $\mathbf{D}$  as described in Lemma 1. Note that if there exists an infinite execution of  $\mathcal{A}$  on  $\mathbf{D}$ , then following Lemma 1 there exists an infinite execution of  $\mathcal{A}$  on  $\mathbf{G}$ . Finally,  $\mathcal{A}$  is not a leader election algorithm for  $\mathbf{G}$ .

Suppose that there exists a finite execution of  $\mathcal{A}$  on  $\mathbf{D}$  which yields a digraph  $\mathbf{D}'$ . In the final configuration every message has arrived and no process has to emit a message. Thus, each vertex has its final label. Following Lemma 1, there exists a lifted execution of  $\mathcal{A}$  on  $Dir(\mathbf{G})$  that yields a configuration  $\mathbf{G}'$  such that  $\mathbf{G}'$  is fibred over  $\mathbf{D}'$  via  $\varphi$ . Since  $\mathbf{G}'$  is non-trivially fibred over  $\mathbf{D}'$  it implies that for every vertex  $v \in V(\mathbf{G})$ , there exist at least two vertices in  $\varphi^{-1}(\varphi(v))$  that have the same label in  $\mathbf{G}'$ . Hence, there exists no vertex  $v \in V(\mathbf{G})$  that has a unique label. The algorithm  $\mathcal{A}$  is not a leader election algorithm for  $\mathbf{G}$ .

### 4.2 Initial Knowledge

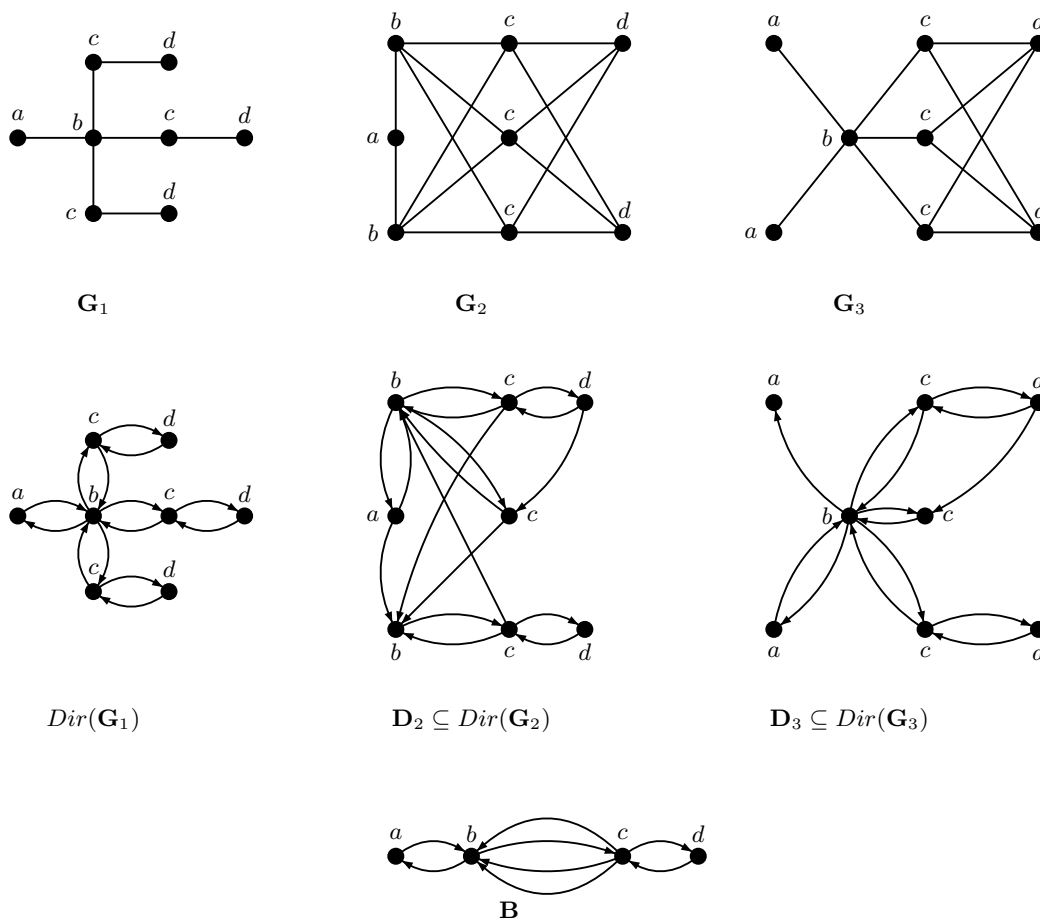
We here underline the importance of the initial knowledge. In the previous algorithm  $\mathcal{M}$ , every process only knows the size of the network. Using this initial knowledge, we ensure that at the end of the execution, each process locally knows that each vertex has been given a unique identity even though some messages are delayed and some processes are isolated. Boldi *et al.* [BCG<sup>+</sup>96] and Yamashita and Kameda [YK96b] also show that knowing the size of the graph allows to solve election problem whenever it is possible. However, in their models, each vertex initially knows its degree (or can compute it easily) and their initial knowledge is actually used in their views construction algorithm.

