

Research Article

Updating Neighbour Cell List via Crowdsourced User Reports: A Framework for Measuring Time Performance

A. Checco ¹, C. Lancia,² and D. J. Leith³

¹University of Sheffield, Sheffield, UK

²Leiden University, Leiden, Netherlands

³Trinity College Dublin, Dublin, Ireland

Correspondence should be addressed to A. Checco; a.checco@sheffield.ac.uk

Received 19 July 2017; Accepted 27 November 2017; Published 21 January 2018

Academic Editor: Stefano Savazzi

Copyright © 2018 A. Checco et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In modern wireless networks deployments, each serving node needs to keep its Neighbour Cell List (NCL) constantly up to date to keep track of network changes. The time needed by each serving node to update its NCL is an important parameter of the network's reliability and performance. An adequate estimate of such parameter enables a significant improvement of self-configuration functionalities. This paper focuses on the update time of NCLs when an approach of crowdsourced user reports is adopted. In this setting, each user periodically reports to the serving node information about the set of nodes sensed by the user itself. We show that, by mapping the local topological structure of the network onto states of increasing knowledge, a crisp mathematical framework can be obtained, which allows in turn for the use of a variety of user mobility models. Further, using a simplified mobility model we show how to obtain useful upper bounds on the expected time for a serving node to gain Full Knowledge of its local neighbourhood.

1. Introduction

Neighbour Cell List Discovery (NCLD) is a core process of modern wireless networks, especially when deployed in an unplanned and decentralised manner like WiFi hotspots and LTE femtocells [1]. In these scenarios, each node needs to independently construct the NCL. Further, appropriate knowledge of network topology, that is, the neighbourhood structure of each node in the network, allows the design of more efficient routing and interference-avoidance algorithms and improved allocation of limited network resources. In a number of common situations, relying on explicit communication or on a central controller may be impractical or even impossible, for instance, when neighbouring devices belong to a different operator. Local knowledge of network topology is enough to produce distributed algorithms for channel allocation in WiFi networks, code selection in small cell networks, and distributed graph colouring and routing and also for problems of joint power and channel allocation optimisation (see Section 2).

Though related to location discovery, the topic of this manuscript is the discovery of existing neighbours without

targeting their actual geographical position. We focus on the process of NCLD via *crowdsourcing*, meaning that the task of detecting and reporting the existence of conflicting neighbours is delegated to users. In this framework, each user periodically reports to the serving node information about the set of neighbouring nodes observed; see, for example, Figure 1. Exploiting User Equipment (UE) measurements is appealing because such technique is easy to implement and virtually cost-free. Nevertheless, the information received from UE measurements is disregarded by the serving node in most implementations [2, Section 7.4.1].

Keeping the NCL updated is fundamental for a number of reasons:

- (i) Neighbouring cells can be added, removed, or temporarily offline.
- (ii) The handover to a new cell might be problematic whenever it is not contained in the NCL of the serving cell.
- (iii) Some cells should not be added to the NCL list because they might reflect spurious measurements, yielding nonreliable handovers.

- (iv) Neighbours with the same PCI should be handled with specific solutions.

With these reasons in mind, this manuscript studies the time T necessary to achieve confident knowledge of the NCL through UE measurements. Estimating T enables optimal tuning of neighbour cell list management schemes. T is also important for the design and deployment of decentralised optimisation schemes. A key example is *self-organisation*, a problem where the network nodes need to optimise their configuration without a central controller, that is, relying on local information only. There exist many fast and efficient decentralised algorithms to self-organise a WiFi/femtocell network; these algorithms are generally fed with a NCL, which needs to be constantly kept up to date. At implementation level, this means that each node needs to periodically estimate with a sufficient level of confidence which nodes of the network are potential conflicting neighbours. The majority of decentralised schemes require that each serving node needs local knowledge of the local neighbourhood [3–6], and any attempt to relax this hypothesis comes at the expense of performance, as shown in Section 2.

In the literature, it is usually assumed that neighbourhood information may be instantaneously acquired [7–13]; that is, the time T is considered negligible. In fact, this assumption may often not be valid, either because it is necessary to listen to the channel long enough to get a high-confidence estimation or because hidden nodes/second-hop NCLs need to be known as well, and thus it is necessary to use communication with users or other nodes to obtain such information. When the time needed by each serving node to update its NCL is larger than the time to execute the optimisation algorithm, a decentralised approach might not be the best solution.

In a framework where the NCL is built via crowdsourced user reports, our main goal is to rigorously characterise T and study its properties and bounds. This is a problem that, to the best of our knowledge, has not yet been addressed in the literature. Our main contributions are the following: (i) the problem of user-reports-based NCLD is stated for the first time through a crisp mathematical formulation; (ii) we introduce a simple mobility model that is useful for gaining insight into those situations where crowdsourcing via user reports is likely to yield the greatest benefit to a decentralised approach; (iii) we show that this model can provide an upper bound on the time to topology discovery; thus it can be used as a design tool (see Section 5.5).

The rest of the paper is organised as follows: in Section 2, we present the related work; then, in Section 3 we show some practical use cases where our approach can be applied. In Section 4, we provide a mathematical model for the discovery process and in Section 5 we define the problem in the context of such model and give some useful bounds. We present simulation results to validate the model and show how it can be used as a network design tool in Section 6. Finally, in Section 7 we draw the conclusions.

2. Related Work

In the field of decentralised algorithm design, it has been shown that local knowledge of network topology is enough to

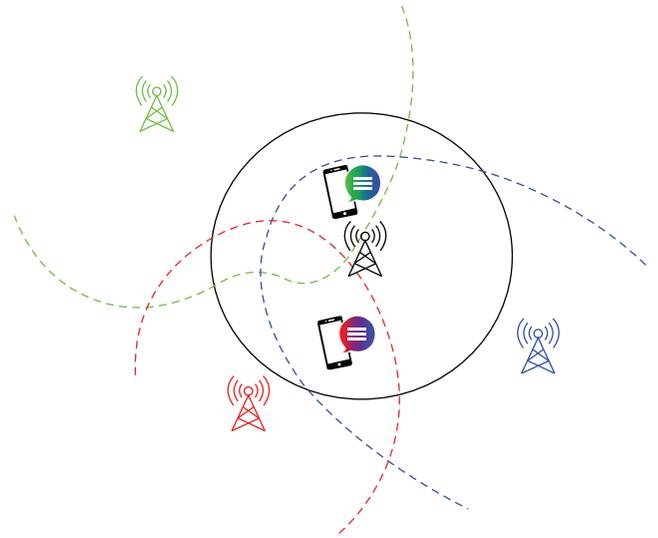


FIGURE 1: Example of NCLD. Each user reports to the serving node which neighbouring nodes they observe.

produce a distributed algorithm for resource allocation; such local knowledge also allows the minimisation of scrambling-code collision and confusion in small cell networks; see [14] and references therein. This knowledge is sufficient and in a certain sense necessary to build efficient algorithms, as the attempts to relax the hypothesis that each serving node needs knowledge of the local neighbourhood will result in an extreme loss of performances [15, 16], which can be prevented only in specific scenarios where the interference model can be described with a simple graph [17].

Perhaps the main motivation of this manuscript is the work of [18], where the authors propose a neighbour cell list management scheme based on the long-term statistics of UE measurements. A key parameter of this scheme is the *forgetting factor* γ , which weighs the longitudinal UE measurements. Such parameter is clearly not trivial to tune, especially in settings, very common in wireless and cellular networking, where instantaneous cell list acquisition cannot be assumed. The optimal tuning γ is only possible by studying the statistical properties of the time T necessary to achieve confident knowledge of the NCL through UE measurements.

The user report function is already available in commercial femtocells [19] and small cells networks, and its implementation for code confusion and interference reduction is recommended in [14, 20].

