Adaptive Redundancy for Data Propagation Exploiting Dynamic Sensory Mobility *

Athanasios Kinalis
Computer Technology Institute (CTI) and
University of Patras, Patras, Greece
kinalis@cti.gr

Sotiris Nikoletseas
Computer Technology Institute (CTI) and
University of Patras, Patras, Greece
nikole@cti.gr

ABSTRACT

Motivated by emerging applications, we consider sensor networks where the sensors themselves (not just the sinks) are mobile. Furthermore, we focus on mobility scenarios characterized by heterogeneous, highly changing mobility roles in the network. To capture these high dynamics of diverse sensory motion we propose a novel network parameter, the mobility level, which, although simple and local, quite accurately takes into account both the spatial and speed characteristics of motion. We then propose adaptive data dissemination protocols that use the mobility level estimation to optimize performance, by basically exploiting high mobility (redundant message ferrying) as a cost-effective replacement of flooding, e.g., the sensors tend to dynamically propagate less data in the presence of high mobility, while nodes of high mobility are favored for moving data around. These dissemination schemes are enhanced by a distance-sensitive probabilistic message flooding inhibition mechanism that further reduces communication cost, especially for fast nodes of high mobility level, and as distance to data destination decreases. Our simulation findings demonstrate significant performance gains of our protocols compared to non-adaptive protocols, i.e., adaptation increases the success rate and reduces latency (even by 15%) while at the same time significantly reducing energy dissipation (in most cases by even 40%). Also, our adaptive schemes achieve significantly higher message delivery ratio and satisfactory energy-latency trade-offs when compared to flooding when sensor nodes have limited message queues.

Categories and Subject Descriptors: C.2.1 [Network Architecture and Design]: Wireless Communication

General Terms: Algorithms, Design, Performance

Keywords: Adaptation, Data Propagation, Mobile Sensors, Performance Evaluation, Wireless Sensor Networks

*This work has been partially supported by the IST Programme of the European Union under contract number IST-2005-15964 (**AEOLUS**). Also, by the Programme **PENED** under contract number 03ED568, co-funded 75% by European Union – European Social Fund (ESF), 25% by Greek Government – Ministry of Development – General Secretariat of Research and Technology (GSRT), and by Private Sector, under Measure 8.3 of O.P. Competitiveness – 3rd Community Support Framework (CSF).

1. INTRODUCTION

Until recently, wireless sensor networks where considered as very large static collections of smart sensor nodes. In the standard scenario, sensors are largely deployed, i.e., placed at some position and remain static, in areas of interest for fine grained monitoring in different classes of applications [2, 5]. The collected data is disseminated to a *static* control point – data sink in the network, using node to node – *multihop* data propagation, [13, 21]. The vast majority of research in the state of the art that introduces mobility in sensor networks, restrict mobility to the sink(s), e.g., the sink(s) are mobile *but the sensors themselves are assumed to be static.* (see [11, 25]).

Recent advances in robotics and low power embedded systems will make in the near future wireless sensors ubiquitous, ever present in the ambient environment where people live and work. Such availability of sensing information is expected to drive several categories of applications such as surveillance, habitat environment monitoring (sensors on animals), pollution and eco-system monitoring, health care, traffic management etc. In these settings *mobility is a dominant characteristic of the system.* Sensors are embedded in clothes, portable devices, vehicles, robots etc, that move as their owners go about their everyday lives. A feature to note about these applications is that they usually aim at gathering significant amounts of data for statistic processing; immediate reporting of recorded measurements is exceptionally difficult and prohibitively expensive. Instead, measurements can be cached on the sensors and delivered to the reporting center per chance as the nodes move within range; these types of networks are sometimes referred to as delay tolerant networks. Secondly, the information gathering process must not interfere with the system, for example people should not be asked explicitly to move near the reporting center to offload the collected data. Thus, the mobility is in this sense *uncontrollable.* Third, the mobile nodes may follow many *diverse mobility patterns* that also change dynamically with time.

2. RELATED WORK AND COMPARISON

In such highly mobile settings, the protocols and findings of previous research on static wireless sensors networks can not be directly applied (if at all). Routing protocols based on multihop data propagation paths [21] or clustering algorithms [20] can not work; in mobile wireless sensor networks the topology changes constantly thus maintaining paths or clusters is a very expensive operation (and even impossibility results have been conjectured). Also, coverage and localization problems in networks with mobile nodes become more difficult to cope with when compared to the static networks. Efficient solutions in the state of the art become inefficient or even inoperable in networks with mobility. Even well studied algorithms need to be redesigned, in [4] the authors propose a leader election algorithm suitable for mobile wireless networks. Also, in [7] the authors propose a mobility-aware routing protocol, using zone-base information and a cluster-like communication protocol that operates on two different stages: route creation and route preservation. Similarly, in [14] the authors propose minimal path routing protocol based on pseudo clusters for networks with mobile and static nodes. In [9] the authors assume a mixed network of static-anchor nodes that act as support and mobile nodes that organize into clusters. By using a color based scheme for encoding approximate node positions, the proposed routing scheme optimizes the intra-cluster paths to the anchor nodes reducing the energy consumption. In our approach we use a simple limited broadcast based scheme eliminating the need to maintain complex routing structures.

Additionally, new problems arise due to the high dynamics, maintaining system integrity [16] and security [6] becomes more difficult. However, randomized propagation protocols may prove to be more appropriate in such settings since they require a minimum amount of network knowledge. Also, at a local level topology changes are less frequent, thus protocols that require limited local knowledge may be more effective. Furthermore, randomness better balances the load in the network [31, 22].

Sink Mobility. Mobility in wireless sensor networks has been studied mainly from the perspective of having one or more special purpose mobile elements that somehow take over or assist the data collection process. In [25] the authors discuss several interesting issues of mobility and implement and evaluate a small sensor network with one mobile entity that moves back and forth on a straight line. This idea is extended in [23] where multiple mobile sinks move in parallel following linear trajectories, and an algorithm to load balance the data collection process is proposed, under the assumption of full coverage of the network. Using one mobile sink to collect information from a wireless sensor network under different data collection and randomized or fixed trajectory movement strategies is examined in [11]. The results demonstrate significant energy savings on the sensor nodes but at the same time latency tends to significantly increase.

[29] 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. In [19] the authors propose a scheduling algorithm for adjusting the movement of mobile elements to visit nodes according to the message load they produce, thus avoiding message loss due to buffer overflows. [17] investigates the network lifetime when multiple mobile sinks periodically are repositioned with respect to the energy consumption in the network. The authors propose an integer linear program to determine the new locations and a flow-based routing protocol to ensure energy efficient routing. In [30] the sink collects network information from the sensors and alters its trajectory into line segments that are closer to the sensors thus minimizing the data propagation cost.

[28] calculates the optimal trajectory of a mobile sink that minimizes energy consumption while considering multi-hop propagation effects. Under the assumption that all sensor nodes can move, optimization of a single propagation path is examined in [18]. In the case of several mobile sinks, latency can be significantly reduced since the network area can be covered faster. In [26], randomized distributed coordination techniques are examined and show improved performance mainly with respect to latency which decreases. We here note that the use of mobility has also been considered in ad hoc mobile networks [12]. Such works were among the first to introduce the concept of allowing the protocol to dictate the motion of a small team of mobile hosts in order to improve the overall performance of the network.