In our model, vertices do not initially know their degree and in this case, the initial knowledge of the size of the graph is not sufficient to solve the election problem on graphs where it can be solved. For instance, assume that there exists a leader election algorithm for the three graphs  $\mathbf{G}_1$ ,  $\mathbf{G}_2$  and  $\mathbf{G}_3$  of Figure 2. In  $\mathbf{G}_1$  (resp.  $\mathbf{G}_2$ ,  $\mathbf{G}_3$ ), there exists a unique vertex of degree 4 (resp. 2, 4). Hence, similarly to the graph of Figure 1, one can elect in these three graphs when we assume that each process initially knows the graph. Consider the digraph  $\mathbf{B}$  such that  $Dir(\mathbf{G}_1)$  is non-trivially fibred over  $\mathbf{B}$  via a not proper fibration  $\varphi$ . When executed on  $\mathbf{B}$ , a leader election algorithm for  $\mathbf{G}_1$  has to elect a process such that its fibre is trivial. Thus, there exist two vertices  $a, b \in \mathbf{B}$  such that  $|\varphi^{-1}(a)| = |\varphi^{-1}(b)| = 1$  and which can be declared as elected. Assume that several messages are arbitrary delayed, i.e., several communication links are not yet established. One can

find two graphs  $\mathbf{G}_2$  and  $\mathbf{G}_3$  and two digraphs  $\mathbf{D}_2$  and  $\mathbf{D}_3$  such that  $\mathbf{D}_2 \subseteq \text{Dir}(\mathbf{G}_2)$  and  $\mathbf{D}_3 \subseteq \text{Dir}(\mathbf{G}_3)$  and such that  $\mathbf{D}_2$  and  $\mathbf{D}_3$  are also non-trivially fibred over  $\mathbf{B}$ .

From Lemma 1, if there exists a finite and maximal execution of an algorithm that elects a leader in  $\mathbf{B}$  then there exists a finite and maximal execution on  $\text{Dir}(\mathbf{G}_1)$ ,  $\mathbf{D}_2$  and  $\mathbf{D}_3$  that also elect a leader. Hence, if the vertex  $b$  is declared as elected in  $\mathbf{B}$ , then there exists an execution on  $\text{Dir}(\mathbf{G}_2)$  where arcs not in  $\mathbf{D}_2$  are removed and at the end of this execution two vertices are elected. Similarly, if the vertex  $a$  is declared as elected in  $\mathbf{B}$ , then there exists a particular execution on  $\text{Dir}(\mathbf{G}_3)$  such that two vertices are marked as elected. Therefore, we cannot find a universal leader election algorithm for all graphs of size  $n$  where election problem can be solved. In the following, we provide a leader election algorithm  $\mathcal{M}_e$  which assumes that each process knows a map of the network.

### 4.3 A Leader Election Algorithm



**Fig. 2.** The labelled digraph  $\text{Dir}(\mathbf{G}_1)$  is fibred over the digraph  $\mathbf{B}$ . This fibration is non-trivial and  $\mathbf{B}$  is minimal; the sub-digraphs  $\mathbf{D}_2$  of  $\text{Dir}(\mathbf{G}_2)$  and  $\mathbf{D}_3$  of  $\text{Dir}(\mathbf{G}_3)$  are also non-trivially fibred over the minimal base  $\mathbf{B}$ . From Lemma 1, an execution of a leader election algorithm on  $\mathbf{B}$  can be lifted to an execution on  $\text{Dir}(\mathbf{G}_1)$  and an execution on  $\mathbf{D}_2$  and  $\mathbf{D}_3$ . Thus, the vertex  $a$  can be declared as elected in  $\mathbf{B}$ ,  $\mathbf{G}_1$  and  $\mathbf{G}_2$  and the vertex  $b$  can be declared as elected in  $\mathbf{B}$ ,  $\mathbf{G}_1$  and  $\mathbf{G}_3$ . If the algorithm chooses  $a$  (resp.  $b$ ), then two vertices in  $\mathbf{D}_2$  (resp.  $\mathbf{D}_3$ ) are declared as elected: that is not possible.

We present how to use  $\mathcal{M}$  to solve the leader election problem on digraphs that are minimal for the non-trivial fibration relation.

Consider a graph  $\mathbf{G}$  such that  $\text{Dir}(\mathbf{G})$  is non-trivially fibred over a digraph  $\mathbf{D}$  in which there exists at least one vertex with a trivial fibre. Our aim is to provide an extension of our previous algorithm by using

the termination detection algorithm of [SSP85]. The idea is to execute this algorithm and to reconstruct a graph from the contents of the vertices mailboxes and check if all processes are involved in the execution, i.e., if there is no isolated process.

**The SSP Algorithm.** Initially, this algorithm was devised to detect the termination of an algorithm. As stated in Section 3.3, each process is able to determine its termination condition. The SSP algorithm detects an instant in which the entire computation is achieved.

Let  $G$  be a graph, to each node  $v$  is associated a predicate  $P(v)$  and an integer  $a(v)$ , its confidence level. Initially,  $P(v)$  is false and  $a(v)$  is equal to  $-1$ . If a vertex  $v$  has finished its computation of the initial algorithm, then it changes its value  $P(v)$  to true. Each time a vertex changes the value of  $P(v)$  or  $a(v)$  then it informs its neighbours.

The modification of the value of  $a(v_0)$  only depends on the value of  $P(v_0)$  and the informations  $v_0$  has about the values  $\{a(v_1), \dots, a(v_d)\}$  of its neighbours:

- if  $P(v_0) = false$  then  $a(v_0) = -1$ ,
- if  $P(v_0) = true$  then  $a(v_0) = 1 + \min\{a(v_k) \mid k \in [0; d]\}$ .

We will adapt this algorithm using the ideas of the algorithm GSSP [GMM04]. For every vertex  $v$ , the value of  $P(v)$ , instead of being boolean, will be a graph reconstructed from the contents of the mailbox of  $v$ . An important property of the function  $P$  is that it is constant between two moments where it has the same value.

In our models, a vertex cannot distinguish its neighbours: therefore we will use the members that appear in the local view. A vertex  $v$  will increase its confidence level  $a(v)$  only if for each neighbouring vertex  $v'$ , it has heard a message such that  $M(v') = M(v)$  and  $a(v') \geq a(v)$ .