Crowdsourcing approaches have been investigated for different applications, for example, for estimating both density and number of attendees of large events [21]. Many works pertain to the use of crowdsourcing for NCL discovery. In [22], the use of mobile measurement to update the NCL of macrocells after deployment has been studied: since the intrafrequency reporting function, known as Detected Set Reporting (DSR), is energetically costly for the mobile device, the use of it is suggested only in critical situations where a problem with the current NCL is known. A similar case, where the NCL needs to be updated when a new macrocell

is deployed, is studied in [23]. Other ways to dynamically build the NCL via crowdsourcing are presented in [18, 24]. A similar work applied to WiMax is presented in [25], and a closely related approach for the femtocell case is presented in [26].

With this work, we address the problem of estimating the NCL construction time, which is necessary to assess whether crowdsourcing is effective in a particular network deployment. However, to the best of our knowledge, this problem has not been addressed in the literature yet.

3. Use Cases

We show in this section some timely use cases where our proposed framework can be applied as a network design tool.

3.1. 3G Network Optimisation. In order to provide seamless mobility and a satisfactory service, the optimisation of the handover function is fundamental in modern 3G cellular networks. To achieve that, the construction of a reliable NCL is one of the most critical tasks. While in the past this was achieved by drive and walk testing, the needs to adapt to changes in the network and to reduce the cost require different solutions [12, 18, 22].

The so-called Detected Set Reporting (DSR) is an intrafrequency 3GPP functionality that allows users to report cells not defined in the NCL. In this way, whenever a macrocell detects a problem, or when a new cell is deployed [23], such a function can be activated. The only disadvantage is that such functionality is energetically costly for the mobile device, so its use is recommended for short periods of time and only in critical situations where a problem with the current NCL is known. Therefore, an estimation of the optimal time to keep the DSR active is required. Our work provides an effective framework to make such estimation possible.

3.2. Small Cells Self-Configuration. An important problem that affects the small cells deployment for residential use is code selection. In 3G, base stations have only few scrambling codes available, making the task of selecting the optimal allocation challenging. Moreover, communication with a central controller is discouraged, to avoid signalling overhead. In 4G and 5G, Physical Cell Identity codes and 5G scrambling codes have similar problems.

A fully decentralised algorithm that can converge to the optimal confusion- and collision-free code allocation has been devised in [14]. However, it relies on the assumption that small cells are able to construct their NCL. Unfortunately, small cells are often not able to detect first- and second-hop neighbours reliably due to hidden-node effects and the absence of an efficient sniffing Common Pilot Channel (CPICH) mechanism. A technique to construct the NCL via crowdsourcing has been proposed in [26]. However, the implementation of such a technique would first require the evaluation of the time scale of the NCL construction and its comparison with the time scale of the convergence of the code allocation algorithm.

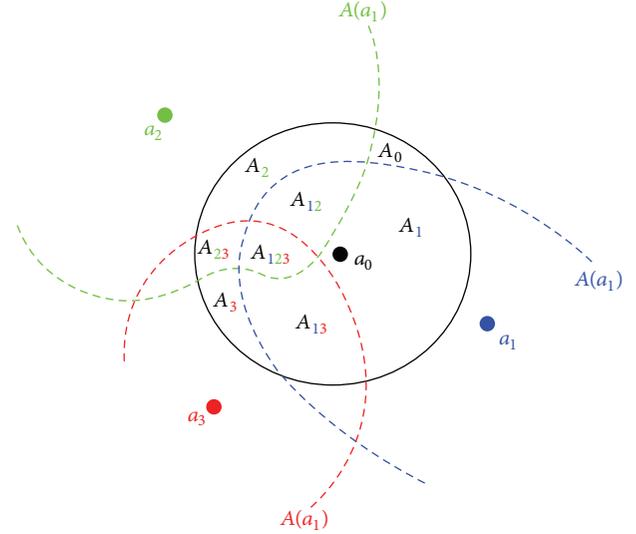


FIGURE 2: Example of tessellation corresponding to scenario of Figure 1. The scenario is modelled with a serving node, a_0 , with three interfering neighbours, a_1 , a_2 , and a_3 ; the coverage area of the serving node, $A(a_0)$, can be tessellated with the sets A_0 , A_1 , A_2 , A_3 , A_{12} , A_{13} , A_{23} , and A_{123} . For simplicity of notation, we will write $A_0 := A_\emptyset$.

4. Neighbour Cell List Discovery Model

Given a set of wireless nodes $\mathcal{A} = \{a_0, \dots, a_N\}$, let $A(a_i) \subset \mathbb{R}^2$ denote the coverage area of access point a_i . Please note that $A(a_i)$ generally depends on the transmission power of a_i and on the radio propagation properties of the medium. We focus on serving access point a_0 and let \mathcal{B} denote the neighbouring nodes that have nonvoid intersection with $A(a_0)$; that is,

$$\mathcal{B} = \{a_i \in \mathcal{A}, i > 0 : A(a_0) \cap A(a_i) \neq \emptyset\}. \quad (1)$$

We will hereafter use the symbol N to denote the cardinality of \mathcal{B} ; that is, $N = |\mathcal{B}|$.

Let $\mathcal{P}(\mathcal{B})$ denote the powerset of \mathcal{B} . A *tessellation* of the area $A(a_0)$ is the collection of tiles $\{A_{\mathbf{i}}\}_{\mathbf{i} \in \mathcal{P}(\mathcal{B})}$ such that

$$A(a_0) = \bigcup_{\mathbf{i} \in \mathcal{P}(\mathcal{B})} A_{\mathbf{i}}, \quad (2)$$

where

$$A_{\mathbf{i}} = \bigcap_{j \in \mathbf{i}} A(a_j) \cap A(a_0) \setminus \bigcup_{j \notin \mathbf{i}} A(a_j), \quad \mathbf{i} \neq \emptyset, \quad (3)$$

$$A_\emptyset = A(a_0) \setminus \bigcup_{\mathbf{i} \in \mathcal{P}(\mathcal{B}) \setminus \emptyset} A_{\mathbf{i}}.$$

In what follows, each element $A_{\mathbf{j}}$ composing the tessellation is referred to as a *tile*, and we will use the vector notation \mathbf{j} to represent a set of neighbouring nodes. Let us consider, for example, $\mathbf{i} = \{a_1, a_2\}$; then, the tile $A_{\mathbf{i}}$ is the portion of $A(a_0)$ that is covered by a_1 and a_2 only; see Figure 2.

Whenever a user is in $A_{\mathbf{j}}$, it will *report* \mathbf{j} to access point a_0 . In other words, a_0 will be aware of the existence of those

neighbouring nodes $a_i \in \mathbf{j}$. The rate of these reports depends on the mobility model assumed (see Section 5).

To keep the model as conservative as possible and to encompass the frequent case of half-duplex nodes, we assume a_0 cannot detect the existence of any neighbour even though a_0 lies in one of the neighbours' coverage area.

Let \mathcal{K}_t denote the *knowledge set* of access point a_0 , that is, the set of neighbours that a_0 is aware of, at time t . Given a sequence of reports $\{\mathbf{j}^1, \dots, \mathbf{j}^t\}$, we have that $\mathcal{K}_t = \bigcup_{i=1}^t \mathbf{j}^i$. \mathcal{K}_t is a sequence of sets that satisfies

$$\mathcal{K}_t = \bigcup_{s=0}^t \mathcal{K}_s; \quad (4)$$

in particular, $|\mathcal{K}_t|$ is nondecreasing in t . Clearly, the knowledge state at time t , \mathcal{K}_t , takes values in $\mathcal{P}(\mathcal{B})$.