Sensor Mobility. [24] presents a case study of applying peer-to-peer techniques in mobile sensor networks designed for wildlife position tracking for biology research. This is one of the first papers examining multiple mobile sensors; the authors assume varying mobility and propose a protocol for propagating data to the most likely node to meet the sink based on the history of previous encounters. The authors in [33] propose a data dissemination mechanism for delay tolerant wireless sensor networks. They also choose the fittest nodes to propagate the data based on history. Furthermore, they assume limited queues on each node and propose a mechanism to drop messages from the queues based on the likelihood of delivery of each message. Our approach is similar, we also try to select the best candidates for delivering messages but we assume that node behavior changes, thus instead of using history we choose the best nodes based on their, dynamically calculated, mobility level. Also, we propose and examine much more elaborate variations of mobility patterns, while we adaptively select the amount of redundancy (i.e., the number of message ferrying nodes), in terms of the mobility levels in the network (to benefit from high mobility by reducing redundancy).

In [27] the authors propose a geometric data dissemination mechanism for delivering data to a mobile sink. The authors assume that the motion of the sink is bounded in a specific but arbitrary area of the network and characterize the motion using geometric criteria. In our work we propose more elaborate methods for characterizing mobility that capture more subtle variations both in speed and trajectory. Finally, in [32] the authors investigate a related problem, the trade-off between mobility of nodes and coverage of the network area. Our approach in fact exploits such tradeoffs in the sense that we handle high mobility as a "replacement" for connectivity and coverage. Note that in [3] the authors perform an interesting comparison between randomized nodal movement approaches and schemes proposing controlled mobility of the sink depicting the differences, strengths and weaknesses of these two approaches. Our approach has been inspired by the research in [1, 15] which, although referring to a different network type (radio), emphasizes the importance and impact of high network dynamics.

Our Contribution. While most research on mobility in sensor networks assumes mobility of the sinks only, we here study sensor networks where the sensors themselves move.

Furthermore, we choose to focus (motivated by critical, emerging application scenarios) on diverse, dynamic mobility, characterized by heterogeneous sensor motion profiles that, additionally, highly change over time. In particular, (a) we propose a novel network parameter, the *mobility level*, that, as we show, captures accurately enough several important characteristics (speed, dislocation, change) of motion. (b) While sensory mobility introduces major complications, we in fact exploit it as a replacement for connectivity and data propagation redundancy: we propose adaptive protocols, that propagate less data in the presence of high, diverse mobility and favor relay-sensors with higher mobility levels. (c) We propose a mobility-and progress-sensitive probabilistic message flooding inhibition scheme that further reduces communication cost. (d) We implement our protocols along with diverse mobility profiles and several transition types between them. We perform extensive simulation examining realistic scenarios where sensors have limited queues for buffering messages. The simulation findings demonstrate the efficiency of our mobility-sensitive adaptation schemes, since they manage to reduce latency (even by 15%) while significantly reducing energy dissipation (by even 40%) compared to the non-adaptive protocols. Our adaptive schemes outperform flooding with respect to the successful message delivery ratio and at the same time achieve satisfactory energy-latency trade-offs.

3. MODEL

Mobile sensor networks are comprised of a vast number of ultra-small homogeneous sensor devices, which we also refer to as sensors. Sensors are fully-autonomous computing and communication devices, characterized mainly by the available power supply (battery) and the energy cost of data transmission and by the (limited) processing capabilities and memory. Sensors are equipped with a set of hardware monitors that can measure several environmental conditions.

The network area $\mathcal{A}$ is a flat square region of size $D \times D$; this assumption can be easily relaxed to include general network areas of arbitrary shapes. The initial positions of sensor nodes within the network area are random and in the general case follow a uniform distribution. *Let n be the number of sensors* spread in the network area and *let d be the density of sensors in that area* (usually measured in numbers of $sensors/m^2$). There is a special node within the network area, which we call the sink $\mathcal{S}$, that represents a control center where data should be collected. $\mathcal{S}$ is immobile and passively awaits nodes to pass by it and transmit their data. In order to be detected by the nodes the sink transmits beacon messages at a rate of $\lambda_{Beacon}\ \frac{messages}{sec}$.

Each sensor device has a *broadcast* (digital radio) *beacon mode* of fixed wireless transmission range R, and is powered by a battery. Also a sensor is equipped with *a general purpose storage memory (e.g., FLASH) of size $\mathcal{C}$.* This storage is used to cache messages that need delivery or forwarding.

Let $\mathcal{E}_i$ be the available energy supplies of sensor i at a given time instance. 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 hardware, while ϵ_{trans} is also depended on the square of 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_{elec} (the time unit is 1 second). Overall, there are three different types of energy dissipation: (a) E_T, the energy dissipation for transmission, (b) E_R, the energy dissipation for receiving and (c) E_{idle}, the energy dissipation for idle state. We note that in our simulations we *explicitly measure the above energy costs.*

An important modelling assumption that differentiates our approach from most standard models in the state of the art is the *mobility* of the sensors. Sensor nodes can calculate their position in some common coordinate system (e.g., by using navigational equipment, such as GPS) and are aware of the dimensions of the network area. Sensors are attached to mobile objects; we model their movement through a high level mobility function which we symbolize by $\mathcal{M}$. Note that nodes generally follow different mobility functions and in fact a single node may follow different mobility functions from one time to another. We consider several types of random motion with respect to speed (low, medium, high) as well as with respect to "locality" (local motion within a limited area or global motion covering a large part of the network. We discuss several aspects of mobility modeling in Section 6. The movement of each sensor node i at time t is characterized by a mobility level $M\ell_i(t)$. Since the mobility pattern of a node is subject to change over time, $M\ell_i(t)$ is dependent on the current time instance.

The mobility function returns a position p_t that the node should move to and the speed of the node to reach p_t. Note that the mobility function can be invoked at anytime even before reaching the designated point. The actual mechanism that moves the mobile entity from position p_{t-1} to position p_t is beyond the scope of this paper. However, in order to simplify our model we assume that all changes in speed and direction can be done instantly. More information about the calculation of $M\ell_i(t)$ and the movement of nodes is given in the following sections.

Finally, we assume that a specific, high-level, application is executed by the sensors that form the network. Applications for wireless sensor networks fall in three major categories [5]: (i) *Periodic Sensing* (the sensors are always monitoring the physical environment and continuously report the recorded measurements to the control center $\mathcal{S}$), (ii) *Event driven* (to reduce energy consumption, sensors operate in a silent monitoring state and are "programmed" to notify about events) and (iii) *Query based* (queries can be propagated to the sensors arbitrarily by the control center $\mathcal{S}$, according to the application and/or user's will). We model in an abstract way the kind of high-level application by the message generation rate λ_i in each sensor i.

