

# Efficient Collection of Sensor Data via a New Accelerated Random Walk \*

C.M. Angelopoulos   S. Nikolettseas   D. Patroumpa   C. Raptopoulos  
{aggeloko, patroump, raptopox}@ceid.upatras.gr, nikole@cti.gr

University of Patras and  
Research Academic Computer Technology Institute (RACTI),  
Greece

## ABSTRACT

Motivated by the problem of efficiently collecting data from wireless sensor networks via a mobile sink, we present an accelerated random walk on Random Geometric Graphs. Random walks in wireless sensor networks can serve as fully local, lightweight strategies for sink motion that significantly reduce energy dissipation but introduce higher latency in the data collection process. In most cases random walks are studied on graphs like  $G_{n,p}$  and Grid. Instead, we here choose the Random Geometric Graphs model (RGG), which abstracts more accurately spatial proximity in a wireless sensor network. We first evaluate an adaptive walk (the Random Walk with Inertia) on the RGG model; its performance proved to be poor and led us to define and experimentally evaluate a novel random walk which we call  $\gamma$ -stretched random walk. Its basic idea is to favour visiting distant neighbours of the current node towards reducing node overlap and accelerate the cover time. We also define a new performance metric called *Proximity Cover Time* which, along with other metrics such as visit overlap statistics and proximity variation, we use to evaluate the performance properties and features of the various walks.

## Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Distributed networks, Network topology, Wireless communication

## Keywords

Wireless Sensor Networks, Random Walks, Data Collection, Sink Mobility

## 1. INTRODUCTION

Wireless Sensor Networks are envisioned as large ad-hoc collections of very small autonomous devices, that can sense

\*This work was partially supported by EU/FIRE HOBNET project - STREP ICT-257466

environmental conditions in their immediate surroundings while having limited processing, communication capabilities and energy reserves (see e.g. [3] for a comprehensive coverage of the fundamental algorithms and protocols for wireless sensor networks). The collected sensory data is usually disseminated to a static control center (called data sink) in the network, using node to node *multi-hop* data propagation. Such settings have increased implementation complexity and sensor devices consume significant amounts of energy. This is due to the fact that multi-hop data propagation protocols in sensor networks, where the sink is static, leads the sensor devices to consume significant amounts of energy in operations other than sensing (e.g. inter-node communication for synchronisation purposes, exploratory messages, etc). Moreover, the data exchanged are used as input to several complex distributed algorithms and protocols (such as localization) run by the sensor devices. Even though those algorithms are designed so as to consume as less resources as possible (e.g. [5][16]) the problem still exists. Furthermore, in the area around the control center, nodes need to heavily relay the data from the entire network, thus a bottleneck of increased energy consumption emerges and failures due to strained energy resources of these nodes leads to an early disconnected and dysfunctional network (see e.g. [11]). One of the approaches for more efficient data-centric routing in wireless sensor networks is provided in [4] where an energy aware distributed heuristic builds a special rooted broadcast tree with many leaves that facilitates the routing. An Energy Balancing Protocol is presented in [7] where the presented algorithm decides in each step of the transmission whether to propagate data one-hop towards the sink, or to send data directly to the sink.

Sink mobility can be used as a simple and energy efficient alternative for data collection. The sink is lifting the burden of inter-node coordination by traversing the network itself. A mobile sink may also substitute connectivity as it is capable of bypassing obstacles and reaching disconnected components of the network. Sensor nodes maintain a rather passive role, in terms of data propagation, by simply waiting to contact the sink. When contact is established they deliver data via cheap (energy wise) one-hop transmissions.

Apparently, in such data collection schemes, new critical issues emerge regarding the pattern of movement to be adopted by the sink. The network has to be traversed in a timely

and efficient manner. Also, the mobility pattern of the sink should guarantee that the entire network area, or at least the vast part of it, will eventually be visited.

In the light of the above discussion, random walks in wireless sensor networks can serve as fully local, very simple, distributed strategies for sink motion that significantly reduce energy dissipation and also probabilistically guarantee that eventually the entire network area will be covered. Several real life applications exploit sink mobility for more efficient data collection, e.g. robots that move inside a building that needs to be evacuated or foresters moving through a forest where sensors have been deployed.

In this study, we choose to model wireless sensor networks via Random Geometric Graphs (RGG). We choose this model instead of other graph models (e.g.  $G_{n,p}$  and Grid), that are well established in random walk studies, as RGG better capture certain relevant characteristics of real WSN's such as link existence dependencies of neighbouring nodes due to geometric proximity.

Our goal is mainly to accelerate the random walk coverage of the network via reducing the node overlaps. Also, to improve other (network related) features of the walk, such as how fast it gets close to the network nodes.

A preliminary version of this research has appeared in [1].

## 2. RELATED WORK AND COMPARISON

Random Walks have been extensively studied in the past decades in the context of several disciplines. However, despite their numerous applications in ad hoc, vanets (see e.g. [6][12]) and overlay networks, not much research has been done on how they can be particularly applied in sensor networks with mobile entities, in a way that addresses the peculiarities of such networks (such as severe computing and communication constraints, the small memory and constrained battery, as well as the time-criticality of important applications).

A comparison of different random walk strategies for ad hoc networks is performed in [9]. The authors investigate the effectiveness of each strategy in terms of the expected hop count and the occurrence of deadlocks. In our work, we propose a new random walk strategy and compare two variations of it with two known walks. We study the performance of the strategies in terms of cover time, approximate cover time, proximity variation and a new metric that we define here, proximity cover time.

In [13] the problem of data gathering in a large-scale wireless sensor network with static nodes and a mobile patrol node is formulated as a classical random walk on a random geometric graph. The authors derive analytical bounds for the performance of the random walk in terms of node coverage. In order to improve this performance they propose an algorithm to constrain the random walk using the available side information, such as the awareness of previously visited sites. We also use some information (just the information of geometric distance) in order to speed up the network traversal and our new walks are different.

In [18] authors investigate the *Random Waypoint Model*, which is widely used in the simulation studies of mobile ad hoc networks and show that it fails to provide a steady state in that the average nodal speed consistently decreases over time; therefore it should not be directly used for simulation. They also propose a simple fix of the problem and discuss a few alternatives. In our study, the moving element of the network (the sink) is visiting the nodes in a sequential way. Therefore we abstract its motion by considering that it is moving on the edges of a Random Geometric Graph, that models a wireless sensor network.

Nice research has also been conducted for deterministic sink mobility, such as in [14, 15] where authors address the problem of maximising the network lifetime. In [14] authors suggest that the base station can be mobile and conclude that the best mobility strategy consists in following the periphery of the network (assuming that the sensors are deployed within a circle). In [15] authors investigate the approach that makes use of a mobile sink for balancing the traffic load and in turn improving network lifetime. They engineer a routing protocol that effectively supports sink mobility and through intensive simulations they prove the feasibility of their mobile sink approach.