Definition 1 (Full Knowledge). Given an integer T and a finite sequence of reports $\{\mathbf{j}^1, \dots, \mathbf{j}^T\}$, the node a_0 is said to have *Full Knowledge (FK)* of its neighbours at time T if

$$\mathcal{K}_T = \bigcup_{s=1}^T \mathbf{j}^s = \mathcal{B}. \quad (5)$$

Remark 2. If a_0 has Full Knowledge (FK) of its neighbours at time T , so it has at all times $T + t$ for $t \geq 0$. In other words, once a_0 has reached FK, it cannot lose it.

Definition 3 (first time to FK). Given a sequence of reports $\{\mathbf{j}^1, \mathbf{j}^2, \dots\}$, the *first time to FK* τ for the node a_0 is the first time the latter reaches FK of its neighbours; that is,

$$\tau := \min \{T \geq 0 \text{ such that } \mathcal{K}_T = \mathcal{B}\}. \quad (6)$$

Remark 4. The characterisation of the first time to FK generally depends on the realisation of a sequence of user reports; this means that τ is a random variable. More precisely, by (6), τ is a *stopping time*; see, for example, [27].

We end the section with a note on the tessellation.

Remark 5. A generic tessellation of \mathcal{B} can be represented as a hypercube $H = \{0, 1\}^N$ by identifying the vertices of H with the tiles A_j that the tessellation is composed of. The number of tiles of a generic tessellation of \mathcal{B} is 2^N as well as the vertices of a hypercube, represented as vectors of size N . The tiles of the tessellation can be mapped onto the vertices of the hypercube by identifying the i th component of the vertices $x \in H$ with $a_i \in \mathcal{B}$. In other words,

$$\begin{aligned} A_j &\longleftrightarrow x \iff \\ x_i &= \mathbb{1}_{\{a_i \in \mathbf{j}\}}, \\ i &= 1, \dots, N, \end{aligned} \quad (7)$$

where $\mathbb{1}$ is the indicator function. We define the *order* of a tile as the number of neighbours a report from that tile would give knowledge of; the number of k th order tiles is $\binom{N}{k}$. A report

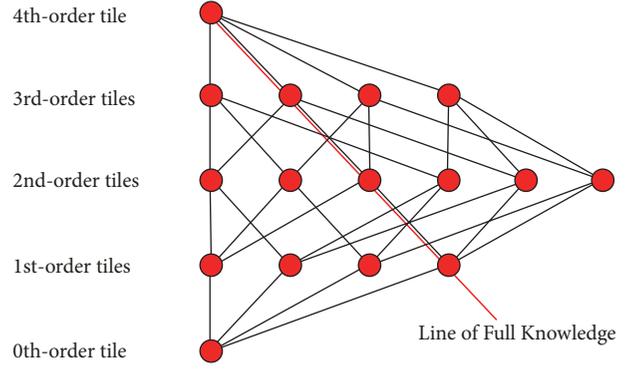


FIGURE 3: Hypercube representation of the tessellation for $N = 4$. There is one zeroth order tile, namely, A^C ; four first-order tiles, A_1, A_2, A_3 , and A_4 ; six second-order tiles, $A_{12}, A_{13}, A_{14}, A_{23}, A_{24}$, and A_{34} ; four third-order tiles, $A_{123}, A_{124}, A_{134}$, and A_{234} ; and a fourth-order tile, that is, A_{1234} .

from a k th order tile is equivalent to k first-order reports. In particular, FK is attained with a report from the N th order tile, or at least two reports from two distinct $(N - 1)$ th tiles, and so on. This property can be graphically represented by what we call the *Line of Full Knowledge*; see Figure 3. The line of FK is clearly not unique (e.g., there are N tiles of order $N - 1$, but only 2 are part of a given line of FK); the aim of Figure 3 is only to illustrate that a sequence of T reports $\{\mathbf{j}^1, \dots, \mathbf{j}^T\}$ is a path on the hypercube H and that FK is attained whenever a line of FK is reached at a time smaller than T .

Since \mathcal{K}_t , the knowledge state at time t , takes on values in the same set $\mathcal{P}(\mathcal{B})$, we can also map the knowledge states on the hypercube H . That is, a sequence of reports $\{\mathbf{j}^1, \mathbf{j}^2, \dots, \mathbf{j}^t\}$ is equivalent to a single report from tile $\bigcup_{s=1}^t \mathbf{j}^s = \mathcal{K}_t$.

We can now define the main problems of this work.

Problem 6 (expected first time to Full Knowledge). Given an access point a_0 , a set \mathcal{B} of neighbours with given position and coverage area, and a sequence of user reports, we want to characterise the expectation of the first time to FK; that is,

$$\mathbb{E}(\tau) = \sum_{t \geq 1} t \mathbb{P}(\tau = t). \quad (8)$$

Obviously, the way the user(s) moves inside the coverage area $A(a_0)$ heavily affects the difficulty of the problem and its answer. However, the formulation of Problem 6 has the great advantage of decoupling the notion of FK from the user mobility model; addressing the mean value of the first time to FK is also an enabler to the estimate of the tail of the distribution of τ —through Markov's inequality, for example. Further, from a numerical point of view, the expected time to FK may be achieved via a Monte Carlo simulation once the set \mathcal{B} and the mobility model in use are fixed.

There may exist cases where it is only necessary to characterise the first time to attain partial knowledge of the local topology. For example, we may be interested in the first moment when the neighbouring nodes that have been already discovered, that is, the elements of the knowledge set \mathcal{K}_t , are

enough to describe a given fraction of the local topology. This idea motivates the following.

Problem 7 (expected first time to δ -knowledge). Let ϱ be a measure over $\mathcal{P}(\mathcal{B})$ and fixed $\delta \in (0, 1]$. Given an access point a_0 , a set \mathcal{B} of neighbours with given position and coverage area, and a sequence of user reports, we want to characterise the expectation of the first time to δ -knowledge $\mathbb{E}(\tau_\delta)$, where

$$\tau_\delta = \min \left\{ T \geq 0 \text{ such that } \frac{\sum_{\mathbf{k} \in \mathcal{P}(\mathcal{K}_T)} \varrho(A_{\mathbf{k}})}{\sum_{j \in \mathcal{P}(\mathcal{B})} \varrho(A_j)} \geq \delta \right\}. \quad (9)$$

When $\delta = 1$ and $\varrho(A_j) > 0$ for each $j \in \mathcal{P}(\mathcal{B})$, Problem 7 is equivalent to Problem 6. Indeed, $\sum_{\mathbf{k} \in \mathcal{P}(\mathcal{K}_T)} \varrho(A_{\mathbf{k}}) / \sum_{j \in \mathcal{P}(\mathcal{B})} \varrho(A_j) \geq 1$ if and only if $\mathcal{K}_T \equiv \mathcal{B}$.

We will hereafter consider the Lebesgue measure $\varrho(A_{\mathbf{k}}) = \|A_{\mathbf{k}}\|$. This leads to the following interpretation: δ -knowledge is attained when the knowledge set \mathcal{K}_t defines for the first time a tessellation that covers a fraction of $A(a_0)$ larger than or equal to δ . Equivalently, τ_δ is the first time when the tiles that would give new information (in the sense that the cardinality of the knowledge set \mathcal{K}_T would increase) cover a fraction of $A(a_0)$ that is smaller than $1 - \delta$.

Remark 8. The concept of δ -knowledge is fundamental in the simulation phase, when we want to know whether user reports can effectively be used to give knowledge of the local topology. Indeed, it is likely that the neighbours a_i whose coverage area do not overlap with $A(a_0)$ save for a nearly negligible portion will be discovered after a very long time; in other words, the leading contribution to $\mathbb{E}(\tau)$ will be represented by the mean first-visit time of the user(s) to $A(a_i)$. Discarding a_i from the picture, the concept of δ -knowledge lets us focus on the quantitative analysis of NCLD; see Section 6.

