REDUNDANT DISTRIBUTED DATA STORAGE
Experimentation with the SensLab Testbed

Pietro Gonizzi[1], Gianluigi Ferrari[1], Vincent Gay[2] and Jérémie Leguay[2]
[1]*Department of Information Engineering, University of Parma, Viale G.P.Usberti 181/A, Parma, Italy*
[2]*Thales Communications and Security, 160 Bd de Valmy, Colombes Cedex, France*

Keywords: Distributed Data Storage, Wireless Sensor Networks, Data Replication, SensLab.

Abstract: Wireless sensor network (WSN)-based applications typically require to store data in the network. For instance, in the surveillance of isolated areas, if no sink nodes are present, WSNs may archive observation data that are periodically retrieved by an external agent. In contrast to conventional network data storage, storing data in WSNs is challenging because of the limited power, memory, and communication bandwidth of WSNs. In our study, we review the state-of-art techniques for data replication and storage in WSNs, and we propose a low-complexity distributed data replication mechanism to increase the resilience of WSN storage capacity against node failure and local memory shortage. We evaluate our approach through experimental results collected on the SensLab large-scale real testbed. In particular, we show how the performance is affected by changing the configuration of several key system parameters, such as (*i*) the transmission power of the nodes; (*ii*) the control message overhead; (*iii*) the number of deployed nodes; and (*iv*) the redundancy. To the best of our knowledge, this is one of the first works presenting experimental results at a really large scale on SensLab.

1 INTRODUCTION

In contrast to conventional network data storage, storing data in wireless sensor networks (WSNs) represents a challenge because of the limited power, memory, and communication bandwidth of WSNs. Nowadays, sensors have reached higher capabilities, in terms of speed processing and local storage than in the past years (Mathur et al., 2006). This makes them more attractive for in-network storage deployments.

WSNs usually consist of: on one hand, unattended nodes that sense the surrounding environment; on the other hand, a sink node which is in charge of collecting data measurements and relaying them to a management entity. There are many reasons for which a sensor node may not be able to transmit data to the sink right after acquisition. First, the sink node may not be always reachable from sensor nodes. For instance, a mobile node can be used to periodically pull out all the collected data. Second, when applications do not require real-time collection, storing data units and sending aggregate data bursts can contribute to reduce the amount of radio transmissions, thereby increasing the lifetime operation of the WSN. Illustrative applications include habitat monitoring, such as tracking animal migrations in remote-areas (Juang et al., 2002), studying weather conditions in national parks (Beaufour et al., 2002), etc. Such scenarios require to collect and store as much data as possible between two consecutive data retrievals performed by an external agent. However, storing data on the sensor node leads to local memory overflow if data retrieval is not timely performed by the sink. To avoid data dropping or overwriting, sensor nodes can cooperate with each other by sharing acquired data.

Node failure is also a critical issue in WSNs. Periodic inactivity (e.g., for energy saving purposes), physical destruction, and (software) bugs are likely to appear in WSNs, leading to data loss. Thus, redundancy by means of data replication (i.e., by storing copies of the same data onto various nodes) contributes to increasing the resilience of the WSN. However, distributed data storage across nodes, with or without redundancy, is a challenge as it requires to properly select "donor nodes" (i.e., nodes available to store data of other nodes) and entails communication overhead to transmit data to the selected nodes. Therefore, network storage capacity and resilience have an energy cost and this limits the network lifetime.

Although distributed data storage and replication have been studied in the literature (Neumann et al.,

Gonizzi P., Ferrari G., Gay V. and Leguay J..
REDUNDANT DISTRIBUTED DATA STORAGE - Experimentation with the SensLab Testbed.
DOI: 10.5220/0003803900150023
In *Proceedings of the 1st International Conference on Sensor Networks* (SENSORNETS-2012), pages 15-23
ISBN: 978-989-8565-01-3

2009; Piotrowski et al., 2009), there is still a number of challenges to tackle. First, to the best of our knowledge, most of previous studies do not encompass both the distribution and the replication aspects. Second, the proposed solutions are usually not tested experimentally at large-scale, while we stress how real deployment is of primary importance when developing WSN applications.

