



**HAL**  
open science

## A GPS-less on-demand mobile sink-assisted data collection in wireless sensor networks

Nicola Roberto Zema, Nathalie Mitton, Giuseppe Ruggeri

### ► To cite this version:

Nicola Roberto Zema, Nathalie Mitton, Giuseppe Ruggeri. A GPS-less on-demand mobile sink-assisted data collection in wireless sensor networks. Short Paper at Wireless Days (WD), IFIP / IEEE, Nov 2014, Rio de Janeiro, Brazil. hal-01074486

**HAL Id: hal-01074486**

**<https://hal.inria.fr/hal-01074486>**

Submitted on 28 Sep 2015

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

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

# A GPS-less on-demand mobile sink-assisted data collection in wireless sensor networks

Nicola Roberto Zema

Università Mediterranea di Reggio Calabria  
Email: nicola.zema@unirc.it

Nathalie Mitton

Inria, France  
Email: nathalie.mitton@inria.fr

Giuseppe Ruggeri

Università Mediterranea di Reggio Calabria  
Email: giuseppe.ruggeri@unirc.it

**Abstract**—The autonomous data collector is a role recently introduced to improve the performance of Wireless Sensor Networks. When a prompt response for data processing and offloading is necessary, i.e. in the case of event-driven networks, a mobile flying sink could be a good option for that role. In this paper, we introduce FreeFall, a distributed algorithm for the autonomous navigation of a mobile collector through a WSN for on-demand data offloading that does not rely on an absolute coordinate system. We show that, under fairly common circumstances, it is possible to set the trajectory of the mobile sink and fulfill the offloading requests without the needs of additional equipment installed on nodes. We show how our system is preferable over more classical routing solutions especially in the presence of localized generation of large amounts of information.

**Keywords**—WSN, Controlled Mobility, GPS-less.

## I. INTRODUCTION

The use of controlled mobility as a degree of freedom has been found as one of the most prominent methods to ease the energy consumption, increase the lifetime and enhance the performances in a wireless sensor network [1]. For instance, sensors nodes around fixed data sinks deplete sooner their energy because of an increased traffic flow through them, and it is created an *energy hole*. This problem could be solved using a mobile collector [2], [3]. In a WSN where data generation is linked to “events”, there would be nodes where the generation rate increases whenever their sensors detect environmental changes [4], and suddenly an huge amount of data has to be rapidly offloaded to a sink. Again the use of a mobile collector has proven to be effective to improve the offloading process. In fact, by using the mobile collector, most of the transmission are carried on single hop connection. As an immediate consequence, the data offloading rate is increased and the energy consumption is reduced.

In [5] it has been shown that a good possibility for the implementation of a controlled mobility framework is when the network itself drives the sink trajectory in a completely distributed manner exploiting geographic and geometrical properties. In this paper, following these previous studies, we introduce FreeFall, a novel distributed GPS-less method for a mobile collector which trajectory is driven by the ‘off-loading’ requests issued by the network sensor nodes without the aid of a global positioning system installed in them. More in detail, according to the FreeFall approach the nodes interested to transmit their data explicitly advertise their position to a mobile sink. The mobile sink will be capable to set its own

waypoints in order to reach the interested areas using only an inertial navigation system. To the best of our knowledge, the only strictly similar work in the literature is our previous Directed Movement approach, introduced in [2], which relies on the same principles than FreeFall but which assumes that nodes are aware of their absolute geographical coordinates. The location discovery system used in FreeFall does not use complex and expensive absolute positioning system (GPS) nor direction finding devices. FreeFall has been evaluated through a simulation study where the performances of our system have been compared with significant related works. Our approach shows a decreased energy expenditure over more common solutions for data delivery and an improvement in the end-to-end delay when the number of sources is limited.

