Continuous monitoring in the dynamic sensor field model

Carme Alvarez, Josep Diaz, Dieter Mitsche, Maria Serna

Continuous Monitoring in the Dynamic Sensor Field Model*

C. Àlvarez[1], J. Díaz[1], D. Mitsche[2], and M. Serna[1]

[1] LSI Dept. Universitat Politècnica de Catalunya, Barcelona.
alvarez,diaz,mjserna@lsi.upc.edu
[2] MA4 Dept. Universitat Politècnica de Catalunya, Barcelona.
dmitsche@upc.edu

Abstract. In this work we consider the problem of continuously monitoring a collection of data sets produced by sensors placed on moving or static targets. We propose a modification of the static sensor field model (SSSF) [2] to model computation in a dynamic network of tiny artifacts. We consider the presence of movement in two ways, the mobility of the communication devices and data mobility. The mobility of devices is simulated by a dynamic communication graph. Data mobility, is due to measurements performed by sensing devices that are not placed on fixed positions but attached to mobile agents. Accordingly, we introduce additional performance measures the traveled distance and the gathering period. We study the *Continuous Monitoring* problem providing bounds on performance for algorithms that use mobility in different ways.

1 Introduction

The use of networks of heterogeneous tiny artifacts is becoming a key ingredient in the technological development of our society. The study of such systems involves several and very different areas of computing. The general sensing setting can be described by two elements: the observers (end users) and the phenomenon, the entity of interest to the observers that is monitored and analyzed by a network with sensors. We concentrate here on the computational issues, extending the *Static Sensor Field* model (SSSF) [2], to scenarios in which dynamicity is of relevance.

The computational system arising from the ad-hoc computation network point of view has been modeled by combining the notion of distributed data streams [5] with classic distributed approaches to solve problems on particular topologies [6] as the SSSF model [2]. The sensor field model, captures some characteristic differences of networks with

* Partially supported by the ICT Program of the European Union under contract number 215270 (FRONTS). The first, second and, fourth authors were also supported by the Spanish project TIN-2007-66523 (FORMALISM).

sensors, it is composed by *actuator devices*, which communicate one to the other and also they can measure and signal the environment. The SSSF model assumes that those devices synchronize at barriers marking rounds, in a similar way to the BSP model [7]. During a computation round, a device accesses the received messages and the data provided by the environment, performs some computation, and finally sends messages to its neighbors and to the environment.

In this paper we continue the study of computational issues for networks of tiny artifacts in the presence of mobility. We analyze two potential sources of mobility: The *passive mobility of the targeted data*, where we assume that a set of sensors is attached to mobile agents and that the collection of input data streams is not originated in a fixed location, and the *active mobility of the network devices* which are able to move in order to obtain readings from far away sensors. This model is different from the asynchronous ad-hoc mobility models surveyed in [3] This model is different The fundamental features of the *Dynamic Sensor Field* (DSSF) model introduced in this paper are the following: A device is able to receive readings from any sensor in its sensing range, instead of being attached to one sensor. At the same time that the device performs its local computation, it may move. Apart from the worst case performance measures of interest for the SSSFs, as latency, message number, o message length, etc. (see [2]) we consider the traveled distance or the gathering period. We analyze the *Continuous Monitoring* problem in which at every period we are interested in reporting an aggregate measure obtained from one reading from each sensor continuously. We propose different static and dynamic sensor fields for solving this problem, when the data and the device mobility follow different mobility patterns. Our objective is to understand the trade-offs between complexity measures due to mobility.

As part of the analysis, we provide an upper bound on the number of steps for a walker being detected by a nearest device. The proof uses a nice coupling argument with two random walks on the truncate integer line. Due to lack of space proofs and algorithms are delayed to the appendix.

2 The Dynamic Sensor Field Model

In the following, the notation is taken from [2, 1]. A *data stream* w is a sequence of data items $w = w^1 w^2 \dots w^i \dots$ possibly infinite. For any $i \geq 1$, $w[i]$ denotes the i-th element of w. For any i, j, $1 \leq i \leq j$, $w[i, j]$ denotes the subsequence of w composed by all data items between the i-th and j-th positions. For any $n \geq 1$, an *n-data stream* $\mathbf{w}$ is an n-tuple of data streams; $\mathbf{w} = (w_1, \dots, w_n)$. For any $i \geq 1$, $\mathbf{w}[i]$ denotes the n-tuple composed by

all the i-th elements of each data stream, $\mathbf{w}[i] = (w_1[i], \ldots, w_n[i])$. For any i, j s.t. $1 \leq i \leq j$, $\mathbf{w}[i, j] = (w_1[i, j], \ldots w_n[i, j])$.

In a Static Sensor Field the data items in a data stream were assumed to be produced as readings of some sensor placed in a fixed location and attached to a device in the same position. However, the data stream abstraction allows us to consider that any of the data items can be obtained at different locations at different time steps. Observe that, additional information, like location of the target at the moment of the reading, could be attached to the data items. In this paper we consider a set W of g data streams from the sensors together with a collection N of n devices that can move according to the network computation. The sensors either do not move at all or each of them moves following an independent random mobility pattern.

We use the standard notation for graphs. A *communication graph* is a directed graph $G = (N, E)$. Each $k \in N$ is associated to a device. Each edge $(i, j) \in E$ specifies that device i can send messages to device j. In a *Dynamic Sensor Field* the communication graph might change during the computation. Let $G_t = (N, E_t)$ denote the communication graph at time t. Due to mobility, the subset of input data items accessible by a device may change along time. Let $D_t \subseteq N \times W$ denote the data stream accessibility relation at time t. We denote by $(k, \alpha) \in D_t$ the event that sensor α can be detected by device k at time t.