## 3. THE NEW RANDOM WALK AND PERFORMANCE METRICS

### 3.1 The network model

Sensor networks comprise of a vast number of ultra-small homogeneous sensor devices (which we also refer to as sensors), whose purpose is to monitor local environmental conditions. Each sensor is a fully-autonomous computing and communication device, characterized mainly by its available power supply (battery), its transmission range  $r$ , the energy cost of data transmission and the (limited) processing and memory capabilities. Sensors (in our model here) do not move. The positions of sensors within the network area are random and follow a uniform distribution. We focus on data collection methods, so we assume that initially all sensors have some data to deliver to the sink. For clarity, we also assume that no data is generated during the network traversal. That is if a node is called “visited”, then it has no data to send to the sink.

There is a special node within the network region, which we call the sink  $S$ , that represents a control center where data should be collected. Here, we assume that the sink is *mobile*. The sink is not resource constrained i.e. it is assumed to be powerful in terms of computing, memory and energy supplies.

We consider that the random uniform placement of the sensors inside the network area is abstracted by a *Random Geometric Graph*. Random Geometric Graphs are formed by  $n$  vertices that are placed uniformly at random in the  $[0,1]^2$  square. An edge  $(u,v)$  exists iff the Euclidean distance of vertices  $u$  and  $v$  is at most  $r$ , where  $r$  corresponds to the wireless communication radius  $r$  of the sensors. This holds assuming a disc radio model; two sensors can communicate with each other iff each one lies inside the communication range of the other. Random Geometric Graphs also have an important nice property: unlike other random graphs, like

$G_{n,p}$ , edges are not statistically independent of each other. That is, the existence of an edge  $(u, v)$  is not independent of the existence of edges  $(u, w)$  and  $(w, v)$ . This property makes RGG a quite realistic model for wireless sensor networks that captures to a great extent the communication structure of real WSNs (at least their spatial aspects).

More strictly, consider an area  $\mathcal{A} \subset \mathbb{R}^2$  in two dimensional space. An instance of the *random geometric graphs model*  $\mathcal{G}(\mathcal{X}_n; r)$  is constructed as follows: select  $n$  points  $\mathcal{X}_n$  uniformly at random in  $\mathcal{A}$ . The set  $V = \mathcal{X}_n$  is the set of vertices of the graph and we connect two vertices if their euclidean distance is at most  $r$ . For any vertex  $v \in V$  we will denote by  $N(v)$  the set of neighbours of  $v$  and by  $\deg(v) = |N(v)|$  its degree. Furthermore, we will denote by  $\|u - v\|$  the euclidean distance between the points corresponding to vertices  $v, u$ .

In [10, 17] it is shown that the connectivity threshold for  $\mathcal{G}(\mathcal{X}_n; r)$  is  $r_c = \sqrt{\frac{\ln n}{\pi n}}$ . In this paper we will consider random instances of  $\mathcal{G}(\mathcal{X}_n; r)$  of varying density, by selecting  $r = \sqrt{\frac{c \ln n}{\pi n}}$ , for different values of  $c > 1$ , which guarantees that the produced random instance is connected with high probability.

Besides the information about the set of neighbours of each vertex  $v \in V$ , an instance  $G(\mathcal{X}_n; r)$  of the random geometric graphs model also contains extra information about the exact euclidean distance between  $v$  and any of its neighbours. We used this information in order to define a new random walk, namely the  $\gamma$ -stretched random walk on  $G(\mathcal{X}_n; r)$ , which is described below. This random walk aims to accelerate the data collection process while keeping the memory requirements restricted. The basic idea is that “more distant” neighbours of the current vertex (state of the walk) are favoured in a probabilistic manner. Therefore, our new walk is in fact a *biased transitions* random walk (see e.g. [2] for a nice discussion of several types of random walks based on the assumptions they made).

### 3.2 The $\gamma$ -Stretched Random Walk

Let  $G(\mathcal{X}_n; r)$  be a random instance of the random geometric graphs model with vertex set  $V = \mathcal{X}_n$ , where  $|V| = n$ . Consider a particle moving on the vertices of the graph  $G(\mathcal{X}_n; r)$ . Given that it occupies a specific vertex  $v \in V$  at time  $t \geq 0$ , it decides where to move at time  $t + 1$  by choosing a vertex  $u \in V$  with probability

$$p_{u,v} = \begin{cases} \frac{\|u-v\|^\gamma}{\sum_{w \in N(v)} \|w-v\|^\gamma} & \text{if } u \in N(v) \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The Markov chain describing the above process will be called  $\gamma$ -stretched random walk on  $G(\mathcal{X}_n; r)$  and will be denoted by  $\mathcal{W}_{G(\mathcal{X}_n; r)}^{(\gamma)}$ . Furthermore, we will denote by  $\mathcal{W}_v^\gamma(t)$  the state of the walk that begins at  $v$  at time  $t$ . More formally, the state space of the  $\gamma$ -stretched random walk is the set of vertices  $V = \mathcal{X}_n$  of the graph and its transition probability matrix is given by  $P = [p_{u,v}]_{u,v \in V}$ .

This new random walk has minimum memory requirements,

since every step is decided only by using information of the current state. Notice also that the larger  $\gamma$  is, the more distant neighbours are favoured over neighbours that are close by. In the simulation, we will often set  $\gamma = \frac{r}{r_c}$ , so that the bias to visit distant neighbours is stronger as the density of  $G(\mathcal{X}_n; r)$  increases. The special case  $\gamma = 1$  will be referred to simply as stretched random walk.

### 3.3 Random Walk with Inertia

In the heuristic Random Walk with Inertia the walk tries to maintain the same direction while traversing the graph  $G(\mathcal{X}_n; r)$ . Let  $v$  denote the current state of the walk, i.e. the current vertex. Let  $S^v$  denote the set of vertices that are 1-hop neighbours of  $v$ . Finally, let  $v_{previous}$  denote the previous state of the walk. At the beginning of the walk given its current state, the next hop is chosen uniformly at random among the neighbouring nodes and  $v_{previous}$  is not defined. From the second hop and so forth the next state of the walk is chosen among the vertices of the set  $S^v / v_{previous}$  according to the following probability distribution function:

$$P\{next\_state == i\} = \begin{cases} \frac{\phi_i^v}{\sum_{j \in S^v} \phi_j^v} & \text{if } i \in S^v \\ 0 & \text{otherwise} \end{cases}$$

where  $\phi_j^v$  is the angle defined by vertices  $v, v_{previous}$  and  $j$  and  $0 \leq \phi_j^v \leq \pi$ .

Intuitively, this walk tries to change the subregions of the graph it visits frequently by favouring the vertices lying towards the same direction of its motion. The choice of the next vertex is probabilistic, in an effort to reduce overlaps by avoiding to move on the same path.

