W^R-Grid: A Scalable Cross-Layer Infrastructure for Routing, Multi-Dimensional Data Management and Replication in Wireless Sensor Networks

Gabriele Monti[1], Gianluca Moro[1], and Claudio Sartori[2]

Dept. of Electronics, Computer Science and Systems
University of Bologna
[1]Via Venezia, 52
Cesena (FC), I-47023, Italy
gmonti,gmoro@deis.unibo.it
[2]Viale Risorgimento, 2
Bologna (BO), I-40126, Italy
csartori@deis.unibo.it

Abstract. In this paper we propose a fully decentralized cross-layer infrastructure that creates self-organizing networks where wireless communications occur through multi-hop routing among sensors. The resulting sensor network is able to efficiently index and query multi-dimensional data without reliance either on Global Positioning System (GPS) or flooding/broadcasting operations. It does not suffer from the dead-end problem, which occurs in geographic-based approaches, and both routing and querying operations can scale to networks of millions of sensors. The efficiency and robustness of resulting sensor networks is controlled through a data replication strategy, which helps also in balancing the energy consumption among devices by regulating the workload among them.

1 Introduction

Wireless sensor networks are revolutionizing remote monitoring applications because of their ease of deployment and ad-hoc connectivity. They are generally formed by a large set of low cost miniaturized radio devices with a limited battery resource. Most of the energy consumption is due to radio transmissions and hence protocol design for sensor networks is directed towards reducing communications in the network. Since such large-scale sensor networks would be expected to serve a substantial number of queries simultaneously for several applications (e.g. humidity, temperature, light etc. for weather monitoring application; temperature, light, presence of chemicals etc. for precision agriculture application and so on.), it has been proven that multi-dimensional data indexing structure can greatly improve query processing efficiency [1]. Data indexing can efficiently work if there is an underlying level of the network performing physical routing without propagating each message to the entire network. The routing service could exploit Global Positioning System (GPS), however, due to its high cost, huge power consumption and unavailability in some environments, GPS is not always a good solution for sensor networks. In fact, in environments where the satellite signal can be obstructed or in indoor environments, the GPS device is unable to provide localization and, consequently, the routing. In this work we introduce the W^R-Grid that extends the infrastructure developed in [2] with data replication. The infrastructure allows multi-dimensional data management and routing, and it is based on the generation and indexing of *virtual coordinates*. The indexing technique supports both routing and query processing in a cross-layer manner. This infrastructure does not suffer from dead-ends problem caused by the inherent greedy nature of algorithms based on geographic routing (such as [3]), besides, it does not require GPS for performing the routing. Moreover it ensures the storage load balancing among nodes. The replication strategy offers improvements and new features with respect to the preceding solution. As will be illustrated in the experimental results the replication reduces the average number of hops in the network up to 50%, improving significantly both the energy consumption and the workload balancing among sensors. Finally, thanks to the replications, whose number can be arbitrarily chosen, the resulting sensor network tolerates node disconnections/connections due for instance to failures or switching off/on of sensors. The paper is organized as follows. Section 2 discusses related works. In Section 3 we briefly describe the original infrastructure while in Section 4 we address data management issues and we describe the replication extension. Section 5 illustrates some application scenario and experimental results compared with GPSR, one of the most efficient solution for message routing without GPS. Section 6 concludes the paper with open issues and perspective works.

2 Related Works

There have been different approaches for storing data in sensor networks. Earlier sensor network systems stored sensor data externally at a remote base station (External Storage) or locally at the nodes which generated them (Local Storage). In wireless sensor networks, data or events will be named by attributes or represented as virtual relations in a distributed database. Many of these attributes will have scalar values: e.g., temperature and light levels, soil moisture conditions, etc. In these systems one natural way to query for events of interest will be to use multi-dimensional range queries on these attributes. A different approach has been taken by works on data-centric routing in sensor networks to cope with such requirements [1, 4–6], in particular where data generated at a node is assumed to be stored at the same node, and queries are either flooded throughout the network [4]. In a GHT [7], data is hashed by name to a location within the network, enabling highly efficient rendezvous. GHTs are built upon the GPSR [3] protocol and leverage some interesting properties of that protocol, such as the ability to route to a node nearest to a given location. In DIFS [8], Greenstein et al. have designed a spatially distributed index to facilitate range searches over attributes. Like us, Li et al. [1] have built a distributed index (DIM) for multidimensional range queries of attributes but they require nodes to be aware of their physical location and of network perimeter; moreover they exploit GPSR for routing.