We assume that all devices are able to receive information from the environment (input data stream) and send information to the environment (output data stream). Moreover each device executes its own process, communicates with their actual neighbors (devices associated to adjacent nodes) and, if required, changes location. All the devices work in a synchronous way, at each time step they receive data from their neighbors and from the environment, apply their own transition function changing in this way their actual configuration, possibly move, and send data to their neighbors and to the environment. We assume that the devices can change position while they are performing their local computation.

A Dynamic Sensor Field (DSSF) is a tuple $\mathcal{F} = (N, W, U, V, X, (Q_k, \delta_k)_{k \in N})$ where

- N is the set of devices and W the set of sensors.
- U is the alphabet used to represent the input data streams.
- V is the alphabet used to represent the output data streams.
- X is the alphabet used to communicate among devices. $U, V \subseteq X$. We denote by data items the elements of alphabets U and V and by

communication items (or items) the elements of X. Each $m \in X^*$ is called message or packet.

- (Q_k, δ_k) defines for each device associated to a node $k \in N$ (device k) its set of local states and its transition function, respectively.

The *local computation of each device k in $\mathcal{F}$* is defined by (Q_k, δ_k) and depends on the communication with its neighbors and with the environment. Given a device k and $t \geq 0$, let us denote by $I_t(k) = \{i \in N \mid (i,k) \in E_t\}$, $O_t(k) = \{j \in N \mid (k,j) \in E_t\}$ and $S_t(k) = \{\alpha \in W \mid (k,\alpha) \in D_t\}$. The transition function δ_k depends on its local state $q_k \in Q_k$, the communication items received by k from $i \in I_{t-1}(k)$, and the data items that k receives from sensor $\alpha \in S_{t-1}(k)$. δ_k provides the communication items sent by device k to $j \in O_t(k)$, and the data item that k sends to the environment. Formally, $\delta_k : Q_k \times (X^*)^n \times (U^*)^g \longrightarrow Q_k \times (X^*)^n \times V$.

There are two differences with the definition of the Static Sensor Field model. The first one is that the set of neighbors at the beginning and at the end of step t could be different. We assume that synchronization takes place after the local computation and the position change are performed. The second difference is that a device can access data items from more than one data stream, all those within range of targets. In the Static Sensor Field model each device is associated to a unique data stream. Thus, we require a modification in the definition of the transition function with respect to the Static Sensor Field model.

A *computation* of $\mathcal{F}$ is a sequence $\mathbf{c}^0, \mathbf{d}^1, \mathbf{c}^1, \mathbf{d}^2, \ldots, \mathbf{c}^{t-1}, \mathbf{d}^t, \mathbf{c}^t, \ldots$, where $\mathbf{c}^0 = (q_k^0)_{k \in N}$ is the n-tuple of the initial local states of the n devices, and for each $t \geq 1$, $\mathbf{c}^t = (q_k^t)_{k \in N}$ is the n-tuple of the local states after t computation steps. $\mathbf{d}^t = (d_k^t)_{k \in N}$ represents the input/output data of the t-th computation step. In particular, for device k the input/output data of the t-th step is represented by $d_k^t = ((x_{ik}^t)_{i \in I_{t-1}(k)}, (u_\alpha)_{\alpha \in S_{t-1}(k)}^t, (y_{kj}^t)_{j \in O_t(k)}, v_k^t)$.

The *stream behavior of* $\mathbf{c}^0, \mathbf{d}^1, \mathbf{c}^1, \mathbf{d}^2, \ldots, \mathbf{c}^{t-1}, \mathbf{d}^t, \mathbf{c}^t, \ldots$ *of* $\mathcal{F}$ is defined as $(\mathbf{u}, \mathbf{v})$, where $\mathbf{u} = (u_\alpha)_{\alpha \in W}$ is the tuple composed by the input data streams, $u_\alpha = u_\alpha^1 u_\alpha^2 \ldots u_\alpha^t \ldots$ and $\mathbf{v} = (v_k)_{k \in N}$ is the tuple composed by the output data stream of each device $v_k = v_k^1 v_k^2 \ldots v_k^t \ldots$ We say that *the sensor field $\mathcal{F}$ computes* the tuple of output data streams $\mathbf{v} = (v_k)_{k \in N}$ given the tuple of input data streams $\mathbf{u} = (u_k)_{k \in N}$. Note that $\mathbf{u}$ and $\mathbf{v}$ have in general infinite length, but the behavior of a DSSF is defined in terms of all the finite prefixes of $\mathbf{u}[1,t]$ and $\mathbf{v}[1,t]$. However, each device will output only one data item ($\mathbf{v}[t]$) per step.

We define the *function $f_\mathcal{F}$ computed by $\mathcal{F}$*, for any given pair of data streams $\mathbf{u}$ and $\mathbf{v}$ and any $t \geq 1$ as $f_\mathcal{F}(\mathbf{u}[1,t]) = \mathbf{v}[1,t]$ iff $\mathcal{F}$ computes

$\mathbf{v}[1,t]$ given $\mathbf{u}[1,t]$.
A function $f_{\mathcal{F}}$ *is computed by a sensor field* $\mathcal{F}$ *with latency* d if for any tuple of data streams $\mathbf{u}$, and for all $t \geq 1$, $f_{\mathcal{F}}(\mathbf{u}[1,t+d])[t+d] = f(\mathbf{u}[1,t])[t]$.

We consider the following worst case complexity measures on the computation of a DSSF.