We note that this walk has light-weight requirements. It assumes zero knowledge of the network and is relatively simple with low computational complexity. Furthermore, it requires a small, constant sized memory since the next step of the walk depends solely on the previous one. By favouring vertices lying towards the same direction, the sink makes long paths and traverses many different sub-regions of the network area very quickly, thus avoiding early overlaps. However, after most of the network area has been covered, there exist small unvisited sub-regions that are hard for the walk to discover. The fact that many different sub-regions are visited very soon makes this walk suitable for time critical applications, such as reactive event detection.

### 3.4 Known Random Walks

In this section we present two known random walks.

#### 3.4.1 Blind Random Walk

This is the usual random walk model. The blind random walk on a random instance  $G(\mathcal{X}_n; r)$  of the random geometric graphs model is the simplest of all possible sink mobility patterns, since the next move of the sink is stochastically independent to the previous ones. Furthermore, given that the current vertex is  $v \in V$ , the probability of moving to any neighbouring vertex  $u \in N(v)$  is  $p_{v,u} = \frac{1}{\deg(v)}$ . This method is very robust, since it probabilistically guarantees that eventually all network regions and nodes will be visited and thus all data will be collected given that the network

is connected. However, in some network structures it may become inefficient, mostly with respect to high latency, since the sink uses no memory of the past movements in order to select the next one and thus overlaps (i.e. visits to already visited vertices) occur.

### 3.4.2 Random Walk with Memory

The performance of the blind random walk can be improved using some memory of past visits. In random walk with memory  $K$ , the sink maintains a first-in-first-out (FIFO) list  $\mathcal{M}$  which contains the last  $K$  nodes visited during the random walk, i.e.  $M = \{c_1, c_2, \dots, c_K\}$ . The next hop is chosen uniformly at random among the neighbours of the node that are not in the memory list  $\mathcal{M}$ . The use of memory eliminates the possibility of short loops in random walks. Setting  $K = 0$ , we simply get the blind random walk. In this study the random walk with memory 1 will be used for comparison purposes. Note that this walk has stronger memory requirements and overhead than both the blind and the  $\gamma$ -stretched random walk.

## 3.5 Metrics

In this section we define and discuss the metrics that will be used in the evaluation of our new random walk, when compared to the blind random walk and the random walk with memory 1. In the following, let  $\mathcal{W}$  denote any random walk model defined on a random instance  $G(\mathcal{X}_n; r)$  of the random geometric graphs model.

### 3.5.1 Cover time and approximate cover time

Consider the random walk  $\mathcal{W}_v$  that begins on vertex  $v$  of a random instance of the random geometric graphs model  $G(\mathcal{X}_n; r)$ . Define  $T_v \stackrel{\text{def}}{=} \inf\{t | \forall u \in V, \exists t' \leq t : W_v(t') = u\}$  to be the time needed until  $\mathcal{W}_v$  has visited all the vertices in the graph. The *cover time*  $C$  of  $G(\mathcal{X}_n; r)$  is defined as  $C = \max_{v \in V} \mathbb{E}[T_v]$ , where  $\mathbb{E}$  denotes the expected value of the random variable  $T_v$ . It was shown in [8] that when  $r = \sqrt{\frac{c \ln n}{\pi n}}$ , for  $c > 1$ , the cover time of  $G(\mathcal{X}_n; r)$  for the blind random walk is asymptotically  $c \ln \left( \frac{c}{c-1} \right) n \ln n$ .

For  $\epsilon \in (0, 1)$  define also the  $\epsilon$ -approximate cover time as the mean number of steps that  $\mathcal{W}$  needs in order to visit a fraction  $1 - \epsilon$  of the vertices. More formally, define  $T_v^{(\epsilon)}$  to be the time needed until  $\mathcal{W}_v$  has visited  $(1 - \epsilon)n$  vertices. The  $\epsilon$ -approximate cover time  $C$  of  $G(\mathcal{X}_n; r)$  is defined as  $C^{(\epsilon)} = \max_{v \in V} \mathbb{E}[T_v^{(\epsilon)}]$ .

These cover times are related to latency, as they capture the time the sink needs to collect the sensory data from the entire network. The approximate cover time metric is of great interest as the majority of overlaps occur while the sink tries to locate the last few unvisited sub-regions that are scattered in the network area; however, in most sensor network applications it is sufficient to collect a vast percentage of the total sensory data, so this metric is relevant and informative.

### 3.5.2 A new metric: Proximity Cover Time

In real-life sensor networks, the sink is capable of collecting data not only from the currently visited node, but also

from all nodes that are inside the communication range  $r$  (via a single-hop data transmission). Therefore, there is strong motivation from real-life WSN to consider as visited not only the vertex of the current state of the walk, but also all neighbouring vertices. Following is the formal definition of the corresponding metric, Proximity Cover Time.

Denote by  $D(v, \rho)$  the disc of radius  $\rho > 0$ , centered at  $v$ . Consider the random walk  $\mathcal{W}_v$  that begins on vertex  $v$  of a random instance of the random geometric graphs model  $G(\mathcal{X}_n; r)$ . We will say that  $\mathcal{W}_v$  is *within distance*  $\rho$  to vertex  $u$  at time  $t$  if  $W_v(t) \in D(u, \rho)$ , i.e. the random walk started at  $v$  occupies a vertex that lies in  $D(v, \rho)$  at time  $t$ . Let  $T_v(\rho) \stackrel{\text{def}}{=} \inf\{t | \forall u \in V, \exists t' \leq t : W_v(t') \in D(u, \rho)\}$  be the time needed until  $\mathcal{W}_v$  has come within distance  $\rho$  to all vertices of the graph. We define the  $\rho$ -proximity cover time of  $G(\mathcal{X}_n; r)$  as  $C(\rho) = \max_{v \in V} \mathbb{E}[T_v(\rho)]$ .

A similar but different metric is presented in [2] where each time the random walk visits an already visited vertex, then u.a.r. picks an unvisited neighbour (if one exists), marks it as visited and then continues the random walk from its current vertex (i.e. it does not make a transition to the marked neighbour).

### 3.5.3 Proximity Variation

For a vertex  $v \in V$  and time  $t_0 \geq 0$  let

$$\text{dist}(v, t_0) = \min_{0 \leq t \leq t_0} \|\mathcal{W}(t) - v\|$$

be the function that returns the minimum distance between the random walk  $\mathcal{W}$  up to time  $t_0$  and node  $v$ . Then the *proximity variation*  $PV(t_0)$  at time  $t_0$ , is given by

$$PV(t_0) = \frac{\sum_{v \in V} \text{dist}(v, t_0)}{n}$$

where  $n$  is the total number of nodes.

The rationale behind this metric is the following: In a real network, if the PV metric converges to zero quickly with  $t$ , this means that the sink gets close to all sensors quite soon and data collection progresses fast; this is especially relevant in case when the role of sensors is not completely passive but includes some limited multi-hop propagation of data to accelerate data propagation at a reasonable energy cost. On the contrary, when the PV converges slowly to zero, it means that the network traversal is performed in a way that some areas may stay unvisited for long time.