In this paper, we propose a low complexity distributed data replication mechanism to increase the resilience and storage capacity of a WSN-based observation system against node failure and local memory shortage. We evaluate our approach through experimental tests conducted on the SensLab real platform (SensLAB, 2008). We show how careful configuration of key system parameters has a positive effect on the performance. The portfolio of parameters includes: *(i)* the node transmit power, which should be reduced to save energy; *(ii)* the communication overhead, i.e., the amount of control messages exchanged by the nodes; *(iii)* the number of deployed nodes; and *(iv)* the data redundancy, i.e., the number of copies to be stored per sensed data unit.

This paper is organized as follows. Section 2 is dedicated to related works. In Section 3, we motivate and detail the design of our greedy mechanism for distributed data replication. Section 4 is devoted to the large scale experiments performed on the SensLab testbed, by considering the aforementioned scenarios. At last, Section 5 concludes the paper.

2 RELATED WORK

Various schemes to efficiently store and process sensed data in WSNs have been proposed in the past years (Hongping and Kangling, 2010).

In distributed WSN storage schemes, nodes cooperate to efficiently distribute data across them. Previous studies focused on *data centric storage* (Ratnasamy et al., 2003; Madden et al., 2005; Awad et al., 2009). According to a centric storage approach, data collected in some WSN regions are stored at super nodes that are responsible for these regions and/or according to the type of data. Hash values are used to distinguish data types and the corresponding storage locations. Even if this approach is based on node cooperation, it is not fully distributed, since super nodes store all the contents generated by the others.

In a *fully distributed data storage* approach, all nodes participate in sensing and storing in the same way. All nodes, first, store their sensor readings locally and, once their local memories have filled up, delegate other nodes to store them. A first significant contribution in this direction is given by Data Farms (Omotayo et al., 2007). The authors propose a fully data distributed storage mechanism with periodical data retrieval. They derive a cost model to measure energy consumption and show how a careful selection of nodes offering storage, called “donor nodes”, optimizes the system capacity at the price of slightly higher transmission costs. They assume the network is built in a tree topology and each sensor node knows the return path to a sink node, who periodically retrieves data. Another study is proposed in EnviroStore (Luo et al., 2007). The authors focus on data redistribution when the remaining storage of a sensor node exceeds a given threshold, by means of load balancing. They use a proactive mechanism, where each node maintains locally a memory table containing the status of the memory of its neighbors. Furthermore, mobile nodes (called *mules*) are used to carry data from an overloaded area to an offloaded one and to take data to a sink node. The major lack of the above studies pertains to the absence of large scale experiments to evaluate the system performance, which is a key contribution of our work.

Data replication consists in adding redundancy to the system by copying data at several donor nodes (within the WSN) to mitigate the risk of node failure. It has been widely studied for WSNs and several works are available in the literature (Neumann et al., 2009; Hamed Azimi et al., 2010). A scoring function for suitably choosing a replicator node is proposed in (Neumann et al., 2009). The function is influenced by critical parameters such as the number of desired replicas, the remaining energy of a replicator node and the energy of the neighbors of the replicator node. In TinyDSM (Piotrowski et al., 2009), a reactive replication approach is discussed. Data burst are broadcasted by a source node and then handled independently by each neighbor, which decides to store a copy of the burst or not. Unlike the above approaches, we propose a replication-based distributed data storage mechanism with lower complexity, since the responsibility of finding donor nodes is not centralized at the source node but handed over to consecutive donor nodes in a recursive manner.

The main contributions of this paper are the following. First, we encompass, with a fully distributed mechanism, both data replication and distributed storage. Second, we conduct large scale experiments on the Senslab real testbed in order to evaluate our solution. Even if the Senslab platform is suitable for testing WSN-based applications (Ducrocq et al., 2010), to the best of our knowledge, no previous studies with results collected in Senslab have been published.

3 GREEDY REPLICATION-BASED DATA STORAGE

3.1 Design Principles

We assume that a WSN is deployed to measure environmental data and store it until a sink node performs a periodic retrieval of the whole data stored in the WSN. Between two consecutive data retrievals, the objective is to distribute multiple copies of every data unit amongst nodes in order to avoid data losses caused by local memory shortages or by node failures. Data distribution relies on the proactive announcement, by every node, of its memory status. Each node periodically broadcasts a memory advertisement message containing its current available memory space. It also maintains an updated memory table containing the memory statuses of all its detected 1-hop neighbors. Upon reception of a memory advertisement, a node updates its local memory table with the new information received. The memory table contains an entry for each neighbor. Each entry contains the address of the neighbor, the last received value of its available memory space, and the time at which this value has been received. When sending a replica of acquired data, a node looks up in its table the neighbor node with the largest available memory and most recent information. Such neighbor is called *donor node*. If no donor node can be found and there is no available space locally, then the acquired data is dropped. At the end of the sensing period, a sink node is meant to gather all data present in the WSN by sending requests to all nodes.[1] Upon sending the data to the sink, a node clears its local buffer and resumes sensing the environment.