4. ESTIMATING THE MOBILITY LEVELS

The usual simple metrics studied, like distance traveled or current speed are not enough to fully capture the characteristics of mobility. Nodes with the same speed will travel the same distance independently of the trajectory followed, thus overall distance traveled is not characteristic. Furthermore, depending on the type of the mobility pattern, the area covered may vary significantly. For example, consider a node whose position fluctuates around an anchor point against a node that travels in a straight line. Clearly, the latter can traverse the network area much faster than the former, even if it moves at slower speed. Here we define a new metric for characterizing mobility, the *mobility level*, that captures not only the differences in speed but also the differences in the trajectory followed. Our metric assigns a higher value to nodes that move fast and tend to traverse new areas, while assigns smaller values to nodes moving slow, that traverse the same or neighboring areas frequently.

The computation of the mobility level can be done easily with information locally obtainable by the node. Each sensor node i is responsible for computing its own mobility level. Consider a time interval t_l; node i records its position, let us call it p_0 at the beginning of t_l. Consider an integer $K > 1$, we define $\delta = \frac{t_l}{K}$. Every δ seconds, the current position p_k of the node is recorded. Thus, we record K positions; by measuring the distance from position p_0 of the other $K-1$ recorded positions we obtain $K-1$ measurements of distance $d(k)$ from the origin, for $k = 1, 2 ... K-1$. Let $d_i{}^{max}(t) = max_k\{d(k)\}$; effectively $d_i{}^{max}(t)$ is the maximum dislocation of node i from p_0 during the interval t_l. Similarly, every δ seconds the node records the speed of the node producing a series of measurements $v(k)$ for $k = 1, 2 ... K-1$. Let $v_i(t) = avg\{v(k)\}$. The mobility level of node i at time t is then calculated as:

$$M\ell_i(t) = v_i(t) \cdot d_i{}^{max}(t)$$

Note that after the interval t_l passes the whole process is repeated, at time $t_l + \delta$ the current position is recorded as the next measurement. The oldest recording is erased (i.e., p_0) and the second oldest recorded position becomes the new p_0 and $d_i{}^{max}(t)$ and $v_i(t)$ are recalculated. Effectively, this means that $d_i{}^{max}(t)$ and $v_i(t)$, and subsequently $M\ell_i(t)$ are quantified, their value changes only every δ seconds. Also, note that K is a granularity parameter, the network operator sets K and t_l to appropriate values to adjust the level of detail in the calculation of $M\ell_i(t)$. In Section 6 (and in Fig. 1 and Tab. 1 in particular) we demonstrate how the mobility level $M\ell(t)$ indeed captures mobility diversity.

5. ADAPTIVE DATA DISSEMINATION

To increase the probability of data delivery and at the same time reduce delivery delay, we choose to disseminate a data message to several nodes. Each node disseminates the information it records to a number β of its neighbors at the given time. Thus, nodes carry copies of data recorded from other nodes and deliver them along with their own as soon as they encounter the sink. When a node receives a beacon from the sink it immediately starts to unicast to the sink the contents of its data cache. After a data message is successfully delivered, it is removed from the cache and no further attempts to disseminate this message are done.

However, the selection of β and which particular β neighbors get a copy of a datum requires careful design. By setting $\beta = \infty$ the protocol degenerates to flooding, thus expending a lot of energy of the nodes because of the many redundant packets, however delay should be minimal since the data will follow all possible paths to the sink. On the other hand, setting β to a small number will decrease communication cost a little but at the same time the data will take a long time to reach to the sink.

Intuitively, slow nodes or nodes confined in a small area (i.e., nodes i with small mobility level $M\ell_i$) should choose a larger β to speed up the delivery of data. This is because such nodes will take a long time to move close enough to the sink to deliver the data on their own. By spreading out the information to other nodes the probability of meeting the sink increases and consequently the delay of delivery decreases. On the other hand, faster nodes that cover more area, can benefit of their high mobility and choose a smaller value for β. Such nodes are more capable at locating the sink, thus their messages have a high probability of being delivered rapidly. Similarly, when selecting to which nodes to disseminate a message we face again a dilemma. Intuitively, the faster nodes are more appropriate for message ferrying because they can travel the network faster and thus may approach the sink more frequently.

The general dissemination algorithm followed by the nodes in our approach is the following:

1. New messages describing events that need reporting or messages that where forwarded by other nodes are stored in a FIFO queue, called the *forward queue*. The queue has limited size depending on the cache memory of the node. In case the queue fills, older messages (i.e., at the front of the queue) are discarded to make room for new messages.

2. The node pops the next message from the front of the forward queue, selects β neighbors and transmits the message to them. This process is repeated as long as there are messages in the forward queue.

3. Forwarded messages are then stored in a *delivery queue* that has the same characteristics (size, FIFO etc.) as the forward queue.

4. If a beacon from the sink is received, then the node switches to a *connected* operational state and begins the delivery of the messages. First it selects messages from the delivery queue to transmit directly to the sink. If the delivery queue is empty it selects messages from the forward queue. Note that in this case messages are not send to β neighbors. After successful delivery messages are erased from the memory of the nodes thus freeing resources.

5. If the delivery of a message to the sink fails, the node reverts to the *disconnected* state and operates according to steps 1,2,3.

Note that messages received from other nodes are discarded, in case they already exist in the forward or delivery queue. Next, we present two protocols that use the mobility level to refine the calculation of β and three methods to select the neighbors that will receive a copy of the message.

5.1 Calculation of redundance β

Below we propose two methods for selecting the number of neighbors β to disseminate a message. The cornerstone of our methods is the use of the mobility level of the nodes involved in the process to estimate the requirement for redundancy of message transmissions.

Completely local protocol. Assuming that each node has a maximum mobility capacity, that is its speed is bounded by v_{max}, then the maximum mobility level a node can reach is

$$M\ell_{max} = v_{max} \cdot (v_{max} \cdot t_l)$$

where $(v_{max} \cdot t_l)$ gives the maximum dislocation achieved during time t_l.

A node that moves at maximum mobility level is considered capable of delivering messages, practically without disseminating them to the rest of the nodes. Normally though, nodes will achieve a smaller mobility level, hence cooperation is required between them to deliver the messages. We define β for node i in terms of the current and maximum mobility levels as follows:

$$\beta_i = \left\lceil \frac{D}{M\ell_{max}} \cdot \left(1 - \frac{M\ell_i(t)}{M\ell_{max}}\right) \right\rceil \quad (1)$$

where D is the dimension of the $D \times D$ network area. Eq. 1 is composed of two terms. The fraction $\frac{D}{M\ell_{max}}$ estimates the amount of time an ideally moving node will need to cover the network area, and thus deliver the message. The second term $1 - \frac{M\ell_i(t)}{M\ell_{max}}$ estimates how close node i is to the ideal case. The rational of this function is to calculate large values of β for "slow", locally moving nodes (i.e., with low mobility level $M\ell$), upper bounded by $\frac{D}{M\ell_{max}}$. The opposite happens for "fast", globally moving nodes: as $M\ell_i(t)$ approaches $M\ell_{max}$ the value of β approaches zero, meaning that the node will not redundantly disseminate the message to other nodes but instead transmit directly to the sink as soon as it is within range. Note also that since β_i is dependent on $M\ell_i(t)$, its value also changes over time to reflect the changes in $M\ell_i(t)$, thus the behavior of the node is adapting to its mobility as this is captured by $M\ell_i(t)$. We note that this protocol necessitates completely local network knowledge, e.g., each sensor selects, b using only information about its own mobility level.