- *Size*: The number of devices that take part in the computation.
- *Time*: The maximum number of operations.
- *Space*: The maximum memory space used by any device.
- *MessageLength*: The maximum number of data items sent in a message.
- *MessageNumber*: The maximum number of sent messages.
- *Distance*: The maximum distance traversed by a device.

Here the maximum is taken over all devices and steps. In general we analyze these complexity measures with respect to the *Size* of the communication graph. We denote by $\mathcal{T}(n)$ the *Time*, by $\mathcal{S}(n)$ the *Space*, by $\mathcal{L}(n)$ the *MessageLength*, by $\mathcal{M}(n)$ the *MessageNumber* and, by $\mathcal{D}(n)$ the *Distance*.

Computational problems that are susceptible of being solved by sensor fields can be stated in the following way [2]:
Sensing Problem Π: Given an n-tuple of data streams $\mathbf{u} = (u_k)_{1\leq k\leq n}$ for some $n \geq 1$, compute an m-tuple of data streams $\mathbf{v} = (v_k)_{1\leq k\leq m}$ for some $m \leq n$ such that $R_\Pi(\mathbf{u}[1,t], \mathbf{v}[1,t])$ is satisfied for every $t \geq 1$. R_Π is the relation that output data streams have to satisfy given the input data streams. A function f between data streams is *consistent* with a relation R when for every pair of data streams $\mathbf{u}$ and $\mathbf{v}$, and every $t \geq 1$, if $f(\mathbf{u}[1,t]) = \mathbf{v}[1,t]$ then $R(\mathbf{u}[1,t], \mathbf{v}[1,t])$.
A DSSF solves the problem Π (with latency d) if $f_{\mathcal{F}}$ computed by $\mathcal{F}$, (with latency d) is consistent with relation R_Π.

3 The continuous monitoring problem

We are interested in solving problems in which it is needed to monitor continuously a wide area in which data is obtained at fixed or mobile locations. This implies "sensing locally" and "informing locally" about environmental phenomena, for instance the average temperature. When data is static the problem has been formulated in [1] as follows:
Average Monitoring: Given g data streams $(u_k)_{1\leq k\leq g}$ for some $g \geq 1$, compute m data streams $(v_k)_{1\leq k\leq m}$ such that $v_k[t] = (u_1[t]+\cdots+u_g[t])/g$.

In the above formulation a problem arises if the data is originated in mobile targets or when the devices that have to collect the data can move. In those situations, the precondition that the network has access

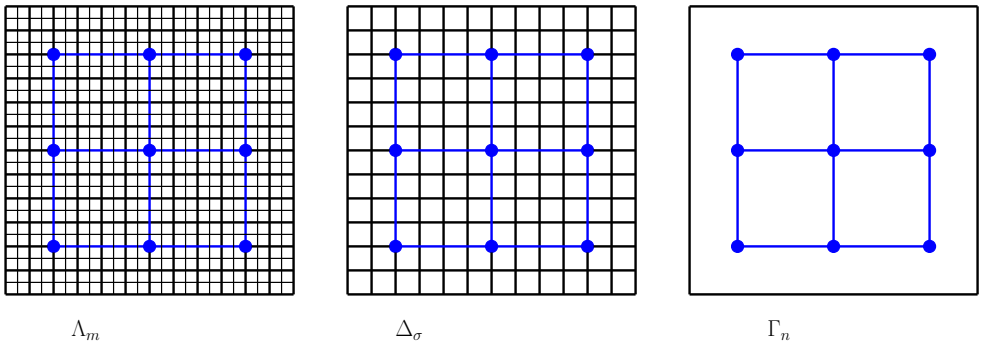

Fig. 1. The three fundamental grids embedded in a terrain $\mathcal{T}$ with $n = 3$, $\sigma = 13$ and $m = 25$.

to all the sensors at any time step might not be possible. Therefore, to monitor continuously a wide area where the g targets move, we relax this condition and require only to get a reading from any sensor inside a reporting period:

Continuous Monitoring: Given a set of g mobile data streams $(u_\alpha)_{1\leq\alpha\leq g}$ for some $g \geq 1$, compute m data streams $(v_k)_{1\leq k\leq m}$ such that, for some $p > 0$, any $1 \leq k \leq m$, and $t > 0$, $v_k[tp] = (u_1[t_1] + \cdots + u_g[t_g])/g$ for some $(t-1)p \leq t_1, \ldots, t_g < tp$.

We refer to p as the *gathering period*. We analyze the complexity of several dynamic sensor fields for the problem when devices and sensors act according to the following scenario.

We assume that the data of interest is accessible in a predetermined square shaped area discretized as a grid. The devices have two associated ranges, a *sensing range* s and a *communication range* r. We assume that the three squared grids (mobility, sensing and communicating) are embedded in the terrain (see Figure 1). The *mobility grid* Λ_m is formed by $m \times m$ nodes that serves as reference positions for the movement of the targets with attached sensors and the computing devices. We assume that sensors and devices stop at grid nodes labeled by coordinates (i, j), $1 \leq i, j \leq m$, and they move following paths on the grid. W.l.o.g assume the distance among to neighboring nodes in Λ_m is a unit length. As a subgrid of Λ_m, we have the *sensing grid* Δ_σ of size $\sigma \times \sigma$, where $2s$ is the distance between nodes in Δ_σ. As subgrid of Δ_σ we have the *communicating grid* Γ_n, with $n \times n$ nodes, where r is the distance among nodes in Γ_n. In the case that $r \leq 2s$ we have that Γ_n is also a sensing grid, in this case we

take $\sigma = n$. Therefore, we can assume that $n \leq \sigma \leq m$. Let $\mathcal{T}(n, \sigma, m)$ denote the scenario.