5. Teleport Mobility

The characterisation of τ , the first time to FK, depends on the assumed user's mobility model: it describes how users enter, exit, and move within $A(a_0)$. The users evolution can then be represented as a pair $U_t = (n_t, X_t)$, where n_t is the number of users that lie in $A(a_0)$ at time t and $X_t = (x_t^1, x_t^2, \dots, x_t^{n_t})$ is a vector with the position of the n_t users. We assume the evolution of U_t to be driven by a discrete-time Markov chain (MC) throughout the paper.

The realisation of $\{U_t\}_{0 \leq t \leq T}$ completely determines the sequence of user reports $\{j^1, \dots, j^T\}$ to the access point a_0 (cf. Remark 4). Since \mathcal{K}_t only depends on \mathcal{K}_{t-1} and U_t , then the bivariate process (U_t, \mathcal{K}_t) is a MC.

It will prove useful to consider a simplified mobility model in which a single user continuously teleports between tiles, without leaving $A(a_0)$ (this model will be extended to many users and to more general models in Section 5.5).

Model 1 (teleport mobility). A single user moves within $A(a_0)$ according to a discrete-time MC taking on values in $\mathcal{P}(\mathcal{B})$. At any time the user cannot abandon the whole region, that is, it is constrained within $A(a_0)$. At each step, the user instantaneously teleports with a probability that is proportional to the measure of the destination tile (note that the actual position within a tile is undefined in this model). The destination tile can also be the same tile of previous step, meaning that the user would remain on the same tile during that discrete-time step. Assuming that all tiles are Lebesgue-measurable plane sets, the transition probabilities are

$$\mathbb{P}(\mathbf{i} \rightarrow \mathbf{j}) = \frac{\|A_j\|}{\|A(a_0)\|}, \quad (10)$$

where $\|\cdot\|$ denotes the Lebesgue measure.

Remark 9. Model 1 greatly simplifies the characterisation of τ , the first time to FK. Indeed, in this mobility model, \mathcal{K}_t is independent of U_t , and the sole process \mathcal{K}_t is hence sufficient to describe the process of gathering knowledge from the user reports. We will hereafter refer to \mathcal{K}_t as the *knowledge chain*.

Assuming Model 1, we can easily describe the process of gathering knowledge from user reports as a discrete-time random walk on the hypercube $H = \{0, 1\}^N$ (which we have introduced in Remark 5); having knowledge of n neighbouring nodes is in fact equivalent to receiving a report from the n th order tile that gives information about all of them.

Let $P(\cdot, \cdot)$ be the transition kernel of the knowledge chain. If $\mathbf{k} \not\subseteq \mathbf{l}$, then (4) guarantees that $P(\mathbf{k}, \mathbf{l}) = 0$ because such transition would mean a loss of knowledge. Conversely, when $\mathbf{k} \subseteq \mathbf{l}$, a transition from \mathbf{k} to \mathbf{l} happens if the user moves to a tile that contains the missing information ($\mathbf{l} \setminus \mathbf{k}$) and does not add more information than that. Therefore,

$$P(\mathbf{k}, \mathbf{l}) = \begin{cases} \sum_{\mathbf{m} \in \mathcal{P}(\mathbf{k})} \frac{\|A_{\{\mathbf{m} \cup (\mathbf{l} \setminus \mathbf{k})\}}\|}{\|A(a_0)\|}, & \text{if } \mathbf{k} \subseteq \mathbf{l}, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

The following result holds.

Lemma 10. *The matrix P is upper triangular.*

Proof. Let us consider the following partial ordering relation among the states:

$$\begin{aligned} \mathbf{k} \leq \mathbf{l} &\iff \\ \mathbf{k} &\subseteq \mathbf{l}. \end{aligned} \quad (12)$$

By (11), $P(\mathbf{k}, \mathbf{l}) \neq 0$ only if $\mathbf{k} \leq \mathbf{l}$. Therefore, any mapping

$$\begin{aligned} \mathcal{P}(\mathcal{B}) \ni \mathbf{l} &\longleftrightarrow \\ l &\in \{1, 2, \dots, 2^N\}, \end{aligned} \quad (13)$$

such that

$$\begin{aligned} \mathbf{k} \leq \mathbf{l} &\iff \\ k &\leq l, \end{aligned} \quad (14)$$

$$P = \frac{1}{\|A(a_0)\|} \begin{pmatrix} \|A_0\| & \|A_1\| & \|A_2\| & \|A_3\| & \|A_{12}\| & \|A_{13}\| & \|A_{23}\| & \|A_{123}\| \\ 0 & \|A_0\| + \|A_1\| & 0 & 0 & \|A_2\| + \|A_{12}\| & \|A_3\| + \|A_{13}\| & 0 & \|A_{23}\| + \|A_{123}\| \\ 0 & 0 & \|A_0\| + \|A_2\| & 0 & \|A_1\| + \|A_{12}\| & 0 & \|A_3\| + \|A_{23}\| & \|A_{13}\| + \|A_{123}\| \\ 0 & 0 & 0 & \|A_0\| + \|A_3\| & 0 & \|A_{13}\| + \|A_{11}\| & \|A_2\| + \|A_{23}\| & \|A_{12}\| + \|A_{123}\| \\ 0 & 0 & 0 & 0 & \|A_0\| + \|A_1\| + \|A_2\| + \|A_{12}\| & 0 & 0 & \|A_3\| + \|A_{13}\| + \|A_{23}\| + \|A_{123}\| \\ 0 & 0 & 0 & 0 & 0 & \|A_0\| + \|A_1\| + \|A_3\| + \|A_{13}\| & 0 & \|A_2\| + \|A_{12}\| + \|A_{23}\| + \|A_{123}\| \\ 0 & 0 & 0 & 0 & 0 & 0 & \|A_0\| + \|A_2\| + \|A_3\| + \|A_{23}\| & \|A_1\| + \|A_{12}\| + \|A_{13}\| + \|A_{123}\| \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (15)$$

5.1. Expected Time to Full Knowledge. Let $\mathbf{k}^* = \{1, 2, \dots, N\}$ be the state of FK. By formula (11), $P(\mathbf{k}^*, \mathbf{k}^*) = 1$. This means that the chain has an absorbing state, and the hitting time of this state is just τ , the first time to FK. Hence, we can compute the expected time to FK simply by

$$\mathbb{E}[\tau] = (I - Q)^{-1} \mathbf{1}, \quad (16)$$

where Q is obtained from P by removing the row and the column relative to state \mathbf{k}^* and $\mathbf{1}$ is the column vector of ones [28]. In a similar way, it is possible to compute the other moments of τ .

Even if $I - Q$ is upper triangular and can be block decomposed, the computation of its inverse may not be affordable when the cardinality of \mathcal{B} grows. In Section 5.4, we will bound the probability of the event $\{\tau > t\}$.

5.2. Expected Time to δ -Knowledge. Regarding Problem 7, we can easily modify matrix P to obtain the expected time to δ -knowledge. Every state $\mathbf{k} \in \mathcal{P}(\mathcal{B})$ such that

$$\sum_{\mathbf{l} \in \mathcal{P}(\mathbf{k})} \frac{\|A_{\mathbf{l}}\|}{\|A(a_0)\|} \geq \delta \quad (17)$$

can be aggregated in the absorbing state, summing the corresponding column of P in the last column, and then eliminating the column and row corresponding to state \mathbf{k} . In this way, it is possible to compute $\mathbb{E}[\tau_\delta]$ using (16).