### 3.5.4 Visit overlap statistics

Other metrics that will be used in order to further characterize each walk's evolution include the number of visits to a specific vertex  $v$ , as well as the distribution of the number of visits for every vertex in the graph. These overlap statistics will provide an insight on how exactly each walk traverses the graph, how different components of the graph are visited and in what rate.

## 4. PERFORMANCE EVALUATION

### 4.1 Simulation Set-up

We conducted our experimental evaluation using Matlab R2009a as our simulation environment. We evaluate the walks on several instances of the Random Geometric Graphs model in order to address its random nature. More particularly, we construct Random Geometric Graphs consisting of  $n = 2000$  nodes placed in a  $100 \times 100$  square area. For such RGG graphs, the connectivity threshold is computed to be of radius  $r = 100\sqrt{\frac{\ln 2000}{\pi 2000}} = 3.47$ . For our performance evaluation we use three characteristic values for  $r$ :  $r_1 = 5$ ,  $r_2 = 7$ ,  $r_3 = 10$ , corresponding to sparse, average and denser graphs respectively. For every value of  $r$ , we construct 50 RGG instances on which we evaluate all walks for 100 iterations per instance. We consider these randomly constructed RGG instances to be representative of the RGG space.

For the performance evaluation of the protocols we used two network topologies. At the first topology sensor nodes are deployed uniformly at random over the planar area. This topology constitutes the base of our performance evaluation study by providing a first insight on the performance of each protocol. At the second network topology a portion of the sensors (in our experiments 60%) is deployed uniformly at random in order to establish connectivity. The rest of the sensors are deployed over the network area following the normal distribution with mean value  $\mu = 50$  (i.e. half the dimension of the network area) and standard deviation  $\sigma = 7$  for the  $x$  and  $y$  co-ordinates (independently of each other) of each sensor. This set-up leads to a heterogeneous network deployment in which a high concentration of sensor nodes is located around the center of the network area.

## 4.2 Simulation Findings

We first experimentally evaluate the performance of the Random Walk with Inertia on the Random Geometric Graph and compare it with the Blind Random Walk (only in the uniform topology). Then we compare four random walks; the well known Blind Random Walk; two versions of our newly proposed walk: the Stretched Walk where  $\gamma = 1$  and the  $\gamma$ -Stretched Walk, where  $\gamma = \frac{r}{r_1}$ , in order to investigate the impact of  $\gamma$ ; the Random Walk with Memory 1, that somehow represents an upper bound in terms of memory usage and therefore is considered to be more powerful than the rest of the walks.

### 4.2.1 Uniform Topology

Figure 1 presents the comparison between the Random Walk with Inertia and the Blind Random Walk. Although the Random Walk with Inertia on the Grid performs better than the Blind, however the RGG model causes the Inertia to present a really weak performance. This result led us to the design of the newly proposed random walk, the so called  $\gamma$ -Stretched Random Walk.

Figure 2 depicts the mean number of hops each walk needs in order to visit all the nodes of the network for three representative values of the communication range  $r$ . For all the random walks we observe that as the density of the graph increases, the total cover time is reduced even by nearly 50%. This is due to the fact that in a dense network the walk has more edges to traverse on, and therefore there is higher probability to visit any given vertex. In other words, a random walk reaches its stationary distribution much faster.

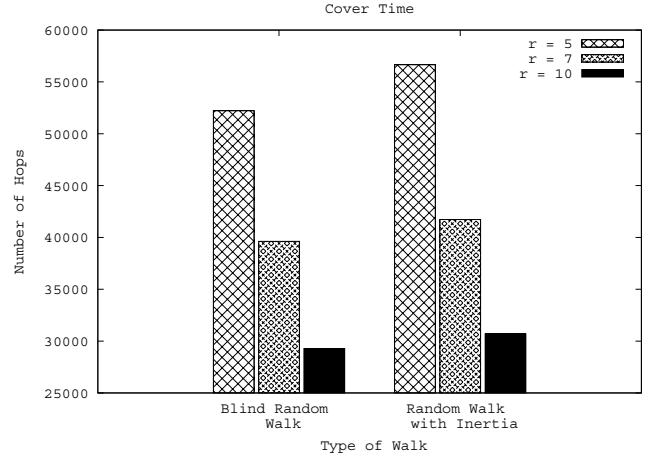


Figure 1: Cover Time of the Blind Random Walk and the Inertia Random Walk

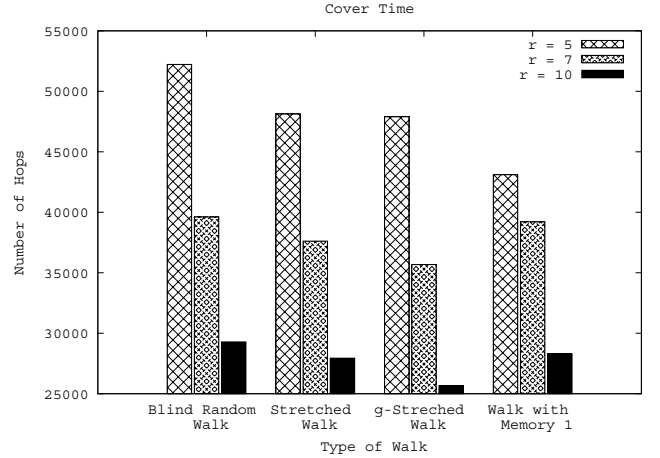
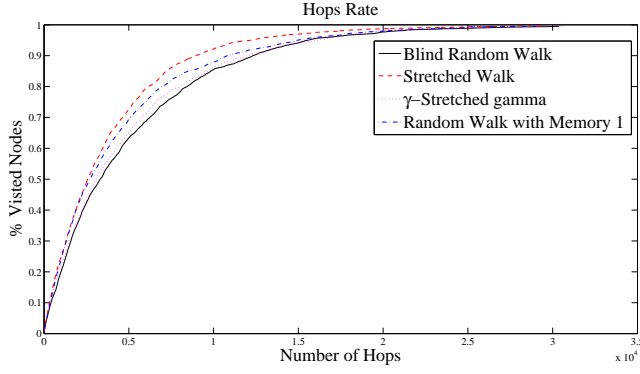


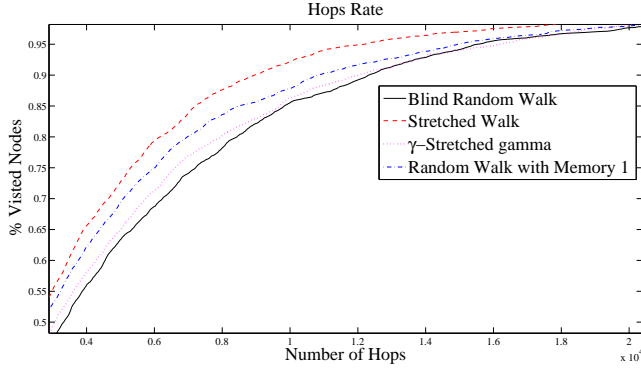
Figure 2: Cover Time of the Blind Random Walk, Stretched Walk,  $\gamma$ -Stretched Walk and Walk with Memory 1

We also observe that for all graph densities, the Blind Random Walk has the biggest cover time, while the impact of the fixed-sized memory of the Walk with Memory declines as the density of the graph increases. In fact, in sparse graphs, where the available options are limited, avoiding 1-hop overlaps by not visiting the last position significantly reduces the cover time. In our Stretched Walks, on the contrary, the impact of the  $\gamma$  factor is proportional to the number of neighbouring nodes as the graph density increases. This is why for small values of radius  $r$  the performance of the Stretched Walks is very similar. However, in more dense graphs, biasing the walk towards the more distant neighbours seems to have a well noticeable effect on the speed the random walk traverses the graph.