By placing σ^2 devices, one in each node of Δ_σ, we can detect in one step any signal originated in a sensor placed at any position of Λ_m. Therefore, the continuous monitoring problem can be solved with gathering period 1, using an algorithm for the average monitoring for the bidirectional grid. Therefore from [2] we have the following result,

Proposition 1. *There is a sensor field that solves the continuous monitoring problem terrain $\mathcal{T}(n, \sigma, m)$ and g sensors with $N = \sigma^2$ devices , latency σ, gathering period* 1*, $\mathcal{T}(N) = \mathcal{L}(N) = \mathcal{S}(N) = \mathcal{O}(g)$, $\mathcal{M}(N) = 2$ and, $\mathcal{D}(N) = 0$.*

In the following sections, in order to analyze the trade-off between size, latency and gathering period due to mobility, we propose a DSSF. The sensor fields are designed with two parts. The *gathering* part solves the problem of obtaining a reading from any sensors and will determine the gathering period. The *averaging* part computes the average of the measures taken during a gathering period. As we will see later on, the gathering part requires more steps than the averaging parts. Therefore the devices, just after the first gathering period finishes, run in parallel both algorithms. When both finalize the process is repeated with the new gathered data. The computing devices are arranged either as a line or as a grid.

4 Continuous monitoring of static data

In the static data setting we assume that input data streams are originated from some positions in the grid Λ_m and that devices move on top of Γ_n. Our first solution, for the particular case $n = \sigma$, is the Line sweeping sensor field (see appendix). Devices are arranged as a line, initially placed on the bottom row of the communicating grid and move upwards until they reach the top row. Every time that the sweeping line of sensors reaches the top or the bottom row of Γ_n the network has collected at least one reading from each sensor. Thus we have that $p = n$. The averaging phase requires also n steps, by running a "send to center" and broadcast protocol on the line. Both phases can be run together after the first gathering period finishes. The Line sweeping algorithms solves the continuous monitoring problem only when $n = \sigma$. For the case in which $n < \sigma$, devices have to move closer to the sensors to obtain readings. For doing so in a reasonable time we have to use more sensors than before. We use the sensing grid Γ_n as a mobility grid for the devices. Finally, we assign to a node in Γ_n the part of the partition of the sensing grid surrounding it, its

surveillance area. This assignment is done in such a way that pieces have the same size and with the property that if we place the devices in the bottom left corner of the surveillance they form a communicating grid.

Our second algorithm, the Surveillance grid, places n^2 devices initially in the bottom-left corners of the surveillance area. For the gathering phase the devices follow a *snake* walk covering all the nodes in the assigned subgrid, synchronously. Once the devices reach the final point in the walk they walk backwards towards the initial position. The averaging part is an extension of the averaging for the line, data is collected by the central line. The central node in the central line computes the average, finally the average is broadcasted to the devices. The number of devices can be reduced to n, increasing the gathering period. The *surveillance strip* sensor field arranges the devices in the form of a sensing line. Each device receives as surveillance area a vertical strip. Again each device follows the same snake-like walk covering the assigned strip. The following theorem presents the results obtained for the algorithms in the previous discussion.

Theorem 1. *The table below, presents the resources bounds of Line sweeping, Surveillance grid and, Surveillance strip for solving the continuous monitoring problem on a terrain $\mathcal{T}(n, \sigma, m)$ and g sensors at unknown but fixed positions. For the two surveillance's algorithms, we require $n < \sigma$ and for the sweeping algorithm $\sigma = n$,*

Algorithm	N	*latency*	*gathering period*	$\mathcal{T}(N)$	$\mathcal{L}(N)$	$\mathcal{S}(N)$	$\mathcal{M}(N)$	$\mathcal{D}(N)$
Line Sweeping	n	n	n	$O(g)$	$O(g)$	$O(g)$	2	r
Survei. grid	n^2	n	$2\max\{(r/2s)^2, n\}$	$O(g)$	$O(g)$	$O(g)$	2	$2s$
Survei. strip	n	n	$nr/2s$	$O(g)$	$O(g)$	$O(g)$	2	$2s$

5 Continuous monitoring of dynamic data

For the case of dynamic data we assume a mobility pattern of the targets based on the *walkers* model introduced in [4]. We keep the term *walker* to refer to the moving targets with attached sensors. We consider a set of g walkers $\mathcal{W}$ moving on the mobility grid Λ_m, under the following random mobility model.

Initially g walkers are sprinkled uniformly at random on the m^2 vertices of the grid. At each step, every w_i, not on the boundary, chooses with probability $\frac{1}{4}$ one of the four possible directions and makes a step in the chosen direction. If w_i is in the corner, it chooses with probability $\frac{1}{2}$ any of the two possible directions, and if it is touching the boundary in one dimension only, w_i chooses with probability $\frac{1}{2}$ the only available

direction in the dimension touching the boundary, and with probability $\frac{1}{4}$ the other two directions in the perpendicular dimension.

We propose three different sensor fields for solving the continuos monitoring problem when the input data streams follow the walker mobility model presented before. Again we assume that devices are aware of the network topology which will be either a line or a grid.

For the case in which $n = \sigma$, we consider the Central line sensor field in which n devices are placed on the central line of Γ_n and remain there. Devices collect data until the gathering period finalizes and combine this protocol with averaging protocol for the line. The crucial part of the analysis requires an analysis of the number of steps needed to finalize the gathering period. For doing so we consider the weakest detection model: a sensor is detected by a device if it passes through the position $(i, \frac{m}{2})$ for some i.

