Accelerated Sensory Data Collection by Greedy or Aggregate Mobility-based Topology Ranks

Constantinos Marios Angelopoulos
University of Patras and CTI, Greece
aggeloko@ceid.upatras.gr

Sotiris Nikoletseas
University of Patras and CTI, Greece
nikole@cti.gr

ABSTRACT

We investigate the problem of efficient data collection in wireless sensor networks where both the sensors and the sink move. We especially study the important, realistic case where the spatial distribution of sensors is non-uniform and their mobility is diverse and dynamic. The basic idea of our protocol is for the sink to benefit of the local information that sensors spread in the network as they move, in order to extract current local conditions and accordingly adjust its trajectory. Thus, sensory motion anyway present in the network serves as a low cost replacement of network information propagation. In particular, we investigate two variations of our method: a)the greedy motion of the sink towards the region of highest density each time and b)taking into account the aggregate density in wider network regions. An extensive comparative evaluation to relevant data collection methods (both randomized and optimized deterministic), demonstrates that our approach achieves significant performance gains, especially in non-uniform placements (but also in uniform ones). In fact, the greedy version of our approach is more suitable in networks where the concentration regions appear in a spatially balanced manner, while the aggregate scheme is more appropriate in networks where the concentration areas are geographically correlated.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*

General Terms

Algorithms, Design, Performance

Keywords

Wireless Sensor Networks, Routing, Aggregation, Performance Evaluation

PE-WASUN'09, October 28–29, 2009, Tenerife, Canary Islands, Spain.

1. INTRODUCTION

At the beginning, studying wireless sensor networks was limited in very large static collections of smart sensor nodes (i.e. placed at some position and remain static) that are deployed over an area of interest. Their mission is to monitor the environment, collect application data and transmit them back to a *static sink* i.e. a control centre, by using sensor to sensor routing. Motivated by important applications where mobility is a dominant characteristic since sensors are attached on mobile entities and sinks can be robotic, later studies have investigated schemes in which *either* the sink *or* sensors are mobile. Many protocols for efficient data gathering/delivery have been proposed alongside with various novel metrics, such as the mobility level that takes into account the speed and dislocation of a sensor. Our research is of the first few that assume that *both* the sink and sensors to be mobile. Particularly, sensors are assumed to be moving in a dynamic and a highly diverse way.

1.1 Related Work

In mobile settings, the protocols and findings of previous research on static wireless sensors networks can not be (at least directly) applied. Efficient solutions in the state of the art become inefficient or even inoperable. Even well studied algorithms need to be redesigned; as an example, in [1] authors propose a leader election algorithm suitable for mobile networks. Also, in [3] authors propose a mobility-aware routing protocol, using zone-based information and a cluster-like communication protocol. Additionally, new problems arise due to the high dynamics, e.g. maintainning system integrity becomes more difficult [2].

Our approach is one of the first few that considers a sensor network where both the sensors and the sink are mobile. For networks of sink mobility, [6] investigates the network lifetime when multiple mobile sinks are periodically repositionned with respect to the energy consumption. Authors propose an integer linear program to determine the new locations and a flow-based routing protocol. [10] proposes a routing scheme where a single sink stops at certain anchor positions while collecting data. The sink samples the global power consumption while in an anchor point and determines the optimal visiting time.

For networks of mobile sensors, [5] suggests exploiting the sensor motion to adaptively disseminate data, e.g. propagate redundant data when mobility is low while propagating less data in the presence of high mobility; in contrast to our approach the sink is assumed to be static, and data travels many hops towards the sink. Also, [8] presents a case study of mobile sensor networks designed for wildlife position tracing. Authors assume varying mobility and propagate data to the node most likely to meet the sink, based on previous history; in contrast, our method considers current dynamics.

In this work, we compare our methods to the following two relevant, characteristic state of the art approaches:

Blind Random Walk

In this scenario the sink is simply moving according to a random walk process and serves any sensors that it may reach. It requires zero knowledge about the deployment of sensors and makes almost no assumption about the network, but its latency is high because of unnecessary visit overlaps and complete ignorance of network topology. This protocol in full mobility schemes is expected to present high latency as the sink does not consider the movement of sensors. Somehow, the blind random walk represents an upper bound on latency, while its energy dissipation is very low and can be considered as minimum since the role of sensors is completely passive.