Figures 3 and 4 depict the rate at which each walk discovers unvisited nodes inside the network area for range  $r = 5$  and  $r = 10$  respectively. For sparse networks ( $r = 5$ ) we note that the Stretched Walk outperforms even Random



(a) Nodes' Discovery Rate for  $r = 5$



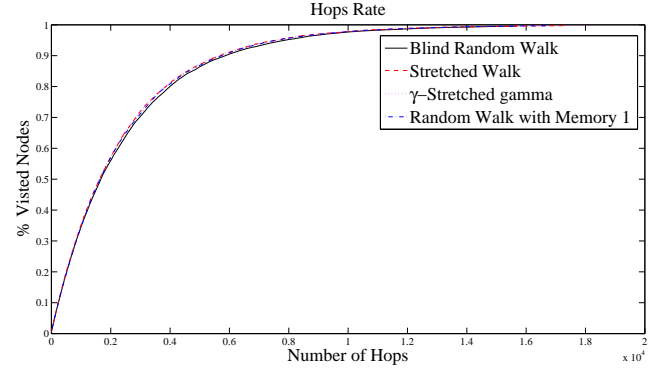
(b) Nodes' Discovery Rate for  $r = 5$  (zoom-in)

**Figure 3: Percentage of discovered nodes over number of hops with range  $r = 5$  and uniform distribution**

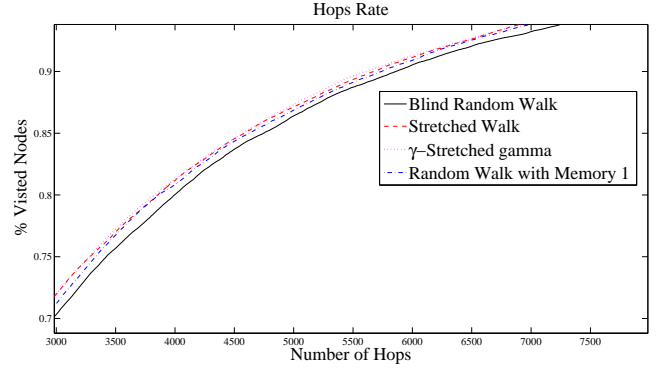
Walk with Memory. The Random Walk with Memory, while avoiding its previous position, may move towards a neighbouring node that geometrically is very approximate; thus allowing a high probability for early overlaps. On the other hand, the Stretched Walk, by favouring the more distant neighbours, quickly changes the subregions of the networks it traverses. This way it avoids early overlaps, thus discovering unvisited nodes much faster. However, in sparse networks, if the bias factor towards distant neighbours is very crude it may have an opposite effect. As seen in the performance of the  $\gamma$ -Stretched Walk, a crude bias factor may narrow the available options for the next step down to the point the walk is forced to make long cycles during its traversal, thus visiting already visited nodes.

In dense networks ( $r = 10$ ) the rate at which unvisited nodes are discovered is accelerated for all walks. However,  $\gamma$ -Stretched Walk performs slightly better as the bias factor strongly favours distant nodes over geometrically approximate ones. We note that for all walks while they all discover relatively quick the 95% of the total number of vertices in the graph, they spend nearly half of the cover time in order to visit the last 5% of the vertices, thus creating a long convergence tail at the end of the traversal.

As discussed in subsection 3.5.2, there is strong motivation from real-life wireless sensor networks to consider as visited

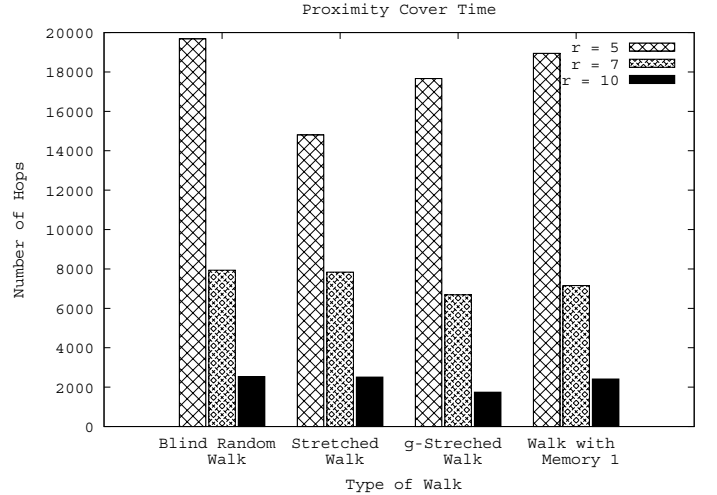


(a) Hops Rate for  $r = 10$



(b) Hops Rate for  $r = 10$  (zoom-in)

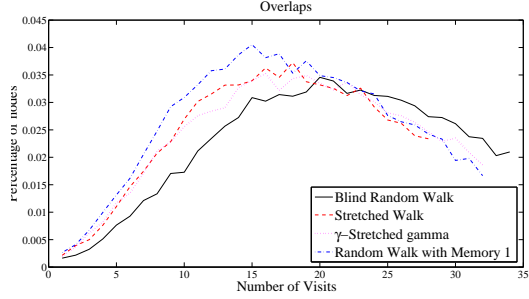
**Figure 4: Percentage of discovered nodes over number of hops with range  $r = 10$  and uniform distribution**



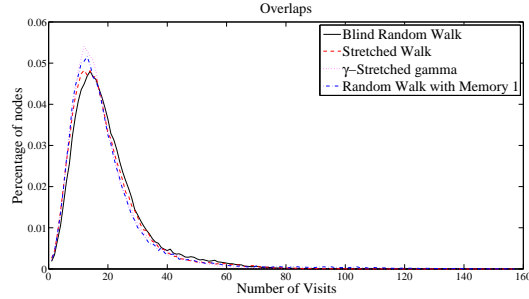
**Figure 5: Proximity Cover Time**

not only the vertex of the current state of the walk, but also all neighbouring vertices in the RGG network. Figure 5 depicts the performance evaluation of all walks in RGG for the Proximity Cover Time metric. Intuitively, the dominant factor for this metric is the dislocation attribute of the motion (i.e. the mean geometric distance the sink covers per

