DEEP: Density-based proactive data dissemination protocol for wireless sensor networks with uncontrolled sink mobility

Massimo Vecchio a, Aline Carneiro Viana a, Artur Ziviani b,*, Roy Friedman c

a INRIA-Saclay, 91893 Orsay, France
b National Laboratory for Scientific Computing – LNCC, 25651-075 Petropolis, RJ, Brazil
c Computer Science Department, Technion, Israel

Abstract

This paper investigates proactive data dissemination and storage schemes for wireless sensor networks (WSNs) with mobile sinks. The focus is on schemes that do not impose any restrictions on the sink’s mobility pattern. The goal is to enable the sink to collect a representative view of the network’s sensed data by visiting any set of x out of n nodes, where x ≪ n. The question is how to obtain this while maintaining a good trade-off between the communication overhead of the scheme, the storage space requirements on the nodes, and the ratio between the number of visited nodes x and the representativeness of the gathered data. To answer this question, we propose density-based proactive data dissemination Protocol (DEEP), which combines a probabilistic flooding with a probabilistic storing scheme. The DEEP protocol is formally analyzed and its performance is studied under simulations using different network densities and compared with a scheme based on random walks, called RuWMS.

1. Introduction

A wireless sensor network (WSN) may consist of hundreds or even thousands of low cost nodes communicating among themselves [1]. Classical application scenarios for WSNs include, e.g., environmental and structural monitoring, event detection, and target tracking. In many of these applications, tiny sensor nodes are deployed in an area to be monitored. Each node is a small device able to collect information from the surrounding environment through one or more sensors, to process this information locally, and to exchange data through a wireless channel. Since nodes are typically powered by small batteries, which in general cannot be easily changed or recharged, one of the main concerns in deploying real and functional WSN applications is saving energy to extend the network lifetime.

The data flows in WSNs are targeted towards special nodes called sinks (also referred to as base stations) using multi-hop communication among sensor nodes in the monitored area. A sink may link the WSN to another network (like a gateway) to propagate the sensed data for further processing. Sinks usually have enhanced computational, storage, and power capabilities over simple sensor nodes since they must do more complex data processing. Many WSN applications consider fixed sinks collecting data sent by the sensor nodes within the monitored area. Nevertheless, nodes close to the sink, acting as relays in a multi-hop scenario, typically consume more energy due to data forwarding than far away nodes and thus risk running out of battery earlier than the average node. Once these nodes nearby the sink die, the network becomes disconnected, thus compromising the network lifetime. Additionally, fixed sinks can become vulnerability points of the network.

To alleviate the problems of fixed sinks, a new paradigm has been promoted in several recent works: instead of the data flowing towards fixed sinks, mobile sinks traverse the network to collect the data from the sensor nodes. Here, we focus on such WSNs with mobile sinks.

Data management proposals in WSNs with mobile sinks are characterized by their data dissemination strategy (how data is distributed in the network) combined with the data collection strategy (how the mobile sink gathers the monitored data made available by the sensors in the network) [2]. Data dissemination may be performed in two ways:

1. Proactively, if the monitored data is proactively distributed and stored throughout the network for being later retrieved by the mobile sink [3–5].
2. Reactively, if data is sent towards the mobile sink as a reaction for the detection of sink’s presence or queries [6–8].

Depending on how data is disseminated, the mobile sink may need to follow a predetermined trajectory in which it needs to visit specific nodes or locations in the network. Most previous work has...
been carried out on determining trajectories for mobile sinks in conjunction with proactive and reactive data dissemination approaches [9,10,5,3,4]. However, avoiding restricting the mobile sink to specific trajectories is both beneficial to the mobile sink and to the sensor nodes. This is because it frees the network to self-organize and to better respond to changing conditions. Moreover, various aspects of the environment and terrain that can only be discovered at deployment time, and may change over the lifetime of the network, may prevent a mobile sink from following a previously computed trajectory.

In this paper, contrary to previous work, we investigate proactive data dissemination strategies that allow a mobile sink to gather a representative view of the monitored region covered by n sensor nodes by visiting any x nodes, where x \ll n. In particular, in the strategies we consider, the mobile sink is free to choose its own trajectory in any way it sees fit at any given moment. This approach does not require a priori knowledge of all network nodes or any mechanism to track the sink. Still, it enables the mobile sink to follow any trajectory while collecting the data. The only condition imposed on the mobile sink in order to retrieve a representative view of the monitored data is on the total number of nodes the mobile sink needs to visit.

A key point to notice is that enabling a mobile sink to follow a completely uncontrolled visiting trajectory does not limit the applicability of the proposed solutions in real-world WSN scenarios. Indeed, a mobile sink's trajectory may often be constrained by physical obstacles (i.e. buildings, streets, trees, and so on). Yet, in our approach, the information is proactively disseminated such that the mobile sink will be able to gather a representative view of the data regardless of its mobility pattern and the constraints imposed on it by real-world effects.