Optimized Deterministic Motion

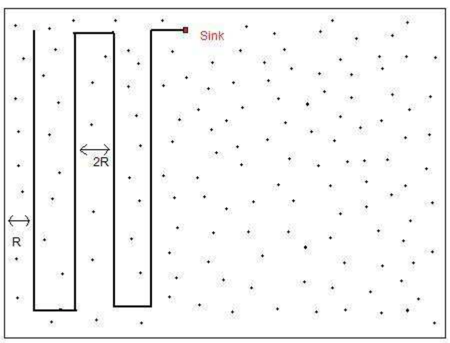


Figure 1: Sink trajectory in Optimized Deterministic Motion

In optimized deterministic motion the route of the sink inside the network area is predetermined. The sink sweeps the entire area in a way that no overlaps occur (see figure 1). This way also guarantees that eventually all sensors are going to be reached, soin uniformplacements this protocol tends to minimize latency. However, in heterogeneous settings, if high density areas are in distance from the sink, it will take a lot of time to reach them. Even then, the sink will have to travel through the entire network area before visiting them again.

Both approaches (however in the case of static sensors) have been investigated in [12, 13]. Also, in [7] a similar idea of inferring topology using mobility has been used; however no aggregation methods had been investigated.

1.2 Our Contribution

Already existing protocols are based on the assumption that given enough time, eventually the sink will reach all sensors inside the network area, and therefore almost all data will be delivered. This strategy for collecting data from sensors works fine in homogeneous placements with respect to success rate (i.e. the number of delivered messages over the total number of messages) but results in high delivery latency per message, especially under placement heterogeneity. Our approach, introduces a novel way of guiding the sink inside the network area in order to collect data faster, while keeping the energy consumption and the success ratio at very satisfactory levels (almost the same, or even better levels as the already existing protocols). The sink gradually gets informed about the sensor placement distribution based on local topology information that sensors measure in a distributed way, and deliver to the sink as they move in the network. Thus, sensory motion anyway present in the network serves as a low cost replacement of network information propagation. As a result, the sink adjusts (either greedily or using aggregation) its trajectory to the current distribution of sensors and collects data very fast, reducing message delivery latency by even 300% for heterogeneous placements and by 30% for uniform placements.

2. MODELLING ASSUMPTIONS

We study wireless sensor networks in which mobile sensor nodes are deployed over an area of interest and a mobile sink is responsible for collecting data. *Both* the sensors and the sink are assumed to be *mobile* and equipped with localization hardware (or are equivalently running a virtual coordinates method). Motivated by realistic scenarios and applications, we focus on heterogeneous sensor placement distributions. Also, we examine diverse mobility dynamics for the sensor motion.

2.1 Sensor Placement and Capabilities

Sensor placement involves a $D \times D$ plane *network area*. Let n be the total number of sensors deployed. Let d be the *density* of sensors in that area (measured in numbers of sensors per m^2). Sensor devices are equipped with a set of hardware monitors that can measure environmental conditions of interest. Each device has a broadcast (digital radio) beacon mode of fixed transmission range R, and is powered by a battery. Also a sensor is equipped with a general purpose storage memory (e.g. FLASH) of small (constant) size C.

Let E_i be the initial available energy supplies of sensor i. At any given time, each sensor can be in one of three different modes, regarding the energy consumption: (a) transmission of a message, (b) reception of a message and (c) sensing of events. In our model, for the case of transmitting and receiving a message, we assume that the radio module dissipates an amount of energy proportional to the message's size. To transmit a k-bit message, the radio expends $E_T(k) = \epsilon_{trans} \cdot k$ and to receive a k-bit message, the radio expends $E_R(k) = \epsilon_{recv} \cdot k$ where $\epsilon_{trans}, \epsilon_{recv}$ are constants that depend on the radio module and the transmission range R of the sensors. For the idle state, we assume that the energy consumed for the circuitry is constant for each time unit and equals E_{idle}. Finally, the *sink* is a special, very powerful node, representing a control centre where data should be propagated to.

The *sink* is initially placed in the *centre* of the network area. In the heterogeneous topology scenario, the placement distribution of sensor motes includes high density areas (called "pockets"), corresponding to hot-spots in the network. Pockets may not necessarily have the same size or density. For example, assume we have six pockets corresponding to six different "hotspots" inside the network area. Two of them appear in a south-east region of the network area, having a density of $5d$ and $3d$ each. The rest four pockets assume to appear in a north-west region of the network

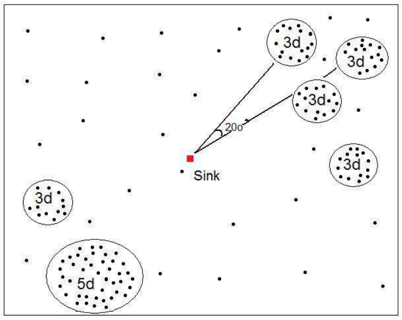


Figure 2: Heterogeneous sensor placement

area, each one having a density of $3d$. The rest of the network area has a uniform random placement (see figure 2). In the uniform scenario, the sensor distribution is uniform all over the network.

