Using Area Hierarchy for Multi-Resolution Storage and Search in Large Wireless Sensor Networks

Konrad Iwanicki[1,2] and Maarten van Steen[1]
[1]Vrije Universiteit, Amsterdam, The Netherlands
[2]Development Laboratories (DevLab), Eindhoven, The Netherlands
E-mail: {iwanicki, steen}@few.vu.nl

***Abstract*—We consider multi-resolution storage, a technique for providing scalable adaptive data fidelity, necessary for many applications of large wireless sensor networks (WSNs). Although the previously proposed design of multi-resolution storage, based on quad trees and geographic routing, is conceptually simple, it exhibits inherent problems if applied to real-world WSNs. To address these problems, we revisit some of the networking assumptions and propose an alternative design that employs an overlay combining area and landmark hierarchies. Simulations and initial experiments with a prototype embedded implementation indicate that our solution can be scalable and can work on real hardware, which motivates further research.**

I. INTRODUCTION

Composed of tiny, low-power embedded devices, wireless sensor networks (WSNs) enable continuous collection of data from the surrounding environment. In many proposed applications, like habitat monitoring, precision agriculture, structural monitoring, and asset tracking, such data collection involves large numbers of nodes continuously sampling their sensors and providing the obtained data samples for querying.

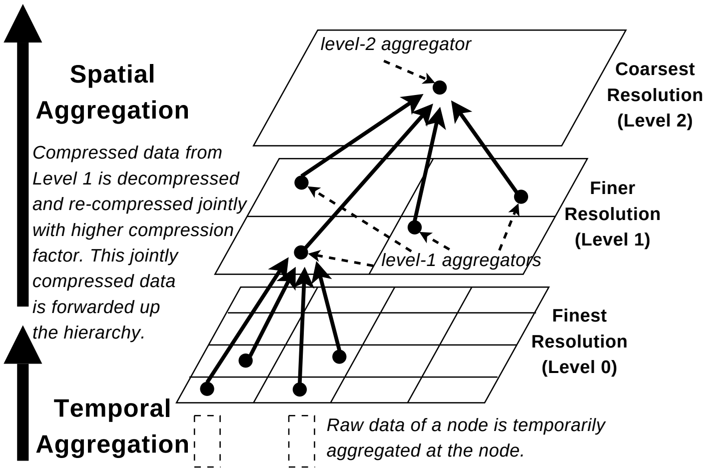
Such a large volume of data forces a trade-off between data fidelity and system scalability. High fidelity requires the ability to query the samples of every single sensor. Collecting such samples in the common, centralized data-collection model entails excessive multi-hop traffic, which essentially precludes system scalability beyond tens of nodes [1]. Therefore, to scale up, centralized systems often employ in-network tree-based aggregation [2]. However, the data compression caused by in-network aggregation to reduce the traffic volume basically precludes high fidelity: one can query the aggregate sensor reading for the whole network, but not the readings of individual sensors. Generally, the centralized data-collection model can provide either high fidelity or scalability, but not both of them simultaneously. This is inadequate for many of the aforementioned application proposals.

A. Related Work and Motivation

To cope with this limitation by exploring the fidelity-scalability trade-off, a concept of so-called *multi-resolution storage* has been introduced [1], [3], [4]. Multi-resolution storage provides scalable adaptive data fidelity by making each sensor node participate in a distributed storage system, effectively abandoning the centralized data-collection model. The principal idea is that the network itself stores a set of multi-resolution spatio-temporal aggregates of sensor data. Nodes compute and maintain these aggregates along a hierarchically decomposed, multi-level, recursive overlay, optimized for efficient querying (see Fig. 1). Queries for such data are issued in a drill-down manner (see Fig. 2). First, they are processed on coarse, highly compressed aggregates corresponding to larger spatio-temporal volumes. Then, the obtained approximate result is used to focus on those regions in the network that are most probable to contain the result data set. This process continues recursively until an accurate enough result is found at some level of the overlay. In this way, the cost of processing a query is proportional to the requested data fidelity. As a result, multi-resolution storage facilitates scalability and simultaneously enables queries for data of high fidelity, which makes it suitable for most of the aforementioned large-scale WSN application proposals.