Intuitively, in order to achieve this full decoupling of data dissemination and storage from the mobile sink's trajectory, the sensors' data need to be distributed in a uniform manner in the network. In a previous work [11], we have proposed to obtain this by applying a reverse sampling technique called RaWMS [12] for data dissemination based on random walks (RWs). The choice of applying RaWMS in this case was made since it is a proven technique for uniform data dissemination. As an outcome for data collection efficiency, we have shown that if every node stores the information of \(\sqrt{n}\) nodes, then having the mobile sink visit any \(2.3\sqrt{n}\) nodes enables it to gather the data of at least 90% of the nodes. For example, in the case of a WSN with 1000 nodes, this means visiting only 7% of the nodes (about 73 nodes) to retrieve a representative view of the monitored area (i.e. data from at least 90% of nodes). Note that this percentage decreases further as the number of nodes in the system increases.

Although the results of adopting RWs to uniformly distribute the monitored data throughout the network are quite encouraging in terms of data collection efficiency, they incur a high communication overhead. The key problem we tackle in this current paper is how to obtain effective uniform distribution of the monitored data for efficient data collection at a reduced communication overhead.

An alternative to RWs for data dissemination could be to flood the information throughout the network, and have each node decide to locally store each piece of information with a probability that is proportional to the ratio between its buffer size and the total number of nodes. However, flooding is known to also generate a large number of messages, and suffers from the broadcast storm phenomenon [13], which causes a large number of collisions and message losses. As a consequence, various probabilistic approaches for controlled flooding have been recently proposed, including for example push-based gossip [14,15], counter-based flooding [16,13,17], and RAPID [18], which combines these approaches with an additional pull-based gossip facet.

In this paper, we explore the feasibility and actual efficiency of utilizing a variant of RAPID for data dissemination coupled with a probabilistic storage policy and propose density-based proactive data dissemination Protocol (DEEP). In particular, the challenges of DEEP are how to properly set the forwarding and storing probabilities used in the proactive data dissemination approaches. These should achieve efficient data dissemination and storage for effective later data collection by the mobile sink, yet at a reduced communication overhead.

The adopted storing probability of DEEP presents a trade-off between the replication level (how many times each data item is stored in the system), which translates to memory requirements, and the number of nodes the mobile sink has to visit, which is related to the data collection efficiency. Loosely speaking, the average replication level in the system is proportional to the storing probability. On the other hand, the number of nodes the sink has to visit is inversely proportional to the storing probability. In this paper, we analyze these relationships both formally and by simulations. We also provide a detailed performance evaluation of our new proactive data dissemination protocol and compare it with RaWMS. The results show that DEEP manages to distribute the data well enough so that the ratio between the collected data and the number of visited nodes is similar to the one provided by RaWMS. Thus, we have a system in which a mobile sink can freely visit a relatively small number of nodes and still collect a representative view of the sensed network. Yet, the communication cost of this system is relatively small.

The remainder of this paper is organized as follows. In Section 2, we discuss related work in the context of WSNs with mobile sinks. In Section 3 we present the system model and discuss the design goal of this paper. Section 4 introduces our proactive data dissemination strategy called DEEP as well as the RaWMS strategy, used as a theoretical upper bound in our performance analysis. The experimental results of our performance analysis are presented in Section 5. Finally, in Section 6 we summarize the outcomings of our analysis.

2. Related work

Data management in WSNs with mobile sinks includes both data dissemination (i.e. proactive or reactive) and data collection (i.e. controlled or uncontrolled) [2]. In this section, we briefly overview the related work in this context considering the combinations of different data dissemination and data collection strategies.

In reactive data dissemination strategies, sensors react to the presence of the mobile sink by, for instance, receiving a query from it. The sensors react by disseminating (actually routing) data towards the current position of the sink or to other nodes that are located close to the predicted sink trajectory. References [19,20,6] are examples of reactive dissemination approaches where the mobile sink follows a controlled, and thus predictable, trajectory as it traverses the monitored area. In this case, the sink must visit some predefined nodes to retrieve a representative view of the monitored data. On the other hand, reactive dissemination approaches where the mobile sink follows a free, i.e. uncontrolled, trajectory, require the sensors to track the sink mobility in order to adapt or influence the data dissemination. Hence, the sink only collects data from sensors along its trajectory or from sensors that route their data towards the sink. Examples of reactive data dissemination with uncontrolled sink mobility are [8,21,7,22–24]. In particular, some authors have investigated approaches that learn (i) the past movement patterns of the sink(s) in order to predict its future locations or (ii) the node's past success rate of transmitting data packets to the sink in order to use the fittest nodes as relays to the sink. These approaches, however, do not guarantee any desired data
delivery ratio. In summary, reactive proposals dealing with sink mobility, both controlled and uncontrolled, must dedicate a significant amount of resources to track the sink and to forward the data on-the-fly to be collected towards the mobile sink.

In proactive data dissemination strategies, however, sensors disseminate their monitored data towards a subset of sensors that play the role of storage nodes. Under controlled sink mobility, this subset of nodes typically form a virtual structure within the WSN that make the data available to be retrieved later by the mobile sink. The mobile sink should then visit nodes belonging to the formed virtual structure to collect a representative view of the monitored data, i.e. the mobile sink has its trajectory determined (controlled) by the location of the nodes forming the virtual structure. Such proactive data dissemination solutions with a rendezvous virtual structure basically differ on how the virtual structure is formed within the WSN. Some examples of virtual structures for this purpose are a grid structure such as TTDD [5] and a line-based one [4]. Hamida and Chelius [3] provide a recent survey of proactive data dissemination strategies using virtual structures for WSN with mobile sinks. Proactive proposals based on rendezvous virtual structures must however keep knowledge about the nodes relative position. This is needed to forward data towards the nodes that participate in the virtual structure. Furthermore, nodes in the virtual structure and nodes close to them are subject to a higher demand in terms of storage and transmission resources, leading to an imbalance in energy consumption among the sensor nodes within the WSN.