5.3. Eigenvalues. The following result fully characterises the spectrum of the matrix P .

will put the matrix P into an upper triangular form. In particular, we can order the states by increasing cardinality and in lexicographic order (for N neighbouring nodes, that is, with 2^N different tiles, this would mean the sequence $\{1\}, \{2\}, \dots, \{N\}, \{1, 2\}, \dots, \{N-1, N\}, \dots, \{1, 2, \dots, N\}$). \square

The explicit computation of the whole matrix P using (11) is expensive in general since P is a $2^N \times 2^N$ matrix. However, as stated above, P is upper triangular. In Section 5.3, we show that it is possible to explicitly characterise its spectrum. For the reader's reference, the following equation shows the example of transition matrix P for $N = 3$:

Theorem 11. For $\mathbf{k} \in \mathcal{P}(\mathcal{B})$, the eigenvalues of P have the form

$$\begin{aligned} \lambda_{\mathbf{k}} = \frac{1}{\|A(a_0)\|} &\left(\|A_0\| + \sum_{\substack{\mathbf{l} \subset \mathbf{k} \\ |\mathbf{l}|=1}} \|A_{\mathbf{l}}\| + \dots + \sum_{\substack{\mathbf{l} \subset \mathbf{k} \\ |\mathbf{l}|=m}} \|A_{\mathbf{l}}\| \right. \\ &\left. + \dots + \sum_{\substack{\mathbf{l} \subset \mathbf{k} \\ |\mathbf{l}|=|\mathbf{k}|}} \|A_{\mathbf{l}}\| \right). \end{aligned} \quad (18)$$

Proof. The matrix P being upper triangular by Lemma 10, the entries $P(\mathbf{k}, \mathbf{k})$ are the eigenvalues of the matrix. Let us then imagine to have the knowledge chain in state \mathbf{k} . The only way for the chain to undergo a self-transition ($\mathbf{k} \rightarrow \mathbf{k}$) is that the user reports any combination of neighbouring nodes that have already been discovered. In other words, the knowledge chain undergoes a self-transition if and only if the user reports an element of $\mathcal{P}(\mathbf{k})$. Therefore,

$$\lambda_{\mathbf{k}} = \frac{1}{\|A(a_0)\|} \sum_{\mathbf{l} \in \mathcal{P}(\mathbf{k})} \|A_{\mathbf{l}}\|. \quad (19)$$

Last formula is equivalent to the thesis. \square

Since each eigenvalue is a sum of positive elements, the second-largest eigenvalue $\tilde{\lambda}$ can be obtained by maximising over the tiles of order $N - 1$:

$$\tilde{\lambda} = \max_{\mathbf{k}: |\mathbf{k}|=N-1} \lambda_{\mathbf{k}}. \quad (20)$$

5.4. *Convergence Properties, Bounds.* Using (20), it is possible to obtain the following result.

Lemma 12. *Given $\varepsilon > 0$, let*

$$S(1 - \varepsilon) = \frac{\log \varepsilon}{\log \bar{\lambda}}. \quad (21)$$

Then, $S(1 - \varepsilon)$ reports are sufficient to achieve FK with probability greater than or equal to $(1 - \varepsilon)$.

Proof. Using Lemma 10 and (20) on P ,

$$\mathbb{P}(\tau > t) \leq \bar{\lambda}^t. \quad (22)$$

For a small target tolerance ε of not achieving FK,

$$\begin{aligned} \text{if } t \geq \frac{\log \varepsilon}{\log \bar{\lambda}} \implies \\ \mathbb{P}(\tau > t) \leq \varepsilon. \end{aligned} \quad (23)$$

□

δ -Knowledge Convergence Bounds. Using the same manipulation of the matrix P described in Section 5.2, Lemma 12 in Section 5.4 can be applied to the modified matrix to obtain a bound for the number of steps to have δ -knowledge with high probability.

5.5. *δ -Knowledge and Other-Than-Teleport Mobility.* Model 1 is equivalent to a single user teleporting instantaneously to a random point within the coverage area of the node; time is discrete. Thus, at each time node a_0 receives a report from a location that is sampled from the uniform probability distribution over the coverage area $A(a_0)$. Bearing in mind the numerical characterisation of the first time to δ -knowledge, the teleport model is particularly convenient. This task could be in fact carried out within the Monte Carlo paradigm by simply throwing sufficiently many points at random inside the coverage area $A(a_0)$. In other words, it is possible to numerically study the process through which δ -knowledge is achieved by sampling sufficiently many times a probability density function that is uniform over the coverage area $A(a_0)$.

Model 1 may prove itself unsatisfactory in a real life scenario. The main problem is that if we generate a sequence of user reports according to it, any two elements of the sequence are independent, whereas in general they are not. In each mobility model where the trajectory taken by the user is physically feasible, the user positions communicated by two successive reports are in fact correlated due to the motion constraints.

Let us imagine that a single user travels inside the coverage area $A(a_0)$ according to an unknown mobility model, and let $T(t)$ be the trajectory taken by the user. Sampling the trajectory at equally spaced discrete times, we obtain an embedded sequence of user locations, which correspond to an embedded sequence of user reports. Next, we can analyse the sequence and understand after how many steps δ -knowledge has been reached. By multiplying this number of steps by the time lapse between two consecutive reports

(inter-report time), the time to δ -knowledge can be obtained for that particular realisation of the user-reports sequence. Finally, the procedure above can be repeated sufficiently many times to estimate with a Monte Carlo method the expected time necessary for a_0 to reach δ -knowledge.

As mentioned above, in a general mobility model it is likely that two successive user reports are correlated. These correlations may decay as the inter-report time grows larger and larger. As an example, let us imagine that a single user travels inside the coverage area $A(a_0)$ according to a MC. Let π be the equilibrium probability measure of the chain and let t_{mix} be the mixing time of the chain, that is, the time needed for the chain to reach equilibrium. If the inter-report time is chosen comparable to t_{mix} then the time lapse between two successive reports will be sufficient for the MC to forget the past trajectory; in other words, the correlations between consecutive reports will be negligible. As a consequence, the user locations will be independently drawn from the probability measure π , and the matrix P describing the knowledge evolution will become

$$P(\mathbf{k}, \mathbf{l}) = \begin{cases} \sum_{\mathbf{m} \in \mathcal{P}(\mathbf{k})} \pi(A_{\{\mathbf{m} \cup \{\mathbf{l}\}\}}), & \text{if } \mathbf{k} \subseteq \mathbf{l}, \\ 0, & \text{otherwise.} \end{cases} \quad (6')$$

Therefore, the formulation and the results developed in Sections 5.1–5.4 are still valid if we consider a single-user mobility model based on a MC, provided that the time lapse between two consecutive reports is of the order of the mixing time of the chain. Under the assumption that user reports are sent at a frequency comparable with the inverse mixing time of the mobility MC, we can compute an upper bound on the time to δ -knowledge. Any reporting rate higher than $1/t_{\text{mix}}$ will in fact still guarantee that a_0 achieves δ -knowledge of its neighbourhood in at most $\mathbb{E}[\tau_\delta] \cdot t_{\text{mix}}$ seconds on average (recall that $\mathbb{E}[\tau_\delta]$ is measured in number of reports).

5.5.1. *Multuser Scenario.* We end this section by briefly mentioning a straightforward application of Model 1 in a multuser scenario. Let us imagine that n users may enter, move within, and exit $A(a_0)$ according to a hidden mobility model. We assume that n is a very large number and that it is possible to statistically characterise the stationary user-density by means of a probability measure π over $A(a_0)$. At each time every user may independently send a report with a very small probability p . Then, the number of reports received by a_0 in a given time interval is approximately Poissonian and the time lapse between two successive reports is exponential with parameter $\lambda = np$. Next, let $m = \mathbb{E}[\tau]$ be the expected time to FK, expressed in number of reports, returned by (6') and (16); the expected time to achieve FK is the expectation of the first time for a Poisson process of parameter λ to hit the state m . A practical example for this kind of scenario is presented in Section 5.6.2.