3 $\mathbf{W}^R$-Grid

W^R-Gridis an extension of W-Grid, which has been presented in [9] and [2], in this paper we will briefly describe the concept of virtual coordinate and the process of virtual coordinate generation, we will give some hints on how routing works and how sensors failure are managed. Then we will study in depth data replication, which is the contribution of this paper. The reader is referred to the above cited works for technical details. We consider the case of sensors equipped with a wireless device. Each one is, at the same time, client of the network (e.g. submitting queries and generating data), and responsible for managing others sensors communications (e.g. routing queries and data). The main idea in W^R-Grid is to map sensors on a binary tree and to build a total order relationship among them. Each node of the tree is assigned a W^R-Grid virtual coordinate which is represented by a binary string. From now on we will refer to the participants of the network as nodes or sensors indistinctly.

3.1 Virtual Coordinate Selection and Generation

When a sensor, let us say s, turns on for the first time, it starts a wireless channel scan (beaconing) searching for any existing W^R-Grid network to join (namely any neighbor device that already holds W^R-Grid virtual coordinates). If none W^R-Grid network is discovered, s creates a brand new virtual space coordinate and elects itself as root by getting the virtual coordinate "$*$"[1]. On the contrary, if beaconing returns one or more devices which hold already a W^R-Grid coordinate, s will join the existing network by getting an appropriate virtual coordinate.

Coordinate Setup. Whenever a node needs a new W^R-Grid coordinate, an existing one must be split. The term "split" may seem misleading at the moment, but its meaning will become clear in Section 4. A new coordinate is given by an already participating sensor s_g, and we say that its coordinate c is split by concatenating a 0 or a 1 to it. The result of a split to c will be $c' = c+1$ and $c'' = c + 0$. Then, one of the new coordinates is assigned to the joining sensor, while the other one is kept by the giving sensor that will then hold two coordinates. No more splits can be performed on the original coordinate c since this would generate duplicates. In order to guarantee coordinates' unambiguousness even in case of simultaneous requests, each asking sensor must be acknowledged by the giving sensor s_g. Thus, if two nodes ask for the same coordinate to split, only one request will succeed, while the other one will be cancelled.

[1] It is conventional to label " $*$ " the root node

Coordinate Selection. At coordinate setup, if there are more neighbors which already participate the W^R-Grid network, the joining sensor must choose one of them from which to take a coordinate. The selection strategy we adopt is to choose the shortest coordinate [2] in terms of number of bits. If two or more strings have the same length the sensor randomly chooses one of them. Experiments have shown that this policy of coordinate selection reduces as much as possible the average coordinates length in the system. In the resulting tree structure, parent-child relationships can be set only by nodes that are capable of bi-directional direct communication. This property is called *integrity* of coordinates and it is crucial for the network efficiency:

Definition 1 *Let c be a coordinate at a sensor s that has been split into c' and c'' and let $NEIGH(s)$ be the set of its neighbors. We say that c has integrity if the child that has been given away by s is held by a sensor $s' \in NEIGH(s)$.*

If each coordinate satisfies this constraint, it will be possible to route any request or message by following the paths indicated by the tree structure and without dead-ends.