For the generation rate of events, we assume that it is constant ($\lambda = 0.05 msg/sec$) and same for each sensor. This way, high density areas imply high sensory data generation. Also, we assume that all sensors are equipped with a positionning system (GPS or similar) and a synchronized clock. Note that our approach only requires relative distance from the sink and also relative time intervals (ΔP, ΔT in ranking function). No absolute localization and synchronization are needed. Thus, the assumptions on GPS and synchronized clocks can be relaxed.

2.2 Modelling Dynamic Sensory Mobility

In most real world scenarios most sensor nodes will move in many different and diverse ways. Also, a node will most likely change the type of movement it follows after some time, varying not only the average speed but also the type of trajectory it follows. Consider a person riding a bicycle to go to work, then spending several hours working (low mobility), then riding back home. Such examples demonstrate the diversity and variability that may arise in networks of mobile sensors. Modelling real life movement patterns is a subject of active research. Simplistic mobility patterns, such as random walk or random way-point alone, can not accurately capture the heterogeneous mobility characteristics we described. Here we try to mimic several main types of movements inspired from the above observations. Using well defined mobility models, we define a few characteristic mobility roles that are used to construct more complex mobility behaviours.

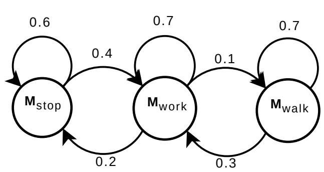
Working movement. We parametrize a version of random walk [4] to achieve slow, located movement. We define the mobility function M_{work} with parameters $[0.5, 1.5] m/sec$ for choosing speed and by setting the movement distance towards a direction to be small, $[1, 5]m$.

Walking movement. We use a variation of the Boundless Area mobility model [4] to define M_{walk}, which is more rapid and less local than M_{work}. When a node reaches the boundaries of the network area we force it to react, i.e., take a left turn of 45^o. We bound the speed to vary between $[1, 2]m/s$, we set the time step $\Delta t = 2s$; at each time step we allow the speed to vary by $\Delta v = 0.25m/s$ and the direction to vary by $\Delta\alpha = 30^o$.

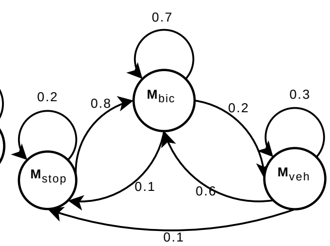
Vehicular movement. Vehicular movement M_{veh} [4] is the fastest of all; we use the Probabilistic Random Walk. In this mobility model, nodes move only towards predefined directions north, north east, east, etc. We vary the speed between $[5.55, 10]m/s$ $(20 - 36km/h)$.

Mobility transitions. Assigning a mobility role is enough to diversify the mobility levels of the nodes. However, in realistic scenarios nodes will change mobility roles. To model such dynamic mobility, we use a state transition diagram to change between mobility models. Each state of the diagram corresponds to a mobility role as defined above. From each state a set of outgoing edges to one or more of the other states exist; each edge is associated with a probability of transition. Also, there is an outgoing edge that returns to the same state. The sum of all outgoing edges from a state is equal to 1. While on a state the node follows the mobility defined by the corresponding mobility model. As soon as a new position needs to be selected a probabilistic experiment is performed to choose a new state according to the state transition diagram, then the corresponding mobility function is invoked to select the position and speed of the node. We also define a special state called the stop state in which the node remains still for a small period of time (see figure 3).

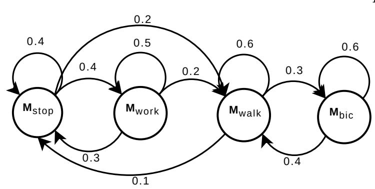
Sink Mobility. The sink moves with constant speed directly towards the position provided each time by our topology-sensitive protocols.



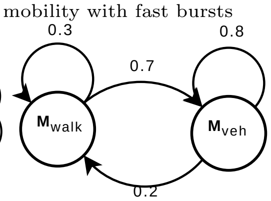
C1: Transitions between slow mobility roles



C3: Transitions for medium mobility with fast bursts



C2: Transitions for medium mobility level



C4: Transitions for fast mobility

Figure 3: Mobility Transitions Diagrams

3. OUR APPROACH

Our approach tries to exploit sensor movement in order to inform the sink about the current network topology. Sensors, by carrying lightweight time-stamped information about local densities across the network, can "guide" the sink towards dense areas. This way, the sink can adjust its route and serve more sensors and higher data traffic in less time, with very limited overhead and very low additional energy dissipation. We below describe the basic components of our method.