3.2 The Greedy Algorithm

The main parameters are listed in Table 1. Without loss of generality, we consider a WSN with N fixed nodes randomly deployed over an area whose surface is A (dimension: [m^2]). Nodes only interact with 1-hop neighbours, i.e., with nodes within radio transmission range, denoted as d (dimension: [m]). The number of 1-hop neighbors of node i ($i \in \{1,\ldots,N\}$) is denoted as $V_1^{(i)}$. The i-th node has a finite local buffer of size B_i (dimension: [data units]) and sensing rate r_i (dimension: [data units/s]). Each node is scheduled to broadcast, without acknowledgement and every T_{adv} (dimension: [s]), its memory status to all nodes within the transmission range (i.e., 1-hop neighbours). Each memory advertisement consists of 4 fields relative to the sending node: node ID; up-to-date available memory space; value of sensing rate; and a sequence number identifying the memory advertisement. Each node maintains a table which records the latest memory status received from neighbor nodes. Upon reception of a memory advertisement from a neighbor, a node updates its memory table, using the sequence number field to discard multiple receptions or out-of-date advertisements. An illustrative scenario, with advertisement of the memory status by node 1, is shown in Figure 1. The remaining parameters (R, T and P_t) will be described later.

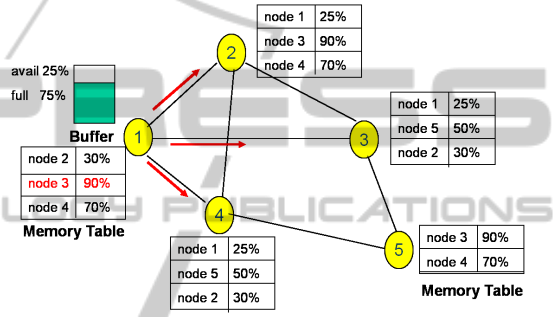


Figure 1: Memory tables at a group of neighbors and advertisement from node 1.

The proposed replication-based data storage mechanism is fully decentralized, in the sense that all nodes play the same role. It consists in creating at most R copies of each data unit generated by a node and distribute them across the network, storing at most one copy per node. Each copy is referred to as *replica*. Let us focus on node $i \in \{1,\ldots,N\}$. At time t, the node generates (upon sensing) a data unit. The memory table of node i contains one entry per direct neighbor. Node i selects from its memory table the neighbor node, called *donor*, with the largest available memory space and the most recent information. Denoting the neighbors of node i as $\{1,\ldots,V_1^{(i)}\}$, the donor $D^{(i)}(t)$ is selected according to the following heuristic rule:

$$D^{(i)}(t) = \arg\max_{j \in \{1,\ldots,V_1^{(i)}\}} \left(\frac{B_j(t_j)}{t - t_j} \right) \qquad (1)$$

where t_j denotes the time at which the available memory space $B_j(t_j)$ of node j was received by node i, with $B_j(t_j) \leq B_j$. If there is no suitable neighbor in the memory table (i.e., $B_j(t_j) = 0, \forall j \in \{1,\ldots,V_1^{(i)}\}$), there is no possibility to distribute replicas of the data unit across the network. In this case, only one copy can be stored in the local memory of node i, if i has

[1] We do not detail the data retrieval phase, as this is not the focus of our work.

Table 1: Main system parameters.

Symbol	Description	Unit
N	Number of nodes	scalar
A	Surface of deployment area	m^2
d	Node transmission range	m
$V_1^{(i)}$	Number of 1-hop neighbours of node i	scalar
B_i	Node i's buffer size, $i \in \{1,\ldots,N\}$	scalar
r_i	Node i's sensing rate, $i \in \{1,\ldots,N\}$	s^{-1}
P_t	Common node transmit power	W
T_{adv}	Period of memory advertisement (from each node)	s
R	Maximum number of replicas per sensing data unit	scalar
T	Period of data retrieval (from the sink)	s

some space locally. If a donor node can be selected, node i sends to it a copy of the data unit, specifying how many other copies are still to be distributed in the WSN. In particular, the number of required copies is set to either $R-1$ (if node i can store the original data locally) or R (if node i's local memory is full). Upon reception of the copy, the donor node D stores the copy in its memory and selects the next donor node among its neighbors, according to (1), and discarding the sending node and the source node from the candidate nodes. Then, it sends the copy to the chosen donor node, decrementing the number of required copies by 1. The replication process continues recursively until either the last (R-th) copy is stored or stops when one donor node cannot find any suitable next donor node. In the latter case, the final number of copies actually stored in the WSN is smaller than R.