Limited awareness protocol. Using simply the mobility level of a node to calculate β may result in unnecessary large values. An extreme example is the case where a "slow" node is surrounded by several "fast" ones. Then transmitting a copy of the message to many of them will result in unnecessary overhead since only a few fast nodes are required to deliver the message efficiently. In such cases, it is more meaningful to determine distributedly β in terms of local observation and comparison of the different mobility levels between the nodes. The method we present here relies on the acquisition of limited knowledge about the network neighborhood, and use of this knowledge to calculate β. A node i before disseminating a message initiates a simple neighbor discovery protocol.

Neighbor discovery protocol. Node i transmits a beacon message announcing its mobility level and its `id`. Nodes that receive the beacon of i respond with a message containing their `id` and mobility level.

Node i then calculates the average mobility level in the neighborhood; assuming $neigh_i(t)$ is the set of all neighbors of node i at time t we have:

$$M\ell_i{}^{avg}(t) = \frac{\left(\sum_{j \in neigh_i(t)} M\ell_j(t)\right) + M\ell_i(t)}{|neigh_i(t)| + 1}$$

In essence, $M\ell_i{}^{avg}(t)$ captures the available mobility at the neighborhood of i at time t. Using $M\ell_i{}^{avg}(t)$ node i can calculate its β as follows:

$$\beta_i = \left\lceil \frac{D}{M\ell_{max}} \cdot \left(1 - \frac{M\ell_i{}^{avg}(t)}{M\ell_{max}}\right) \right\rceil \quad (2)$$

Note that $M\ell_i{}^{avg}(t)$ encapsulates the mobility level of node i, thus it is also taken into account in the selection of β_i. As before, β_i is bounded by $\frac{D}{M\ell_{max}}$ when the available mobility in the neighborhood is low and approaches 0 when the average mobility approaches $M\ell_{max}$.

5.2 Neighbor selection

After calculating β_i, node i needs to select the particular β_i neighbors to deliver the messsage to. As mentioned earlier this selection can influence the overall performance of the protocol, intuitively "fast" nodes at high mobility level should be preferred. However, always selecting the same "fast" nodes will result in uneven workload and strain their resources. Below we present three different strategies for selecting the nodes to disseminate a message to.

Completely Random Selection. Node i selects β_i of its neighbors randomly. In order to do so the node uses the neighborhood information gathered by the neighbor discovery protocol. This simple method probabilistically quarantees that the load distribution will be equally shared by the nodes. It is also particularly relevant in cases of limited network knowledge.

Fittest Candidate Selection. Node i selects β_i of its neighbors such that $M\ell_i(t) < M\ell_j(t)$ where j a neighboring node to i. In this way the fastest neighbors are selected, hoping to reduce latency. In the case where no neighbors with higher mobility level can be found, node i waits for a short period of time and repeats the neighbor discovery process in the hope that it either reaches a new neighborhood or new neighbors approach it.

Probabilistic Candidate Selection. To avoid long delays until finding suitable nodes with higher mobility level and also to reduce the strain imposed on these nodes, we compromise our selection criterion. Again node i selects β_i of its neighbors such that $M\ell_i(t) < M\ell_j(t)$, however if no such neighbors are found the rest of the nodes are examined probabilistically, in a way that favors nodes with high mobility. Let p_j the probability of sending a message to node j; p_j is calculated as follows:

$$p_j = \begin{cases} \frac{M\ell_j(t)}{M\ell_i(t)} & M\ell_j(t) \leq M\ell_i(t) \\ 1 & M\ell_j(t) > M\ell_i(t) \end{cases}$$

Thus, the node examines the neighborhood information and for each neighboring node it performs a probabilistic choice using p_j until the message is send to β_i neighbors.

5.3 Message Flooding Inhibition

Note that although selecting only β neighbors at a time will have the effect of reducing the rate a messsage spreads throughout the network, the propagation of a message is arbitrary and eventually it may be transmitted to every single node. Even when a node k delivers the message to the sink, the rest of the nodes that have a copy of the message will propagate the message to about β neighbors each. Nodes that already store a copy of the message will discard it, however the messsage may still be flooded through the network at a slow pace. Here we present a mechanism to reduce the spread of a message.

We introduce a hop counter h_c contained in each message transmitted; a node i before transmitting a message increases its h_c. Each node j that receives a message performs a probabilistic experiment to decide whether to further propagate the message to its neighbors or to simply store the message in its delivery queue until a sink is located. Let p_{fwd} the probability of forwarding:

$$p_{fwd} = \left(1 - \frac{M\ell_j(t)}{M\ell_{max}}\right) \cdot \left(1 - \frac{h_c}{h_{opt}}\right)$$

where h_{opt} is the optimal number of hops between node j and the sink as given by $h_{opt} = \lceil \frac{dist_{sink}(j)}{R} \rceil$. $dist_{sink}(j)$ is the Euclidian distance of node j from the sink and R the wireless transmission range of nodes. This formula is used when $h_c \leq h_{opt}$; when $h_c > h_{opt}$ then p_{fwd} is set to 0. Since, the location of the sink (hence also $dist_{sink}(j)$) may not always be known, h_{opt} can be calculated by using another distance, for example by setting $dist_{sink}(j) = \frac{D}{2}$. In this way the forward probability depends on the distance the message has traveled (as given by h_c) with respect to the overall required distance (as given by h_{opt}) and the mobility of the node compared to the maximum mobility. Thus a message will not be propagated further than h_{opt} hops and will not be propagated further by a node that moves with maximum mobility capacity ML_{max}. On the other hand, messages that performed few hops are more likely to enter the forward queue. We note that this inhibition mechanism is continued with all strategies for estimating β and the particular β neighbors for data propagation.

6. MODELING DIVERSE MOBILITY

In most real world scenarios most 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. For example, consider a person working in a university campus; for long periods of time she moves slowly in a confined space (e.g., 10 by 10 meters) as she goes about her work in the office. At some point, the person may start walking faster towards a specific direction as she goes to the next building where she continues her work reverting to the previous type of movement. These examples demonstrate the diversity and variability that may arise in networks of mobile sensors. Modeling real life movement patterns is a subject of active research. Clearly simplistic mobility patterns, such as random walk or random waypoint alone, can not accurately capture the heterogeneous mobility characteristics we described before. Here we try to mimic, in a coarse way, several main types of movements inspired from the above observations. Using well defined mobility models, below, we define a few characteristic mobility roles that are used to construct more complex mobility behaviors.