3.1 Carrying and Ranking Local Topology Information

Let d_{local} be the local density of a given sub-region of the network area. At the beginning, each sensor gets informed about d_{local} corresponding to its region, by broadcasting a *hello* message and counting the sensors that respond to it. Clearly, this function is not energy efficient if every node computes d_{local} and responds to every *hello* message it receives (complexity $O(n^2)$, where n is the total number of sensors). A mechanism for reducing the number of broadcasts is described.

Intuitively, we would like fast sensors to have accurate topological information, because there is high probability that they will carry this information further in the network. On the other hand, we could allow slow sensors to have less accurate information as their dislocation is small. Let v_i be the speed of sensor i and let v_{max} be the top speed of sensors in the network. Then, sensor i broadcasts a *hello* message with probability $p_i = \frac{v_i}{v_{max}}$ This way, fast sensors, almost certainly, will measure d_{local} and slow sensors will measure d_{local} with low probability. When a sensor is measuring d_{local} along with the *hello* messages, it sends its speed as well. Let this speed be denoted as $sender_{sp}$. When a sensor receives a *hello* message, *unless* there is *no* sensory data to deliver, it decides if it is going to respond via speed ratio:

$$a_i = \frac{sender_{sp}}{receiver_{sp}}$$

where $receiver_{sp}$ is the speed of the receiving sensor. If $a_i > 1$ then the receiver responds, as we have a faster sensor asking (i.e. sending *hello* message) a slower one. Else the receiver decides with probability a_i if it will respond. Notice that the faster is a sensor that receives a *hello*, the lower is the probability to respond. Also, note that only sensors with sensory data respond to *hello* messages. This way areas of no interest or areas that have been recently served by the sink correspond to low values for d_{local} and therefore to low ranks. This way the sink doesn't get "trapped" inside dense areas (i.e. pockets-hotspots), but leaves them as soon as all data is collected.

When sensors start moving, each one carries this local topology information. As said, our goal is to gradually inform the sink about the local distribution of sensors in different regions of the network. If the sink is aware of a high concentration of sensors in a sub-region (thus, high local density and high data traffic), it will try to reach this sub-region as quickly as possible. So, each sensor carries information about the sub-region corresponding to its d_{local}, as it moves in the network.

Let P be the relative coordinates of a given position inside the network area. This type of information can easily be acquired via navigation hardware. Intuitively, the value of information carried by d_{local} degrades over distance. The further a region is, the more time will take for the sink to reach it, and given the dynamic mobility of sensors, the less likely it is for this information to be up to date by the time the sink gets there. The same degradation of quality occurs over time as well. The older d_{local} is, the less accurate (i.e. obsolete) the information it carries. Let T be a denotion of time, i.e. when d_{local} was obtainned. Again, this information can be acquired from a synchronised clock.

By now, the necessity of a ranking function denoting the quality and importance of carried information should be obvious. Let

$$R = \frac{d_{local}^2}{\Delta P \Delta T} \tag{1}$$

where ΔP is distance change (dislocation) and ΔT a time interval, be the ranking function, that ranks information of d_{local} with respect to the distance from origin and the time it was measured. Based on this function, the sink decides towards which direction it will move via a mechanism described in following sections.

3.2 Updating Ranked Information

Previously, we have seen that each sensor can carry local topology information for a sub-region of the network area simply by taking measurements for local density and corresponding position and time. Then, it estimates the importance of these information through the ranking function $R = f(d_{local}, \Delta P, \Delta T)$. As the quality of carried information degrades, each sensor should try to replace it with new information of better quality. So, periodically, each sensor gets informed about d_{local} and marks current position and time. Because of memory restrictions, information of degraded quality is discarded, allowing to use minimal space in memory.

Consider a single sensor. The update process begins with a first set of measurements. Let them be denoted as d_{local}, P and T. The sensor is moving in the network and after a period of time (that is defined by the protocol) a second set of measurements is taken. Let d'_{local}, P' and T' be the second set. When a third set of measurements is to be taken, sensor must decide which of the two old sets it should discard. This can be easily decided via the ranking function R. Let $P_{current}$ and $T_{current}$ be the current values for position and time. Let

$$r = \frac{d_{local}}{||P_{current} - P||(T_{current} - T)}$$

be the rank corresponding to the first set and

$$r' = \frac{d'_{local}}{||P_{current} - P'||(T_{current} - T')}$$

the rank for the second. Then, the set of measurements corresponding to $\min\{r, r'\}$ is to be discarded and replaced by the new measurements. Continuously performing this process guarantees that each sensor will carry the best ranked information available.

Notice that by following the described scheme, each sensor needs very small, constant space in memory (enough to store two sets of information only). Our method can be extended to carry a higher number of triplets (say β), that clearly introduces a performance vs cost trade-off, since a large β would provide a more accurate topology information, but at the same time more memory would be needed.