5.6. Examples

5.6.1. *Femtocells Deployment for Residential Use.* Regarding the use case of femtocell self-organisation presented in

Section 3.2, each serves a very small number of devices. Using data of typical residential densities and coverage areas, a statistic of the tessellation can be devised. If it is possible to establish a time \tilde{T} after which the user position can be considered as drawn from a uniform distribution, then $S(\delta)$ is an upper bound of the time to δ -knowledge for all the inter-report times smaller than or equal to \tilde{T} .

5.6.2. Cells Deployed in Congested Areas. Opposite to the previous example, cells deployed in congested places like a mall have an extremely large basin of potential users. However, in situations where users main interest is other than connecting to the Internet, it is reasonable to expect the single-user reporting-activity to be rather sporadic. Therefore, the Poissonian approximation that we have mentioned at the end of Section 5.5 may be applicable. In this case, characterising the time to achieve δ -knowledge is possible through a statistic of the typical (or worst case) tessellations.

6. Simulations

6.1. Teleport Model on Random Positioned Nodes. In this section, we offer a preliminary assessment of the possibility of using the machinery developed so far in real applications. To this purpose, we developed a simulation framework in MATLAB and studied a scenario where 8 nodes are positioned on a plane at random according to a uniform (bivariate) probability distribution, that is, *uniformly at random*. Each node has a circular coverage area of the same size. We considered 350 different configurations, with the constraint that the coverage area of a_0 has nonvoid intersection with the coverage area of the remaining nodes, meaning that FK is achieved as soon as all 7 neighbours are reported to a_0 . We compute the tessellation of each configuration using a classical Monte Carlo sampler. For each of these 350 configurations, we computed the expected time to 0.9-knowledge $\mathbb{E}[\tau]$ together with the number of steps sufficient to guarantee 0.9-knowledge with 90% confidence, that is, $S(0.9)$. The inter-report time being fixed during this first experiment, the amount of time in seconds to achieve 0.9-knowledge is directly proportional to the number of steps just evaluated.

Figure 4 displays the empirical probability mass function of these two quantities. $\mathbb{E}[\tau]$ is centred around 10 steps, while $S(0.9)$ is shifted on higher values, as expected being an upper bound. Figure 5 shows the empirical cumulative distribution function of $\mathbb{E}[\tau]$ and $S(0.9)$. We see that 16 steps are sufficient to achieve 0.9-knowledge for nearly all scenarios (95%), while we need 22 steps using $S(0.9)$. We also notice that the bound obtained from (21) is a conservative estimation, because it uses only the second-largest eigenvalue $\tilde{\lambda}$. Indeed, it takes into account only the slowest way to reach the desired knowledge, while the problem has a rich combinatorial structure that cannot be completely captured by (22).

Roughly speaking, a user moving at 0.5 m/s according to a random walk model, and providing *at least* one report every hour, can guarantee the node will have 0.9-knowledge with high probability in less than 5 h and in less than two hours

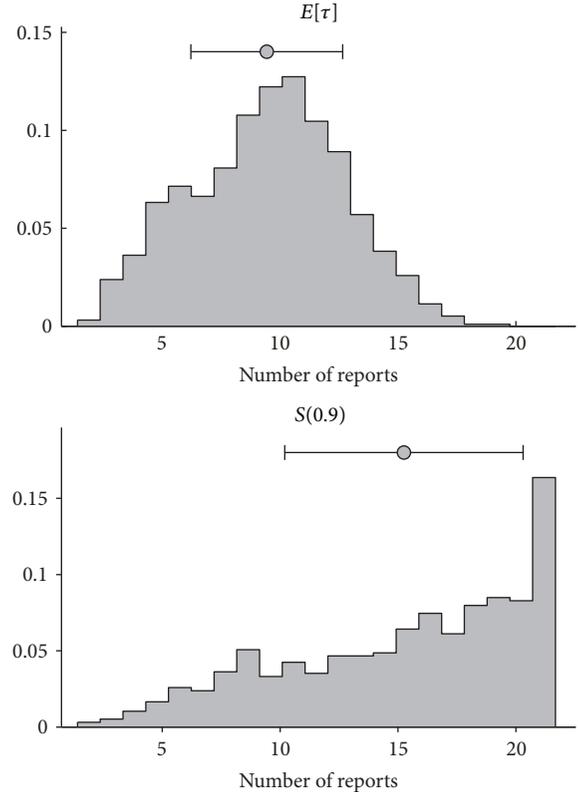


FIGURE 4: Empirical probability mass function of the expected time to 0.9-knowledge, and the number of steps to have 0.9-knowledge with 90% confidence, for the teleport model on random positioned nodes. Since the inter-report time is fixed, the simulation time is directly proportional to the number of reports.

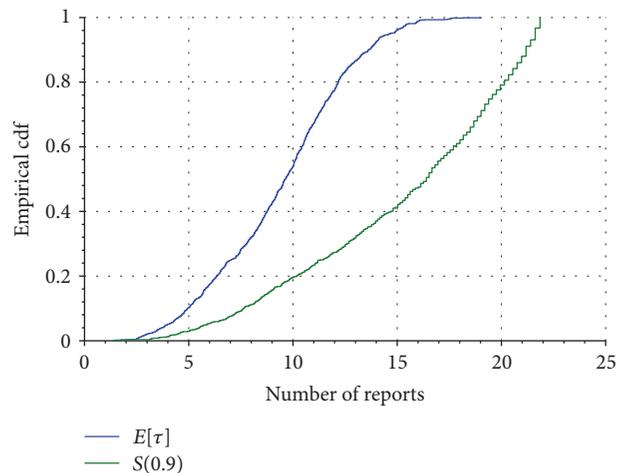


FIGURE 5: Empirical cumulative distribution function of the expected time to 0.9-knowledge, and the number of steps to have 0.9-knowledge with 90% confidence, for the teleport model on random positioned nodes. Since the inter-report time is fixed, the simulation time is directly proportional to the number of reports.

if reports are sent *at least* every 15 minutes (see next section for a more detailed analysis on the interaction between

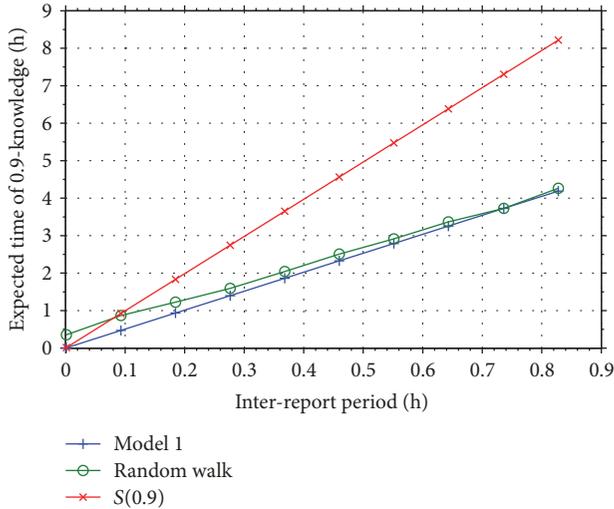


FIGURE 6: Empirical mean 0.9-knowledge time (in hours) of a random walk versus inter-report period, compared with Model 1 and its bound $S(0.9)$ in a femtocell grid of 8 nodes.

report frequency and our bound). If the local topology is not typically expected to change often, these are acceptable times.

To summarise, simulation on random scenarios show that our proposed bound can be used to estimate the time to δ -knowledge. Using realistic values, the expected time to δ -knowledge is reasonably small.

6.2. Random Walk on a Grid. In order to investigate and confirm the ideas of Section 5.5, we simulated the reports sent with different inter-report times by a random walker that moves within $A(a_0)$ under the condition of reflective boundary and compared this mobility model with Model 1 (see Section 5) for a set of 8 nodes positioned as described at the beginning of this section.