For a fixed $w \in \mathcal{W}$, let T_w be the random variable counting the number of steps it takes to detect walker w and denote by T the random variable counting the number of steps to detect all walkers. We prove an upper bound on $\mathbf{E}[T]$.

Lemma 1. $\mathbf{E}[T] \leq g(m^2/2 - 2m + 2)$.

Proof. Consider some arbitrary walker w, and w.l.o.g. assume w starts on a position (i, j) for some $j \leq \frac{m}{2}$. Consider the walk on the mobility grid as the following random walk on the truncated integer line with $m/2$ positions: denoting by $p_{i,i+1}$ the probability to go from position i to $i + 1$, set $p_{i,i+1} = p_{i,i-1} = \frac{1}{4}$ for any $i = 2, \ldots, \frac{m}{2} - 1$, $p_{i,i} = \frac{1}{2}$ for any $i = 2, \ldots, \frac{m}{2} - 1$, $p_{1,2} = \frac{1}{2}$, $p_{1,1} = \frac{1}{2}$, $p_{\frac{m}{2},\frac{m}{2}} = 1$. Let U_i be the random variable counting the number of steps it takes to hit position $\frac{m}{2}$ in this line, starting from position $i \leq \frac{m}{2}$. Then, for any $j \leq \frac{m}{2}$, $p_{j,j+1}$ is equal to the probability of the walker w in position (i, j) to go to $(i, j+1)$ for any $1 \leq i \leq m$, and thus, in particular, $\mathbf{E}[T_w] \leq \mathbf{E}[U_1]$. We have $\mathbf{E}[U_1] = 2+\mathbf{E}[U_2]$, and by forward substitution we obtain $\mathbf{E}[U_i] = 4i-2+\mathbf{E}[U_{i+1}]$ for $2 \leq i \leq \frac{m}{2} - 1$, and $\mathbf{E}\left[U_{m/2}\right] = 0$. By backward substitution, this yields $\mathbf{E}[U_1] = m^2/2 - 2m + 2$, and thus $\mathbf{E}[T_w] \leq m^2/2 - 2m + 2$. Now, $\mathbf{E}[T] = \mathbf{E}[\max_i T_i] \leq \mathbf{E}[\sum_{i=1}^{g} T_i] \leq g(m^2/2 - 2m + 2)$.

Therefore we have:

Theorem 2. *The sensor field* Central line *solves the continuous monitoring problem on a terrain $\mathcal{T}(n, n, m)$ and g mobile sensors, with n devices, latency n, expected gathering period $gm^2/2$, $\mathcal{T}(n) = \mathcal{L}(n) = \mathcal{S}(n) = \mathcal{O}(g)$, $\mathcal{M}(n) = 2$ and, $\mathcal{D}(n) = 0$.*

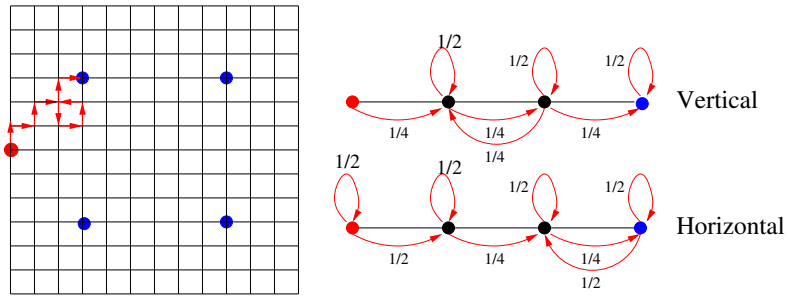

Fig. 2. An illustration of the coupled walks on the grid and on the two lines.

In Lemma 3 in the appendix we provide a lower bound on the expected gathering for the Central line sensor field showing that, even for constant g, it is of the same asymptotic order as the upper bound.

For the general case we consider the Communicating grid sensor field that places a device in any node of Γ_n. Again devices gather data until the gathering period finalizes and combine this protocol with the averaging protocol for the grid. Assuming wlog that $2n$ divides $m-1$, the sensors are placed on positions $(\frac{m-1}{2n}+1+i\frac{m-1}{n}, \frac{m-1}{2n}+1+j\frac{m-1}{n})$, for $0 \leq i \leq n-1, 0 \leq j \leq n-1$. Define by $M := \frac{m-1}{n}$ as the maximum (grid) distance between a sensor and its closest device $\mathcal{S}$. Again we assume the weakest detection nodel: a sensor w is detected by a device, if w passes through the position of that device. Define T_w as the rv counting the number of steps until w is detected and denote by T the rv counting the number of steps until all $w \in \mathcal{W}$ are detected. We provide upper and lower bounds to $\mathbf{E}[T]$, which are polynomial in M.

Lemma 2. $\mathbf{E}[T] \leq 2gM^{20}$.