Notice that information concerning regions with high density values (i.e. pockets) will achieve high rankings, they will be carried by the sensor for a longer time period and eventually traverse a longer distance inside the network area. In next paragraph we will define the process followed by the sink to exploit this ranked information in order to adjust its route, eventually reducing latency in a power-efficient way.

3.3 Our Greedy Topology Rank Method

The sink starts with no information regarding the distribution of sensors over the network. It starts moving in a random direction, waitting to encounter sensors. When a sensor gets inside the communication range of the sink, along with sensory data, it is asked to deliver its ranked information regarding network topology. That is, the stored triplet (d_{local}, P, T) that the very moment of communication with the sink achieves the higher value for function R. However, a sensor responds to this requirement only if it has data to deliver. Hence, areas of no interest or areas that have been recently served by the sink are ignored. This way, the sink doesn't get "trapped" inside dense areas (i.e. pockets), but leaves them as soon as all data is collected. Sink collects topology information for a short period of time (round) from any sensor that it may contact during this period. Suppose that the sink has collected m triplets of information, each one having a rank. The sink will move along the direct line defined by its current position and the position corresponding to the best-ranking triplet. If there is no topological information available, the sink simply follows the blind random walk until it contacts at least one sensor again.

Having moved towards this direction for a distance, say of 3 hops, the sink starts, again, to collect topology information from nearby sensors. This way, it can re-evaluate its route. It is expected that the closer the sink gets to a subregion of high density, the more sensors will urge it to move towards this region. If for some reason (i.e. mobility, failures, etc) the local density of that region drops, the sink will collect high ranks regarding different locations, thus redefining its route.

3.4 Our Aggregate Topology Rank Method

In the method (Greedy) presented in the previous subsection, the sink by using the R function, decides in a greedy way its trajectory trying to reach big sensor concentrations as quickly as possible. However, this greedy choice may not always be the best. Consider the sensor placement shown in figure 2. The sink, would choose to go towards the south-east region of the network, as there lies the pocket with the higher concentration of sensors with density of 5d. However, in the north-west side we have a total density of 12d, but the sink "sees" individual pockets of 3d density. If the sink could aggregate information about pockets that lie towards the same direction, it should further improve its performance.

Aggregation function

Suppose the sink already has some triplets in its memory and is about to decide the direction of its trajectory. As has been mentioned above, it should aggregate the information of triplets corresponding to the same or nearby network regions, thus, referring to the same direction. So, in order for the sink to aggregate triplets it should check the direction they point to. Since both the sink and sensors are mobile in a diverse way this check should be dynamic.

Intuitively, given a position S (i.e. of the sink), two positions P_1 and P_2 (i.e. where two triplets point at) are said to be in the same direction if the angle $\angle P_1SP_2$ is relatively small (i.e. sharp). Computing the angle of two positions, been referred to by two triplets in relation to the current position of the sink, is easy through simple geometry and computation of the tangent of the angle (also C++ provides function atan2 for this purpose). If the angle is smaller than a threshold then the sink aggregates the two triplets. As a result of many experiments and fine tuning, we set this threshold to be $angle_{thres} = 20^o$. Note, that the smaller this threshold is the bigger the distance between two positions P_1, P_2 must be in order to be aggregated. This is why we have chosen a small threshold; we want the sink to aggregate information concerning different locations (and therefore far-away) from the current location of the sink.

As mentioned, a triplet consists of d_{local}, position P (pair of coordinates) and timestamp T. Consider two triplets, A and B, that the sink has decided to aggregate, each one with corresponding measures. From the aggregation one new triplet will result, say C. The aggregated density measure will be $C_{d_{local}} = A_{d_{local}} + B_{d_{local}}$. The aggregated position will be

$$C_P = \frac{A_P * A_{d_{local}} + B_P * B_{d_{local}}}{A_{d_{local}} + B_{d_{local}}}$$

Note, that this formula is a straight analogue to the center of mass of physics. Also, the aggregated timestamp is computed in a similar way. So we have that,

$$C_T = \frac{A_T * A_{d_{local}} + B_T * B_{d_{local}}}{A_{d_{local}} + B_{d_{local}}}$$

The two former triplets are now obsolete so the sink erases them from its memory.

The sink performs the aggregation process before deciding the direction of its trajectory by parsing the stored triplets in its memory in pairs and until there are no more triplets to aggregate. If three or more triplets corresponding to pockets that are positioned towards the same direction have been delivered to the sink, eventually they will result to a single aggregated triplet.