From the networking perspective, the two key components in multi-resolution storage are: (1) the hierarchical recursive overlay, which determines how the multi-resolution aggregates are constructed, and (2) the accompanying point-to-point routing algorithm, which is responsible for delivering queries, replies, and aggregates. In the proposals for multi-resolution storage [1], [3], [4], these two components are a distributed quad tree [5] and geographic routing [6], respectively.

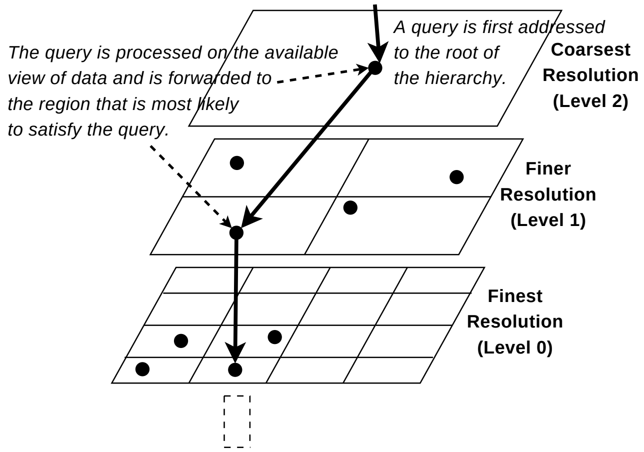
Although conceptually simple, such a design poses inherent problems when applied in real-world WSNs. The problems stem mainly from a tension between low-power hardware of sensor nodes and the consequences of using geographic coordinates as the basis for the hierarchical overlay and point-to-point routing. First, a quad tree and geographic routing both implicitly assume that physical proximity of two nodes implies connectivity between these nodes. However, due to the short range of low-power sensor node radios, multipath effects, and physical obstacles blocking radio signal, this assumption often does not hold in practice [7], [8], [9]. In the testbed used in our experiments, for instance, some nearby nodes sharing an office room are unable to communicate, but they are able to hear nodes in more distant offices [10]. As a result, the planarization of the connectivity graph, adopted by most geographic routing protocols to guarantee route existence [6], must include special mechanisms that are expensive in terms of energy and bandwidth [11], thus being of limited applicability to resource-constrained sensor devices. Second, geographic routing, either using planarization [6], [11] or other techniques [12], cannot be easily ported to three dimensions.



Compressed data from Level 1 is decompressed and re-compressed jointly with higher compression factor. This jointly compressed data is forwarded up the hierarchy.

Raw data of a node is temporarily aggregated at the node.

Fig. 1. Multi-resolution aggregation (source: Ganesan et al. [1]).



A query is first addressed to the root of the hierarchy.

The query is processed on the available view of data and is forwarded to the region that is most likely to satisfy the query.

Fig. 2. Drill-down querying (source: Ganesan et al. [1]).

Consequently, the proposed multi-resolution storage cannot be implemented for many applications, such as structural monitoring or volumetric indoor networks. Finally, obtaining geographic coordinates necessitates special hardware (e.g., a GPS receiver) or localization algorithms. Again, these solutions consume additional energy or bandwidth, introduce localization errors with disruptive effects on geographic routing [11], and often cannot be applied in many environments, like cities with skyscrapers, canyons, or underground parking lots.

B. Contributions and Roadmap

To address the above limitations, in this paper we revisit the networking aspects of large-scale multi-resolution storage. We observe that in real deployments although node proximity often does not imply connectivity, the reverse holds: *connectivity usually implies proximity*. Based on this observation, we propose an alternative design of a multi-resolution storage system. Our design abandons *artificially imposed* geographic coordinates, and instead, uses a hierarchical overlay based on *actual physical* internode connectivity, namely an overlay being a combination of landmark [13] and area hierarchy [14]. We present multi-resolution aggregation and drill-down queries on this overlay and introduce a self-organizing algorithm for bootstrapping and maintaining the overlay. Therefore, our design not only combines various known concepts, but also introduces new ideas and solutions for the occurring problems. We evaluate our design through simulations and preliminary experiments with a prototype embedded implementation.

The rest of the paper is organized as follows. We first discuss our system in Section II. Then, in Section III, we evaluate its performance. Finally, in Section IV we conclude and set directions for future work.

II. SYSTEM OVERVIEW

Our solution is based on a recursive overlay in which sensor nodes, depending on radio connectivity, are grouped into a multi-level hierarchy of nested areas. We assume that the nodes constituting the overlay are immobile. This is a valid assumption in the considered applications, such as habitat monitoring, precision agriculture, structural monitoring, and asset tracking, in which the main task of the nodes is to monitor a given physical region. However, the connectivity and the population of the nodes can change over time, for instance, if a node runs out of battery power or a communication obstacle emerges. Therefore, the protocol for maintaining the overlay must be able to account for such changes.