Proof. Consider a walker w at position (i,j), whose distance to $\mathcal{S}$ is M (see Fig. 2) At every step, w moves horizontally or vertically with prob. $\frac{1}{2}$. We couple the walk of w on the grid as follows: At each step w decides whether the next move is vertical or horizontal, with probability $\frac{1}{2}$. Then, with prob. $= \frac{1}{2}$ w moves up or down (left or right), unless the distance of w to its $\mathcal{S}$ in that dimension is either 0 or $M/2$, in which case the distance increases or decreases by 1. So w's movements (up-down and right-left) are coupled by two random walks on a line with vertices $\{M/2, \ldots, 0\}$. Let S_0 be the rv counting the number of steps to hit 0 in the vertical (horizontal) direction, and let $S_{M/2}$ be the rv counting the steps to hit $M/2$. Observe that $\mathbf{E}[S_0] \leq M^2$ and $\mathbf{E}\left[S_{M/2}\right] \leq M^2$. Using Markov's

inequality we see that if in both directions $\geq \frac{M^{12}}{2}$ steps are made, with probability at least $\geq 1 - \frac{4}{M^{10}}$, in that period both random walks visit 1 or more times 0 and $M/2$. Consider M^8 consecutive blocks of M^{12} steps. The event that in all these blocks both walks always visit both ends happens with probability $\geq 1 - \frac{4}{M^2}$, . Hence, for one block, with prob. $\geq 1 - \frac{4}{M^{10}}$ there is a *meeting* point where at the same step, both walks have the same position from 0. As $\mathbf{E}[S_0] \leq M^2$, $\exists t \in 1, \ldots, 2M^2$ s.t. the probability that after exactly t steps the random walk is in position 0 is $\geq \frac{1}{M^3}$. The value of t depends on the position of the meeting, and this t is the same for both random walks. Moreover, as both random walks are independent, the probability that after exactly t steps, both are at 0, is at least $\geq \frac{1}{M^6}$. Consider M^8 consecutive blocks (each with M^{12} steps), which start at a meeting, and assume that among the $2t$ steps following the meeting, exactly t are horizontal and t are vertical steps. Let $\mathcal{C}$ to the event that in at least one of the blocks both walks simultaneously end in position 0. Then, $\mathbf{Pr}\left[\bar{\mathcal{C}}\right] \leq (1 - \frac{1}{M^6})^{M^8} \leq e^{-M^2}$.

By Chernoff, with prob. $\geq 1 - \Theta(e^{-\frac{1}{3}M^{2/3}})$, w on the grid starting in an arbitrary position, out of the $2t \leq 4M^2$ steps, $t \pm 2M^{4/3}$ will be vertical and $t \pm 2M^{4/3}$ will be horizontal steps. So, with prob. $\geq \Theta(\frac{1}{M^{4/3}})$ exactly t out of the next $2t$ steps are horizontal. Consider M^8 consecutive blocks of length M^{12}, and let R be the rv counting the number of times, where for the appropriate t, among the $2t$ steps after the meeting, exactly t horizontal steps are made. Then $\mathbf{E}[R] \geq M^{20/3}$, and since all the consecutive blocks are independent, $\mathbf{Pr}\left[R \leq e^{-\Theta(M^{20/3})}\right]$. Let $\mathcal{D}$ be the event that $R \geq \frac{1}{2}M^{20/3}$. Combining the previous expressions, we obtain $\mathbf{E}[T_w] \leq \mathbf{E}[T_w|(\mathcal{F} \wedge \mathcal{D} \wedge \mathcal{C})] + \mathbf{Pr}\left[\bar{\mathcal{F}} \cup \bar{\mathcal{D}} \cup \bar{\mathcal{C}}\right]\mathbf{E}[T_w]$. Using the previous expression together with the facts that $\mathbf{E}[T_w|(\mathcal{F} \wedge \mathcal{D} \wedge \mathcal{C})] \leq M^{20}$, and $\mathbf{Pr}\left[\bar{\mathcal{F}} \cup \bar{\mathcal{D}} \cup \bar{\mathcal{C}}\right] \leq \mathbf{Pr}\left[\bar{\mathcal{F}}\right] + \mathbf{Pr}\left[\bar{\mathcal{D}}\right] + \mathbf{Pr}\left[\bar{\mathcal{C}}\right]$, we get

$$\mathbf{E}[T_w] \leq ((M^{20} + 5M^{18})(\sum_{i=0}^{\infty}(\frac{5}{M^2})^i) \leq 2M^{20}.$$

Therefore, $\mathbf{E}[T] = \mathbf{E}[\max_i T_i] \leq \mathbf{E}[\sum_{i=1}^{g} T_i] \leq 2gM^{20}$.

Therefore we have:

Theorem 3. *The sensor field* Communicating grid *solves the continuous monitoring problem for g mobile sensors on a terrain $\mathcal{T}(n, \sigma, n)$, with $N = n^2$ devices, latency $2n$, expected gathering period $2g\left(\frac{m-1}{n}\right)^{20}$, $\mathcal{T}(N) = \mathcal{L}(N) = \mathcal{S}(N) = \mathcal{O}(g)$, $\mathcal{M}(n) = 2$ and, $\mathcal{D}(n) = 0$.*

In Lemma 4 in the appendix we provide a lower bound of the same asymptotic order for the expected gathering period.

Finally, we analyze a variation of the Line sweeping sensor field in which the per step traveled distance is halved, called Slow line sweeping. In our model we assumed that a device is unable to get readings from a sensor while moving. Therefore, by advancing r positions, some of the sensors can cross the sweeping line without being detected. However if we reduce the distance traveled to $r/2$ all the sensors are detected in a sweep of the terrain. Therefore we have the following:

Theorem 4. *The sensor field Slow line sweeping solves the continuous monitoring problem with g mobile sensors on a terrain $\mathcal{T}(n, n, m)$ with n devices, latency n, gathering period $2n$, $\mathcal{T}(n) = \mathcal{L}(n) = \mathcal{S}(n) = \mathcal{O}(g)$, $\mathcal{M}(n) = 2$ and, $\mathcal{D}(n) = r/2$.*