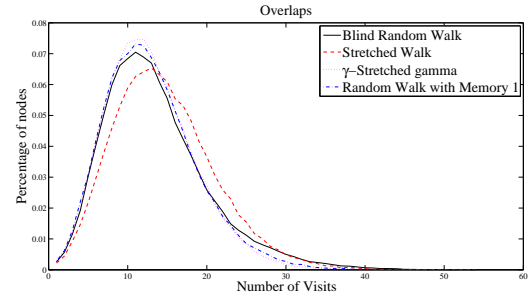
hop). In sparse networks the use of memory by the Random Walk with Memory slightly improves the Proximity Cover Time of the Blind Random Walk. The Stretched Walk with  $\gamma = 1$  shows the best performance from all other walks. The  $\gamma$ -Stretched Walk is outperformed due to the crude way it is biased towards the few distant neighbours. However, for higher network densities this factor aids the walk to outperform even the Walk with Memory. In a network consisting of 2000 nodes, in less than 2000 hops the sink has been inside the communication radius of every node in the network.



(a) Overlaps distribution for  $r = 5$



(b) Overlaps distribution for  $r = 7$



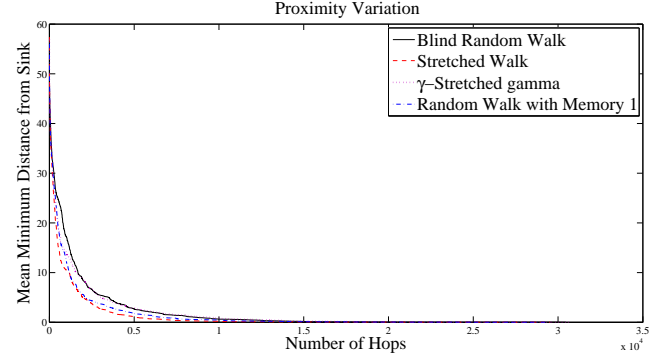
(c) Overlaps distribution for  $r = 10$

**Figure 6: Overlaps distribution over the nodes of the graph for the random uniform topology.**

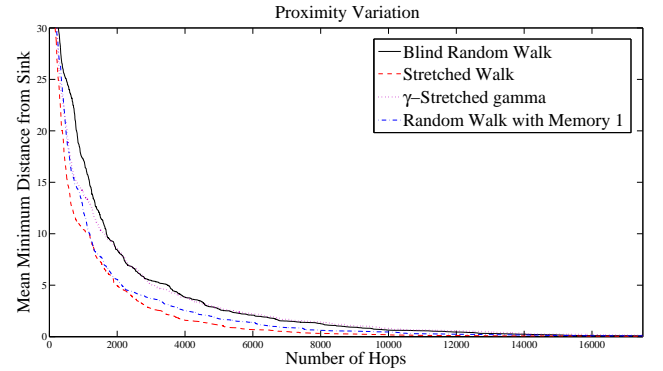
Figure 6 depicts the overlaps distribution over the nodes of the graph. That is what percentage of the total number of the nodes has been visited  $i$  times by the sink during the entire network traversal process. In sparse graphs the Random Walk with Memory outperforms the two Stretched Walks, that is more nodes of the network have been visited fewer times (i.e. the curve corresponding to Walk with Memory is above and to the left than the rest). At first glance this contradicts the fact that the Stretched Walk is discovering unvisited nodes in sparse networks at a higher rate than the Walk with Memory (Fig. 2). We note however that the

Stretched Walk avoids early overlaps, while it needs more time to cover the entire network (Fig. 2). This leads to the conclusion that most of the overlaps occur towards the end of the network traversal process, thus the converging tails in Fig. 2.

For more dense networks the impact of the bias factor of  $\gamma$ -Stretched Walk is increasing along with the value of  $r$ , thus eventually demonstrating the best performance for this metric. Also, as the network graph density increases, the number of revisits for the majority of the nodes is more or less the same (10-15 revisits). However, the total number of revisits is increased significantly, as the last few unvisited nodes are hard to find in a dense network.



(a) Proximity Variation for  $r = 5$

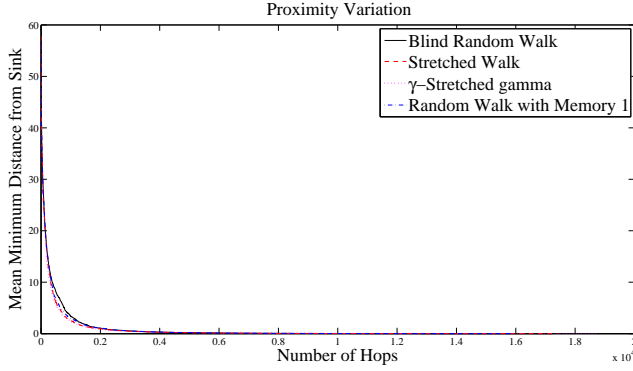


(b) Proximity Variation for  $r = 5$  (zoom-in)

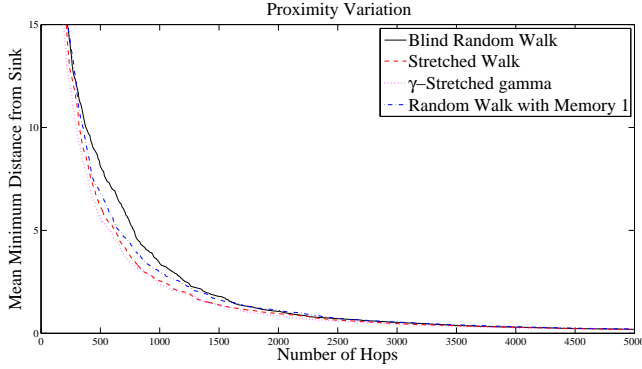
**Figure 7: The Proximity Variation Metric for  $r = 5$  and uniform topology**

There lies a different intuition behind each walk on how the network area is traversed, regardless of the total number of hops needed. For example the strategy of avoiding previous positions is getting less effective when the density of the network increases, while the  $\gamma$  factor in Stretched walks is allowing the sink to visit many different network subregions very early in the traversal process. The proximity variance metric captures these differences. In Fig. 6 we note that the two walks which perform better in sparse graphs are the Stretched Walk (where  $\gamma = 1$ ) and the Random Walk with Memory. As there are relatively few neighbouring nodes slightly favouring distant nodes or just avoiding previous position seems to suffice. On the other hand, the  $\gamma$ -Stretched Walk (where  $\gamma = \frac{r}{r_c}$ ) favours distant nodes in a more crude

manner (for  $r = 5$  and  $r_c \approx 2.3$ ,  $\gamma \approx 2$ ). This way, the walk is left with very few available options, forced to make several cycles during its traversal process.