3.2 Routing Algorithm

As we stated in the previous subsection, the coordinate creation algorithm of W^R-Grid generates an order among the nodes and its structure is represented by a binary tree. The main benefit of such organization is that messages can always be delivered to any destination coordinate, in the worst case by traveling across the network by following parent-child relationship. The routing of a message is based on the concept of distance among coordinates. The distance between two coordinates c_1 and c_2 is measured in logical hops and correspond to the sum of the number of bits of c_1 and c_2 which are not part of their common prefix. For instance:

$$d(\mathbf{*0}011, \mathbf{*0}11) = 5$$

Given a message and a target binary string c_t each sensor s_i forwards it to the neighbor that present the shortest distance to c_t. It is important to notice that each sensor needs neither global nor partial knowledge about network topology to route messages, its routing table is limited to information about its direct neighbors' coordinates. This means **scalability** with respect to network size.

3.3 Sensors Failure

Sensors usually have scarce resource, they especially suffer of power constraints and this can lead to failures that could affect routing efficiency. During a routing operation it may happen that a sensor cannot find any neighbor that improves its distance from the destination coordinate (dead end). This means that a link

[2] among the ones that still can be split, see Coordinate Setup

has broken since W^R-Grid total order relation guarantees delivery in any case. A solution to sensors failures is described in [2], therefore here we just specify that every recovery operation is only triggered on routing failures detection, in order to avoid any network efficiency loss.

4 Data Management and Replication in W^R-Grid

W^R-Grid distributes data (tuples of attributes) gathered by sensors among them in a data-centric manner. Values of surveys are hashed[3] into binary strings and stored at nodes whose W^R-Grid coordinates have the longest common prefix with those strings. Thus, a W^R-Grid network acts directly as a distributed database in which data proximity is preserved, i.e. logically close sensors store similar data. Probably, the most important feature that a distributed database must satisfy is storage load balancing among participants, especially in case of not uniform distributions of data. In fact, if the managed information do not distribute uniformly in the domain space it can happen that virtual coordinates store different number of data. Nodes that manage more data will likely receive a higher number of queries than the others causing bottlenecks and loss of efficiency for the entire network. In order to improve the data distribution balance we implemented a storage load balancing algorithm (SLOB).

4.1 Storage Load Balancing in W^R-Grid

We introduced a maximum number of data that a region/coordinate can manage, defined as *bucket size* (b). The value for b can be the same for each node or, in environments where devices have different characteristics, it can be proportional for instance to the storage and/or communication bandwidth capabilities. Whenever a sensor receives a new data it checks whether the space represented by the coordinate that must store the data is full or not. In case it is full the coordinate is split, but, differently from what it happens when a new node joins the network, in this case both the resulting subspaces are stored at the sensor. The bucket size guarantees that each coordinate contains at most the same quantity of information. However, this trick does not balance the storage load on its own. In fact, nodes holding spaces with a higher number of data will split more frequently that the others. The result will be that those nodes will manage more coordinates if we do not find a way for them to give away the ones in excess, which is exactly the goal of the SLOB Algorithm. Periodically each node evaluates the average storage load and the correspondent Root Mean Square Error. The purpose of this evaluation is discovering local unbalanced situations and trying to fix them through coordinates transfers. By solving local unbalancing the algorithm is able to create a balanced network storage load, please refer to [2] for a detailed description of it.

[3] See [10] for details about hashing function

4.2 $\mathbf{W}^{R}$-Grid Replication

Our previous work [2] was intended for use into Ad-hoc networks, which have different characteristics from sensor networks. In sensor networks the most important operations are data gathering and querying, therefore is necessary to guarantee the best efficiency during these tasks. In particular, data sensed by the network should be always available for users' queries and query execution latency must be minimized. In order to achieve these results we introduced replication of data in W^{R}-Grid. Data replication is obtained by generating multiple virtual coordinate spaces (namely multiple trees T). In this way, each information is replicated on every existing space, resulting in more than one benefit for network performances:

- **higher resistance to sensors failure**. Having multiple virtual spaces implies the existence of different paths for each coordinate and the possibility of changing routing space in case of dead-end;
- **reduction of query path length and latency**. Multiple realities mean multiple order relationship and therefore a reduction of the probability that two nodes physically close have very different virtual coordinates. Which may happen whenever a multi-dimensional space is translated into a one-dimensional space.