References

1. C. Àlvarez, A. Duch, J. Gabarro, O. Michail, M. Serna, and P. Spirakis. Computational models for networks of tiny artifacts: a survey. *Computer Science Review*, 2011. In press.
2. C. Àlvarez, A. Duch, J. Gabarro, and M. Serna. Sensor field: A computational model. In *Algorithmic Aspects of Wireless Sensor Networks: 5th International Workshop, ALGOSENSORS 2009, Rhodes, Greece, July 10-11, 2009. Revised Selected Papers*, pages 3–14, Berlin, Heidelberg, 2009. Springer-Verlag.
3. J. Díaz, D. Mitsche, and P. Santi. Theoretical aspects of graph models for MANETs. In S. Nikoetseas and J. Rolim, editors, *Theoretical Aspects of Distributed Computing in Sensor Networks*, Monographs in Theoretical Computer Science, pages 161–190. Springer, 2011.
4. J. Díaz, X. Pérez, M. Serna, and N. Wormald. Walkers on the cycle and the grid. *SIAM Journal on Discrete Mathematics*, 22(2):747–775, 2008.
5. P. Gibbons and S. Tirthapura. Estimating simple functions on the union of data streams. In *SPAA*, pages 281–291, 2001.
6. F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes.* Morgan Kaufmann, San Mateo, CA, 1993.
7. L. Valiant. A bridging model for parallel computation. *Commun. ACM*, 33(8):103–111, 1990.

Appendix

A Sensor fields for static data

We provide here a detailed description of the sensor fields Line sweeping, Surveillance grid and Surveillance strip.

Line sweeping for a terrain $\mathcal{T}(n, n, m)$ and g sensors at unknown but fixed locations.
Initially we place n devices in the bottom row of Γ_n. This guarantees that any sensor placed in the m/n bottom rows of Λ_m can be detected.
Devices keep two arrays `A` and `B` with g positions, a variable `avg`, a boolean variable `up` initially set to true.
Keep a variable `cn` keeping information about its position in the line holding one of the values `L,R,C,LX,RX` indicating whether they are left, right to the central node, the central node or the leftmost or rightmost point.
Devices are also able to detect when they are in the bottom/top row of Γ_n.
Initial step ($t = 0$)

- Collect all detected readings in table `A`.
- If `cn` = `LX` or `cn` = `RX` send `B`.
- Move upwards.

Step ($t > 0$)

- Collect all detected readings in table `A`.
- If `cn` = `L`
  - If a message is received from the left neighbor merge the received table with `B`.
  - If a message is received from the right neighbor update `avg` with the received value, set `B` =`A` and `A` =0.
- If `cn` = `R` the code is the same as in the case `cn` = `L`, changing left for rigth and viceversa.
- If `cn` = `LX` the code is the same as in the case `cn` = `L`, with the change that `B` is sent after a message from the right neighbor is received.
- If `cn` = `RX` the code is the same as for the case `cn` = `LX`, changing right for left.
- If `cn` = `C`
  - If a message is received from the left neighbor and from the right neighbor, merge the received tables with `A`.
  - Set `avg` to the average of the data stored in `A`.

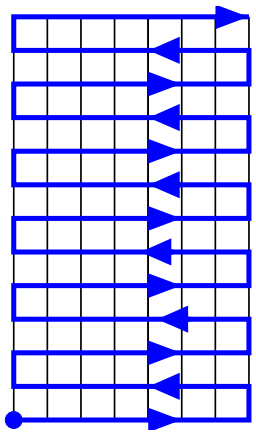

Fig. 3. A snake walk in a strip.

  - Set `A` =0.
- If `up` move upwards, otherwise move downwards for distance r
- If in top row, set `up` = $false$
- If in bottom row, set `up` = $true$
- Send the variables `A` if updated and `avg` if computed or received.
- Output to the environment the value of `avg`.

Observe that in the Line sweeping sensor field we only require that the devices have some knowledge of the local network topology, have sense of direction (up/down) and, that they are able to detect proximity. In particular, the number of devices in the network is not assumed to be known by the devices, although they could perform a count at the expense of increasing the memory usage to $\mathcal{O}(\log n)$.

For a $a \times b$ grid, define the *snake walk* as the covering path that starts on the bottom left position and traverses all the rows in the grid (see Fig. 3. We assume that a device placed in the bottom left position of the grid and given its dimension a and b is able to move step by step on the snake walk keeping the actual positions and being able to detect the direction to reach the next/previous position. They are also able to detect when they reach the first/last position in the snake walk. We use the instructions *move upwards (downwards)* for the movement to the next (previous) position on the snake walk.

Surveillance grid for a terrain $\mathcal{T}(n, \sigma, m)$ and g sensors at unknown but fixed locations.

Initially we place n^2 devices in the bottom left position of its surveillance grid with dimension $k \times k$.
The devices keep two arrays `A` and `B` with g positions, a variable `avg`, a boolean variable `up` initially set to true.
Furthermore they keep a variable `cn` keeping information about their position in the grid holding one of the values `L,R,LX,RX, CU,CD,CT, CB, CC` indicating where they are left, right to the central column, the leftmost or rightmost point of their row, in the upper (lower) part of the central column or, the central node in the central column.
We assume that devices know the side a of the surveillance area and are aware of
Initial step ($t = 0$)

- Collect all detected readings in table `A`.
- Move upwards.