4 EXPERIMENTAL VALIDATION

We present a large-scale validation of our distributed storage mechanism on the SensLab testbed (SensLAB, 2008). The SensLab platform offers more than 1000 sensors at four sites in France (Grenoble, Strasbourg, Lille, Rennes) where researchers can deploy their codes and run experiments. At each site, nodes are installed on an almost regular grid, as depicted in Figure 2. Each node's platform embeds a TI MSP430 micro-controller and operates in various frequency bands depending on the radio chip (either CC1100 or CC2420).

All the experiments have been run in the Grenoble site of SensLab. The deployed WSN platform is the *wsn430v13*, which adopts the CC1100 radio chip. The duration of each experiment is set to 15 min. The sensing period of the nodes is chosen randomly and independently in the interval [0.1,5.1] s. All nodes have the same buffer size, which equals $B = 250$ data

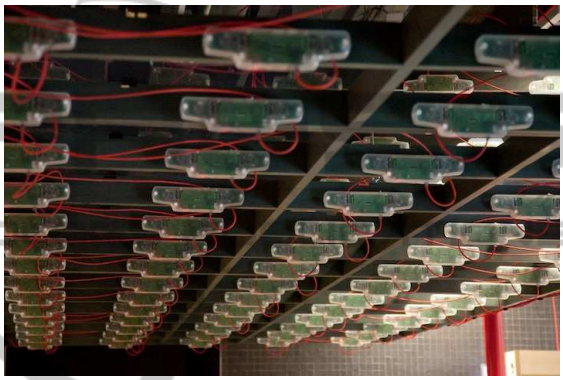


Figure 2: The Lille site of SensLab.

units. Nodes at the Grenoble site of SensLab are deployed in a square room of 14x14 m^2. The chosen operating system is TinyOS 2.1.0 with a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) protocol at the MAC layer, which is the default one included in the TinyOS distribution.

Several scenarios are presented in the following. In particular, we study how the system performance is affected by (*i*) the (common) transmission power P_t; (*ii*) the memory advertisment period T_{adv}; (*iii*) the number of deployed nodes N; and (*iv*) the number of replicas R.

4.1 Impact of Transmit Power

We argue that our greedy approach is significantly influenced by the network topology. For instance, in a dense network, where nodes have several neighbours, our data distribution mechanism could work better than in a sparse network with only a few direct neighbours.

The network topoloogy depends on the transmission power of the nodes. We run 4 experiments setting the transmit power P_t to 8.9 dBm, 0 dBm, -15 dBm, and -20 dBm, respectively, deploying $N = 78$ nodes.

We compute the number of detected neighbours in each case, by counting the number of memory advertisements received by each node. We assume node x is a neighbor of node y if node y detects at least one memory advertisement from node x, within the overall experiment duration. Note that the larger the number of received memory advertisments, the higher the radio link quality. In Figure 3, the number of detected neighbours (per node) and the corresponding average value are shown considering two values of the transmit power: (a) 0 dBm and (b) -20 dBm, respectively. As expected, the average number of detected neighbours decreases for decreasing transmit power. In Figure 4, the average number of detected neighbours, evaluated experimentally and analytically, is shown as a function of the transmit power. Analytical results are obtained as follows. Under the assumption of omnidirectional antennas (as is the case for SensLab nodes), the average number of 1-hop neighbours, denoted as $\overline{V}_1$, can be expressed as