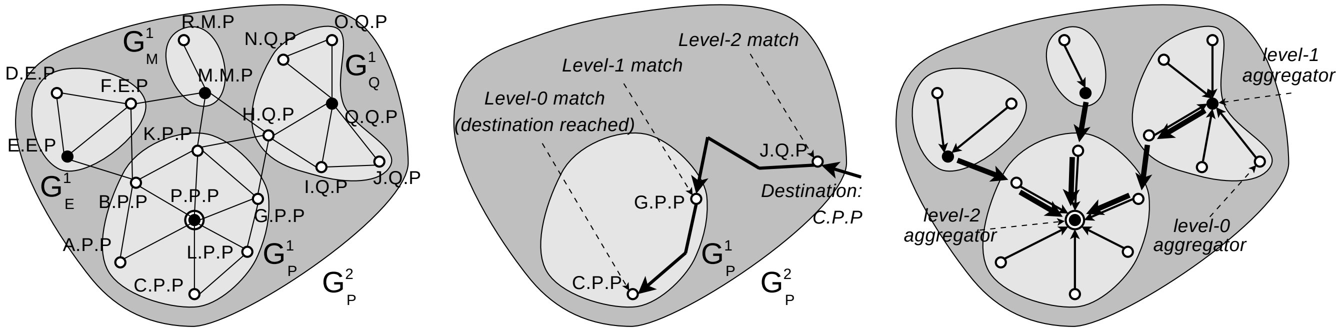
Note that the static overlay nodes do not preclude mobile clients. Handling client mobility, however, is beyond the scope of this paper, and thus, the term *node* in the remainder of the paper always refers to an immobile sensor node.

A. Hierarchical Naming

The overlay consists of multiple levels and is formed by grouping connected nodes into areas at level 0, grouping such areas into superareas at level 1, and so on at higher levels. To enable reasonable granularity of multi-resolution aggregates the number of levels should be at least logarithmic with respect to the node population size [1], [3], [4]. Therefore, the diameter of an area at level i (in terms of wireless hops) should be a multiple of the diameter of an area at level $i-1$, for instance, the diameter can double at each level.

The nodes and their areas/groups are labeled based on their membership in the hierarchy. The labels are synthesized by the hierarchy maintenance algorithm and enable scalable, efficient routing. The labeling and routing provide multi-resolution aggregation and drill-down querying.

An example of a recursive hierarchical overlay is shown in Fig. 3a. Each node has a unique identifier and each group has a dynamically appointed aggregator node. The label of a node is composed by concatenating identifiers of the aggregator nodes for the groups the node is member of at each level. For instance, the label of node J is $J.Q.P$, because node J: at level 0, is its own aggregator (of singleton group G_J^0), at level 1, belongs to a group with aggregator Q (group G_Q^1), and at level



(a) a recursive hierarchy (b) hierarchical routing (c) multi-resolution aggregation

Fig. 3. Basic system concepts and principal operation.

2, belongs to a group with aggregator P (group G_P^2). Likewise, the label of node Q is $Q.Q.P$, while the label of node P is $P.P.P$. In general, the label of a level-i aggregator node has the identifier of this node at positions from 0 to i. Importantly, to ensure that from a label of an individual node we can identify the multi-resolution aggregates the node contributes to, the group hierarchy is recursive, that is, each non-top-level group is completely nested in precisely one higher-level group. This implies that if the labels of two nodes are equal at level i, they must also be equal at all levels above i.

B. Point-to-Point Routing

Based on its label, each node maintains a routing table that enables hierarchical routing [14], [13], [15]. The structure of the table allows for maintaining $O(logN)$ rather than $O(N)$ entries, which ensures scalability and facilitates deployment on memory-constrained sensor devices. The routing table consists of rows corresponding to the hierarchy levels. The level-i row of a node's routing table contains pointers to the sibling groups of the node's level-i group. Consider the routing table of node $J.Q.P$ from Fig. 3a: row 0 contains entries for G_H^0, G_I^0, G_N^0, G_O^0, and G_Q^0; row 1 contains entries for G_E^1, G_M^1, and G_P^1; and row 2 has no entries as no other level-2 group exists.