In Figure 6, we let the inter-report time increase and compare the average time to achieve 0.9-knowledge according to both the random walk (green line) and the teleport model (blue). We see that if the inter-report time is sufficiently large, the empirical mean time to achieve 0.9-knowledge for the random walk model is well approximated by that of Model 1.

We assume typical femtocell parameters, that is, that coverage radius is 50 m and that the user does a step in a grid of 2.5 m every 5 s. Figure 6 also shows that when reports are sent each 6 min or less, the time to 0.9-knowledge is smaller than 1 h, but at such high frequency the bound $S(0.9)$ (red line) is not valid anymore. The reason why more reports than Model 1 are needed in the case of high-frequency reports is the following: since the inter-report time is short, it is likely that many reports will be sent from the same tile; that is, the knowledge chain will undergo many self-transitions.

It is important to notice that the inter-report times used in Figure 6 are far from the theoretical order of magnitude of the random walk mixing time. Yet, Figure 6 suggests that, for a family of scenarios, it should be possible to determine the value of the inter-report time such that the average time

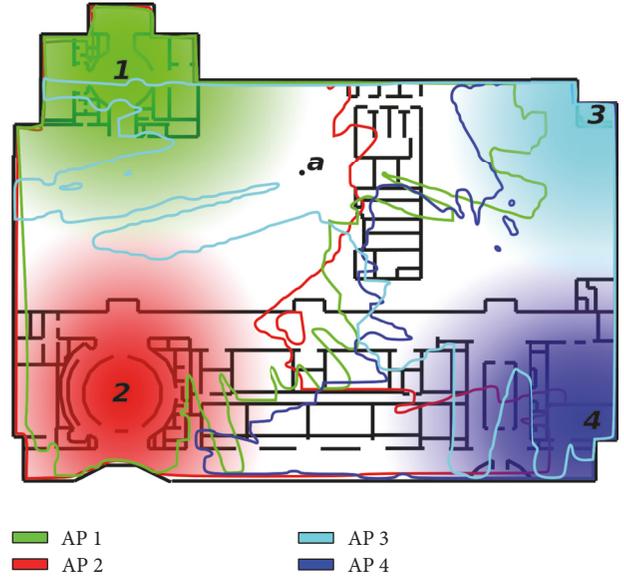


FIGURE 7: Coverage areas at Hynes convention centre. The coloured lines delimit the extension of the coverage areas. Base stations are transmitting at 2.1 GHz with a power of 34 mW. Point a lies in the tile A_{123} .

to achieve 0.9-knowledge may be well predicted by Model 1. Once that value of the inter-report time is found, the value of $\mathbb{E}[\tau_{0.9}]$ returned by Model 1 may serve as an upper bound to the actual time to achieve 0.9-knowledge when smaller inter-report times are implemented.

To summarise, simulation on random walks corroborate the analysis of Section 5.5

6.3. A Realistic Scenario. A received power map for 4 base stations in the Hynes convention centre have been generated using the Wireless System Engineering (WiSE) [29] software, a comprehensive 3D ray tracing based simulation package developed by Bell Laboratories. Base stations are assumed transmitting at a frequency of 2.1 GHz with a power of 34 mW. We assume there is a macrocell that covers the whole building, and we estimate its time to Full Knowledge. As before, a Monte Carlo simulation has been made to estimate the tessellation, and then the expected time to δ -knowledge has been computed using a teleport mobility model (Model 1), as explained in Section 5.2.

Figure 7 shows the corresponding coverage areas when the power detection threshold is -70 dBm. Although the shape of the coverage areas and their intersection is much more complex than the simple scenario depicted in Section 6.1, it is still possible to construct the tessellation by considering which coverage areas each spatial point lies in. For example, point a lies in the coverage area of nodes 1, 2, and 3, so it belongs to the tile A_{123} .

Figure 8 displays the expected time to δ -knowledge, $\mathbb{E}[\tau_\delta]$, when δ is varied. We notice a step-function-like behaviour, with a new step that is added every time a new state becomes absorbing, as explained in Section 5.2.

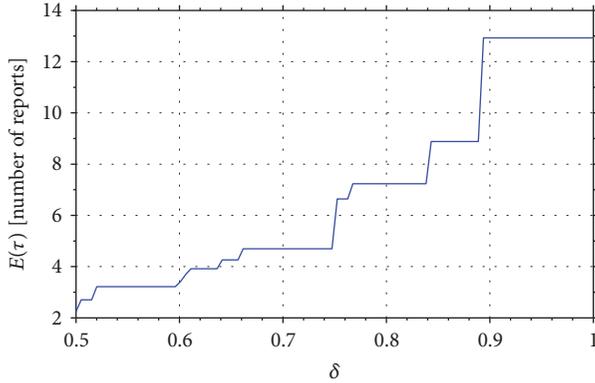


FIGURE 8: Expected time to δ -knowledge, $E[\tau_\delta]$, using Model 1 in Hynes convention centre for different values of the parameter δ . The inter-report time is set to the very same value used for Figures 4 and 5.

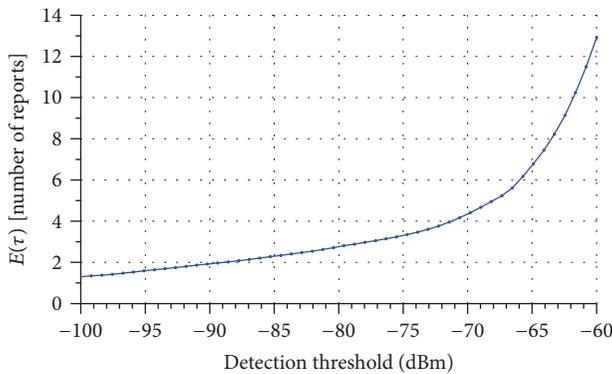


FIGURE 9: Expected time to FK, $E[\tau]$, using Model 1 in Hynes convention centre, for different values of the user-detection threshold. The inter-report time is set to the very same value used for Figures 4 and 5.

Figure 9 shows the behaviour of $E[\tau]$, the expected time to FK, when the user-detection threshold varies from a very conservative value of -60 dBm to a more realistic one of -100 dBm. When the users are more sensitive, the coverage areas, and specifically the higher order tiles, are bigger, leading to better performance. In particular, we see that an average of 14 steps are enough to achieve FK.

To summarise, these results seem to confirm that the values obtained placing random nodes with circular coverage areas in Section 6.1 are compatible with real world scenarios, so the use of statistics obtained from macroscopic parameters as densities of deployment and distribution of coverage radii can be used as a tool to bound the time to δ -knowledge.

7. Conclusions

In this paper, we have introduced the problem of user-reports-based Neighbour Cell List Discovery and provided a crisp mathematical formulation of it for a simple mobility model. We have also shown that such mobility model can be effectively used as an upper bound for a wide range of

mobility models when the user-reports frequency is lower than the inverse mixing time of the Markov chain of the actual mobility model. Additionally, we have provided a useful method to estimate the time to δ -knowledge when the problem is too complex to be solved exactly.

Simulations on random scenarios with typical small cells parameters show that the expected number of reports in order to have a high degree of knowledge of the local topology is very small. Roughly speaking, a user moving at 0.5 m/s according to a random walk model, and providing at least one report every hour, can guarantee the serving node will have 0.9-knowledge with high probability in less than 5 h, and in less than two hours if reports are sent at least every 15 minutes. Since we do not expect the network topology to be affected by high network dynamics, these are acceptable times for the problems of interest. We encourage the adoption of the presented framework to assess the possibility of employing crowdsourced user reports in other self-configuration problems, comparing the time to δ -knowledge with the expected time to convergence of a given decentralised algorithm.

Simulations in more realistic scenarios show that the bounds obtained are compatible with the ones obtained from statistics on random scenarios with similar parameters. This seems to confirm that the use of statistics obtained from macroscopic parameters, such as densities of deployment and distribution of coverage radii, can be used as a tool to bound the time to δ -knowledge.