If the mobile sink follows an uncontrolled mobility pattern, no structure can be defined since no indication is provided a priori about the trajectory of the sink. In this case, the ideal dissemination would be a uniform data distribution throughout the WSN in a way that allows a sink following an uncontrolled trajectory to retrieve a representative view of the monitored area by visiting a relatively small number of any nodes in the network. An example of such an approach is a previous work of ours [11] in which the data dissemination is performed based on random walks (RWs). This approach leads to a full decoupling of the data dissemination from the sink trajectory management. However, it incurs in high communication overhead. To counter this issue, in this paper, we introduce our probabilistic dissemination strategy called DEEP and present a detailed simulation-based analysis of the trade-offs involved in adopting it to deal with uncontrolled sink mobility.

3. Rationale

In this section we present the considered system model, the design goals, and the adopted evaluation methodology.

3.1. System model

We consider a large number \(n\) of sensor nodes placed at uniformly random locations in a square universe given by a geographical area, for collecting data or monitoring events. Data can be collected by a single mobile sink travelling through uncontrolled trajectories in the monitored region. We assume that the sink has no significant resource limitation – in particular, as compared to ordinary sensor nodes. All sensors and sink are uniquely identified. Sensors are all equal in the sense that they have the same attributes, i.e. computational, memory, and communication capabilities. No synchronization in the transmission of nodes is assumed nor required.

A sensor node is a device owning an omni-directional antenna that enables wireless communication and thus, exchange of messages. Each node \(i\) is able to directly communicate wirelessly with a subset of reachable nodes we refer to this subset as the neighbors of the node \(i\), denoted \(N(i)\). In other words, each node \(i\) can communicate with another node \(j\) if \(j\) is located within \(i\)’s transmission range: a transmission disk centered on \(i\) with radius \(r_c\). We assume bidirectional communication and that the network is dense enough so that it is connected [25].

Each sensor node in the network is equipped with a given buffer space that is utilized to store proactively disseminated data from other nodes for later retrieval by the mobile sink. This buffer space is referred to as the node’s partial view. Hence, each node may act as a storage node for some other nodes in the WSN, but not for all of them. The goal is thus to limit the storage requirement in each individual node and to enable each node to determine the contents of its own partial view independently of other nodes, without impairing the ability of the sink to collect data. After partial views are provided to nodes by the data dissemination strategy, a mobile sink can visit the nodes and retrieve the information stored in their partial views. By slight abuse of terminology, we interchangeably use the term partial view both for the actual information stored at a given node \(p\) and for the IDs of the nodes whose information is stored at \(p\).

Clearly, for space efficiency, the data stored in partial views as well as data transmitted between nodes can be compressed, e.g., using techniques such as those described in [26–28]. Generally, if low latency is not required by the application (e.g., in continuous monitoring), the data collected by sensor nodes can be locally compressed before being disseminated, reducing the network traffic and thus prolonging the network lifetime. For example, in [26] the authors described a lossless compression algorithm that can be implemented in a few lines of code, requires very low computational power, compresses data on-the-fly, and uses a very small dictionary whose size is determined by the resolution of the analog-to-digital converter on board sensor nodes. These main characteristics make the algorithm particularly suited to the reduced storage and computational resources of habitat monitoring WSN nodes, reaching compression ratios up to 70% on environmental datasets. As compression is orthogonal to our current work, we do not further elaborate on it in this paper.

3.2. Design goals and evaluation methodology

We are seeking a solution to the proactive dissemination and storing problem that will excel in the metrics listed below. In addition, for scalability and robustness reasons, the solution should only utilize local information. Finally, the considered solution cannot rely on any information about the mobile sink mobility pattern, as doing so would restrict the latter’s movements.

3.2.1. Metrics

We evaluate the quality of the considered solutions both analytically and by simulations based on the following metrics, as elaborated in the sequel:

**Data gathering efficiency**: The percentage of nodes whose information has been gathered after the sink has visited a given number of distinct nodes. In particular, the data gathering of the mobile sink is more efficient as each visited node provides more new information with respect to the information the sink already has. A data gathering scheme is considered efficient if the mobile sink is able to gather a representative amount of network’s monitored information by visiting only a relative small number of nodes.

**Message overhead**: the number of messages transmitted by every node in the network.

**Quality of data distribution**: the number of storage nodes that hold each node’s information (also called replication level) and the uniformity of the data distribution in the network. Both
measures influence the data gathering efficiency and the robustness of the solution, as will be described later.

4. Dissemination protocols

In this section, we present RaWMS, a previous data dissemination protocol based on RWs, that we use for comparison purposes, and we also introduce DEEP, our proposed protocol.

4.1. RaWMS-based dissemination

This strategy was introduced in [12,11]. Briefly, the goal of this strategy is to ensure that the partial view of each node will correspond to a uniform sample of the entire information (or entire set of nodes) in the system. To construct these partial views, an appropriate number of maximum degree Random Walks (MD RW) [12] is started, where each RW carries the information and ID of the node that started it.