## II. MAIN ASSUMPTIONS

The main assumptions for our framework to be effective are: (i) the network is composed by static sensor nodes and a mobile sink (either a ground robot or a flying drone); (ii) it is dense enough to uniformly cover the whole sensing area given the capabilities of a general sensor networking device; (iii) each node is able to infer the relative position of its neighbors using an *r*-map [6]. The *r*-map is the implementation of a polar coordinate system specifying the relative location of the neighbors of a node from its point of view. Thanks to the implementation of a *r*-map, every node  $i$  sets its own reference originate axis, later referred as  $\vec{x}_i$ . Through the relative positions of nodes inferred by the *r*-map, each node  $i$  is able to estimate in this reference system for every of its neighbor  $j$  both the angle  $\alpha_{ij} = \angle \vec{x}_i, \vec{i}_j$  and the Euclidean distance  $d_{ij}^i$  between nodes  $i$  and  $j$  in the reference system of  $i$ . A detailed algorithm for the construction of an *r*-map is described in [6].

## III. OUR CONTRIBUTION

When a source  $s$  has to offload data, it needs to advertise the sink  $S$ . As the sink is mobile and  $s$  has no information about its mobility pattern,  $s$  is not able to reach it directly. It will thus send a request in the network using a *source advertisement* packet. Specifically we suppose that each advertisement packet is forwarded along two perpendicular directions as in [7]. Also the sink advertises its position by using a *sink advertisement* packet. Once again sink advertisement packets are forwarded along two perpendicular directions. Eventually, both *source advertisement* packet and *sink advertisement* packet reach a common node that will forward the request to the sink with a *backtrace advertisement* packet, as illustrated on Figure 1-b. More details about the aforementioned process will be given in the following sections, however, for the sake of

clarity the following description is restricted to a single sending direction which forms an angle  $\beta$  with the reference axis  $\vec{x}_i$  of the node  $i$ .

*Neighbor of interest:* Given a node  $i$ , a node  $j$  is said to be a *Neighbour of Interest* of node  $i$  in respect to the propagation angle  $\beta$  if  $|\beta - \alpha_{ij}^i| < 90^\circ$  and  $d_{ij}^i \times \cos(\beta - \alpha_{ij}^i) < \Delta_{width,i}$ ; with  $\Delta_{width,i}$  set as the communication range of the nodes. As an example, in Figure 1-a, the source node has three neighbors:  $N_1$ ,  $N_2$  and  $N_3$  but only  $N_1$  and  $N_2$  are selected as *Neighbour of Interest* since laying in the  $\Delta_{width}$ -wide strip along the direction chosen by the source.

#### A. Source and sink advertisement packets

Let's assume a source  $s$  needs to offload data. First of all node  $s$  determines a random direction along which to send its request, identified by the angle  $\beta$ , then determines all its *Neighbor of Interest* in respect the same angle  $\beta$  and, finally a *source advertisement* packet is sent every  $\Delta T_{adv}^s$  to the *Neighbors of Interest*. By properly selecting the *Neighbor of Interest* it is assured a *source advertisement* packet is forwarded following the direction which has been previously determined. The *source advertisement* packet contains the source node ID, the propagation angle  $\beta$ , the list of the forwarders,  $d_{trans}(si)$  and  $d_{long}(sj)$ . The meaning of source node ID, the propagation angle  $\beta$  is straightforward. The list of the forwarders is meant to contain the list of the nodes which have already forwarded the packet and it is used for the backtrace process. Of course when a packet is freshly generated this list is empty and it will be populated by the forwarders themselves.  $d_{trans}(sj)$  and  $d_{long}(sj)$  are the longitudinal and transversal distance between the last node  $j$  which processed the packet and the Source  $s$  which had generated the packet. More in details, the transverse distance is the distance between node  $j$  and its projection on the  $\beta$  axis. The longitudinal distance is the distance between the source and the projection of the node on the  $\beta$  axis. For instance, on Figure 1-a, the transverse distance of  $N_1$  is  $D_1$  and its longitudinal distance is  $D_2$ . Again when a packet is freshly generated those distances are zero. Every time a packet is forwarded by a node  $j$   $d_{trans}(sj)$  and  $d_{long}(sj)$  are updated accordingly. It should be noted that in very dense networks, where the list of *Neighbors of Interest* is long it could be important to properly prune that list in order to avoid jeopardizing the scalability of FreeFall. In this work the number of neighbors has randomly selected at each  $\Delta T_{adv}^s$  and a smarter pruning strategy is left as a future work. From its side, the sink applies the same algorithm and issues a *sink advertisement* packet.