**Labels.** As in the enumeration algorithm, we start with a labelled graph  $\mathbf{G} = (G, \lambda)$ . During the computation vertices  $v$  will get new labels of the form  $(\lambda(v), n(v), N(v), M(v), a(v), A(v))$ . Thus, we add to the label of each vertex two items:

- $a(v) \in \mathbb{N}$  is the confidence level of the vertex  $v$ ,
- $A(v) \in \mathcal{P}_{\text{fin}}(\mathbb{N}^3)$  is a set maintained by each vertex  $v$ . It contains the confidence level of its neighbours in the form  $(n, p, a)$  where  $p$  is the number of the neighbours of  $v$  with  $n$  as identity number and  $a$  as confidence level. If  $(n, p)$  belongs to  $N(v)$ , then  $(n, p, a)$  belongs to  $A(v)$ .

For sake of simplicity, we define a function *confidence* to update the set  $A(v_0)$  of a process  $v_0$  as follows. For every process  $v_0 \in V(G)$ , we denote  $\text{confidence}(n', a', a_{old})$  the operation on set of triples of integers defined as follows. If  $(n', p, a_{old}) \in A(v_0)$  then we first remove  $(n', p, a_{old})$  in  $A(v_0)$  and we add  $(n', p - 1, a_{old})$  to  $A(v_0)$  if  $p > 1$ . Then, if  $(n', p, a') \in A(v_0)$  then we replace  $(n', p, a')$  in  $A(v_0)$  by  $(n', p + 1, a')$ , and otherwise,  $A(v_0) = A(v_0) \cup \{(n', 1, a')\}$ .

Note that in Algorithm 2, the digraph  $\mathbf{B}_{\mathbf{G}}$  is the minimal base of the initial digraph  $\text{Dir}(\mathbf{G})$  on which the algorithm is performed.

**Messages.** A message emitted by a process  $u$  and heard by the process  $v$  has the following form  $\langle (m, n_{old}, M, a) \rangle$  where  $m$ ,  $n_{old}$  and  $M$  are identical to values of messages exchanged in  $\mathcal{M}$ . We add the item  $a$  which is the value of the confidence level  $a(u)$  of  $u$ .

#### 4.4 Correctness of $\mathcal{M}_e$

Let  $\mathbf{G}$  be a simple labelled graph. In the following,  $i$  is an integer denoting a computation step. Let  $(\lambda(v), n_i(v), N_i(v), M_i(v), a_i(v), A(v))$  be the label of the vertex  $v$  after the  $i$ th step of the computation of the algorithm  $\mathcal{M}_e$ . We present some properties satisfied by each execution of the algorithm in the asynchronous radio communication model.

We can easily state by induction that if the mailbox of a vertex  $v$  is the same between two steps, the confidence level of  $v$  increases.

---

**Algorithm 2:** Algorithm  $\mathcal{M}_e$  in the asynchronous radio communication model.

---

```

var:   to_emit : bool init false;
         n_old : int init 0 ;
I : { $n(v_0) = 0$  and no message has arrived at  $v_0$ }
begin
  |  $n_e := -1$ ;
  |  $a(v_0) := 0$ ;
  |  $M_{old} := \emptyset$ ;
  |  $n(v_0) := 1$  ;
  |  $M(v_0) := \{(n(v_0), \lambda(v_0), \emptyset)\}$ ;
  |  $to\_emit := true$ ;
end
S : { $to\_emit = true$ }
begin
  | emit  $\langle (n(v_0), n_{old}, M(v_0), a(v_0)) \rangle$ ;
  |  $to\_emit := false$ ;
end
R : {A message  $\langle (n', n'_{old}, M', a') \rangle$  has arrived at  $v_0$ }
begin
  |  $a_{old} := a(v_0)$ ;
  |  $M_{old} := M(v_0)$ ;
  |  $n_{old} := n(v_0)$ ;
  |  $M(v_0) := M(v_0) \cup M'$ ;
  | if  $n(v_0) = 0$  or  $\exists (n(v_0), \ell, \mathcal{N}) \in M(v_0)$  such that  $(\lambda(v_0), N(v_0)) \prec (\ell, \mathcal{N})$  then
  |   |  $n(v_0) := 1 + \max\{n \mid \exists (n, \ell, \mathcal{N}) \in M(v_0)\}$ ;
  |   |  $N(v_0) := \text{replace}(n', n_{old})$ ;
  |   |  $M(v_0) := M(v_0) \cup \{(n(v_0), \lambda(v_0), N(v_0))\}$ ;
  |   | if  $M(v_0) \neq M_{old}$  then
  |     |  $a(v_0) := 0$ ;
  |     |  $A(v_0) := \{(n, p, -1) \mid (n, p) \in N(v_0)\}$ ;
  |     | if  $M(v_0) = M'$  then
  |       |  $A(v_0) := \text{confidence}(n', a', a_{old})$ ;
  |     | if  $\forall (a, p) \in A(v_0), a(v_0) \leq a$  then
  |       |   | construct  $\mathbf{D}'$  from  $M(v_0)$ ;
  |       |   | if  $\mathbf{D}'$  is fibred over  $\mathbf{B}_{\mathbf{G}}$  then
  |       |     |  $a(v_0) := a(v_0) + 1$ ;
  |     | if  $a(v_0) \neq a_{old}$  or  $M(v_0) \neq M_{old}$  then
  |       |   |  $to\_emit := true$ ;
  |     | if  $a_i(v) > |V(G)|$  then
  |       |   | define  $C_{\mathbf{D}'}$ ;
  |       |   |  $n_e := \min\{n(w) \mid w \in C_{\mathbf{D}'}\}$ ;
end

```

---

**Lemma 7.** For each vertex  $v$  and each step  $i$ , if  $M_i(v) = M_{i+1}(v)$  then  $a_{i+1}(v) \geq a_i(v)$ . Moreover, if  $v$  applies the action **S** at step  $i$  and  $j$ , then  $M_i(v) \neq M_j(v)$ .