More specifically, a random walk is a process in which a carried data item is repeatedly forwarded (as a message) from a node to one of its randomly chosen neighbors until the end of the random walk length (i.e., the number of hops the RW should last). Thus, in the RaWMS-based dissemination, the data carried by each RW is stored at the node in which the RW stops, which is known as reverse sampling. For a RW that starts in a node \( p \) and ends in \( q \), we say that \( q \) sampled (the information of) \( p \). Moreover, \( q \) is the storage node for the corresponding information of \( p \). Fig. 1 illustrates the RaWMS-based dissemination strategy, as elaborated below.

To better understand RaWMS, we remind the reader a couple of additional basic terms in random walks theory. Given the random nature of random walks, the likelihood that the random walk has reached each node in the network can be represented by a probability distribution. This distribution may change after each step of the random walk (in particular, before the first step the probability for being at the sender is 1 and 0 for all other nodes). Yet, for many network graphs, after “enough” steps have been taken, the distribution no longer changes with each additional step. This final distribution is called the stationary distribution of the (network) graph. The minimal number of steps required to reach the stationary distribution, when exists, is called the mixing time of the (network) graph. The network graph of ad hoc networks is often modelled as a random geometric graph. As been shown in [12], these graphs always have a stationary distribution. Moreover, when the random walk is a maximum degree random walk, this distribution is also uniform, meaning that the probability of ending at any node in the network is the same.

In summary, with the RaWMS-based dissemination strategy, a maximum degree RW is started every \( \phi \) time units at each source node by randomly selecting the next neighbor to send the message to; the same process is repeated at each node along the RW until the hop distance \( d \), which should equal the network’s mixing time, is reached. As been shown in [12], the mixing time of sparse connected networks is roughly \( \frac{\sqrt{n}}{d} \). Further, as the network becomes denser, its mixing time is reduced.

As mentioned above, this ensures that the resulting views resemble uniformly chosen views and thus there is no correlation between the views of neighboring nodes. The average view size obtained is of \( \sqrt{n} \) with an expected intersection of \( \sqrt{n} \frac{2}{3} \), for all network sizes, when adopting a sufficiently large random walk length (e.g., \( n/2 \), as explained above). Based on this analysis, one may expect nearby nodes to have little information intersection between them. Note that this feature helps improve the data gathering efficiency (as defined in Section 3.2) because the mobile sink tends to gather a larger amount of new information in each visited node as the intersection of partial views of nearby nodes is small.

Additionally, to deal with message losses, a salvation mechanism is included in this strategy: if a low level ack is not received for a given message sent by some node participating in the RW, then another random neighbor is chosen and the RW is passed to it. In RaWMS, each node may set its partial view size independently. Yet, for the purpose of analysis, we assume that all nodes have the same partial view size \( k = \sqrt{n} \). Since there is no way to a priori know where a RW will end, two or more RWs may end in the same node. Thus, to guarantee a target partial view size of \( k \) distinct nodes, each source node starts \( n \ln (\frac{1}{\alpha}) \) RWs, for \( n \) nodes in the network (refer to [11] for more details). Following the complexity analysis performed in [12], the communication complexity of this strategy (discounting the salvation mechanism, which according to the empirical results of [29] has only a marginal communication overhead impact), equals \( O\left(\frac{\sqrt{n}}{d}\right) \), for \( d = \frac{n}{2} \), and in general \( O(n^2) \), for \( d = O(n) \).

As been shown in [12], when using RaWMS, both the content of each partial view and the replication of each node in the views of other nodes are uniformly distributed. Moreover, as mentioned in [29], the probability that the information of a given node \( p \) will not be collected by the mobile sink after visiting \( l \) nodes is

\[
\prod_{i=0}^{l-1} \frac{n-k-i}{n-l} = \prod_{i=0}^{l-1} \frac{n-k}{n-l} = \left(1 - \frac{k}{n}\right)^l, \tag{1}
\]

where \( k \) is the partial view size. As this is true for any node, the expected number of nodes whose information has been recovered after visiting \( l \) nodes, i.e., the expected data gathering efficiency, is given by

\[
n \cdot \left(1 - \prod_{i=0}^{l-1} \frac{n-k-i}{n-l}\right). \tag{2}
\]

4.2. DEEP probabilistic dissemination

This strategy is based on an efficient probabilistic forwarding and storing strategy. The probabilistic forwarding is inherited from (is set according to) the RAPID-like dissemination mechanism [18]. In particular, the data dissemination strategy of this approach employs a combination of density sensitive probabilistic forwarding with deterministic corrective measures, as described in [18]. Essentially, the goal of RAPID is to ensure that there will be a predefined average number of (re)transmissions of each message in
each neighborhood. This is obtained as follows: each node $p$ that receives a message $m$ for the first time, decides to rebroadcast $m$ immediately with probability $f = \min\left(1, \frac{1}{2\beta n}\right)$, where $N(p)$ is the one hop neighborhood of $p$ and $\beta$ is the desired average number of retransmissions in each neighborhood. Additionally, to overcome situations in which due to the probabilistic nature of this process, no node decides to transmit in a given neighborhood, RAPID complements this with a semi-deterministic corrective measure. Specifically, if a given node $q$ decides not to retransmit $m$ but does not hear any other retransmission of $m$ after a (relatively long) randomly selected period of time, then $q$ rebroadcast $m$ after all.