The routing is performed hierarchically by resolving longer suffixes of the destination label [14], [13]. The organization of the node routing tables guarantees that at each routing step, every node finds the next-hop neighbor. In the example from Fig. 3b, when routing to $C.P.P$, the level-2 element of the destination label, P, already matches the level-2 element of the present node's label, $J.Q.P$. Hence, $J.Q.P$ resolves the level-1 element from Q to P, by first routing toward group G_P^1. As soon as the message reaches a node in G_P^1, node $G.P.P$ in the figure, this node resolves the level-0 element of the destination label from G to C by routing toward G_C^0. The pseudo-code of the algorithm, which we omit for brevity, can be found, for instance, in Appendix B of our technical report [16].

Note that although in our example the shortest path from $J.Q.P$ to $C.P.P$ was used, due to keeping only $O(logN)$ rather than $O(N)$ routing entries, the routes taken may not be optimal. The path overhead was extensively studied in the past [14], [13], and we also evaluate it in our experiments.

C. Multi-Resolution Aggregation and Drill-Down Queries

We use hierarchical naming and routing to provide multi-resolution aggregation and drill-down queries. Multi-resolution aggregation is performed similarly as when using geographic coordinates (see Fig. 1). However, when computing the multi-resolution aggregates, instead of a quad tree, our system exploits the group hierarchy (see Fig. 3c). Likewise, instead of geographic routing, hierarchical routing is used. For this reason, no additional mechanisms, like geographic hashing or periodic multi-hop beaconing, are necessary to allow a level-i aggregator node to obtain the routing address of the parent level-$i{+}1$ aggregator to whom the computed aggregates are forwarded. To get the label of a parent level-$i{+}1$ aggregator, a level-i aggregator node simply substitutes all elements at levels $[0 \ldots i]$ of its label with the $i{+}1$-st element of its label. For instance, the parent aggregator of node $J.Q.P$ from Fig. 3 is a level-1 aggregator node, $Q.Q.P$; the parent aggregator of node $Q.Q.P$, in turn, is a level-2 aggregator node, $P.P.P$.

Similarly, drill-down queries follow the same scheme as with quad trees and geographic routing (see Fig. 2). Again, no special mechanisms for obtaining the routing address of the top-level aggregator are needed. All elements of the top-level aggregator's label are simply equal to the last label element of any node. In Fig. 3, for instance, the top-level aggregator is $P.P.P$. Since the hierarchical routing algorithm is point-to-point, queries can be issued from any place in the network. Thus, when augmented with some hand-off functionality, the system can support mobile clients querying the static network.

D. Label and Route Maintenance

An important component in our design is the protocol for maintaining the hierarchical overlay. Such a protocol must be scalable and robust to failures and message loss, while consuming little bandwidth. Scalability of the protocol is important as it determines the scalability of the whole system. For the same reason, the protocol must gracefully handle node failures and message loss, which are inherent in WSN deployments. Finally, low bandwidth utilization implies low consumption of energy, the scarcest resource in WSNs.

The hierarchical overlay used in our system is a hybrid of area [14] and landmark [13] hierarchies. From an area hierarchy, it borrows the recursive nesting of areas. From a landmark hierarchy, it borrows appointing nodes as landmarks and aggregators. Therefore, to maintain such a hierarchy in a scalable, robust, and energy-efficient manner, we devised a custom protocol, PL-GOSSIP.

Due to space limitations, in this paper, we are unable to present the details of PL-GOSSIP, and instead we give an overview of its most important features. The detailed description of the protocol, its mathematical foundations, experimental evaluation, and comparison against existing solutions can be found in an earlier paper [17] and a technical report [16].

PL-GOSSIP is based on *asynchronous gossiping*, which is simple, robust to failures and message loss, and consumes little bandwidth and energy [16], [17]. Asynchronous gossiping makes PL-GOSSIP nodes operate in logical rounds. In every round, each node broadcasts its state (the label, the routing table, and additional consistency information) in a heartbeat message. It also receives similar heartbeats from neighboring nodes. The state of the neighbors, as received in their heartbeats, is merged with the node's own local state so that when the node broadcasts its next heartbeat, other neighbors can also update their state. In this way, information can be propagated to any node in the system.