(a) Proximity Variation for  $r = 10$



(b) Proximity Variation for  $r = 10$  (zoom-in)

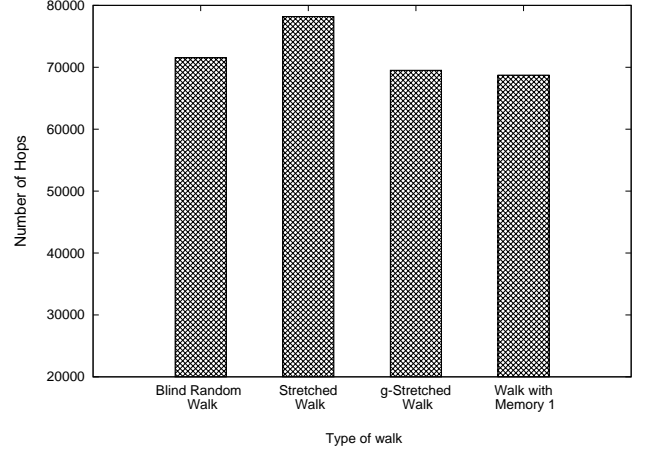
**Figure 8: The Proximity Variation Metric for  $r = 10$  and uniform topology**

In dense networks, where there are many available options for the next step of the walk, this crude selection of the distant nodes allows the walk not to get attracted by approximate neighbours. Therefore, in Fig 8 the  $\gamma$ -Stretched Walk is the one converging to zero at a higher rate.

#### 4.2.2 Normal Distribution

Figure 9 depicts the cover time of each walk for a specific characteristic communication range  $r$ . As shown in the figure, the  $\gamma$ -Stretched Random Walk performs better than the Blind Random Walk. The  $\gamma$  biased factor allows the Stretched Walk not to get trapped inside dense areas of the network, contrary to the Blind Walk. On the other hand, the simple Stretched Random Walk has the worst performance.

A better insight on the performance of our two proposed walks for the normal distribution can be obtained by analysing their performance over the uniform network topology shown in figure 2. We notice that in sparse areas both walks have similar performance regarding their cover time and the impact of the  $\gamma$  factor is negligible. This is due to the fact that in sparse areas both walks have few alternatives for their next move. On the contrary, in dense areas the  $\gamma$  factor allows the  $\gamma$ -Stretched Walk to quickly traverse the network



**Figure 9: Cover Time for heterogeneous topology**

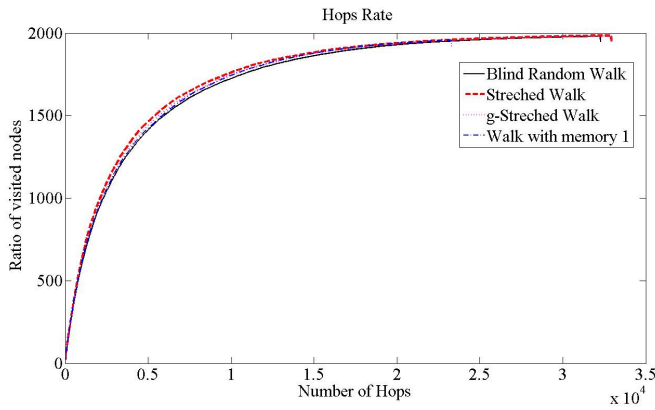
area by favouring the more distant neighbours resulting in better performance. As a result, in the case of a heterogeneous topology, the  $\gamma$ -Stretched Walk behaves similarly in the sparse areas of the network, but much better in the more dense areas, yielding a better overall performance.

Figure 10 presents the rate at which each walk discovers unvisited nodes inside the network area. We note that the  $\gamma$ -Stretched Random Walk outperforms even the Random Walk with Memory. As in the uniform topology, the Stretched Walk, by favouring the more distant neighbours, quickly changes the subregions of the networks it traverses, this way avoiding early overlaps and discovering unvisited nodes much faster.

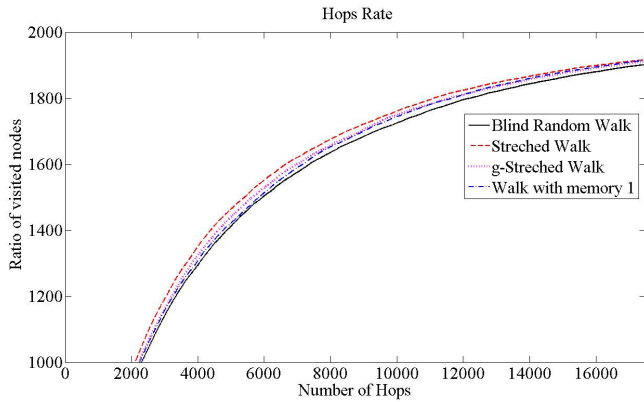
Figure 11 depicts the overlaps distribution over the nodes of the graph. That is what percentage of the total number of the nodes has been visited  $i$  times by the sink during the entire network traversal process. One can see that the Random Walk with Memory outperforms the two Stretched Walks, that is more nodes of the network have been visited fewer times. This does not contradict the fact that the  $\gamma$ -Stretched Walk discovers unvisited nodes at a higher rate than the Walk with Memory. We note however that the Stretched Walk avoids early overlaps, while it needs more time to cover the entire network. Actually, most of the overlaps occur towards the end of the network traversal process, especially in this heterogeneous topology, since it is more difficult for the Stretched Walks to discover the nodes of the dense area thus leading to multiple overlaps.

Finally, figure 12 depicts the proximity variation metric for each walk. As shown in the figure, the Stretched Random Walk ( $\gamma = 1$ ) has the best performance, since it favours distant neighbours, but is not that biased as the  $\gamma$ -Stretched, thus succeeding to visit many different network subregions, both in the sparse but also in the dense areas of the network. On the other hand, the  $\gamma$ -Stretched Walk (where  $\gamma = \frac{r}{r_c}$ ) favours distant nodes in a more crude manner (for  $r = 7$  and  $r_c \approx 3.5$ ,  $\gamma \approx 2$ ). This way, the walk is left with very few available options, forced to make several cycles during





(a) Hops rate for  $r = 7$



(b) Hops rate for  $r = 7$  (zoom-in)

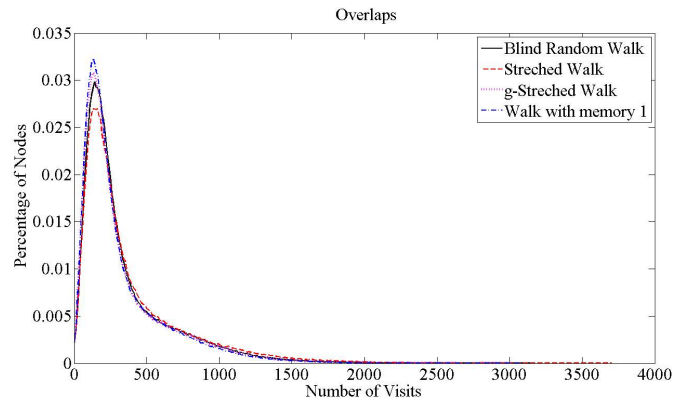
**Figure 10: The Hops rate for  $r = 7$  and heterogeneous topology**

its traversal process.