RAPID includes an additional deterministic pull-based gossip mechanism that we decided not to use here due to its long latency and added communication cost. A study of how the setting of $\beta$ influences the reliability and latency of RAPID appears in [18]. As mentioned before, we take the above described subset of RAPID and couple it with a storing strategy by which every node that receives a message carrying the data sensed by a given node stores this data with probability $s$. As we target partial views of size $\sqrt{n}$, we have also set $s = \frac{n}{\sqrt{n}}$. We denote the resulting mechanism DEEP. Fig. 2 shows a flowchart describing the operation of the DEEP dissemination strategy at each node.

4.2.1. Theoretical analysis

The communication complexity of this strategy depends on the network density and is determined by the number of retransmissions each node performs. This results in a total communication complexity of $(f \times n) \times n = O\left(\frac{n^2}{2\beta n}\right)$.

Assuming that the probabilistic flooding protocol gives extremely high reliability levels, then the information of each node is stored at each other node with probability $s = \frac{n}{\sqrt{n}}$. In other words, the storing of the information of any given node can be viewed as a Bernoulli trial with probability $s = \frac{n}{\sqrt{n}}$. This means that the replication level of the nodes’ information follows the binomial distribution. Moreover, the probability that a given node’s information has not been gathered by the mobile sink after visiting $l$ nodes is

$$(1 - p)^l = \left(1 - \frac{1}{\sqrt{n}}\right)^l. \tag{3}$$

As this is true for any given node, the expected data gathering efficiency after visiting $l$ nodes is given by

$$n \cdot \left(1 - \left(1 - \frac{1}{\sqrt{n}}\right)^l\right). \tag{4}$$

Recalling (2), the following inequality is verified:

$$n \cdot \left(1 - \prod_{i=0}^{l-1} \frac{n - k - i}{n - i}\right) \geq n \cdot \left(1 - \left(1 - \frac{k}{n}\right)^l\right). \tag{5}$$

By substituting $k$ with $\sqrt{n}$ in (5), we obtain

$$n \cdot \left(1 - \prod_{i=0}^{l-1} \frac{n - \sqrt{n} - i}{n - i}\right) \geq n \cdot \left(1 - \left(1 - \frac{1}{\sqrt{n}}\right)^l\right). \tag{6}$$

From (6) we notice that the data gathering efficiency of both RaWMS and DEEP are similar. Yet, in the case of RaWMS, $n \cdot \left(1 - \left(1 - \frac{1}{\sqrt{n}}\right)^l\right)$ is a lower bound approximation, and hence RaWMS is expected to be slightly better than DEEP. As $n$ becomes larger while $l$ is kept much smaller than $n$ (e.g., $l$ is kept on the order of $\sqrt{n}$ or less), the difference between the data gathering efficiency of RaWMS and DEEP diminishes. Therefore, for medium and large networks, the theoretical analysis suggests that DEEP is a very good substitute for RaWMS, provided that it can obtain very high reliability ratios with small values of $\beta$. In the next section, we will prove that this is the case.

An additional issue to consider when evaluating DEEP, or any other flooding-based protocol (probabilistic or not) is the inherent memory requirement of such protocols due to the need to suppress duplicates. Specifically, when a message $m$ is flooded in the network, we can refer to $m$ as alive for the duration of time between $m$’s initial transmission until the last node that is supposed to rebroadcast $m$ has done so. Hence, every node must remember the message $id$ of each message it has received as long as this message might still be alive. The exact memory overhead implied by this depends on various factors. For example, if the frequency of message transmissions by each node is lower than the inverse of the maximum duration a message can be alive, then each node only needs to remember the last message it has received from each other node. This translates into $O(n)$ memory. As the frequency becomes even lower, the memory consumption is further reduced. Moreover, if the content of each message that needs to be stored is much larger than the field length of the message $id$, then the space required for storing message $ids$ becomes negligible compared to the view size. On the other hand, if the content of messages is also short, then the alive message ids storage space may dominate the overall space requirements of the nodes, rendering the benefits of the probabilistic storage policy useless. Consequently, DEEP makes sense mainly when the frequency of message transmissions is low, and the size of each message is considerable.

![Fig. 2. Operation of the DEEP dissemination strategy at each node.](image-url)
5. Performance analysis

This section describes the experiments we have conducted to assess both the performance and the accuracy of the investigated approaches. The results presented here shed light on the studied dissemination strategies, their parameters and cost, and how they are influenced by different network densities.

**Network topologies:** Our simulations involve scenarios with 200 nodes placed at uniformly random locations in a square area. The average number of nodes in the communication range of any node was set to a target average density \( d_{avg} \). By varying \( d_{avg} \), we built two classes of connected density-based topologies, referred here as: Sparse \((d_{avg} = 6)\) and Dense \((d_{avg} = 24)\), each one composed of 30 different topologies.

**Experimental setup:** The experiments have been done using a discrete event simulator implemented in Matlab, with a simplified MAC layer, which includes message retransmission (up to 6 retransmissions) and timeouts that are triggered when unicast messages are lost.