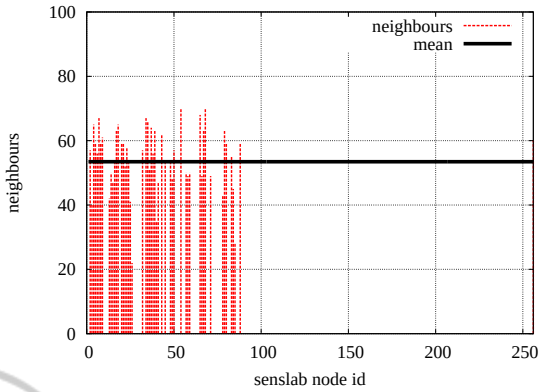
$$\overline{V}_1 = \rho \cdot \pi \cdot d^2 \qquad (2)$$

where $\rho = N/A = 78/14^2 = 0.4$ node/m^2 is the node spatial density and d is the transmit range (dimension: [m]). As the receiver sensitivity of the CC1100 receiver, denoted as $P_{\mathrm{r-min}}$, is -88 dBm (at $f_c =$ 868 Mhz with 500 Kbaud rate), taking into account a strong attenuation (the path loss exponent of the SensLab environment is around 3.75), from Friis formula one obtains:

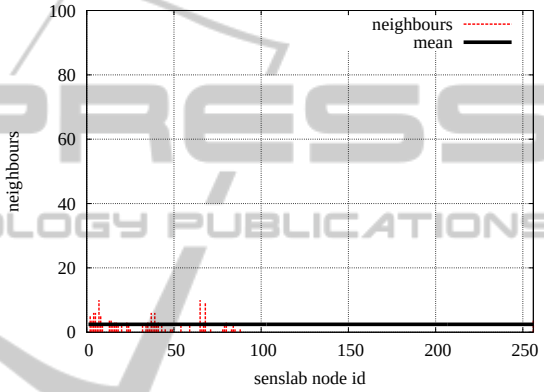
$$d = \frac{c}{4\pi \cdot f_c} \cdot \left(\frac{P_{\mathrm{t}} \cdot G_{\mathrm{t}} \cdot G_{\mathrm{r}}}{P_{\mathrm{r-min}}} \right)^{1/3.75} = 6.9 \cdot (P_{\mathrm{t}})^{1/3.75} \qquad (3)$$

where $G_{\mathrm{t}} = G_{\mathrm{r}} = 1$ (omnidirectional antenna's gain) and $c = 3 \cdot 10^8$ m/s is the speed of light. Using (3) into (2), one obtains the average number of neighbours as a function of the transmit power. As can be seen from Figure 4, the agreement between analysis and experiment is very good.

At this point, it is of interest to evaluate the time required by the system to reach the network storage capacity and the amount of dropped data due to local memory shortage. The network storage capacity denotes the maximum amount of distinct data that can be stored in the WSN. Given N nodes, the buffers $\{B_i\}$ (dimension: [data units]), and the sensing rates $\{r_i\}$ (dimension: [data units/s]), $i \in \{1, \ldots, N\}$, the network storage capacity C (dimension: [data units]) can be expressed as $C = \sum_{i=1}^{N} B_i = N \cdot B = 78 \cdot 250 =$ 19500. In Figure 5, the amount of data stored is shown, as a function of time, for the four considered experimental cases. As one can see, the capacity C is reached later when a lower transmission power is used — for instance at -20 dBm — since fewer neighbors are detected. Consequently, data cannot be distributed efficiently through the network. On the other



(a) 0 dBm.



(b) -20 dBm.

Figure 3: Number of detected neighbours versus SensLab node id. Nodes at the Grenoble site of SensLab are numbered from 1 to 256. $B = 250$ data units, $N = 78$ nodes, $T_{\mathrm{adv}} = 25$ s. The experiment duration is 15 mins.

hand, with a transmit power equal to 8.9 dBm, nodes have a "larger" view of the network and the capacity is reached earlier. For instance, at $t = 400$ s, the stored data at 8.9 dBm and 0 dBm are 17000 data units, about 90% of the capacity. The stored data at -15 dBm and -20 dBm, at the same time istant, are approximately 74% and 68% of the total capacity, respectively. Moreover, two theoretical cases are considered for comparison. In the case with local storage ("Local storage (anal)" curve), nodes fill their own local buffer autonomously, i.e., the fill up time for the i-node is $t_i = B_i/r_i$, where B_i and r_i are the buffer size and the sening rate of node i, respectively. In this case, the time interval to reach the storage capacity corresponds to the longest storage filling time across all nodes. As expected, the analytical curve relative to local storage lower bounds the experimental curves. In the case with ideal distributed storage ("Distr. storage (ideal)" curve), a performance benchmark can be obtained considering an *ideal* WSN where nodes can communicate with any other node, considering in-