Step ($t > 0$)

- Collect all detected readings in table `A`.
- If `cn` = `L`
  - If a message is received from the left neighbor merge the received table with `B`.
  - If a message is received from the right neighbor update `avg` with the received value, set `B` =`A` and `A` =0.
  - If `up` move upwards, otherwise move downwards for distance s .
- If `cn` = `R` the code is the same as in the case `cn` = `L`, changing left for right and viceversa.
- If `cn` = `LX` the code is the same as in the case `cn` = `L`, with the change that `B` is sent after a message from the right neighbor is received.
- If `cn` = `RX` the code is the same as in the case `cn` = `LX`, changing right for left.
- If `cn` = `CU` the code is the same as in the case `cn` = `L`, changing right for down.
- If `cn` = `CD` the code is the same as in the case `cn` = `L`, changing right for up and left for down
- If `cn` = `CT` the code is the same as in the case `cn` = `LX`, changing right for down and left for up
- If `cn` = `CB` the code is the same as in the case `cn` = `LX`, changing right for up and left for down.
- If `cn` = `CC`
  - If a message is received from the up neighbor and from the down neighbor, merge the received tables with `A`.

  - Set `avg` to the average of the data stored in `A`.
  - Set `A` =0.
- If `up` move upwards, otherwise move downwards
- Send the variables `A` if updated and `avg` if computed or received.
- Output to the environment the value of `avg`.

In the previous algorithm we have assumed that the devices are able to detect when they are in the first/last position in the walk. This feature can be implemented in the devices. Alternatively, it can be computed with a counter, having knowledge of the size of the surveillance area.

The Surveillance strip sensor field is the similar to the Line sweeping changing the movement of the devices to sweep the surveillance area, following the snake walk.

B Sensor fields for dynamic data

We provide here the proof of the lower bounds for the expected gathering period corresponding to Theorems 2 and 3.

Now we give a lower bound, which even for constant g and $m \to \infty$ is of the same asymptotic order.

Lemma 3. $\mathbf{E}[T] \geq (1-(\frac{1}{2})^g)(3m^2/8-m)$.

Proof. First observe that the probability to have at least one walker w initially placed at a position (i,j) with $j \leq \frac{m}{4}$ or $j \geq \frac{3m}{4}$ is $1-(\frac{1}{2})^g$. Call this event $\mathcal{E}$. By symmetry, we can assume without loss of generality that $j \leq \frac{m}{4}$ is the case with at least that probability. Consider again the same random walk on the truncated integer line $\{1,\ldots,\frac{m}{2}\}$ as in the proof of the upper bound, and denote also, as before by U_i the random variable counting the number of steps it takes to hit position $\frac{m}{2}$ in this line, starting from position $i \leq \frac{m}{2}$. By the previous analysis, $\mathbf{E}\left[U_{m/4}\right] = \sum_{i=1}^{m/4}(4(\frac{m}{2}-i)-2) = 3m^2/8-m$. Now, for a walker w starting in position (i,j) with $j \leq \frac{m}{4}$, $\mathbf{E}[T_w] \geq \mathbf{E}\left[U_{m/4}\right]$. Since with probability at least $1-(\frac{1}{2})^g$ at least one walker is initially at a position (i,j) with $j \leq \frac{m}{4}$ (without loss of generality), we get $\mathbf{E}[T] \geq \mathbf{E}[T_w] \geq (1-(\frac{1}{2})^g)(3m^2/8-m)$. Hence, $\mathbf{E}[T] \geq \mathbf{Pr}[\mathcal{E}]\mathbf{E}[T|\mathcal{E}] \geq (1-(\frac{1}{2})^g)\mathbf{E}\left[U_{m/4}\right] = (1-(\frac{1}{2})^g)(3m^2/8-m)$.

Using similar ideas as in the case of the Central line sensor field, we give an easy lower bound which is quadratic in M for the expected gathering period of the Communicating grid sensor field.

Lemma 4. $\mathbf{E}[T] \geq (M^2/8-M/4+2)(1-(\frac{1}{2})^g)$.

Proof. As in the case in which all the devices are placed on the central line, note that with probability at least $(1-(\frac{1}{2})^g)$ at least one walker is initially placed at a position (i,j) which is at least at distance $\frac{M}{2}$ from the closest sensor. Assume without loss of generality that in this case the vertical distance is at least $\frac{M}{4}$. Call this event $\mathcal{E}$. Consider the random walk on the truncated line $\{\frac{M}{4},\ldots,1\}$ with the following transition probabilities: $p_{i,i+1}=p_{i,i-1}=\frac{1}{4}$, $p_{i,i}=\frac{1}{2}$ for any $i=2,\ldots,\frac{M}{4}-1$, $p_{1,1}=\frac{1}{2}$, $p_{1,2}=\frac{1}{2}$, $p_{M/4,M/4}=1$. Note that the expected number of steps for a walker on a grid whose vertical distance is at least $\frac{M}{4}$ to hit its closest device is bounded from below by the expected number of steps needed to hit the position $\frac{M}{4}$ in this random walk starting from position 1. Thus, define by L_i the rv counting the number of steps it takes to hit position $\frac{M}{4}$ in this random walk, starting from position i. We have $\mathbf{E}[L_i]=4i-2+\mathbf{E}[L_{i+1}]$ for any $1\leq i\leq\frac{M}{4}-1$, and $\mathbf{E}\left[L_{M/4}\right]=0$. These equations yield $\mathbf{E}[L_1]=M^2/8-M/4+2$.

Now,

$$\mathbf{E}[T]\geq\mathbf{E}[T|\mathcal{E}]\mathbf{Pr}[\mathcal{E}]\geq\mathbf{E}[L_1]\mathbf{Pr}[\mathcal{E}],$$

and therefore $\mathbf{E}[T]\geq(M^2/8-M/4+2)(1-(\frac{1}{2})^g)$.