For what concerns replication implementation in W^{R}-Grid, we must say that the changes to the algorithm (Section 3) are few. Supposing that each sensor is given an unique identifier $ID(s)$, each reality is uniquely identified by the root node ID. Each coordinate c is coupled with its reality identifier so that each couple (ID, c) will be unique. During coordinate creation, sensors take a coordinate from every reality they discover from neighbors. At periodic beaconing, if any new reality is discovered a new coordinate from that reality is taken, allowing a progressive spread of the various realities to every participant of the network. During routing toward a target coordinate, sensors will evaluate their distance with respect to each reality and will route on the reality that takes closer to the target. Nothing else changes from what described in Section 3.

It is well known, from database literature, that replication has also drawbacks. Generally it has a negative impact in case of data updates, since it needs each existing replica to be affected by changes in order to maintain consistency. However we can observe that usually sensor networks are more like a stream of information in which older surveys can be replaced by newer ones or just stored with the newer one to maintain historical information. We can say that updates represent a limited problem and we can therefore focus on new data insertion. Since it is costly (in terms of network traffic) to replicate each tuple/record in each reality, analysis will be presented in section 5.1 in order to find out the best replication configuration which guarantees query efficiency at reasonable costs.

5 Application Scenarios and Experimental Results

Sensor networks are intended for monitoring specific phenomenon or environments, they survey various kind of data such as temperature, humidity, pressure, light, etc. We can see the example of an environment monitoring application in which sensors survey temperature (T) and pressure (P), to which we refer as d_1 and d_2. Each event is inserted in the distributed database implicitly generated by W^R-Grid, reporting for instance date and time of occurrence. Without loss of generality we can define a domain for T and P let us say $Dom(d_1) = [-40, 60]$ and $Dom(d_2) = [700, 1100]$. We present an example of range query submitted to the network.

Return the times at which sensors surveyed a temperature ranging from 26 to 30 Celsius degrees and pressure ranging from 1013 to 1025mbar. We must calculate the correspondent binary string for the four corner of the range query, namely:

(26,1013) (26,1025) (30,1013) (30,1025)
$c_1 = \mathbf{*110110}00$ $c_2 = \mathbf{*110110}01$
$c_3 = \mathbf{*110110}10$ $c_4 = \mathbf{*110110}11$

Now all we have to do is querying the sensors whose coordinate have $*110110$ as prefix.

5.1 Experimental Results

In our previous works [2] we evaluated the performances of our algorithm with respect to the Average Path Length (APL, measured in hops) covered by messages and to the average storage load at each sensor. Simulation results validated the goodness of virtual coordinates idea, routing algorithm and storage load balancing algorithm. In this paper we exploited our Java simulator in order to evaluate

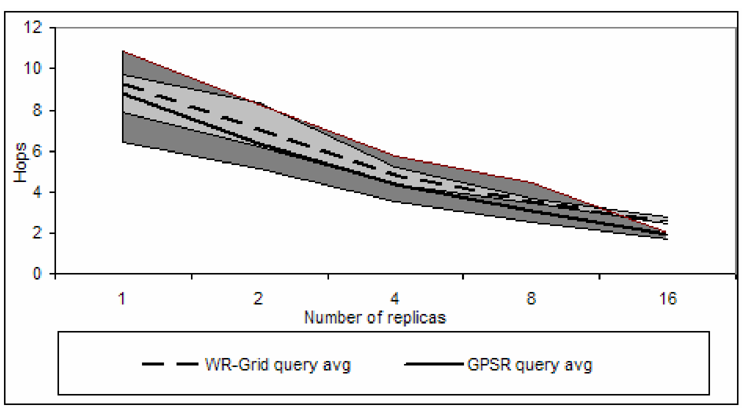

Fig. 1. Query path length for different numbers of realities in the network

the impact of multiple realities policy. We ran simulation on an area of 1500 by 1500 meters in which about 200 sensors with a supposed radio transmission of