#### B. Advertisements processing

Upon receiving an advertisement packet a node  $j$  first it checks the forwarder list to ensure that the packet it is not a duplicate. If the packet is a duplicate it is discarded and no other action is taken. If it is not the case then node  $j$  checks for a matching advertisement, i.e. if the received packet is a *source advertisement* a corresponding sink advertisement is searched, if the received packet is a *sink advertisement* a corresponding *source advertisement* is searched for. If a matching is found a backtrace packet is generated and no other action is taken. If nothing is found in the buffers then node  $j$  forward the packets again. In order to do so node  $j$  has (i) to determine how the direction  $\beta$  relates to its own reference axis  $\vec{x}_j$ . This can be

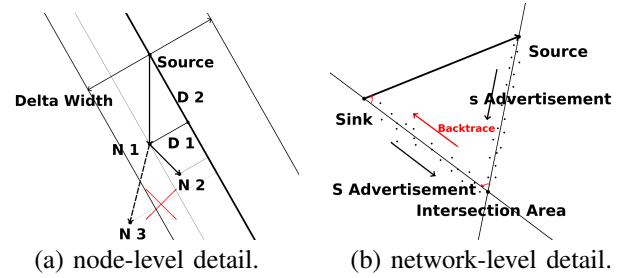


Fig. 1. Working Scenario

done by leveraging the values  $d_{ij}$  and  $\alpha_{ij}$  contained in the r-map; (ii) update the values of  $d_{trans}(sj)$  and longitudinal  $d_{long}(sj)$  or the equivalent in case of a sink; (iii) add itself to the forwarder list. To show how the values of  $d_{trans}(sj)$  and longitudinal  $d_{long}(sj)$  are computed let assume that a node  $j$  receives from node  $i$  a packet which has been produced by the Source  $s$  then:  $d_{trans}(sj) = d_{trans}(si) + d_{ij}^i \times \cos(\beta - \alpha_{ij}^i)$  and  $d_{long}(sj) = d_{long}(si) + d_{ij}^i \times \sin(\beta - \alpha_{ij}^i)$ . The sign of  $d_{ij}^i$  can be negative in case of axis crossing.

#### C. Backtrace packet generation and forwarding

When it have both the *source* and *sink* advertisement packets, node  $j$  has already the estimated transverse and longitudinal distances to the source(s) ( $d_{trans}^j(sj)$  and  $d_{long}^j(sj)$ ) and the sink(s) ( $d_{trans}^j(Sj)$  and  $d_{long}^j(Sj)$ ). From them, it can estimate  $\alpha^j(sS)$  as following:

$$\alpha^j(sS) = \beta^j(S) - \beta^j(s) - \arctan\left(\frac{d_{trans}^j(sj)}{d_{long}^j(sj)}\right) - \arctan\left(\frac{d_{trans}^j(Sj)}{d_{long}^j(Sj)}\right)$$

Node  $j$  generates a *backtrace packet* with the sink Id,  $\alpha^j(sS)$ ,  $d^j(jS)$ ,  $d^j(js)$ . The *backtrace packet* is sent back following the reverse path of the *sink advertisement*.