Depending on the link quality, interference, and other environmental conditions, a number of packets will be dropped. In order to take into account this channel behavior, usually two models are used: a model in which packet losses are independent and identically distributed, and a first-order Markov model for the success/failure process of packet transmissions. The latter is known as the Gilbert–Elliott model [30,31], and as shown in [32] it is able to accurately approximate the behavior of a fading radio channel in bursty applications (multimedia, VoIP etc.). Since our application is highly delay-tolerant, and packets exchanges among nodes do not rely on bursty transmissions/receptions, we decided to adopt a memory-less channel model. Thus, the loss rate is a controllable parameter: each neighbor of a node \( i \) that sends a message \( m \) decides whether to accept \( m \) with a given probability. With such a simple model, we could emulate different loss rates and test the robustness of the dissemination algorithms under different loss rates regimes. Finally, in order to take into account that collisions are more frequent in Dense scenarios rather than in Sparse ones, in our experiments (if not differently specified) we have adopted loss rates equal to 5% and 10% for Sparse and Dense topologies, respectively.

Recall that DEEP has two probabilistic parameters: the storage probability \( s \) and the number of expected retransmissions of each messages in a given neighborhood \( b \). As indicated before, to ensure an average view size of \( \sqrt{n} \), we use \( s = \frac{1}{2} \). As for \( b \), we experiment below with various possible values for it, and chose the most appropriate ones for the Sparse and Dense topologies.

When considering RaWMS, its main parameters are the number of random walks each node initiates in order to disseminate its own data, and the length of each random walk. The latter should roughly equal the mixing time of the network to ensure uniform distribution. As reported in the Appendix A, we have fine tuned these parameters, and have chosen the number of random walks to be 14 for Sparse topologies and 15 for Dense ones. The chosen random walks lengths were set to \( n/2 = 100 \) for Sparse topologies and \( n/8 = 25 \) for Dense ones. Notice that the mixing time of dense networks is shorter than the mixing time of sparse networks, which is why the walk length in Dense topologies can (and should be) shorter than in Sparse ones.

**Simulation composition:** Each simulation comprises two tasks: the data dissemination and the data collection tasks. In the former, the proposed DEEP dissemination strategy and the RaWMS-based strategy were implemented. They are referred in the graphs as: DEEP and RaWMS. The data collection task is then used to investigate the efficiency of data gathering by the mobile sink (see Section 5.3 for more details). For this, the sink performs \( x \) visits and the amount of data collected per visit is recorded.

### 5.1. Setting \( b \)

We study how specific values of \( b \) – i.e. the desired average number of retransmissions in each neighborhood – impact the obtained view size for both Sparse and Dense network classes. For each network in the relative class of density-based topologies, we run the DEEP mechanism such that each node transmits its information once. Each node locally stores the information it receives from others with probability \( s = \frac{1}{2} \), yet in this experiment there is no bound on the buffer size of nodes. Finally, the average values are computed.

Figs. 3 and 4 depict the average percentage of nodes in the network for different resulting view sizes, for Sparse and Dense network classes, respectively. As can be seen, the partial view size seems to have a binomial distribution, as expected from such an experiment. Moreover, results presented in Figs. 3 and 4 clearly show that the \( b \) parameter can be used as a tuning knob to set the desired resulting partial view size distribution. As a consequence, given a target partial view size distribution we may choose the proper \( b \) to achieve it. Yet, in order to obtain an average partial view size of \( \sqrt{n} \) (approximately 14 when the total number of nodes is 200), \( b \) should be set to 3.8 in Sparse topologies and 5.8 in Dense topologies so that the resulting partial view size distribution in each case matches the target average.

### 5.2. View size pruning

Since we consider nodes with limited buffer constraints (in our terminology, a partial view), the presence of a buffer management policy may influence the performances of the analyzed protocols. In order to verify whether a particular buffer management policy could bias the information distribution in the network, and consequently affect the efficiency of data gathering, we have analyzed the results with three different buffer management policies applied on the stored partial view. Thus, in case the view of a node becomes full:
the size-based policy removes the least recent entry to make room for the new information in the view;
the random policy extracts a random natural number within \([1, V_s]\), where \(V_s\) is the node view size, and removes the corresponding entry of the view;
in the competition policy, each new information has to compete with different located entries in the full view. For this, a pointer at the view will indicate which entry will be in competition with the new information. The new information will be then stored with a probability \(p_c = 0.5\) at the position indicated by the pointer. After this operation and independently of the competition result, the pointer is moved to the next position in the view, in a round-robin fashion.

Nevertheless, since the results with different policies showed similar trends, for the sake of brevity, we only report here the results observed when the size-based policy is used.

5.3. Data gathering efficiency

The experiments in this section were run with both the dissemination and the gathering tasks. During the dissemination task and according to the dissemination strategy being evaluated, the nodes simply disseminate and store the information they generate/receive. Then, during the gathering task, a mobile sink visits a set of storage nodes by performing its own RW, trying to avoid revisiting an already visited node. This is known as self-avoiding RW \([33]\) and corresponds to the UNIQUE-PATH quorum access strategy detailed in \([29]\). When the sink visits a node \(i\), it gathers the data of node \(i\) as well as all the information in \(i\)'s view. The procedure is repeated until the sink has collected all the network information. Figs. 5 and 6 show the average amount of accumulated collected information the mobile sink gathers per visited node, in both Sparse and Dense classes of topologies. The results largely confirm the theoretical analysis presented in Section 4: DEEP and RaWMS have similar data gathering efficiency and are marginally affected by the network density. Notice that the theoretical curve described by the right member of Eq. (6) is also depicted as a reference.