Such gossip-based information propagation through periodic state merging is used to advertise hierarchy groups and disseminate changes to the hierarchy membership. The goal of group advertisements is to populate and maintain routing tables. An advertisement of a group is always created at the aggregator node of the group and added to the aggregator's routing table. When the aggregator broadcasts its heartbeat, the aggregator's neighbors can add the group advertisement to their routing tables. When they broadcast their heartbeats, their neighbors can update their routing tables as well, and so on. In the end, all members of a supergroup containing the group will record the advertisement for the group, which ensures that the node's routing tables are correct. Similarly, any change to the membership of the group in the hierarchy is performed by the aggregator node, which updates its label locally to reflect the change. When the aggregator broadcasts its heartbeat, its neighbors also update their labels to adopt the membership change, and so on. To enable consistent label update adoption in the presence of failures, we introduced a special consistency mechanism: *update vectors* [16], [17]. The details of update vectors, however, are beyond the scope of this paper.

Appointing nodes as aggregators is done in a *bottom-up* fashion using *probabilistic election heuristics* and exploiting the round-based pattern of gossiping. Initially, each node is a level-0 aggregator node and belongs to its own singleton level-0 group. When an aggregator node discovers that there exists another group at the same or a higher level, it has to either spawn its own higher-level group or join its group to the higher-level group of the other aggregator. To ensure that there exists only a single top-level group, the aggregators must be prevented from spawning their higher-level groups simultaneously. To this end, an aggregator probabilistically defers spawning its group. In this way, only a fraction of aggregators spawn higher-level groups and other aggregators join their groups to those groups, so that the number of groups at consecutive levels decreases exponentially. The details of this mechanism can be found in our technical report [16].

The usage of asynchronous gossiping differentiates PL-GOSSIP from other related protocols for maintaining hierarchical overlays [18], [19], [20], [21]. The round-based communication minimizes energy and bandwidth consumption: broadcasting one message per round (e.g., every 5 minutes) generates little traffic, which in addition can be efficiently scheduled by the MAC layer to save energy. Moreover, since the heartbeat-based state exchange is asynchronous and the only means of coordination is the local round-based pattern, the algorithm tolerates message loss and handles node failures without all the drawbacks of synchronous communication in the presence of faults (e.g., timeout setting, acknowledgments, retransmissions). For these reasons, PL-GOSSIP can outperform the existing state-of-the-art hierarchy maintenance protocols [17], and thus is a reasonable choice for our system.

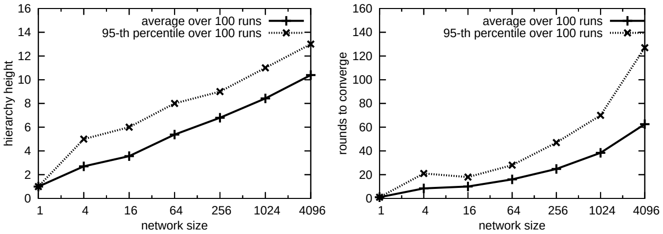
III. EXPERIMENTAL RESULTS

The previously proposed design of multi-resolution storage, based on quad trees and geographic routing, has been evaluated by emulation using data traces obtained from real deployments. The main objective of those results was showing that multi-resolution storage is well suited for many of the proposed large-scale WSN applications. To avoid repeating the experiments and because the query patterns and data-collection requirements vary between applications, in our evaluation we focus on application-independent aspects of hierarchy-based multi-resolution storage. Moreover, in contrast to past work, we present a summary of some preliminary small-scale implementation-based experiments involving solely the hierarchy maintenance algorithm. In this way, our results complement prior work.

A. Simulation Environment

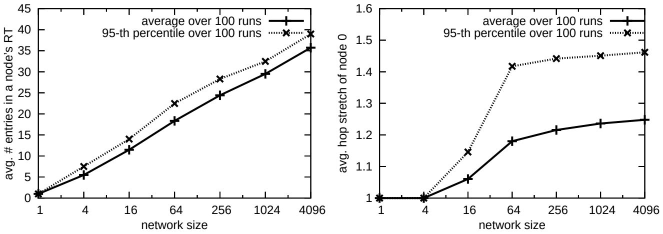
To evaluate various design decisions when prototyping our system, we used a custom high-level event-driven simulator that we had developed. The simulator allows for working with very large networks, repeating experiments multiple times, and isolating impact of various phenomena on the system behavior, thus being a perfect tool at the early development stage of our project. The simulator resembles other high-level simulators for WSNs (e.g., [12]) in that it makes the following common assumptions: a fixed circular node radio range, no network congestion, and pessimistically fixed message loss for all links.