Let us recall that during the execution of the algorithm, each process  $v$  is able to reconstruct a digraph  $\mathbf{D}(M(v))$  from its mailbox  $M(v)$  (see Proposition 2).

In the following lemma, for each vertex  $v$ , if there exists a step  $j$  in which  $M_j(v) = M$ , we define  $i(M, v)$  as the first step where it holds. At this step, we define a digraph  $\mathbf{H}(M, i)$  as follows. For every arc  $a$  from  $v$

to  $v'$  in  $Dir(\mathbf{G})$ ,  $a$  belongs to  $A(\mathbf{H}(M, i))$  if  $v'$  has heard a message from  $v$  before the step  $i(M, v)$ . We have to prove that while  $\mathbf{H}(M, i) \neq Dir(\mathbf{G})$ , then the execution of the algorithm is not terminated.

From Lemma 7, we show that the confidence level of a vertex allows to know how far from  $v$  the vertices have the same mailbox of  $v$ .

**Lemma 8.** *For every vertex  $v \in V(\mathbf{H}(M, i))$  and each step  $i$ , for every vertex  $w \in V(\mathbf{H}(M, i))$  such that  $dist_{\mathbf{H}(M, i)}(v, w) \leq a_i(v)$ , there exists a step  $j \leq i$  such that  $a_j(w) \geq a_i(v) - dist_{\mathbf{H}(M, i)}(v, w)$  and  $M_j(v) = M_i(v)$ .*

We now prove in the following lemma that once a vertex gets a confidence level greater than the size of the graph then all vertices of the graph have the same mailbox and have a confidence level greater than 0.

**Lemma 9.** *If there exists a vertex  $v$  and a step  $i$  such that  $a_i(v) > |V(G)|$ , then there exists a sub-digraph  $\mathbf{H}$  of  $Dir(\mathbf{G})$  such that  $\mathbf{H}$  is fibred over  $\mathbf{B}_{\mathbf{G}}$ .*

Let  $\mathbf{B}_{\mathbf{G}}$  be the digraph such that  $Dir(\mathbf{G})$  is fibred over  $\mathbf{B}_{\mathbf{G}}$  via a non-trivial fibration relation  $\varphi$  and  $\mathbf{B}_{\mathbf{G}}$  is the minimal base of  $Dir(\mathbf{G})$ .

For every vertex  $v$ , since  $a(v)$  and the number of given identities are bounded by  $|V(G)|$ , we know that any execution of  $\mathcal{M}_e$  terminates. Consider an execution in which some messages are delayed, i.e, there exist isolated processes and some communication links are missing. Thus, every process involved in the computation belongs to a sub-digraph  $\mathbf{H}$  of  $Dir(\mathbf{G})$ . Once the enumeration algorithm is terminated on  $\mathbf{H}$ , every process has the same mailbox and is able to construct a labelled digraph  $\mathbf{D}(M(v))$ . We show that if  $\mathbf{D}(M(v))$  is fibred over  $\mathbf{B}_{\mathbf{G}}$  then  $\mathbf{H} = Dir(\mathbf{G})$ .

**Lemma 10.** *Let  $\mathbf{H}$  be a sub-digraph of  $Dir(\mathbf{G})$  and the digraph  $\mathbf{B}_{\mathbf{G}}$  such that  $Dir(\mathbf{G})$  (resp.  $\mathbf{H}$ ) is fibred over  $\mathbf{B}_{\mathbf{G}}$  via a fibration relation  $\varphi_{\mathbf{G}}$  (resp.  $\varphi_{\mathbf{H}}$ ). If  $x_0$  is the vertex with the maximal view in  $\mathbf{B}_{\mathbf{G}}$ , then  $\varphi_{\mathbf{H}}(v) = x_0 \implies \varphi_{\mathbf{G}}(v) = x_0$ . Moreover, for every vertex  $v \in \mathbf{G}$ ,  $T_{\mathbf{G}}(v) \approx T_{\mathbf{H}}(v)$ .*

Thus, if there exists a vertex  $v$  such that it cannot increase its confidence level then the digraph  $\mathbf{D}(M(v))$  reconstructed from its mailbox  $M(v)$  is not fibred over the minimal base  $\mathbf{B}_{\mathbf{G}}$  of  $Dir(\mathbf{G})$ . Hence, there exist vertices which have not heard all messages emitted by their neighbours. Eventually, they stand for these messages: the algorithm is not terminated.

From Lemmas 9 and 10, if there exists a vertex  $v \in V(G)$  and a step  $i$  in which  $a_i(v) > |V(G)|$ , then  $\mathbf{H}(M(v), i) = \mathbf{G}$ . Moreover, it knows that all the vertices will not change their mailbox anymore. Therefore, for all vertices  $w \in V(G)$ , there exists a step  $i$  where  $a_i(v) > |V(G)|$ .