Working movement. Nodes move slowly, the movement is mostly centered in the area around their initial position $\mathcal{M}_{work}$. Such movement can be approximated by a random walk [8] mobility function. We define the function to choose a direction and a distance to move towards that direction. Good parameters for obtaining a local movement are $[0.5, 1.5] m/sec$ for choosing speed and by setting the movement distance towards a direction to be small, e.g., $[1, 5]m$.

Walking movement. Nodes move more quickly than the working mobility and travel in smoother trajectories $\mathcal{M}_{walk}$. Such behavior can be obtained by using a variation of the Boundless Area [8] mobility model. In this model at each time step a random variation of speed and direction are chosen and the next position is calculated by applying the new direction to the current coordinates. When the node reaches the boundaries of the network area we force it to reflect, 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 $\Delta\alpha = 30^o$.

Bicycle ride. This type of movement is similar to the walking movement except that the speed is usually greater and there are less direction changes $\mathcal{M}_{bic}$. Again we use our variation of the Boundless Area [8] mobility model; we bound the speed between $[3, 6]m/s$ (10.8–$21.6km/h$), we set $\Delta t = 3s$, $\Delta v = 0.5m/s$ and $\Delta\alpha = 30^o$.

Vehicular movement. Vehicular movement $\mathcal{M}_{veh}$ is the faster of all with fewer direction changes, we use the Probabilistic Random Walk [8]. 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–$36km/h$). By simulating these mobility roles we obtain the mobility levels shown in Tab. 1.

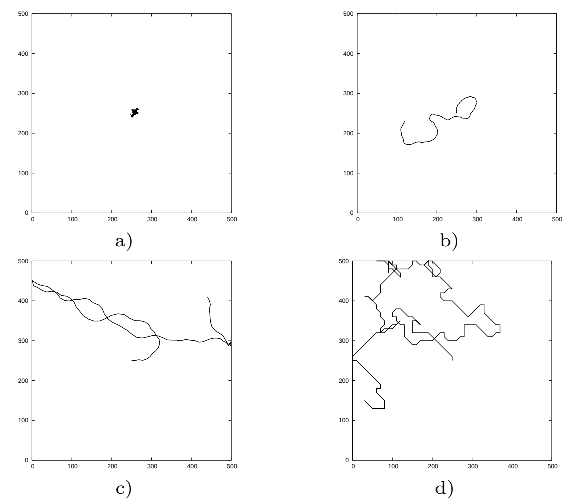

Figure 1: Example movement of a mobile node under the influence of a) $\mathcal{M}_{work}$ b) $\mathcal{M}_{walk}$ c) $\mathcal{M}_{bic}$ and d) $\mathcal{M}_{veh}$.

Role	Avg. $M\ell$	Min. $M\ell$	Max. $M\ell$	$M\ell_{max}$
$\mathcal{M}_{work}$	3.19	0.42	10.79	11.25
$\mathcal{M}_{walk}$	11.37	1.35	19.97	20.00
$\mathcal{M}_{bic}$	99.43	17.05	179.80	180.00
$\mathcal{M}_{veh}$	239.51	37.19	481.05	500.0

Table 1: Average, Minimum, Maximum recorded mobility levels and the theoretical maximum for different mobility roles.

6.1 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, e.g., a person working in an office may walk to his car, drive to another building and carry on working there. To model the dynamics of the mobility of a node, we use a state transition diagram to change between the 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. The following diagrams define various mobility transitions.

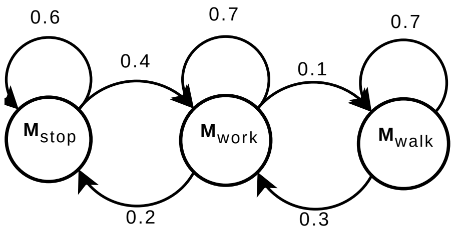

Figure 2: Transition diagram between slow mobility roles.

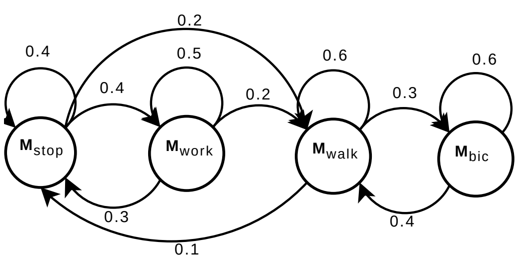

Figure 3: Transition diagram for overall medium mobility level.

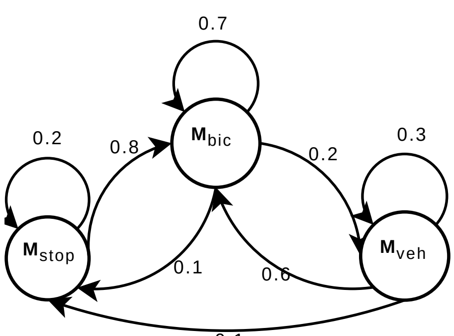

Figure 4: Transition diagram for medium mobility with fast bursts.

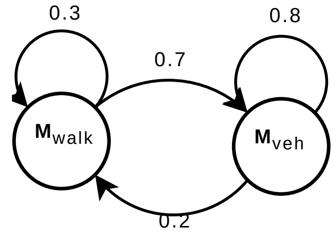

Figure 5: Transition diagram between fast mobility roles.

In Fig.2 the combination of mobility roles produces a slow moving node that covers small distances; we refer to this combination as **C1**. In Fig.3, **C2** is shown that produces a medium movement, resembling a man moving from one place to another at various speeds and staying a little at each point. **C3** in Fig.4 produces a faster movement, resembling a person on a bicycle exercising, circulating through an area, occasionally stopping and occasionally pedaling very fast. Finally, **C4** presented in Fig.5 produces fast movement, resembling a motorcycle while occasionally the driver walks a bit before continuing the ride. By simulating these transitions we obtain the measures of the mobility level shown in Tab. 2, notice how the variance between minimum and maximum values increases while the average mobility level is dominated by the most probable mobility role. The minimum recorded mobility is zero because of the stop state.

Transition	Avg. $M\ell$	Min. $M\ell$	Max. $M\ell$
C1	3.09	0.000000	19.96
C2	35.18	0.000000	179.80
C3	89.85	0.000000	400.36
C4	153.98	1.528439	457.46

Table 2: Average, Minimum and Maximum mobility levels recorded for different mobility transitions.

7. EXPERIMENTAL EVALUATION

We implement our protocols on the **ns-2** simulation platform version 2.30, using the TRAILS toolkit [10] which simplifies the implementation and simulation of complex mobility scenarios. We set the network area to be $1000 \times 1000 m^2$, we always position $\mathcal{S}$ at $(500, 500)$, the center of the network. We deploy in a random uniform manner 300 nodes in the network area. The nodes and sink have significant energy resources (100Joules) to prevent failures due to energy depletion. Also, we do not consider the possibility of other types of node failures.