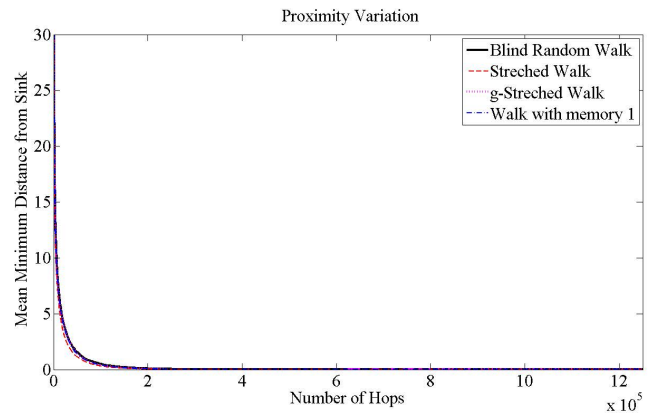
## 5. CONCLUSIONS

In this work we address the problem of efficient data collection in wireless sensor networks via a mobile sink. We model the data collection process by random walks on Random Geometric Graphs as they better capture the geometric dependencies that characterise inter-node wireless communication in WSN's. We use random walks as local, simple motion strategies for the sink that have very low computational and a priori knowledge requirements. We define a new walk called Stretched Walk and evaluate two variations of it. The basic idea of the new walk is to favour distant neighbours of the current node towards avoiding visit overlaps. Also, motivated by real-life WSN's, we define a new metric called Proximity Cover Time; this captures collecting data from all neighbour nodes of a currently visited node.

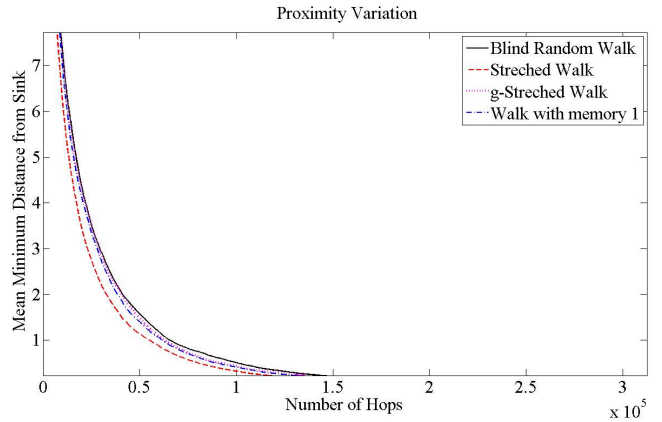
Findings show that the use of constant sized memory in order to accelerate the data collection process has limited results, particularly when the network density increases. Instead, walks that use local (immediate neighbour) information in a smart manner can demonstrate significantly better results. However, their use should be wise as they may lead to opposite results (e.g. strong bias factor in sparse networks leads to bigger cover times).



**Figure 11: The overlaps distribution for  $r = 7$  and heterogeneous topology**



(a) Proximity Variation for  $r = 7$



(b) Proximity Variation for  $r = 7$  (zoom-in)

**Figure 12: The Proximity Variation for  $r = 7$  and heterogeneous topology**

## 6. REFERENCES

- [1] C. M. Angelopoulos, S. Nikolettseas, D. Patroumpa, and C. Raptopoulos. A new random walk for efficient data collection in sensor networks. In *The 9th ACM International Symposium on Mobility Management*

- and *Wireless Access (MOBIWAC)*, 2011.
- [2] P. Berenbrink, C. Cooper, R. Elsasser, T. Radzik, and T. Sauerwald. Speeding up random walks with neighborhood exploration. In *Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010.
  - [3] A. Boukerche. *Algorithms and Protocols for Wireless Sensor Networks*. Wiley&Sons, 2008.
  - [4] A. Boukerche, X. CHeng, and J. Linus. Energy-aware data centric routing in microsensor networks. In *ACM MSWiM*, 2003.
  - [5] A. Boukerche, H. A. Oliveira, E. Nakamura, and A. A. Loureiro. Localization systems for wireless sensor networks. *IEEE Wireless Communication Magazine*, 14(6):6–12, 2007.
  - [6] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro. Vehicular ad hoc networks: A new challenge for localization-based systems. *Computer Communications*, 31:2838–2849, 2008.
  - [7] J. R. C. Efthymiou, S. Nikoletseas. Energy balanced data propagation in wireless sensor networks. *Wireless Networks (WINET) Journal*. Also, in *Proc. 4th International Workshop on Algorithms for Wireless, Mobile, Ad-Hoc and Sensor Networks (WMAN 04), IPDPS 2004.*, 12(6):691–707, 2006.
  - [8] C. Cooper and A. Frieze. Component structure of the vacant set induced by a random walk on a random graph. *Random structures and Algorithms*, 32:401–439, 2008.
  - [9] S. S. Dhillon and P. V. Miegheem. Comparison of random walk strategies for ad hoc networks. In *The Sixth Annual Mediterranean Ad Hoc Networking WorkShop*, June 2007.
  - [10] P. Gupta and P. Kumar. *Critical power for asymptotic connectivity in wireless networks*. Stochastic Analysis, Control, Optimization and Applications, Boston, 1998.
  - [11] A. Jarry, P. Leone, S. Nikoletseas, and J. D. P. Rolim. Optimal data gathering paths and energy balance mechanisms in wireless networks. In *DCOSS*, pages 288–305, 2010. Best Paper Award.
  - [12] K. K. Leung. Modeling and optimization of vehicular wireless ad-hoc networks. In *The 15th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM '12*, 2012.
  - [13] L. Lima and J. Barros. Random walks on sensor networks. In *WiOpt '07: 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, April 2007.
  - [14] J. Luo and J.-P. Hubaux. Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks. In *24th IEEE INFOCOM*, Miami, USA, 2005.
  - [15] J. Luo, J. Panchard, M. Piorkowski, M. Grossglauser, and J.-P. Hubaux. Mobiroute: Routing towards a mobile sink for improving lifetime in sensor networks. In *2nd IEEE/ACM International Conference on Distributed Computing in Sensor Networks (DCOSS)*, volume 4026, pages 480–497, 2006.
  - [16] Y. W. Neal Patwari, Robert J. O Dea and R. J. O. dea. Relative location in wireless networks. in *Proceedings of the 6th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, 2009.
  - [17] M. Penrose. *Random Geometric Graphs*. Oxford University Press, 2003.
  - [18] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *Proceedings of Infocom '03*, pages 1312–1321.