100 meters are spread. Coordinate creation is gradual, the simulator randomly choose one or more sensor to elect as root of realities, then, as described in Section 3 we let periodic beaconing to build the W^R-Grid network. Beside coordinate creation we simulated the survey of events (3000 in each run) by sensors and their consequent insertion in the network. We also simulated the execution

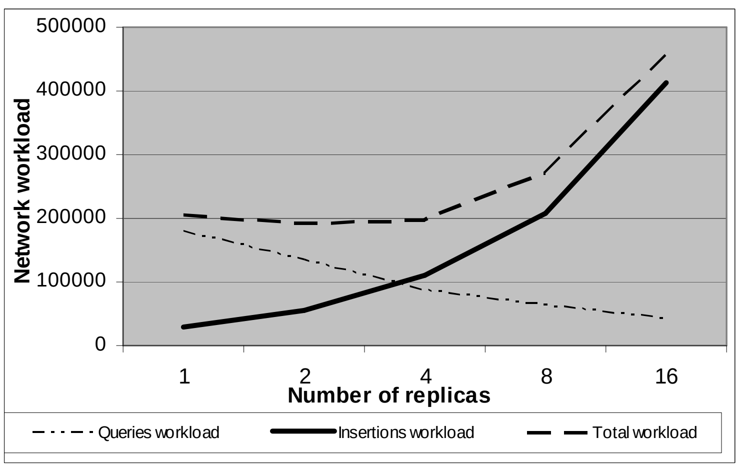

Fig. 2. Sensors workload for different numbers of replicas and Query/insertion ratio (10/1 and 5/1)

of queries of randomly chosen data from randomly chosen sensors. Simulation reported information about the number of hops covered by queries (query path length), the number of data stored per node (storage load) and the number of times each node is request to route a query (workload) during the simulation. We analyzed average and Mean Square Error of those measures with different

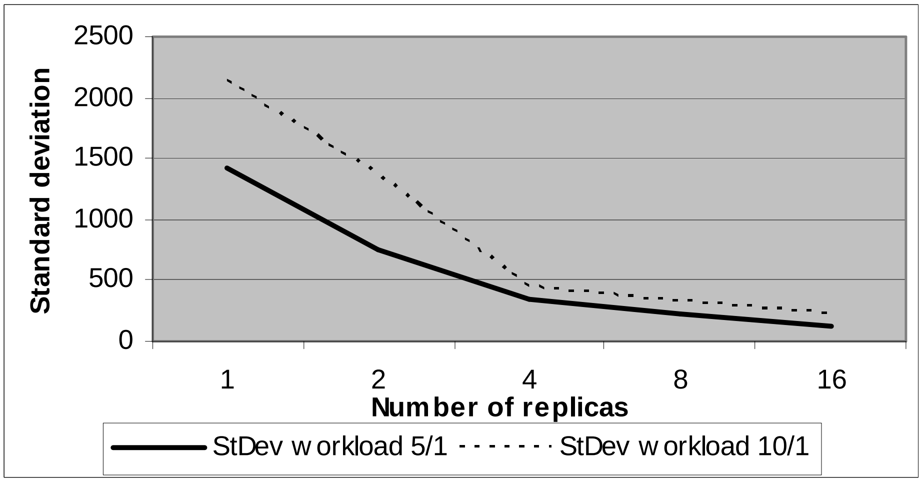

Fig. 3. MSE of sensors workload for different numbers of replicas.