As can be seen, after visiting any \(2.3\sqrt{n} \approx 32\) nodes, the sink is able to collect information from about 89% of the nodes when using the DEEP strategy (the same is achieved in Sparse topologies) and 91.5% of the nodes when using RaWMS strategy (88.5% in Sparse topologies), which closely follows the theoretical analysis.

The minor difference between the theoretical curve and RaWMS's actual performance can be explained as follows: as reported in Section 5.4 below, the average replication level obtained while using RaWMS was only 12 instead of the anticipated \(\sqrt{n} \approx 14\) (the reasons are explained in Section 5.4 below). The obtained results of RaWMS are slightly better than computing the theoretical curve with replication level of 12 (recall that in the case of RaWMS, this curve is a lower bound). Alternatively, it is possible to generate a few additional random walks, at the expense of additional communication, in order to obtain a replication level of 14. When doing so, the performance of RaWMS again matches the theoretical expectation. The details are omitted for brevity, and since they do not reveal any additional insight.

It can be observed that, in Dense topologies, the performance of DEEP is about 3% worse than RaWMS. Still, the results are quite satisfactory: 89% of nodes' information is achieved after visiting any \(2.3\sqrt{n} \approx 32\) nodes in Dense and Sparse networks. This slightly lower data gathering efficiency was anticipated by the theoretical analysis and discussion in Section 4.2. It stems from the slightly worse distribution uniformity inherent in DEEP, which is also echoed in the results that appear below in Section 5.4. Notice that as in RaWMS, the obtained results are marginally worse than anticipated by the theoretical analysis. Here, this is due to the fact that the theoretical analysis assumed that every message is received by every node. Yet, due to the probabilistic nature of the dissemi-
nation task of DEEP, (very) few nodes miss each message. Hence, the distribution of the data and consequently the gathering efficiency suffer a bit.

5.4. Quality of data distribution

Using two different experiments, we investigate here the quality of the data dissemination obtained by each approach. For this, the experiments were performed selecting at random one network topology from each set of Sparse and Dense network topologies. On the selected test topologies, each data dissemination mechanism was performed 30 times (simulation runs). The average of these results are plotted in the graph as histograms for the sake of readability. We verified that within each class of topologies (Sparse or Dense), the actual topology chosen has very little impact on the results compared to other topologies from the same class. Therefore, we conclude that the results are representative for the entire class of topologies.

**Replication level (first experiment):** This experiment consists of computing the average amount of distinct storage nodes that store the information of each node in the test topologies, for each dissemination mechanism. These results are summarized in Fig. 7.

As for DEEP, as expected, the curve here resembles a binomial distribution. We can see that the average is closer to 12 in Sparse topologies, and roughly 13 in Dense ones. The slightly higher replication level in denser topologies is due to the probabilistic retransmission mechanism of DEEP. As the network becomes denser, the chance that a given node will not receive a retransmission by any of its neighbors diminishes. With the RaWMS-based approach, the replication level is distributed fairly uniformly, as expected. That is, the information of almost every node is replicated the same number of times.

The main reason for the difference in the curves for RaWMS between Sparse and Dense networks is that as mentioned above, in Sparse each node starts 14 random walks while in Dense each node starts 15 random walks. Hence, the expected view size in Dense topologies is slightly higher. As reported in the Appendix A, when both are run with the same number of random walks, then both perform very similarly, yet in Dense the uniformity is marginally better. This is due to the fact that in a dense network the chances of having disconnected nodes, or nodes with only a single path to them, is extremely small.

As mentioned before, the theoretically expected replication level was \( \sqrt{\pi} \approx 14 \), while the actual one is 12. We explain this by the fact that due to the limited number of network nodes (200), the number of random walks that ended in the same node was slightly higher than we expected.

**Distribution level (second experiment):** In order to perform this experiment, the test topologies were logically divided into 16 cells. The experiment consists of evaluating in how many different logical cells of the network each node’s information is contained. This experiment gives us an idea on how well the dissemination protocols distribute the nodes’ information over the entire network area. The results of this experiments are shown in Fig. 8.

We can observe that DEEP and RaWMS distribute data in the same number of cells, for sparse (i.e. 8 cells among a total of 16 cells) and dense (i.e. 9 cells among a total of 16 cells) networks, corresponding to a good data distribution. As expected, the RaWMS approach produces a more homogeneous distribution than DEEP, having a higher number of nodes distributing their information in the same number of cells.

5.5. Communication overhead

This section investigates the communication overhead incurred by each node during the data dissemination. This experiment was performed using the same test topologies used in Section 5.4. Fig. 9 depicts the average number of forwarded messages per node in the Sparse and Dense test topologies. The results represent the average number of forwarded messages by each node ID, after 30 dissemination runs over the test topologies.

In both densities, results clearly show that DEEP has a much lower communication overhead than RaWMS, as anticipated in Section 4. Moreover, the number of messages generated by DEEP in Dense topologies is lower than in Sparse topologies. Recall that the number of messages each node \( i \) retransmits depends on \( \frac{1}{\lambda n_i} \). Hence, as the network becomes denser, \( N(i) \) becomes larger, and so each node sends fewer messages.