Hence, every process  $v$  has reconstructed a digraph  $\mathbf{D}(M(v))$ . Thus, there exists a process  $w \in C_{\mathbf{D}}$  with the lowest identity, i.e., such that  $|\varphi^{-1}(w)| = 1$  and  $n(w) < n(w')$  for every  $w' \in C_{\mathbf{D}}$ , which can be declared as the only one *elected* process. Therefore, we have proven the following theorem:

**Theorem 2.** *For every graph  $\mathbf{G}$ , there exists a leader election algorithm on  $\mathbf{G}$  using asynchronous radio communications if and only if the digraph  $Dir(\mathbf{G})$  is minimal for the non-trivial fibration relation  $\varphi$  in which there exists a vertex such that its fibre is trivial.*

## 5 Conclusion

In this paper, we consider a model for computations in radio networks. From this model we have characterized graphs in which we can solve the naming/enumeration and the leader election problems over asynchronous radio communications.

Our characterizations of graphs where we can solve naming or election are similar to the one obtained by Boldi *et al.* [BCG<sup>+</sup>96], even if our model is a priori weaker, since the nodes do not initially know their degree. We also underline the importance of the initial knowledge. In order to give a unique name to each node in fibration-minimal graphs, nodes do not need to initially know their degree if they know the size of the network. On the contrary, this combination of initial informations (each node knows the size but not its degree) is not sufficient to solve the election problem on graphs minimal for the non-trivial fibration relation.



It remains open to determine exactly what initial knowledge about the network is necessary and/or sufficient to solve election in our model.

Our algorithms have a polynomial communication complexity, while the view-based algorithms of Yamashita and Kameda [YK99] and of Boldi *et al.* [BCG<sup>+</sup>96] need that processes exchange messages of exponential size.

## References

- [Ang80] D. Angluin. Local and global properties in networks of processors. In *Proc. of the 12th Symposium on Theory of Computing (STOC 1980)*, pages 82–93, 1980.
- [BB06] B. Balasundaram and S. Butenko. Graph domination, coloring and cliques in telecommunications. *Handbook of Optimization in Telecommunications*, pages 865–890, 2006.
- [BCG<sup>+</sup>96] P. Boldi, B. Codenotti, P. Gemmell, S. Shammah, J. Simon, and S. Vigna. Symmetry breaking in anonymous networks: characterizations. In *Proc. of the 4th Israeli Symposium on Theory of Computing and Systems (ISTCS 1996)*, pages 16–26. IEEE Press, 1996.
- [Bod89] H.L. Bodlaender. The classification of coverings of processor networks. *Journal of parallel and distributed computing*, 6(1):166–182, 1989.
- [BV02] P. Boldi and S. Vigna. Fibrations of graphs. *Discrete Mathematics*, 243(1-3):21–66, 2002.
- [CGG<sup>+</sup>00] B.S. Chlebus, L. Gasieniec, A. Gibbons, A. Pelc, and W. Rytter. Deterministic broadcasting in unknown radio networks. In *SODA*, pages 861–870, 2000.
- [CGÖR00] B.S. Chlebus, L. Gasieniec, A. Östlin, and J. M. Robson. Deterministic radio broadcasting. In *ICALP*, pages 717–728, 2000.
- [Cha05] J. Chalopin. Local computations on closed unlabelled edges: the election problem and the naming problem. In *Proc. of the 31st Conference on Current Trends in Theory and Practice of Informatics (SOFSEM 2005)*, volume 3381 of *Lecture Notes in Computer Science*, pages 81–90. Springer-Verlag, 2005.
- [Chl01] B.S. Chlebus. Randomized communication in radio networks. In P.M. Pardalos, S. Rajasekaran, J.H. Reif, and J.D.P. Rolim, editors, *Handbook of Randomized Computing*, volume I, pages 401–456. Kluwer Academic Publishers, 2001.
- [CM99] I. Cidon and O. Mokryn. Distributed algorithms in multihop broadcast networks. *CoRR*, cs.DC/9909011, 1999.
- [CM04] J. Chalopin and Y. Métivier. Election and local computations on edges. In *Proc. of Foundations of Software Science and Computation Structures, 7th International Conference (FOSSACS 2004)*, volume 2987 of *Lecture Notes in Computer Science*, pages 90–104. Springer-Verlag, 2004.
- [CMZ06] J. Chalopin, Y. Métivier, and W. Zielonka. Local computations in graphs: the case of cellular edge local computations. *Fundamenta Informaticae*, 74(1):85–114, 2006.
- [GMM04] E. Godard, Y. Métivier, and A. Muscholl. Characterization of classes of graphs recognizable by local computations. *Theory of Computing Systems*, 37(2):249–293, 2004.
- [LeL77] G. LeLann. Distributed systems, towards a formal approach. In *Information processing 1977*, pages 155–160. North-Holland, 1977.
- [Maz97] A. Mazurkiewicz. Distributed enumeration. *Information Processing Letters*, 61(5):233–239, 1997.
- [Maz04] A. Mazurkiewicz. Bilateral ranking negotiations. *Fundamenta Informaticae*, 60(1-4):1–16, 2004.
- [Mec07] S. Mecke. Mac layer and coloring. *Algorithms for Sensor and Ad Hoc Networks*, pages 63–80, 2007.
- [Pel07] A. Pelc. Activating anonymous ad hoc radio networks. *Distributed Computing*, 19(5-6):361–371, 2007.
- [San06] N. Santoro. *Design and Analysis of Distributed Algorithms (Wiley Series on Parallel and Distributed Computing)*. Wiley-Interscience, 2006.
- [SSP85] B. Szymanski, Y. Shy, and N. Prywes. Terminating iterative solutions of simultaneous equations in distributed message passing systems. In *Proc. of the 4th Annual ACM Symposium on Principles of Distributed Computing (PODC 1985)*, pages 287–292. ACM Press, 1985.
- [Tan02] A. Tanenbaum. *Computer Networks*. Prentice Hall Professional Technical Reference, 2002.
- [Tel00] G. Tel. *Introduction to distributed algorithms*. Cambridge University Press, 2000.
- [YK96a] M. Yamashita and T. Kameda. Computing functions on asynchronous anonymous networks. *Math. Systems Theory*, 29(4):331–356, 1996.
- [YK96b] M. Yamashita and T. Kameda. Computing on anonymous networks: Part ii - decision and membership problems. *IEEE Transactions on parallel and distributed systems*, 7(1):90–96, 1996.
- [YK99] M. Yamashita and T. Kameda. Leader election problem on networks in which processor identity numbers are not distinct. *IEEE Transactions on parallel and distributed systems*, 10(9):878–887, 1999.