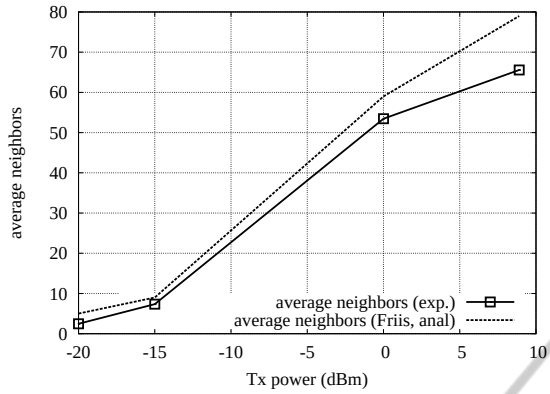


Figure 4: Average number of detected neighbors versus node transmission power. The deployed nodes are $N = 78$.

stantaneous transmission. In this case, the WSN is equivalent to a single super-node with a storage capacity C as defined above and sensing rate equal to $\sum_{i=1}^{N} r_i$.

In Figure 6, the amount of dropped data is shown, as a function of time, in the four cases considered in Figure 5. Nodes drop newly acquired data once their local memory has filled up and no neighbours are available for donating extra storage space. At $t = 400$ s, dropped data at 8.9 dBm and 0 dBm equal 600 data units, about 3% of the stored data. In the cases with lower transmit power, e.g., at -15 dBm and -20 dBm, the amount of dropped data is significantly higher. As expected, data dropping starts in advance with lower transmission power. In the same figure, the amount of dropped data, in the case with local storage, is also shown for comparison.

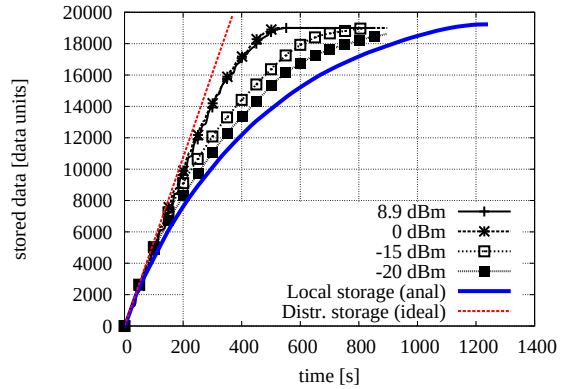


Figure 5: Data stored in the system varying the transmission power of the node. No replication($R = 1$), $B = 250$ data units, $N = 78$ nodes, $T_{adv} = 25$ s. The experiment duration is 15 mins.

We also measure the amount of data distributed throughout the network. The amount of distributed

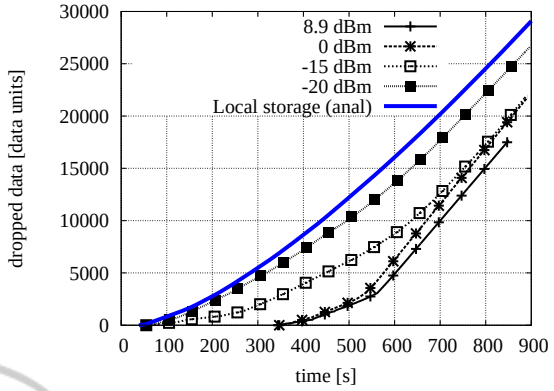
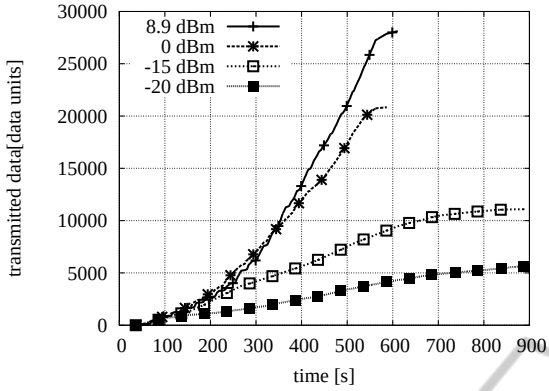


Figure 6: Data dropped in the system varying the transmission power of the node. No replication($R = 1$), $B = 250$ data units, $N = 78$ nodes, $T_{adv} = 25$ s. The experiment duration is 15 mins.