#### D. Backtrace Packet reception at sink

The sink takes a decision about its movements every  $\Delta t_{movement}$  seconds. If one or more *backtrace packets* have been received during this interval each one of the will be processed as follows. Using the laws of sines and cosines the sink extrapolates the remaining angles and sides of the triangle whose vertexes are the original source, the mobile sink and the node which generated the *backtrace packet*. Hence the sink will be able to calculate the direction and the distance to travel for reaching the source as in Figure 1-b. In case of multiple sources, the node stores information about them and creates a distance-ranked list of way points using simple vectorial operations, closest first. Just after the expiration of  $\Delta t_{movement}$  control the sink set its course to the first node in the list and, upon arrival, starts the data transfer procedure. While in the movement phase the sink will not receive any other packets but will resume the network operation upon arrival and store eventual new information. At the end of the data transfer it will move again to a new destination if present.

## IV. RESULTS

We run a set of simulation under the WSNet environment [8]. We set a topology of 600 nodes distributed uniformly on a  $500m \times 500m$  square area. Nodes use a single-channel, rician faded, 2400MHz bpsk 802.15.4 compliant MAC and

radio interfaces. The values of  $\Delta t_{movement\ control}$  is set as a 802.15.4 beacon-enabled time-slot and  $\Delta T_{adv}$  as the former multiplied by the network diameter. We compare FreeFall with (i) our previous *GPS-driven* approach [2], where the nodes use the absolute position system to send the advertisement packets; (ii) the plain Greedy Geographical Routing (iii) a random mobility framework later referred as *RandMob* where the mobile sink will move randomly around the network and, if it gets into contact with a source, it will start the data offloading procedure. The signaling times are kept the same of the other mobility frameworks. Using *FreeFall* and *GPS-driven*, the sinks and the sources signal each other's presence and then the former will reach the selected positions to transfer the data. We are first interested in the capability of our *FreeFall* to handle flows of multiple sizes. To this purpose, a source and a sink are set at the two opposite corners of the topology and a bursty, acked application is activated between them. Using the maximum capabilities, in terms of packets size and transmission rate possible for the 802.15.4 MAC, the source have to deliver to the network a stream of packets which sum is respectively of 0.25, 0.5, 1 and 1.5 MB, representing the application data. Each value is the result of a 30 simulative runs which outputs were processed with 95% confidence intervals and shown where applicable. According to [9] we set the maximum speed of the sink to  $60km/h$ . Figure 2 displays the time difference, at the source, between the first data packet enqueueing (and the activation of the delivery systems) and the reception of the ack for the last packet. *FreeFall* shows a great reduction either versus the routed approach and the random mobility. It even outperforms the *GPS-driven* approach as ,geometrically, the latter configures a specific case of the former when the two corridors intersect at a right angle, maximizing the hop distance. Figure 3 shows the same evaluation in the case of multiple traffic flows with up to 4 source nodes that need to offload 1 MiB of application data. In terms of sheer traffic consumption, the Random Mobility is the best option but, as shown in the figures that the time necessary for complete the transfer up to four sources, is much higher and it represent the possibility of completely missing the source or the data.

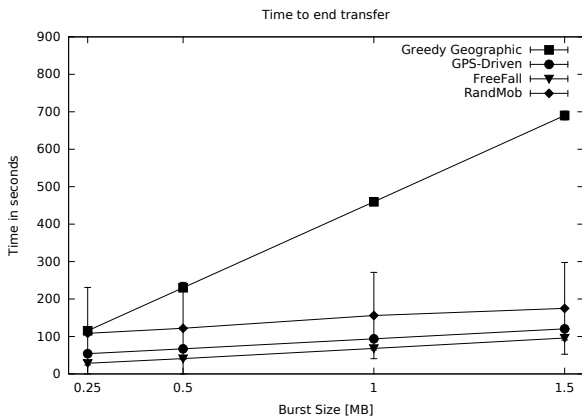


Fig. 2. Time to end data transfer for different burst sizes.