---

\(^{1}\) Since a topology is a 17 × 17 square area, it was divided into 6 cells of 4 × 5, 1 of 5 × 5, and 9 cells of 4 × 4.
Recall that in RaWMS, each node is expected to transmit an average of $\frac{d}{n}$ messages, where $d$ is the RW length, which corresponds to the mixing time of the network. As mentioned before, following the study of tuning RaWMS in the Appendix A, we have chosen $d$ to be $\frac{n}{2} = 100$ for Sparse topologies and $\frac{n}{8} = 25$ in Dense topologies. The impact of this is clearly evident in the graph in Fig. 9.

As far as the communication overhead is concerned, the above results indicate that the both strategies perform better in Dense topologies than in Sparse ones, and that in both topologies DEEP is better than RaWMS. Yet, the benefit of DEEP over RaWMS is much more profound in Sparse topologies.

5.6. Loss resilience

As a final performance validation assessment, we show in Fig. 10 how the efficiency in data gathering is slightly affected by different loss rates. The graphs show the fraction of nodes the sink has gathered information from after having visited $2.3\sqrt{n}$ nodes, under different loss rate regimes (0%, 5%, 10%, and 15%). As expected, RaWMS efficiency is not affected by increasing loss rates regimes, since it relies on unicast packet retransmissions and on the salvation mechanism in the case retransmissions do not help. Fig. 10 shows that DEEP is also quite robust to increasing loss rates. In particular, we can observe a decrease of 3% in the data collection efficiency when passing from ideal scenarios (loss rate equals to 0%) to more severe ones (loss rate equals to 15%) for Sparse topologies. On the other hand, DEEP is unaffected by losses in Dense topologies. We can observe that the gap between DEEP and RaWMS, (with a loss rate equal to 15%) is around 3% and 1.5% for Sparse and Dense topologies, respectively. Finally, considering the low communication overhead and data gathering efficiency attested in previous results and the robustness to losses of the DEEP strategy, we confirm the validity of this strategy for proactive data dissemination in WSNs with uncontrolled mobile sinks.

6. Discussion and conclusions

In this paper, we have investigated how to proactively disseminate sensed data in a WSN so that a mobile sink would be able to collect a representative view of the sensed information while visiting a small number of nodes. In particular, we focused on solutions that do not impose any restrictions on the sink’s mobility. We have introduced the DEEP dissemination protocol, which is based on probabilistic flooding of the data to the entire network and storing of each data item at each node with a given probability. We have used the previously proposed RaWMS strategy, which is known to ensure uniform distribution of the data, as a reference point for comparison.

We have explored DEEP and RaWMS, both analytically and by simulations, in terms of their data gathering efficiency, communication message overhead, and data distribution quality. We have shown that while RaWMS is somewhat better than DEEP in terms of data distribution, the actual data gathering efficiency of DEEP is very close to the one provided by RaWMS. Coupled with the fact that the probabilistic data dissemination mechanism of DEEP is much more efficient than RaWMS in terms of communication overhead, we conclude that DEEP is the more viable solution among the two. This is especially true for sparse networks, when the frequency of sending messages is low, and when the amount of sensed data reported in each message is large.

Acknowledgments

We would like to thank Gabriel Kliot for many helpful discussions. Artur Ziviani acknowledges the support he received from CNPq and FAPERJ.

Appendix A. Tuning RaWMS

In order to ensure that the comparison of DEEP with RaWMS is fair, we have performed several experiments in order to tune the latter’s parameters. Specifically, RaWMS performance mainly depends on the number of random walks each node initiates in order to publish its information, denoted hereafter $N$, and the length of each such walk, denoted $L$, which should correspond to the mixing time of the network.

Recall that the average view size is the same as the average replication level. Our original goal for the average partial view size and average replication level is $\sqrt{n}$, or 14 in the case where $n = 200$. It is known from [12] that in order to end up with a given view size (and therefore also replication level) $k$, each node must start $n \ln(\frac{n}{k}) \approx 14.5$ random walks. Hence, we have experimented with both $N = 14$ and $N = 15$.

The motivation for finding the lowest value of $L$ that still gives good data gathering efficiency is that the value of $L$ greatly impacts the message overhead of the protocol. Also, it is empirically shown
in [12] that the mixing time of sparse, yet connected, uniformly distributed ad hoc networks is $n/2$ (the theoretical bound has a somewhat larger constant). Yet, it also widely known that the mixing time of the network improves as the it becomes denser. For example, the mixing time of a clique is 1. Hence, we have experimented with $L$ equals $n/2 = 100, n/4 = 50, n/8 = 25,$ and $\lfloor n/16 \rfloor = 13$.

All experiments were conducted for both Sparse and Dense topologies. As can be seen from Figs. 11–18, when combining the

---

**Fig. 11.** Data gathering efficiency of RaWMS for Sparse topologies when $N = 14$.

**Fig. 12.** Data gathering efficiency of RaWMS for Sparse topologies when $N = 15$.

**Fig. 13.** Data gathering efficiency of RaWMS for Dense topologies when $N = 14$.

**Fig. 14.** Data gathering efficiency of RaWMS for Dense topologies when $N = 15$.

**Fig. 15.** Message overhead of RaWMS for Sparse topologies when $N = 14$.

**Fig. 16.** Message overhead of RaWMS for Sparse topologies when $N = 15$. 
message overhead with data gathering efficiency, the optimal results for \textit{Sparse} are obtained with \( N = 14 \) and \( L = n/2 = 100 \). In the case of \textit{Dense}, the best is to use \( N = 15 \) and \( L = n/8 = 25 \).