The sink $\mathcal{S}$ transmits beacon messages at a steady pace $\lambda_{Beacon} = 1$, that is a beacon message per second. In our application scenario we assume that all sensor nodes record an instance of the environmental conditions producing a fixed number of messages set to 20. 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 $\lambda_i = 0.025$ messages/second. Thus, the data generation phase lasts for about $800 sec$, we simulate the network for $3600 sec$ 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. Each node uses fixed sized caches for the forward and delivery queues, each cache can accommodate 64 messages, thus there is possibility of message drops due to caches exceeding their maximum size, i.e., we avoid the ideal case of infinite buffers. The transmission range of both nodes and sink is set $R = 70m$. The characteristics of the radio module, i.e., the values of ϵ_{trans}, ϵ_{recv} and E_{idle}, were set to match as close as possible the specifications of the *mica mote* platform.

Node movement. We assign different mobility roles to the nodes of the network, we performed a wide range of experiments combining the mobility roles described above. We examined mixed mode scenarios where 25% of the nodes follow $\mathcal{M}_{work}$, 25% $\mathcal{M}_{walk}$, 25% $\mathcal{M}_{bic}$ and 25% $\mathcal{M}_{veh}$. The assigned mobility functions remain the same for a particular node during the simulation. In the second experimental setup we present here, the mobility of the nodes changes during the simulation using the mobility transition graphs defined earlier, we assign **C1** to 25% of the nodes, **C2** to another 25%, **C3** to another 25% and **C4** to the remaining 25% of the nodes.

Protocol Comparison. In order to examine the effectiveness of the protocols we propose here, we also implemented and evaluated in these settings two other protocols to use as a point of reference. The first one is the *simple flooding protocol*; we obtain this protocol simply by setting $\beta = \infty$ (i.e., the node will send the message to all its neighbors) without any adaptation and without the message flooding inhibition mechanism. Also, note that this protocol does not execute the neighbor discovery protocol but instead uses broadcast transmissions to simultaneously deliver a message to all neighboring nodes. The second test case protocol is the *fixed β protocol*, which is a non-adaptive version of our main protocol that uses the completely random neighbor selection; we set $\beta \in \{2, 4, 6, 8\}$.

Metrics. Conducting these experiments, we measure several metrics that depict the behavior of our protocols. We call *success rate* the percentage of data messages that were received by all sinks 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. Note that we consider the motion of the nodes to be initiated by the objects/persons/vehicles they are attached onto, so the nodes themselves do not consume energy for movement. We also measure the *delivery delay*, which is defined as the average time interval between the creation of a message and the time when it is delivered to the sink.

7.1 Experimental Results

Accuracy of the mobility level estimation method. We defined the mobility level estimation function as a highly configurable method and with the purpose of obtaining different mobility levels even when there is little variability in the motion, especially when considering subtle variations in the trajectory of mobile sensors. In the experiment we describe here we test this ability. We place a single mobile node in the network area and use controllable motion patterns to limit the variability of its movement. More specifically, the node moves back and forth on a linear segment of length $l = 50m$. In the second pattern, the node moves on the circumference of a circle of radius $r = \frac{l}{2\pi}$; i.e., r is such that the length of the circumference is l. Finally, the node moves on the perimeter of a square of side $\alpha = \sqrt{\frac{l^2}{4\pi}}$, thus the surface of the square is equal to the surface of the circle of the previous pattern. For comparison we also include two randomized motion patterns, variations of $\mathcal{M}_{work}$ and $\mathcal{M}_{walk}$ with steady speed (which we call $\mathcal{M}'_{work}$ and $\mathcal{M}'_{walk}$ respectively). In all cases, we fix the speed of the node to $3\frac{m}{s}$ effectively eliminating the speed component of the mobility level estimation function. This also has the effect that all walks travel the same distance at the given time, hence the randomized walks are comparable to the controlled walks. We simulate the motion of the node for $200sec$, by setting $TL = 5sec$ and $K = 30$. We obtain the following average measures of the mobility level.

Line l	Circle $\frac{l}{2\pi}$	Square	$\mathcal{M}'_{work}$	$\mathcal{M}'_{walk}$
39.48	38.98	33.63	21.61	41.53

Table 3: Comparison of mobility levels for similar motion patterns.

We can see that even in these very similar settings, the calculated mobility level captures, to some extent, the diversity in the mobility of the nodes. Also, as shown in Tab. 1 and Tab. 2, our mobility level method clearly captures high discrepancies in mobility profiles. When mobility changes a lot, the mobility level changes a lot as well.

Evaluation of the protocols. In the first set of experiments we present here, the sensor nodes are divided in four groups, where the nodes of each group follow a specific mobility role as defined earlier, for the duration of the simulation. In Fig. 6 we can see that the highest success rate is achieved by our adaptive protocols with the limited awareness adaptation component (93%). The completely local adaptive protocol is almost in a tie with the fixed β protocol for certain values of β. We can observe that when $\beta = 2$ the delivery ratio decreases. This is expected since this protocol and flooding, rely more on message redundancy rather than message ferrying. Flooding achieves the lowest success rate due to the many packets dropped by the limited sized queues. Considering the neighbor selection strategy we can see that favoring the best candidate, i.e., the one with the highest mobility level, can improve the success rate with respect to the purely randomized and probabilistic neighbor selection strategies.

In Fig. 7 we observe that flooding and the fixed β protocol consume great amounts of energy. The adaptive protocols consume about 40% less energy. In fact due to the great volume of messages transmitted and because of the cost of the neighbor discovery protocol, the fixed β protocol, for $\beta = 8$, consumes even more energy than the flooding protocol. The adaptation techniques do not exhibit any significant difference in between them.

The delivery delay is shown in Fig. 8. We can see that flooding has very low delay, but that is expected since messages farther away from the sink, that exhibit long delays, are most likely to be dropped and thus are not considered in the calculation of the delay. The second fastest protocol is the fixed β protocol, for values $\beta > 4$. The adaptive protocols clearly rely more on message ferrying hence they exhibit high delay compared to flooding but almost match the performance of the fixed β protocols and at the same they re-

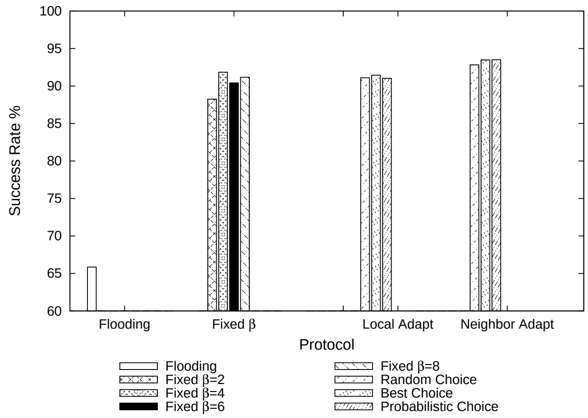

Figure 6: Success rate of the protocols when nodes are assigned a static mobility role.

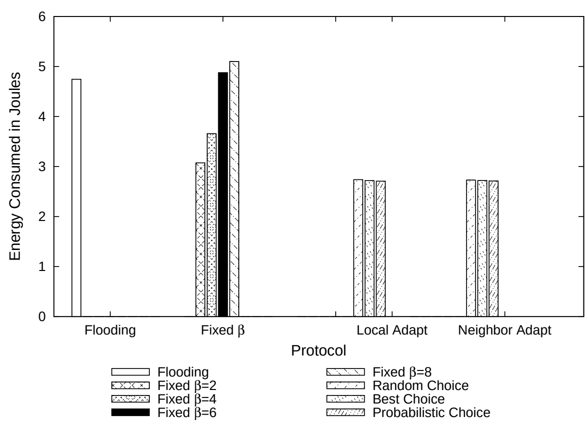

Figure 7: Energy dissipation of the protocols when nodes are assigned a static mobility role.