## APPENDIX

### A Proofs of Section 2

*Proof (of Lemma 1).* Let  $\mathbf{D}_1 = (D_1, \lambda_1), \mathbf{D}_2 = (D_2, \lambda_2)$  be two digraphs such that  $(D_1, \lambda_1)$  is fibred over  $(D_2, \lambda_2)$  via  $\varphi$ . It is sufficient to prove the lemma for one particular run  $\rho$  of  $\mathcal{A}$ .

In our model, consider a particular execution for  $\rho$  in  $\mathbf{D}_2$  in which each emitted message from a process  $v$  is followed by the hearing of all its neighbours. Consider a step of this execution: the process  $v$  emits a message in  $\mathbf{D}_2$  and all its neighbours hear the message just after its emission. Let  $\lambda'_2$  be the labelling of  $\mathbf{D}_2$  after this step. One can lift this execution in  $\mathbf{D}_1$  in which every vertex in  $\varphi^{-1}(v)$  emits the same message (not simultaneously and in any order). Then, all emitted messages are heard. We denote  $\lambda'_1$ , the new labelling of  $\mathbf{D}_1$ . Each vertex  $w \in N_{\mathbf{D}_2}(v)$  hears  $k$  messages, with  $k$  depending on the number of arcs  $a \in A(\mathbf{D}_2)$  such that  $s(a) = v$  and  $t(a) = w$ . Since  $\varphi$  is a fibration relation, for every vertex  $w' \in \varphi^{-1}(w)$ ,  $w'$  has  $k$  neighbouring processes in  $\varphi^{-1}(v)$  and hears  $k$  same messages. In this sense,  $\lambda'_1(w') = \lambda'_2(w)$  and labels of all other vertices are not modified. Note that if there exist any self-loops on  $v$ , then there exist arcs  $a \in A(\mathbf{D}_2)$  such that  $s(a) = t(a) = v$ . Once  $v$  has emitted a message,  $\lambda'_1(v) = \lambda'_2(\varphi^{-1}(v))$ . Thereafter once  $v$  has heard this message, we have also  $\lambda'_1(v) = \lambda'_2(\varphi^{-1}(v))$ . Therefore, the digraph  $(D_1, \lambda'_1)$  is fibred over  $(D_2, \lambda'_2)$  via  $\varphi$ .  $\square$

### B Proofs of Section 3

*Proof (of Lemma 3).* The property is obviously true for the vertices that are active at step  $i$ . It is easy to see that, for each vertex  $v$ , we always have  $M_i(v) \subseteq M_{i+1}(v)$ .

For each vertex  $v$  and each step  $i$  such that  $n_i(v) \neq n_{i+1}(v)$ ,  $n_{i+1}(v) = 1 + \max\{n_1; (n_1, \ell_1, N_1) \in M_i(v)\}$  and either  $n_i(v) = 0 < n_{i+1}(v)$  or  $(n_i(v), \lambda(v), N_i(v)) \in M_i(v)$  as shown in Lemma 2 and therefore  $n_i(v) < n_{i+1}(v)$ .

If  $N_i(v) \neq N_{i+1}(v)$  then  $v$  heard a message  $mess = \langle (n', n'_{old}, M') \rangle$  and  $N_{i+1}(v) = code(N_G(v))$ . If  $n'_{old} \notin N_i(v)$ , it means that  $mess$  is the first message heard by  $v$ . Therefore  $max(N_{i+1}(v) \Delta N_i(v)) = n' \in N_{i+1}(v)$ . Consequently,  $N_i(v) \prec N_{i+1}(v)$ .  $\square$

*Proof (of Lemma 4).* Assume that the number  $m$  is known by  $v$  and let  $U = \{(u, j) \in V(G) \times \mathbb{N} \mid j \leq i, n_j(u) = m\}$ . Consider the set  $U' = \{(u, j) \in U \mid \forall (u', j') \in U, N_{j'}(u') \prec N_j(u) \text{ or } N_{j'}(u') = N_j(u) \text{ and } j' \leq j\}$ . It is easy to see that there exists  $i_0$  such that for each  $(u, j) \in U', j = i_0$ . Since  $(m, \ell, N) \in M_i(v)$ , neither  $U$  nor  $U'$  are empty.

If  $i_0 < i$ , the number  $n_{i_0}(u) = m$  of  $u$  was modified at step  $i_0 + 1$  but by maximality of  $(\lambda(u), N_{i_0}(u))$ , the vertex  $u$  could not modify its number. Hence,  $i_0 = i$  and there exists a vertex  $w \in V(G)$  such that  $n_i(w) = m$ .  $\square$

*Proof (of Lemma 5).* We show this claim by induction on  $i$ . At the initial step the assertion is true. Suppose that it holds for  $i \geq 0$ .

If the rule  $I$  is applied by  $v$ , then,  $M_i(v) = (1, \lambda(v_0), \emptyset)$  and trivially, the property holds.