In conclusion, we provide a useful tool to estimate the time to NCL construction, which is fundamental to assess whether a decentralised algorithm can be employed in a given network scenario.

Conflicts of Interest

The authors declare that there are no conflicts of interest to disclose regarding the publication of this paper.

Acknowledgments

The authors would like to thank Anna Zakrzewska, from Bell Laboratories, Alcatel Lucent Ireland, for her insights and useful suggestions.

References

- [1] Z. Xiao, T. Li, W. Ding, D. Wang, and J. Zhang, "Dynamic PCI allocation on avoiding handover confusion via cell status prediction in LTE heterogeneous small cell networks," *Wireless Communications and Mobile Computing*, vol. 16, no. 14, pp. 1972–1986, 2016.
- [2] Technical Specification Group Radio Access Network, "Home Node B (HNB) Radio Frequency (RF) requirements (FDD)," 3rd Generation Partnership Project, TR 25.967, 2011.
- [3] Ö. Johansson, "Simple distributed $\Delta+1$ -coloring of graphs," *Information Processing Letters*, vol. 70, no. 5, pp. 229–232, 1999.
- [4] F. Kuhn and R. Wattenhofer, "On the complexity of distributed graph coloring," in *Proceeding of the twenty-fifth annual ACM symposium on Principles of distributed computing (PODC '06)*, pp. 7–15, ACM, Denver, Colo, USA, July 2006.

- [5] M. Luby, "Removing randomness in parallel computation without a processor penalty," in *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science*, pp. 162–173, IEEE, White Plains, NY, USA, October 1988.
- [6] M. Szegedy and S. Vishwanathan, "Locality based graph coloring," in *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing (STOC '93)*, pp. 201–207, ACM, San Diego, Calif, USA, May 1993.
- [7] H. Olofsson, S. Magnusson, and M. Almgren, "A concept for dynamic neighbor cell list planning in a cellular system," in *Proceeding of the Seventh IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, (PIMRC '96)*, pp. 138–142, IEEE, Taipei, Taiwan.
- [8] S. Magnusson and H. Olofsson, "Dynamic neighbor cell list planning in a microcellular network," in *Proceedings of the ICUPC 97 - 6th International Conference on Universal Personal Communications*, pp. 223–227, San Diego, Calif, USA.
- [9] V. M. Nguyen and H. Claussen, "Efficient self-optimization of neighbour cell lists in macrocellular networks," in *Proceedings of the 21st International Symposium on Personal Indoor and Mobile Radio Communications, (PIMRC '10)*, pp. 1923–1928, Istanbul, Turkey, September 2010.
- [10] M. Amirijoo, P. Frenger, F. Gunnarsson, H. Kallin, J. Moe, and K. Zetterberg, "Neighbor cell relation list and measured cell identity management in LTE," in *Proceedings of the NOMS 2008 - IEEE/IFIP Network Operations and Management Symposium: Pervasive Management for Ubiquitous Networks and Services*, pp. 152–159, Salvador, Bahia, Brazil, April 2008.
- [11] R. Atawia, M. El Azab, T. Elshabrawy, and M. Ashour, "Ranked overlapping coverage based construction of efficient neighboring cell list for GSM/UMTS cellular networks," in *Proceedings of the International Conference on Communications and Information Technology, (ICCIT '12)*, pp. 254–259, Hammamet, Tunisia, June 2012.
- [12] D. Kim, B. Shin, D. Hong, and J. Lim, "Self-configuration of neighbor cell list utilizing E-UTRAN NodeB scanning in LTE systems," in *Proceedings of the 7th IEEE Consumer Communications and Networking Conference, (CCNC '10)*, pp. 1–5, IEEE, Las Vegas, Nev, USA, January 2010.
- [13] D. Aziz, A. Ambrosy, L. T. W. Ho, L. Ewe, M. Gruber, and H. Bakker, "Autonomous neighbor relation detection and handover optimization in LTE," *Bell Labs Technical Journal*, vol. 15, no. 3, pp. 63–84, 2010.
- [14] A. Checco, R. Razavi, D. J. Leith, and H. Claussen, "Self-configuration of scrambling codes for WCDMA small cell networks," in *Proceedings of the IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications, (PIMRC '12)*, pp. 149–154, IEEE, Sydney, NSW, Australia, September 2012.
- [15] K. R. Duffy, C. Bordenave, and D. J. Leith, "Decentralized constraint satisfaction," *IEEE/ACM Transactions on Networking*, vol. 21, no. 4, pp. 1298–1308, 2013.
- [16] A. Checco and D. J. Leith, "Learning-based constraint satisfaction with sensing restrictions," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 5, pp. 811–820, 2013.
- [17] A. Checco and D. J. Leith, "Fast, Responsive Decentralized Graph Coloring," *IEEE/ACM Transactions on Networking*, vol. 25, no. 6, pp. 3628–3640, 2017.
- [18] Y. Watanabe, Y. Matsunaga, K. Kobayashi, H. Sugahara, and K. Hamabe, "Dynamic neighbor cell list management for handover optimization in LTE," in *Proceedings of the IEEE 73rd Vehicular Technology Conference (VTC '11-Spring)*, pp. 1–5, IEEE, Yokohama, Japan, May 2011.
- [19] P. Sapiano, "Discovering neighbouring femto cells," US Patent EP2 214 434," <http://www.freepatentsonline.com/EP2214434A1.html>.
- [20] J. Edwards, "Implementation of network listen modem for WCDMA femtocell," in *Proceedings of the IET Seminar on Cognitive Radio and Software Defined Radio: Technologies and Techniques*, IET, London, UK, September 2008.
- [21] F. M. Naini, O. Dousse, P. Thiran, and M. Vetterli, "Opportunistic sampling for joint population size and density estimation," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2530–2543, 2015.
- [22] D. Soldani and I. Ore, "Self-optimizing neighbor cell list for UTRA FDD networks using detected set reporting," in *Proceedings of the IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, pp. 694–698, IEEE, Dublin, Ireland, April 2007.
- [23] F. Parodi, M. Kylväjä, G. Alford, J. Li, and J. Pradas, "An automatic procedure for neighbor cell list definition in cellular networks," in *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, (WOWMOM '07)*, pp. 1–6, IEEE, Espoo, Finland, June 2007.
- [24] M. R. Hasan, M. T. Kawser, and M. R. Islam, "An automatic GSM neighbor cell list update procedure enhancing SON in LTE," in *Proceedings of the 6th International Conference on Electrical and Computer Engineering, (ICECE '10)*, pp. 143–146, IEEE, Dhaka, Bangladesh, December 2010.
- [25] J. Li and R. Jäntti, "On the study of self-configuration neighbour cell list for mobile WiMAX," in *Proceedings of the International Conference on Next Generation Mobile Applications, Services and Technologies, (NGMAST '07)*, pp. 199–204, IEEE, Cardiff, UK, September 2007.
- [26] Z. Becvar, M. Vondra, and P. Mach, "Dynamic optimization of neighbor cell list for femtocells," in *Proceedings of the 77th Vehicular Technology Conference, (VTC Spring '13)*, pp. 1–6, IEEE, Dresden, Germany, June 2013.
- [27] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov chains and mixing times*, AMS, 2009.
- [28] J. G. Kemeny and J. L. Snell, *Finite Markov chains: with a new appendix "Generalization of a fundamental matrix"*, Springer-Verlag, Berlin, Germany, 1976.
- [29] S. J. Fortune, D. M. Gay, B. W. Kemighan, O. Landron, R. A. Valenzuela, and M. H. Wright, "WISE design of indoor wireless systems: practical computation and optimization," *IEEE Computational Science & Engineering*, vol. 2, no. 1, pp. 58–68, 1995.