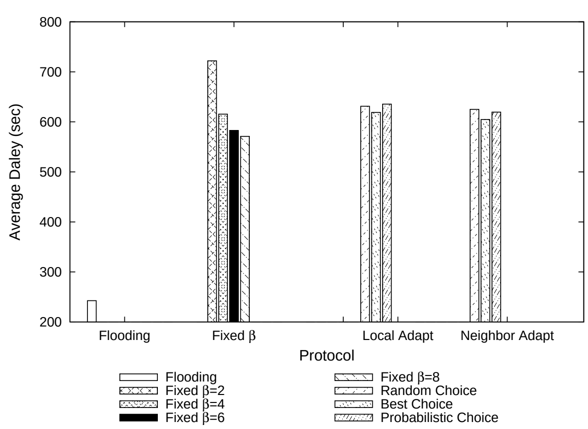

Figure 8: Message delivery delay of the protocols when nodes are assigned a static mobility role.

duce the energy consumption significantly. Also, we observe that the limited awareness protocol achieves slightly lower delay hence our protocols exploit the extra mobility information to improve the delivery process. Considering the neighbor selection strategies, the best candidate selection further reduces the delivery delay, a fact that confirms our intuition about favoring fast nodes. An interesting observation is that the limited awareness and the local adaptation protocols perform very similarly when considering the energy dissipation and delivery delay, but differ in the achieved success rate. This leads us to believe that both protocols roughly produce the same number of messages as this is depicted by the energy dissipation metric, but the limited awareness protocol uses the extra knowledge to more effectively adjust the value of β and thus distribute the redundant messages where needed most.

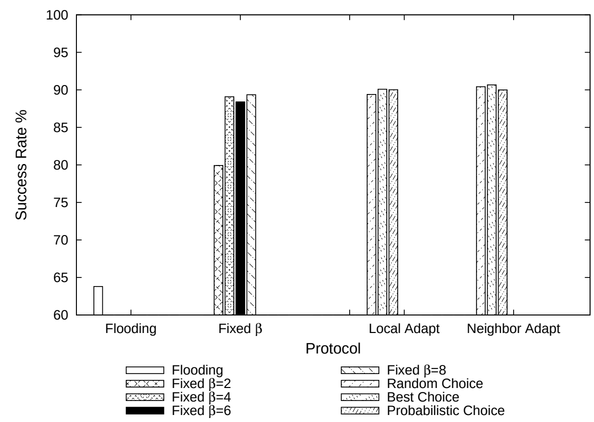

Figure 9: Success rate of the protocols when nodes are assigned dynamically mobility roles.

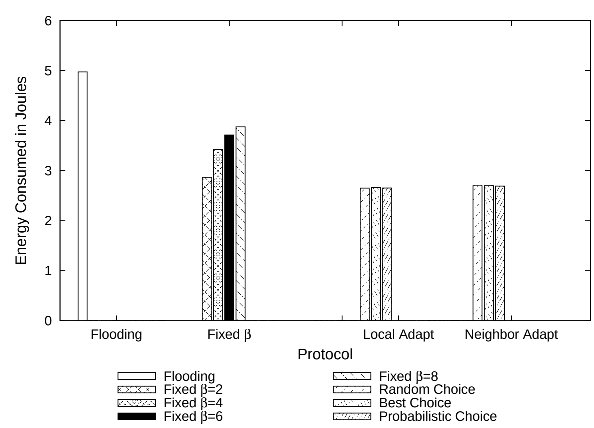

Figure 10: Energy dissipation of the protocols when nodes are assigned dynamically mobility roles.

In the second set of experiments, nodes follow one of the role transition diagrams (**C1**, ..., **C4**) thus their movement is much more dynamic. This highly dynamic mobility has a detrimental effect on the performance of all protocols. The success rate of all protocols is reduced; flooding and the adaptive protocols are less affected by the changes, the fixed β protocol is heavily affected especially for small values of β. The limited awareness protocol is more affected by the dynamics compared to the local adaptation protocol. This is explained by the fact that the more dynamic behavior the nodes exhibit, the more frequent become the changes in

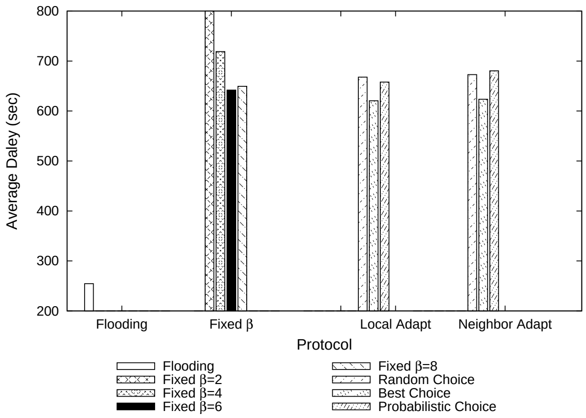

Figure 11: Message delivery delay of the protocols when nodes are assigned dynamically mobility roles.

the neighborhood of nodes, thus the neighbor information cached on a node is quickly outdated. Again the fittest neighbor selection strategy achieves better results; nodes with high mobility, even if completely stop, are more likely to regain the high mobility levels.

Considering the energy dissipation (Fig.10), we observe that protocols generally expend less energy to cope with dynamics; especially the fixed β protocol dissipates significantly less energy with respect to the previous set of experiments. This effect can be explained by the fact that success rate has dropped hence the protocols do not transmit as many messages. The only exception is flooding whose energy dissipation increases a little.

The increased mobility levels and their variance also impact heavily the delivery delay. The fixed β protocol is very affected, it achieves the worst delay of all protocols. The delay of adaptive protocols also increases, however it is still below the fixed β protocol. Flooding also experiences a small increase in the delivery delay. The fittest neighbor choice is again quite reliable since it achieves the lowest delay of all other selection methods.

8. CONCLUSIONS AND FUTURE WORK

In this work we focus on sensor networks where the sensors move in a diverse, highly dynamic manner. We introduced a new network parameter, the mobility level, that captures the heterogeneity and variability of motion in a quite accurate manner. We then proposed adaptive data dissemination schemes that benefit of high mobility levels to reduce energy dissipation, in the sense that sensors mobility serves as a replacement for redundant data propagation. The simulation findings demonstrate significant performance gains and satisfactory trade-offs with respect to non-adaptive protocols as well as flooding.

In future work, we believe that efficient (possibly priority based) message queue-management schemes can further increase the protocol performance. Also, the neighbor discovery process can be further optimized by considering a larger (but still limited) neighborhood for candidate selection. Furthermore, we plan to also rigorously analyze the performance of our protocols by using tools from stochastic processes theory, and also investigate new aspects of random walks on graphs (like cover time when the walk speed varies with time and also with respect to local graph parameters, such as the vertex degree).