We have two reasons to suppose that the assumptions made by the simulator will not necessarily impair real-world operation of the system. First, the hierarchy is based solely on physical links and the measured link quality, and thus, no implicit assumptions on connectivity or message loss of nearby nodes are made. Second, the bandwidth for hierarchy maintenance is small and the bandwidth for aggregation and queries can be tuned to application requirements, which facilitates preventing congestion. We acknowledge, however,



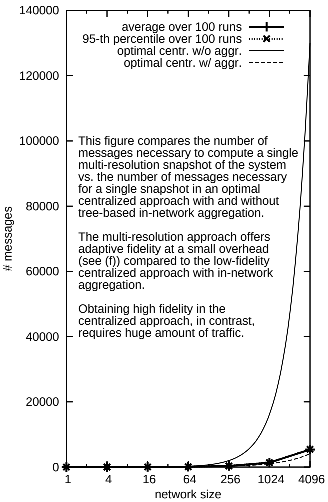
(a) hierarchy height

(c) rounds to converge



(b) average routing table size

(d) average hop stretch

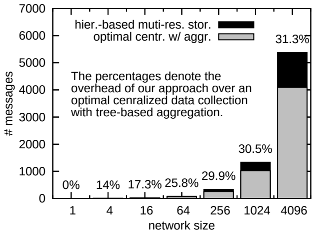


This figure compares the number of messages necessary to compute a single multi-resolution snapshot of the system vs. the number of messages necessary for a single snapshot in an optimal centralized approach with and without tree-based in-network aggregation.

The multi-resolution approach offers adaptive fidelity at a small overhead (see (f)) compared to the low-fidelity centralized approach with in-network aggregation.

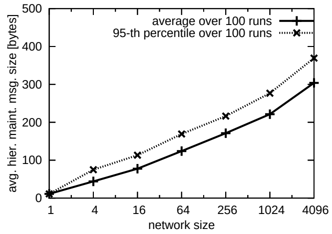
Obtaining high fidelity in the centralized approach, in contrast, requires huge amount of traffic.

(e) cost of multi-res. aggregation



The percentages denote the overhead of our approach over an optimal cenralized data collection with tree-based aggregation.

(f) overhead of multi-res. aggr.



(g) cost of hierarchy maintenance

Fig. 4. Scalability of our solution with respect to the network size. (Note the logarithmic scale of the *x*-axes.)

that only real-world evaluation of a system prototype could ultimately verify these hypotheses.

The simulations were conducted with various network configurations: message loss rates from 1% to 20%, node population increasing exponentially from 1 to 4096, node density varying from ~ 12 to ~ 80 neighbors per node, and topologies like grids, uniform deployments, and random distributions. As the results were consistent in all cases, due to lack of space, we present only a tiny albeit illustrative subset of the experiments. In these experiments, nodes formed a square grid with at least 5 (corners) and at most 12 (center) neighbors per node. All nodes were booted simultaneously in round 0 and operated long enough to enable assessment of the interesting metrics.

B. Simulation Results

Fig. 4 presents the results for different network sizes. Fig. 4a and Fig. 4b confirm that the hierarchy height and the routing table size both grow logarithmically with the number of nodes. This is crucial for the overall system scalability, as a short hierarchy minimizes costs of multi-resolution aggregation and drill-down queries while small routing tables facilitate the implementation for resource-constrained sensors.

The time to bootstrap the hierarchy (Fig. 4c) is also short, considering that it depends on the network diameter, and thus, grows exponentially with the exponentially growing node population. For instance, for a 1024-node network with diameter 32, the hierarchy is formed within 38.4 rounds on average and at most 70 rounds in 95% of the cases. With 5-minute PL-GOSSIP rounds, we need 3.2 hours on average and at most 5.8 hours in 95% of the cases. This is insignificant compared to the expected network lifetime of several weeks or even months, achievable with such extremely sparing communication. Simultaneous node boot, as in the experiments, is also a pessimistic scenario, as the group hierarchy must be constructed from scratch. Normally, deployments are incremental, in which case the hierarchy is ready almost immediately after the last node has been deployed.