numbers of replicas in the system and different query/insertion ratios (10/1, 5/1). Figure 1 shows that as the number of realities increases the routing performances of W^R-Grid improves considerably (average hops are halved compared to W-Grid). This is the demonstration that multiple realities reduce the probability that two nodes physically close are distant according to the order relationship. It is important to notice that this benefit follows a logarithmic curve, therefore, once that a certain number of coordinate (we can say around 10) is reached, it is no more convenient to increase it. In Figure 2 and 3 can be observed a consequence of the improvement in routing efficiency. Since the average hops per query is reduced also the average sensor workload is reduced. At the same time it is possible to see that the MSE of that measure decreases, meaning a better balance in the workload per sensor. By observing Figure 3 we can say that multiple realities improve storage load balancing too and surely this has a positive effect on sensors energy consumption since it implies a more balanced request load per node. On the other side replication implies higher cost at insertion time, more precisely, in case of n realities each event must be inserted in n different indexes. Therefore the number of replica should be limited to the smallest necessary in order to guarantee data availability and routing efficiency. From our simulations and showed graphs we can say that a number of 4-6 realities is the best choice. With a higher number the increase of routing efficiency and balancing cannot be justified by the increase of replication costs.

6 Conclusions and future work

In this paper we presented W^R-Grid which extends our previous work W-Grid by adopting a replication methodology. W^R-Grid acts as a distributed database without needing neither special implementation nor reorganization and any kind of data can be distributed, stored and managed. We have evaluated the benefits of replication on data management, discovering from experimental result that it can halve the average number of hops in the network. The direct consequence of these results are a significant improvement on energy consumption and a workload balancing among sensors (number of messages routed by each node). Finally, thanks to the replications, whose number can be arbitrarily chosen, the resulting sensor network tolerates sensors disconnections/connections due to failures of sensors. Next future works will concern other analysis and experiments and the introduction of path learning capability at nodes in order to improve the W^R-Grid average path length.

References

1. Li, X., Kim, Y., Govindan, R., Hong, W.: Multi-dimensional range queries in sensor networks. In: SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems, New York, NY, USA, ACM Press (2003) 63–75
2. Moro, G., Monti, G.: W-Grid: a self-organizing infrastructure for multi-dimensional querying and routing in wireless ad-hoc networks. In: To appear in P2P 2006: Sixth IEEE International Conference on Peer-To-Peer Computing, available as technical report at http://p2p.ingce.unibo.it/~gmoro/papers/DEIS-LIA-06-001.pdf. (2006)
3. Karp, B., Kung, H.: GPRS: greedy perimeter stateless routing for wireless networks. In: MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking, ACM Press (2000) 243–254
4. Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F.: Directed diffusion for wireless sensor networking. IEEE/ACM Trans. Netw. **11**(1) (2003) 2–16
5. Ye, F., Luo, H., Cheng, J., Lu, S., Zhang, L.: A two-tier data dissemination model for large-scale wireless sensor networks. In: MobiCom '02: Proceedings of the 8th annual international conference on Mobile computing and networking, New York, NY, USA, ACM Press (2002) 148–159
6. Xiao, L., Ouksel, A.: Tolerance of localization imprecision in efficiently managing mobile sensor databases. In: MobiDE '05: Proceedings of the 4th ACM international workshop on Data engineering for wireless and mobile access, New York, NY, USA, ACM Press (2005) 25–32
7. Ratnasamy, S., Karp, B., Shenker, S., Estrin, D., Govindan, R., Yin, L., Yu, F.: Data-centric storage in sensornets with ght, a geographic hash table. Mob. Netw. Appl. **8**(4) (2003) 427–442
8. Greenstein, B., Estrin, D., Govindan, R., Ratnasamy, S., Shenker, S.: Difs: A distributed index for features in sensor networks. In: Proceedings of first IEEE WSNA, IEEE Computer Society (2003) 163–173
9. Moro, G., Monti, G., Ouksel, A.: Routing and localization services in self-organizing wireless ad-hoc and sensor networks using virtual coordinates. In: ICPS'06: IEEE International Conference on Pervasive Services 2006. (2006)
10. Ouksel, A., Moro, G.: G-Grid: A class of scalable and self-organizing data structures for multi-dimensional querying and content routing in P2P networks. In: Agents and Peer-to-Peer Computing, Springer-Verlag (2004)