4. PERFORMANCE EVALUATION

We implement our protocols in the ns-2 simulation platform version 2.33, using the TRAILS toolkit [11], which simplifies the implementation and simulation of complex mobility scenarios and the 802.11 MAC protocol. We have studied two representative scenarios, one including heterogeneous placements for sensors, and a second one including uniform placements. We set the network area to be 500x500m^2, we always position the sink S at (250, 250), the center of the network, except for the Optimized Deterministic Motion in which we position S at (15,15) in order to be fair with the protocol. Otherwise, at the beginning, it would visit two times the same half of the network, and it would take twice such time to visit the other half. For each protocol we run 5 sets of experiments, for 50,100,150,200 and 250 sensors. Each set consists of 20 iterations and computes the mean value of each metric.

For the heterogeneous placement scenario, the sensor placement consists of six pockets, each one of 15x30m^2. In the south-east area of the network lie 2 pockets; one with 23% of all sensor population and the other one with 13% of sensor population. In the north-west area of the network lie four more pockets, each one with 13% of all the sensor population. The remainning sensors are deployed uniformly in the rest of the network area. Neighboring pockets have a distance of 100 meters from each other. This scenario is quite challenging. While the highest density is located in the south-east area of the network, almost half of the sensors population is located in the north-west area (see Figure 2)

The sink has significant energy resources (100Joules) and has a constant speed of $8\frac{m}{sec}$. Each sensor commences with 5 Joules of energy. The sink S transmits beacon messages at a steady pace of $\lambda_{Beacon} = 1$, that is a beacon message per second, and asks for topological information every 3secs. In our application scenario we assume that all sensor nodes record an instance of the environmental conditions producing a fixed number of data messages set to 100. The time interval between two successive messages produced at a node i is not constant, messages are produced at random intervals. However, on the average new messages are produced at rate $\lambda_i = 0,05$ messages/second. Thus, the data generation phase lasts for about 2000sec, we simulate the network for 7000sec, in order to collect delayed data. The data is generated in packets of 36 bytes while the size of a beacon message is 24 bytes.

The transmission range of both nodes and sink is set to R = 15m. The characteristics of the radio module, i.e. the values of e_{trans}, e_{recv} and E_{idle}, were set to match as closely as possible the specifications of the mica mote platform.

4.1 Node movement

We assign different mobility roles to the nodes of the network. We examined cases where the mobility of the nodes changes during the simulation using the mobility transition graphs defined earlier. In particular, we assign C_1 to 25% of the nodes, C_2 to another 25%, C_3 to another 25% and C_4 to the remainning 25% of the nodes, with speed having a mean value of $0.8\frac{m}{sec}$, $3\frac{m}{sec}$, $9\frac{m}{sec}$ and $18\frac{m}{sec}$ accordingly.

4.2 Metrics

Conducting these experiments, we measure several metrics that depict the performance of the protocols. We call success rate the percentage of data messages that were received by the sink over the total number of generated messages. We measure the energy consumed at the sensor network due to communication, as the average number of Joules consumed at each node. We also measure the delivery delay (latency), which is the average time interval between the creation of a message and the time when it is delivered to the sink.

4.3 Performance

Before commenting on the experimental results we would like to point out that the statistical analysis of our findings shows high concentration around the mean. Furthermore, we use constant memory size for each sensor and buffers do not overflow.

Heterogeneous Placement: Figure 4 depicts the success rate achieved by each protocol for the heterogeneous placement scenario. Blind random walk has smaller success ratios as the sink uses no sophistication; it simply walks around the network area and any sensors inside its range deliver sensory data. The other protocols perform relevantly the same, however our-greedy and our-aggregate protocols approximate 100%. This is because our approach ignore sensors that have no sensory data, so the sink tends to move towards sensors that carry data.

In figure 5 we can see the mean latency per data message, that is the average time interval from the time a data message is generated to the time it is delivered to the sink. As expected, blind random walk performs relatively bad under heterogeneous and diverse scenarios. While large populations of sensors are concentrated in small subregions of the network area, the sink by performing random walk wastes much time in areas with very few or no sensors. Latency