If the rule  $R$  is applied by  $v$ , then,  $v$  heard a message  $mess = \langle (n', n'_{old}, M') \rangle$  from another vertex  $v'$ . Let  $j$  be the step in which  $v'$  emitted this message. We know that  $M' = M_j(v')$ . If  $v$  keeps its number at step  $i + 1$ , then,  $M_{i+1}(v) = M_i(v) \cup M_j(v')$  and the assertion is true by induction hypothesis. Besides, if  $v$  modifies its number, then,  $n_{i+1}(v) = 1 + \max\{n \mid \exists (n, l, N) \in M_i(v) \cup M_j(v')\}$  and  $M_{i+1}(v) = M_i(v) \cup M_j(v') \cup (n_{i+1}(v), \lambda(v), N_{i+1}(v))$ . Consequently, the assertion is true.  $\square$

*Proof (of Lemma 6).*

1. By Lemma 4 and Lemma 5 applied to the final labelling.
2. Otherwise, there exists two neighbours  $v, v'$  such that  $M(v) = M(v')$ . However, since the configuration is final, both  $v$  and  $v'$  have sent their mailboxes to their neighbours and thus  $M(v) = M(v')$ .

3. A corollary of the previous point using Lemma 2.
4. A corollary of the previous property and since neither  $v$  nor  $v'$  need to change its number.
5. Since each neighbour of  $v$  that has the number  $n$  has sent a message with its number, and since all messages have been heard, we know that there exists  $(n', p') \in N_p(v)$  with  $p' > p$ . Moreover, due to the design of the function replace, we know that  $\sum_{(p,n) \in N_p(v)} p$  is bounded by the degree of  $v$ . Consequently, the claim holds.  $\square$

*Proof (of Proposition 2).* We use the notation of Lemma 6. Let  $\mathbf{G} = (G, \lambda)$ .

Consider the graph  $\mathbf{D}$  defined as follows. Its set of vertices is  $V(\mathbf{D}) = \{m \in \mathbb{N} \mid \exists v \in V(G), n_\rho(v) = m\}$ . For any  $m, m' \in V(\mathbf{D})$ , there are  $p$  arcs  $a_{m', m, 1}, \dots, a_{m', m, p}$  from  $m'$  to  $m$  if there exists  $v \in V(G)$  such that  $n_\rho(v) = m'$  and  $(m, p) \in N_\rho(v)$ . From Lemma 6, this is independent of the choice of  $v \in V(G)$ . For every vertex  $v, v' \in V(G)$ , if  $n_\rho(v) = n_\rho(v')$  then  $\lambda(v) = \lambda(v')$  and we can define the labelling  $\eta$  of  $\mathbf{D}$ : for every  $v \in V(G)$ ,  $\eta(n_\rho(v)) = \lambda(v)$ .

Let us recall that  $V(\text{Dir}(\mathbf{G})) = V(G)$  and for all arc  $\{v, v'\} \in E(G)$ , there exist two arcs  $a_{v', v}, a_{v, v'}$  such that  $s(a_{v, v'}) = t(a_{v', v}) = v$  and  $t(a_{v, v'}) = s(a_{v', v}) = v'$ . Moreover, for each  $v \in V(G)$ , the label of  $v$  in  $\text{Dir}(\mathbf{G})$  is the same as in  $\mathbf{G}$ .

It remains to define the homomorphism  $\varphi$  from  $\text{Dir}(\mathbf{G})$  to  $\mathbf{D}$ . For every vertex  $v \in V(G)$ ,  $\varphi(v) = n_\rho(v)$ . For every vertex  $v$  such that  $\varphi(v) = n$ , and for each  $(m, p) \in N_\rho(v)$ , we know from Lemma 6 that there exists  $p$  arcs  $a_1, \dots, a_p \in A(\text{Dir}(\mathbf{G}))$  such that  $t(a_i) = v$  and  $n_\rho(s(a_i)) = m$ . For each  $1 \leq i \leq p$ , let  $\varphi(a_i) = a_{m, n, i}$ .

By definition,  $\varphi$  is a fibration and thus  $\text{Dir}(\mathbf{G})$  is fibred over  $\mathbf{D}$ .  $\square$

*Proof (of Proposition 3).* Let  $\mathbf{G}$  be a labelled graph of size  $n$  with  $m$  edges, maximal degree vertex  $\Delta$  and diameter  $D$ . Consider a run  $\rho$  of the algorithm on  $\mathbf{G}$ . According to Lemmas 4 and 5, we know that each vertex modifies its number at most  $n$  times.

For every vertex  $v$ , since numbers of  $v$  and of its neighbours only increase,  $(n(v), N(v))$  can change  $(d(v)+1)n$  times. When  $v$  modifies its number or its local view, it yields at most the emission of  $O(n)$  messages (because vertices that already have  $(n(v), N(v))$  in their mailbox do not emit this message). Thus, any run of the algorithm needs  $O(mn^2)$  messages. Thus, since each message has the form  $\langle (n, n_{old}, (n', l', N')) \rangle$  and since for every vertex  $v$ ,  $N(v)$  contains at most  $\Delta$  elements with numbers lower than  $n$ , the size of each message is  $O(\Delta \log n)$  bits.

From these previous proofs, for each vertex  $v$ ,  $n(v)$  can be represented with  $\log n$  bits while  $N(v)$  can be represented with  $O(\Delta \log n)$  bits. Since, each vertex only keeps useful informations in its mailbox, there exists at most  $n$  elements  $(n_0, l, N)$  in  $M(v)$  and each of these elements can be represented with  $O(\Delta \log n)$  bits. Hence, one can represent the mailbox of each vertex with  $O(\Delta n \log n)$  bits. Thus, the maximum local memory requirement at any vertex is  $O(\Delta n \log n)$ .  $\square$

## C Proofs of Section 4

*Proof (of Lemma 8).* This lemma can be proved by induction on the distance  $k$  between  $v$  and  $w$  in  $\mathbf{H}(M, i)$ . If  $k = 0$ , the assertion is trivially true. We assume that this assertion holds for every vertex  $v, w$  such that  $\text{dist}_{\mathbf{H}(M, i)}(v, w) \leq k$ .