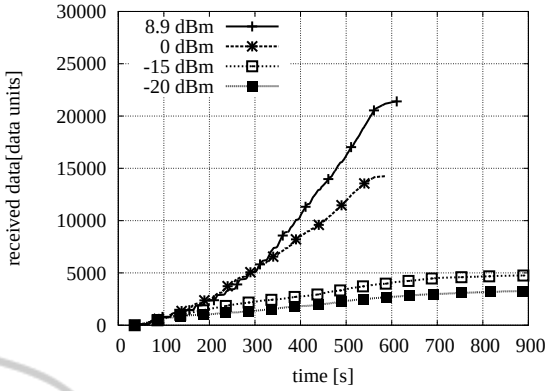
data, both transmitted to and received by donor nodes, is shown in Figure 7 as a function of time. As expected, distributed data increase with higher transmit power, since a node potentially has more donor nodes candidates than in other cases. However, we experience lot of data losses at the receiver. For instance, compare the y-axis of Figure 7(a) and Figure 7(b) at $t = 600$ s. In the ideal case with no collisions, the y-axis should show the same values, i.e., all transmitted data should be received at the donor node side, with a 100% packet delivery ratio (PDR). As one can see, however, in the cases at higher power — for instance at 8.9 dBm and 0 dBm — about 75% of transmitted data are received, while the remaining 25% are lost. For lower power settings, e.g., -15 dBm and -20 dBm, the percentage of lost data is still higher, reaching approximately 55% of the transmitted data. We conclude that several collisions occur at the MAC and physical layers, where a pure CSMA/CA protocol is used. However, we argue that high data losses are caused also by the limited deployment area, which is a room of 14x14 square meters. Such size can be too small for deploying 78 nodes and leads to lots of packet collisions.

4.2 Impact of the Memory Advertisement Period

In this subsection, we show the results obtained varying the memory advertisement period T_{adv} of the nodes. Recall from section 3 that the memory advertisement period is the time interval each node periodically broadcast its memory status to neighbours. We run 3 experiments setting the memory advertisement period T_{adv} to 25 s, 50 s and 80 s, respectively, out of a total experiment duration of 15 min. We deploy



(a) Transmitted data.



(b) Received data.

Figure 7: Distributed data, both transmitted and received, varying the transmission power of the node. No replication($R = 1$), $B = 250$ data units, $N = 78$ nodes, $T_{\text{adv}} = 25$ s. The experiment duration is 15 mins.

58 nodes with 250 data units as buffer, and a transmit power fixed at -15 dBm. Again, we do not perform replication ($R = 1$).

In Figure 8, the data stored is shown, as a function of time, considering the three cases. As one can observe, all curves coincide within the overall experiment duration and the storage capacity C, equal to 14500 data units, is reached simultaneously. We conclude that buffers are filled at the same speed in the three cases. The amount of memory advertisements exchanged across the network (also referred to as *Hello* messages) can be computed analytically considering the advertisment period T_{adv}. In particular, the number of transmitted hello can be expressed as

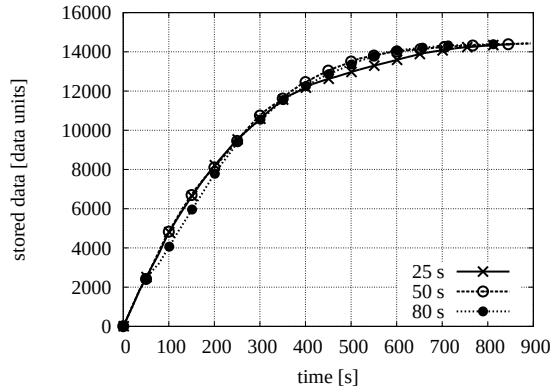


Figure 8: Data stored in the system varying the memory advertisement T_{adv}. No replication($R = 1$), $B = 250$ data units, $N = 58$ nodes, transmit power $P_{\text{t}} = -15$ dBm. The experiment duration is 15 mins.

$$\text{Tx}_{\text{hello}} = N \cdot \lfloor t/T_{\text{adv}} \rfloor . \quad (4)$$