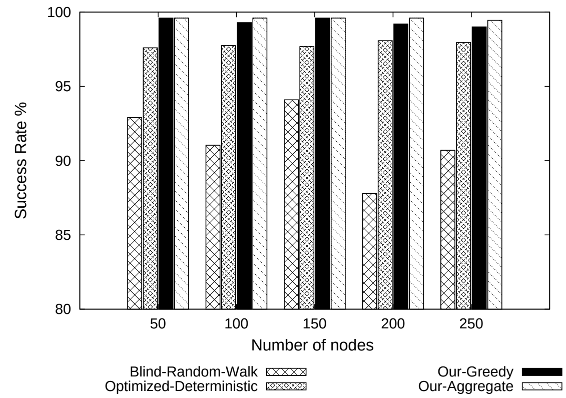


Figure 4: Success rate in heterogeneous placement scenario

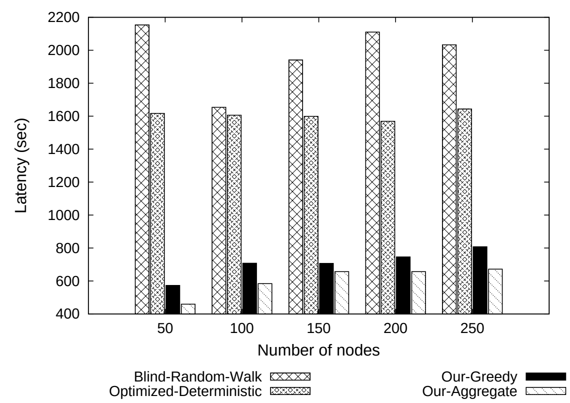


Figure 5: Latency in heterogeneous placement scenario

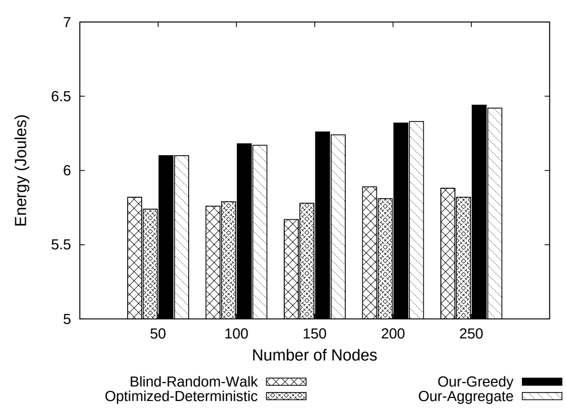


Figure 6: Energy Dissipation in heterogeneous placement scenario

varies from 1630 to 2150 seconds. This diversity exists because of the probabilistic character of blind random walk. Optimized deterministic protocol performs better as it sweeps methodically the network area and eventually all sensors are visited. "Our-greedy" protocol reduces latency by even 330% as it is capable of locating areas with high densities due to the topological information that sensors gather and deliver as they move inside the network. However, "our-aggregate" protocol outperforms all other protocols by even 460%. This is because of the aggregation process that allows the sink to adjust its trajectory and move towards the north-west area of the network, while under the "our-greedy" protocol the sink is first "attracted" by the pocket with 5d density in the south-east area.

Figure 6 depicts the dissipated energy per sensor measured in Joules. The amount of energy dissipated depends on the total number of messages transmitted and received. In blind random walk and optimized deterministic protocols only data messages are transmitted so they perform relatively the same. In our two protocols more energy is dissipated due to the overhead paid by sensors in order to collect topological information. However, this overhead does not exceed 10%.

Uniform Placement

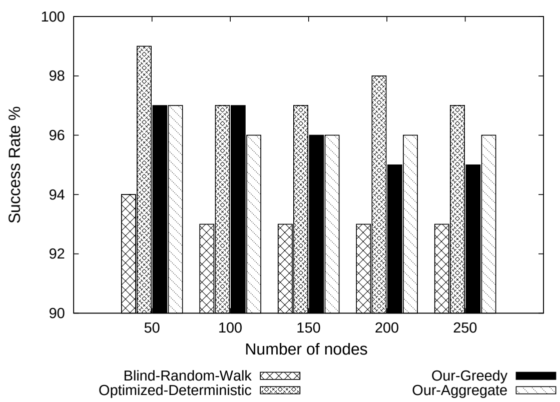


Figure 7: Success rate in uniform placement scenario

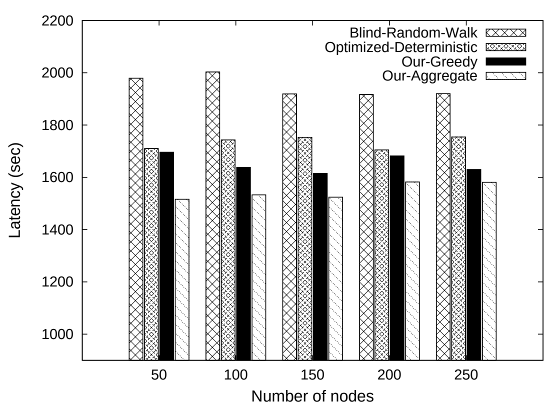


Figure 8: Latency in uniform placement scenario

Our two protocols perform better also in uniform placement because the sink gets "attracted" by sensors which have sensory data to deliver, while it ignores the rest. So it doesn't waste time in areas of no interest or areas that have been visited shortly before. Although the success rate of the deterministic scheme is better than ours, our protocols behave much better w.r.t. latency. Two remarks can be made. First, that the reduction of latency is not so great as in heterogeneous placement (about 30%), which is expected due to the placement. Second, the "our-greedy" and "our-aggregate" protocols have almost identical behaviour. This is because independently of whether the sink uses the aggregation process or not, the expected number of sensors with data around the sink is the same towards all directions and so aggregation does not contribute to the sink trajectory.