The performance of drill-down queries depends mostly on the quality of point-to-point routing. This is measured using a standard metric, the hop stretch: the ratio of hop-length of the route between two nodes to the hop-length of the shortest path in the radio connectivity graph. Fig. 4d shows that the hop stretch is small. Although those results are not directly comparable, protocols for geographic routing report worse [6], [11] or similar [12] hop-stretch values, which supports the claim that our design is an attractive alternative to the multi-resolution storage based on quad trees and geographic routing.

Because the size of a raw sensor reading, the size of an aggregate, and the sensor sampling frequency vary between applications, we measure the cost of multi-resolution aggregation in the number of messages necessary to create a full multi-resolution snapshot of the system that involves readings from all sensors. This metric is further motivated by the fact that in WSNs, for the message sizes required by most of the applications, setting up a message transmission is orders of magnitude more costly (in terms of latency and energy) than the actual data transmission. Fig. 4e compares this cost for our solution with the corresponding costs of optimal centralized approaches supporting low and high data fidelity, that is, with and without in-network aggregation, respectively. To achieve high fidelity, the optimal centralized approach (without aggregation) requires a prohibitive amount of messages, which indeed precludes scalability beyond tens of nodes. In contrast, our approach provides scalable adaptive data fidelity with only small overhead as compared to the optimal centralized approach supporting solely low data fidelity (see Fig. 4f).

Finally, the bandwidth used for hierarchy maintenance is also low and scales logarithmically with the node population (see Fig. 4g). With the aforementioned 5-minute PL-GOSSIP rounds, for instance, every node in a 1024-node network generates less than 6 bits per second of outgoing traffic, which is almost nothing for 250-kbps radios of sensor nodes.

While the above results confirm scalability of our solution in a static network, long-term operation of a system entails some changes in the topology and node population. To test if our design can handle such changes, we conducted some experiments in which the network was constantly changing, for instance, due to node churn. Because of space constraints, the details and the results of these experiments can be found in our technical report [16].

C. Summary of Implementation-Based Experiments

We have also conducted preliminary implementation-based experiments, but solely with the prototype of the hierarchy maintenance protocol, PL-GOSSIP. The experiments were run in TOSSIM, a low-level simulator for the TinyOS sensor node operating system, and on our indoor testbed [10], consisting of 50^+ TelosB wireless sensor nodes spread across six office rooms. The objective of these experiments was demonstrating that our hierarchy maintenance protocol, one of the most crucial and complex elements in our system, can operate in realistic settings on real, resource-constrained hardware. In addition, we wanted to compare it against alternative solutions.

The results of these experiments are a subject of a separate paper [17]. In short, due to a lack of hidden assumptions, real-world operation of PL-GOSSIP does not deviate significantly from the simulations discussed above. The protocol quickly bootstraps the hierarchy and recovers it after failures. Moreover, to achieve this, it uses less energy than the existing alternative state-of-the-art protocols. These results give good prospects for successful real-world deployment of a complete multi-resolution storage system that follows the design presented in this paper.

IV. DISCUSSION AND FUTURE WORK

We proposed an alternative design of multi-resolution storage for wireless sensor networks, which employs an overlay that is a combination of area and landmark hierarchies. We explained how to perform multi-resolution aggregation and drill-down queries on this overlay and introduced a protocol for efficient maintenance of the overlay. Through simulations, we showed that our solution can be scalable and robust. In addition, through preliminary experiments with a prototype implementation, we demonstrated that our hierarchy can be efficiently maintained on real WSN hardware.

Although these preliminary results are encouraging, much more experimentation is necessary to verify if the proposed design is truly practical. While we showed that we can efficiently maintain the area hierarchy in the real world, it is not yet clear if the hierarchy-based multi-resolution aggregation and drill-down queries can also be seamlessly ported from the controlled simulation environment to the unpredictable, hard reality. To this end, a complete system prototype must be implemented. Moreover, the experiments with the prototype must be conducted with networks much larger than 50 nodes. Otherwise, there are no benefits of multi-resolution storage, as a simple centralized solution can probably perform better.

Finally, since multi-resolution storage displays its advantages only in large networks, it is necessary to closer examine the requirements of the potential applications of such networks. This would verify if, from the economic perspective, multi-resolution storage is the best available solution. Nevertheless, irrespective of whether our design finds its way to a production stage or remains an academic exercise, the problems it introduces constitute an exciting research agenda.