## V. CONCLUSIONS AND FUTURE WORKS

In this paper, we have introduced *FreeFall*, a completely distributed, practical signaling and movement protocols for a

mobile sink to set a trajectory across the network and perform

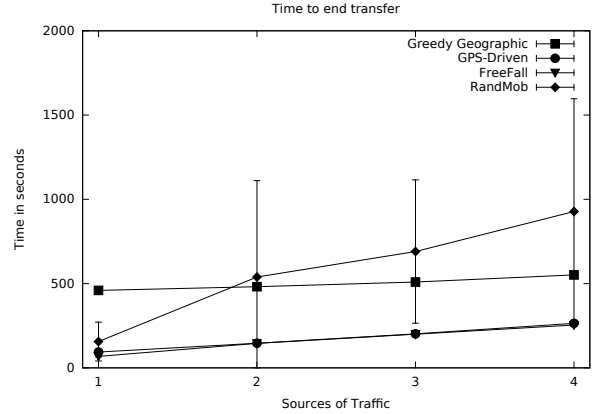


Fig. 3. Time to end data transfer for different number of sources.

data collection in a dense Wireless Sensor Network in which nodes have to offload non-negligible amount of data. Our system is remarkable because it does not use any absolute positioning system and can be employed without additional cost on the nodes and in all those environments where, for instance, GPS is not available. There are, however, some issues that we plan to address in our future research. A multi-sink variant should be investigated together with a framework to make them interoperate, efficiently schedule the source visitation order and the information buffering space needed to avoid starvation and improve fairness. *FreeFall* should implement an obstacle avoidance system that would make the packets route around coverage holes and also a careful tune of the parameters, an accurate evaluation of either the mobility inherent cost and the evaluation of a *noisy r-map* is needed.

## REFERENCES

- [1] S. Gao, H. Zhang, and S. K. Das, "Efficient data collection in wireless sensor networks with path-constrained mobile sinks," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 4, pp. 592–608, 2011.
- [2] N. R. Zema, N. Mitton, and G. Ruggieri, "Using location services to autonomously drive flying mobile sinks in wireless sensor networks," in *Int. Conf. on Ad Hoc Networks (AdHocNets)*, Rhodes, Greece, 2014.
- [3] W. Liang, J. Luo, and X. Xu, "Prolonging network lifetime via a controlled mobile sink in wireless sensor networks," in *Global Telecommunications Conference (GLOBECOM)*, Austin, USA, 2010.
- [4] L. Filipponi, A. Vitaletti, G. Landi, V. Memeo, G. Laura, and P. Pucci, "Smart city: an event driven architecture for monitoring public spaces with heterogeneous sensors," in *Int. Conf. on Sensor Technologies and Applications (SENSORCOMM)*, 2010.
- [5] C. Tunca, S. Isik, M. Donmez, and C. Ersoy, "Distributed mobile sink routing for wireless sensor networks: A survey," *Communications Surveys Tutorials, IEEE*, vol. 16, no. 2, pp. 877–897, Second 2014.
- [6] Y. Bejerano, "Simple and efficient k-coverage verification without location information," in *Int. Conf. on Computer Communications (INFOCOM)*, Phoenix, USA, 2008.
- [7] G. Shi, J. Zheng, J. Yang, and Z. Zhao, "Double-blind data discovery using double cross for large-scale wireless sensor networks with mobile sinks," *IEEE Trans. Vehicular Technology*, vol. 61:5, 2012.
- [8] A. Fraboulet, G. Chelius, and E. Fleury, "Worldsens: Development and prototyping tools for application specific wireless sensors networks," in *Proc. Int. Conf. on Information Processing in Sensor Networks*, ser. IPSN, New York, NY, USA, 2007, pp. 176–185.
- [9] E. Natalizio, R. Surace, V. Loscri, F. Guerriero, and T. Melodia, "Filming Sport Events with Mobile Camera Drones: Mathematical Modeling and Algorithms," Research Report, Dec. 2012.