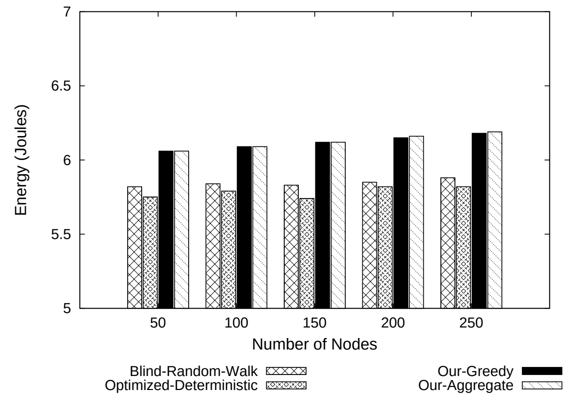


Figure 9: Energy Dissipation in uniform placement scenario

5. CONCLUSIONS AND FUTURE WORK

In this work we studied sensor networks in which *both* sensors and the sink are mobile, and sensors move in a diverse, highly dynamic manner. Motivated by realistic scenarios and applications we focused on heterogeneous sensor placement distributions. We proposed two mobility-based topology exploration protocols, one greedy protocol and one using an aggregation process. In these protocols the sink gets gradually informed about network topology by local information that mobile sensors collect and carry. This way the sink can effectively collect data produced reducing latency even by 460%, compared to relevant solutions, while keeping energy dissipation at low levels. Still, our protocols outperform other ones even in uniform placement scenarios since the sink is able to locate even small traffic discrepancies over time and head fast towards higher traffic regions.

In future work we will further investigate the aggregation aspect; for example how aggressive it is, how the $angle_{thres}$ can be adaptive or even different aggregation functions. Furthermore, we plan to investigate schemes including multiple mobile sinks; for instance, how can multiple mobile sinks be coordinated in an effective way. Finally, to also compare to other state of the art protocols. Also, to investigate the impact of memory size.

Acknowledgements

This work has been partially supported by the IST Programme of the European Union under contract number IST-2005-15964 (**AEOLUS**).

6. REFERENCES

[1] A. Boukerche and K. Abrougui. An efficient leader election protocol for mobile networks. In International Conference on Wireless Communications and Mobile Computing (IWCMC), pages 1129-1134, New York, NY, USA, 2006. ACM.

[2] A. Boukerche, Y. Ren, and Z. Zhang. Performance evaluation of an anonymous routing protocol using mobile agents for wireless ad hoc networks. In 32nd IEEE LCN, pages 893-900, 2007.

[3] L. M. Arboleda C. and N. Nasser. Cluster-based routing protocol for mobile sensor networks. In QShine, page 24, 2006.

[4] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad-hoc network research. Wireless Communications & Mobile Computing, pages 483-502, 2002.

[5] A. Kinalis and S. Nikoletseas, "Adaptive Redundancy for Data Propagation Exploiting Dynamic Sensory Mobility", In the Proceedings of the 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), ACM Press, pp. 149-156, 2008. Also, in the Journal of Interconnection Networks (JOIN), 2009

[6] S.R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In IEEE GLOBECOM, volume 1, pages 377-381 Vol.1, Dec. 2003.

[7] C.M. Angelopoulos and S. Nikoletseas: "Fast Sensory Data Collection By Mobility-based Topology Exploration" in 52nd IEEE International Global Communications Conference 2009 (GLOBECOM)

[8] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design trade-offs and and early experiences with zebranet. In 10th ASPLOS, 2002.

[9] O. Powell, P. Leone, and J. Rolim. Energy optimal data propagation in wireless sensor networks. Journal of Parallel Distributed Computing (JPDC), pages 302-317, 2007.

[10] 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 DCOSS, volume 4026, pages 480-497, 2006.

[11] I. Chatzigiannakis, A. Kinalis, G. Mylonas, S. Nikoletseas, G. Prasinos, and C. Zaroliagis, TRAILS, a toolkit for efficient, realistic and evolving models of mobility, faults and obstacles in wireless networks, 41st ACM/IEEE Annual Simulation Symposium (ANSS), 2008, pp. 23-32.

[12] Ioannis Chatzigiannakis, Athanasios Kinalis, Sotiris E. Nikoletseas: Sink mobility protocols for data collection in wireless sensor networks. MOBIWAC 2006: 52-59.

[13] Ioannis Chatzigiannakis, Athanasios Kinalis, Sotiris E. Nikoletseas, Jose D. P. Rolim: Fast and energy efficient sensor data collection by multiple mobile sinks. MOBIWAC 2007: 25-32