REFERENCES

[1] D. Ganesan, B. Greenstein, D. Estrin, J. Heidemann, and R. Govindan, “Multiresolution storage and search in sensor networks,” *ACM Trans. on Storage*, vol. 1, no. 3, pp. 277–315, August 2005.

[2] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, “TAG: A Tiny AGgregation service for ad-hoc sensor networks,” in *Proc. OSDI 2002*, Boston, MA, USA, December 2002, pp. 131–146.

[3] D. Ganesan, D. Estrin, and J. Heidemann, “Dimensions: Why do we need a new data handling architecture for sensor networks?” *ACM Comp. Comm. Review*, vol. 33, no. 1, pp. 143–148, January 2003.

[4] D. Ganesan, B. Greenstein, D. Perelyubskiy, D. Estrin, and J. Heidemann, “An evaluation of multi-resolution storage for sensor networks,” in *Proc. SenSys 2003*, Los Angeles, CA, USA, November 2003.

[5] R. Finkel and J. L. Bentley, “Quad trees: A data structure for retrieval on composite keys,” *Acta Informatica*, vol. 4, pp. 1–9, March 1974.

[6] B. Karp and H. T. Kung, “GPSR: Greedy perimeter stateless routing for wireless networks,” in *Proc. MobiCom 2000*, Boston, MA, USA, August 2000, pp. 243–254.

[7] A. Woo, T. Tong, and D. Culler, “Taming the underlying challenges of reliable multihop routing in sensor networks,” in *Proc. SenSys 2003*, Los Angeles, CA, USA, November 2003, pp. 14–27.

[8] J. Zhao and R. Govindan, “Understanding packet delivery performance in dense wireless sensor networks,” in *Proc. SenSys 2003*, Los Angeles, CA, USA, 2003, pp. 1–13.

[9] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, “Some implications of low-power wireless to IP routing,” in *Proc. HotNets-V*, Irvine, CA, USA, November 2006, pp. 31–36.

[10] K. Iwanicki, A. Gaba, and M. van Steen, “KonTest: A wireless sensor network testbed at Vrije Universiteit Amsterdam,” Vrije Universiteit, Amsterdam, the Netherlands, Tech. Rep. IR-CS-045, August 2008, available at: http://www.few.vu.nl/~iwanicki/.

[11] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, “Geographic routing made practical,” in *Proc. NSDI 2005*, Boston, MA, USA, May 2005.

[12] B. Leong, B. Liskov, and R. Morris, “Geographic routing without planarization,” in *Proc. NSDI 2006*, San Jose, CA, USA, May 2006.

[13] P. F. Tsuchiya, “The landmark hierarchy: A new hierarchy for routing in very large networks,” *ACM Comp. Comm. Review*, vol. 18, no. 4, pp. 35–42, August 1988.

[14] J. Hagouel, “Issues in routing for large and dynamic networks,” Ph.D. dissertation, Columbia University, 1983.

[15] K. Iwanicki and M. van Steen, “On hierarchical routing in wireless sensor networks,” in *Proc. IPSN 2009*, San Francisco, CA, USA, April 2009.

[16] K. Iwanicki and M. van Steen, “The PL-Gossip algorithm,” Vrije Universiteit, Amsterdam, the Netherlands, Tech. Rep. IR-CS-034, March 2007, available at: http://www.few.vu.nl/~iwanicki/.

[17] K. Iwanicki and M. van Steen, “Multi-hop cluster hierarchy maintenance in wireless sensor networks: A case for gossip-based protocols,” in *Proc. EWSN 2009*, Cork, Ireland, February 2009, pp. 102–117.

[18] S. Kumar, C. Alaettinoglu, and D. Estrin, “SCalable Object-tracking through Unattended Techniques (SCOUT),” in *Proc. ICNP 2000*, Osaka, Japan, November 2000, pp. 253–262.

[19] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocols for wireless microsensor networks,” in *Proc. HICSS 2000*, Maui, HI, USA, August 2000.

[20] A. Manjeshwar and D. P. Agrawal, “TEEN: A routing protocol for enhanced efficiency in wireless sensor networks,” in *Proc. IPDPS 2001 Workshops*, San Francisco, CA, 2001.

[21] S. Bandyopadhyay and E. J. Coyle, “An energy efficient hierarchical clustering algorithm for wireless sensor networks,” in *Proc. INFOCOM 2003*, San Francisco, CA, USA, March-April 2003, pp. 1713–1723.