Consider two vertices  $v, w$  and a step  $i$  such that  $a_i(v) \geq k+1$  and  $\text{dist}_{\mathbf{H}(M, i)}(v, w) = k+1$ . There exists a vertex  $v \in N_{\mathbf{H}(M, i)}(v)$  such that  $\text{dist}_{\mathbf{H}(M, i)}(v, w) = k$ . Consider the last step  $j'$  in which  $v$  increases its confidence level. We know that  $a_{j'}(w) \geq a_{j'}(v) - 1$  and  $M_{j'}(w) = M_j(v) = M_i(v)$ . By induction hypothesis, we deduce that there exists a step  $j < j'$  such that  $a_j(w) \geq a_{j'}(v) - k \geq a_i(v) - (k+1)$  and  $M_j(w) = M_{j'}(v) = M_i(v)$ . Thus, the property is checked for every vertex  $v, w$  such that  $\text{dist}_{\mathbf{H}(M, i)}(v, w) = k+1$ .  $\square$

*Proof (of Lemma 9).* Since  $a_i(v) > |V(G)|$ , from Lemma 8, we know that for every vertex  $w \in V(G)$ , there exists a step  $i_w < i$  such that  $a_{i_w}(w) \geq 1$  and  $M_{i_w}(w) = M_i(v)$ . Assume that there exists a vertex  $w$  such that  $M_i(w) \neq M_{i_w}(w)$ . Let  $j$  be the step in which for the first time, the mailbox  $M_{i_w}(w)$  of a vertex  $w$  has been modified. More precisely, for every vertex  $w \in V(G)$  there exists a step  $j' \geq j-1$  such that

$M_{j'}(w) = M_i(v)$  and there exists a vertex  $w$  such that  $M_{j-1}(w) = M_i(v) \subsetneq M_j(w)$ . Consider the message  $\mathbf{mes} = \langle (n, n_{old}, M, a) \rangle$  heard by  $w$  and processed at step  $j$ . We know that at step  $j'$  in which the message has been emitted by a neighbouring process  $u$ ,  $M_{j'}(u) = M \subseteq M_j(w)$ . Moreover, we know that  $a_{j-1}(w) \geq 1$  and there exists a couple  $(a, p) \in A(w)$  such that  $a \geq 0$ . Thus, there exists a step  $j'' \leq j - 1$  in which the process  $u$  has emitted a message  $\mathbf{mes}' = \langle (n, n_{old}, M_{j-1}(w), a) \rangle$  and then,  $M_{j''}(u) = M_{j-1}(w)$ . Since  $\mathbf{mes}'$  has been arrived before  $\mathbf{mes}$  (let us recall that messages are heard in the same order they were emitted), we know that  $j' \leq j''$ . Thus,  $M_{j''}(u) = M_{j'}(u)$  and  $n_{j''}(u) = n_{j'}(u)$ . Hence, when  $w$  has heard the message  $\mathbf{mes}$ ,  $M(w)$  and  $N(w)$  were not been modified. From Lemma 7,  $n(w)$  was not been modified. We deduce that  $M_j(w) = M_{j-1}(w)$  and for every vertex  $w \in V(G)$ ,  $M_i(w) = M_i(v)$ .  $\square$

*Proof (of Lemma 10).* Since  $\mathbf{H}$  is a subdigraph of  $\mathbf{G}$ , from Remark 2, for each  $v$ ,  $T_{\mathbf{H}}(v) \preceq T_{\mathbf{G}}(v)$ . Since  $\mathbf{H}$  is fibred over  $\mathbf{B}_{\mathbf{G}}$  via  $\varphi$ , for every  $w_0$  in  $\mathbf{B}_{\mathbf{G}}$  that has a maximal view, for every  $v_0 \in \varphi^{-1}(w_0)$ ,  $T_{\mathbf{H}}(v_0)$  is maximal in  $\mathbf{G}$  and thus  $T_{\mathbf{H}}(v_0) = T_{\mathbf{G}}(v_0)$ .

We now prove that for every vertex  $v$  in  $V(\mathbf{G})$ ,  $T_{\mathbf{H}}(v) = T_{\mathbf{G}}(v)$ . Let  $X_0$  be the set of vertices that have a maximal view. Let  $v_0$  be the closest vertex from  $v$  in  $\mathbf{G}$  such that  $T_{\mathbf{G}}(v_0)$  is maximal, and let  $dist_{\mathbf{G}}(v, X_0)$  be the distance from  $v$  to  $v_0$  in  $\mathbf{G}$ . We prove the result by induction on  $dist$ . If  $v \in X_0$ , then we already know the result holds. Otherwise, there exists a neighbour  $u$  of  $v$  such that  $dist_{\mathbf{G}}(u, X_0) = dist_{\mathbf{G}}(v, X_0) - 1$ . By induction, we know that  $T_{\mathbf{G}}(u) = T_{\mathbf{H}}(u)$ , and thus  $u$  has the same degree in  $\mathbf{G}$  and in  $\mathbf{H}$ . Moreover, the multiset of the views of the neighbours of  $u$  should be the same in  $\mathbf{H}$  and  $\mathbf{G}$ . Consequently, if  $T_{\mathbf{H}}(v) \prec T_{\mathbf{G}}(v)$ , there exists another neighbour  $v'$  of  $v$  such that  $T_{\mathbf{G}}(v) \prec T_{\mathbf{H}}(v)$ , which is impossible.  $\square$