9. REFERENCES

[1] A.Clementi, F. Pasquale, A. Monti, and R. Silvestri, *Communication in dynamic radio networks*, 26th ACM Symposium on Principles of Distributed Computing (PODC), 2007, pp. 205–214.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, *Wireless sensor networks: A survey*, Journal of Computer Networks, vol. 38, 2002, pp. 393–422.

[3] S. Basagni, A. Carosi, and C. Petrioli, *Controlled vs. uncontrolled mobility in wireless sensor networks: Some performance insights*, IEEE 66th Vehicular Technology Conference, VTC, 30 2007-Oct. 3 2007, pp. 269–273.

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

[5] A. Boukerche, R. Pazzi, and R. Araujo, *A supporting protocol to periodic, event-driven and query-based application scenarios for critical conditions surveillance*, 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS), 2004.

[6] A. Boukerche, Y. Ren, and Z. Zhang, *Performance evaluation of an anonymous routing protocol using mobile agents for wireless ad hoc networks*, 32nd IEEE Conference on Local Computer Networks (LCN), IEEE, 2007, pp. 893–900.

[7] Liliana M. Arboleda C. and Nidal Nasser, *Cluster-based routing protocol for mobile sensor networks*, QShine '06: Proceedings of the 3rd international conference on Quality of service in heterogeneous wired/wireless networks (New York, NY, USA), ACM, 2006, p. 24.

[8] T. Camp, J. Boleng, and V. Davies, *A survey of mobility models for ad hoc network research*, Wireless Communications & Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications **2** (2002), no. 5, 483–502.

[9] Tai-Jung Chang, Kuochen Wang, and Yi-Ling Hsieh, *A color-theory-based energy efficient routing algorithm for mobile wireless sensor networks*, Computer Networks **52** (2008), no. 3, 531–541.

[10] 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.

[11] I. Chatzigiannakis, A. Kinalis, and S. Nikoletseas, *Sink mobility protocols for data collection in wireless sensor networks*, 4th ACM International Workshop on Mobility Management and Wireless Access Protocols (MobiWac), October 2006, pp. 52–59.

[12] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis, *Distributed communication algorithms for ad hoc mobile networks*, Journal of Parallel and Distributed Computing (JPDC) **63** (2003), no. 1, 58–74.

[13] I. Chatzigiannakis, S. Nikoletseas, and P. Spirakis., *Smart dust protocols for local detection and propagation*, 2005, ACM Mobile Networks (MONET) Journal, Vol. 10 (1), pp. 9–16.

[14] Jae-Young Choi, Jun-Hui Lee, and Yeong-Jee Chung, *Minimal hop count path routing algorithm for mobile sensor networks*, Proceedings of the First International Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06) (Washington, DC, USA), vol. 2, IEEE Computer Society, 2006, pp. 616–621.

[15] A. Clementi, A. Monti, F. Pasquale, and R. Silvestri, *Optimal gossiping in directed geometric radio networks in presence of dynamical faults*, 32nd International

Symposium on Mathematical Foundations of Computer Science (MFCS), 2007, pp. 430–441.

[16] M. Elhadef, A. Boukerche, and H. Elkadiki, *Diagnosing mobile ad-hoc networks: two distributed comparison-based self-diagnosis protocols*, 4th ACM International Workshop on Mobility Management and Wireless Access, 2006, pp. 18–27.

[17] S.R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, *Energy efficient schemes for wireless sensor networks with multiple mobile base stations*, IEEE Global Telecommunications Conference, 2003. GLOBECOM'03., vol. 1, Dec. 2003, pp. 377–381 Vol.1.

[18] D. Goldenberg, J. Lin, A.S. Morse, B. Rosen, and Y.R. Yang, *Towards mobility as a network control primitive*, 5th ACM International Symposium on Mobile Ad-Hoc Networking and Computing (MobiHoc04), 2004.

[19] Yaoyao Gu, Doruk Bozdağ, Robert W. Brewer, and Eylem Ekici, *Data harvesting with mobile elements in wireless sensor networks*, Computer Networks **50** (2006), no. 17, 3449–3465.

[20] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, *Energy-efficient communication protocol for wireless microsensor networks*, 33rd IEEE Hawaii International Conference on System Sciences (HICSS 2000), 2000, p. 8020.

[21] C. Intanagonwiwat, R. Govindan, and D. Estrin, *Directed diffusion: A scalable and robust communication paradigm for sensor networks*, 6th ACM/IEEE Conference on Mobile Computing (MOBICOM 2000), 2000, pp. 56–67.

[22] A. Jarry, P. Leone, O. Powell, and J. Rolim, *An optimal data propagation algorithm for maximizing the lifespan of sensor networks*, Distributed Computing in Sensor Systems (DCOSS), 2006.

[23] D. Jea, A. Somasundara, and M. Srivastava, *Multiple controlled mobile elements (data mules) for data collection in sensor networks*, 1st IEEE Conference in Distributed Computing in Sensor Systems (DCOSS), 2005, pp. 244–257.

[24] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, *Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet*, 10th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS02), 2002.

[25] A. Kansal, A. Somasundara, D. Jea, M. Srivastava, and D. Estrin, *Intelligent fluid infrastructure for embedded networks*, 2nd ACM/USENIX Conference on Mobile Systems, Applications, and Services (MobiSys04), 2004.

[26] A. Kinalis and S. Nikoletseas, *Scalable data collection protocols for wireless sensor networks with multiple mobile sinks*, 40th ACM/IEEE Annual Simulation Symposium (ANSS) (Norfolk, VA, USA), March 2007, pp. 60–69.

[27] C. Liu and J. Wu, *Scalable routing in delay tolerant networks*, Mobihoc, 2007.

[28] J. Luo and J-P. Hubaux, *Joint Mobility and Routing for Lifetime Elongation in Wireless Sensor Networks*, 24th IEEE INFOCOM, 2005.

[29] J. Luo, J. Panchard, M. Piorkowski, M. Grossglauser, and J-P. Hubaux, *Mobiroute: Routing towards a mobile sink for improving lifetime in sensor networks*, IEEE International Conference on Distributed Computing in Sensor Networks (DCOSS), vol. 4026, 2006, pp. 480–497.

[30] M. Ma and Y. Yang, *Sencar: An energy efficient data gathering mechanism for large scale multihop sensor networks*, IEEE International Conference on Distributed Computing in Sensor Networks (DCOSS), vol. 4026, 2006, pp. 498–513.

[31] O. Powell, P. Leone, and J. Rolim, *Energy optimal data propagation in wireless sensor networks*, J. Parallel Distributed Computing **67** (2007), no. 3, 302–317.

[32] W. Wang, V. Srinivasan, and K-C. Chua, *Trade-offs between mobility and density for coverage in wireless sensor networks*, Mobicom, 2007.

[33] Y. Wang and H. Wu, *DFT-MSN: The delay/fault-tolerant mobile sensor network for pervasive information gathering*, INFOCOM, 2006.