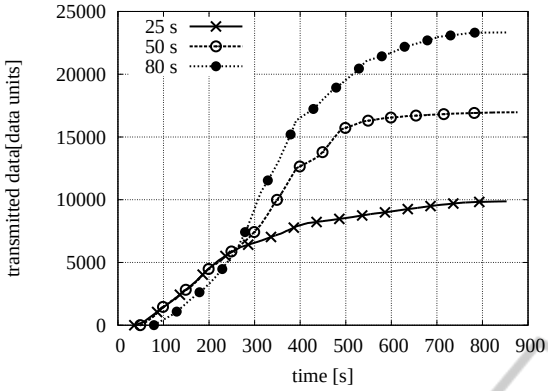
Similarly, the amount of received hello, on average, is given by

$$\text{Rx}_{\text{hello}} = \overline{V}_1 \cdot \text{Tx}_{\text{hello}} = \overline{V}_1 \cdot N \cdot \lfloor t/T_{\text{adv}} \rfloor . \quad (5)$$

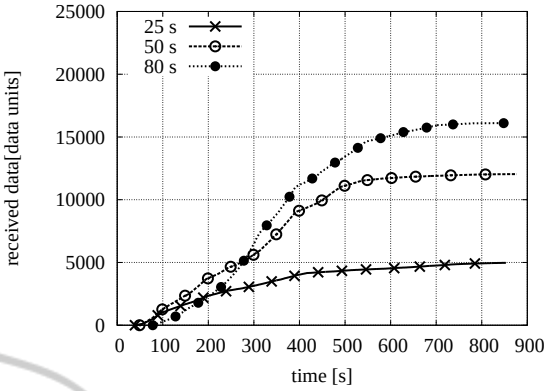
The amount of exchanged hello messages impacts the data distribution across the network. The distributed data, both transmitted and received, are shown in Figure 9. Data distribution increases with larger memory advertisement periods, i.e., $T_{\text{adv}} = 80$ s, while there are less distributed data for shorter T_{adv}. This is due to the fact that, with frequent memory advertisements, the memory table of a node is updated continuously with fresh information, and data is efficiently sent to the best donor node. On the other hand, with sporadic memory advertisements, a node keeps on selecting the same donor until it receives new state information. However, it is likely that data units sent to such a donor are not stored, since the donor has already filled up its local buffer. In this case, the donor node forwards the data unit to another claimed donor node. Data keeps on circulating in the network passing through nodes which have no local space for storage, but leading to energy depletion. Note that, given the node sensing interval in [0.1-5.1] s, between 16 and 400 data units are generated within an 80 s hello interval; therefore, on average, data acquired by a node between two consecutive memory advertisements is much more than the node buffer itself. An advertisement period of 25 s is thus more reasonable. Finally, as already observed in Section 4.1, significant data losses occur at the MAC layer, as it can be observed comparing the y-axis in Figure 9(a)- 9(b).

4.3 Impact of the Number of Deployed Nodes

In order to evaluate the impact of the number of deployed nodes, tests have been executed with 60,



(a) Transmitted data.



(b) Received data.

Figure 9: Distributed data, both transmitted and received, varying the memory advertisement T_{adv}. No replication($R = 1$), $B = 250$ data units, $N = 58$ nodes, transmit power $P_t = -15$ dBm. The experiment duration is 15 mins.

80 and 100 nodes, respectively, with a buffer size equal to 250 data units, a memory advertisment period $T_{adv} = 25$ s, the sensing interval uniformed and randomly distributed in [0.1-5.1] s, and the transmit power $P_t = -15$ dBm, for an experiment duration of 15 min. We have generated the same set of curves as for Section 4.2. Here we show the most relevant results. In Figure 10 the amount of stored data is shown, as a function of time, for the three considered values of N. To evaluate the amount of stored data S with an arbitrary number of deployed nodes N, we have derived a mathematical expression by fitting the three curves in Figure 10 with a 2nd grade polynomial function of t, directly proportional to N:

$$S(N) \simeq \begin{cases} N \cdot \left(-4 \cdot 10^{-4} \cdot t^2 + 0.6 \cdot t + 9.56\right) & t < 900 \text{ s} \\ 226 \cdot N & t \geq 900 \text{ s} \end{cases} \quad (6)$$

where the second line corresponds to the saturation point.

4.4 Impact of the Number of Replicas

As discussed in Section 3, redundancy is introduced by setting the number of copies (referred to as replicas) to a value $R > 1$. Replicas of a sensed data unit follow a hop-by-hop replication from the generator node to subsequent donor nodes. We computed the average hop distance reached by the replicas from the generator node, for various values of R. Our results, depicted in Figure 11, show that replicas do not propagate, on average, beyond 2 hops from the generator node. This is in agreement with the proposed mechanism, since donor nodes are selected on the basis of their available memory but not their physical position.

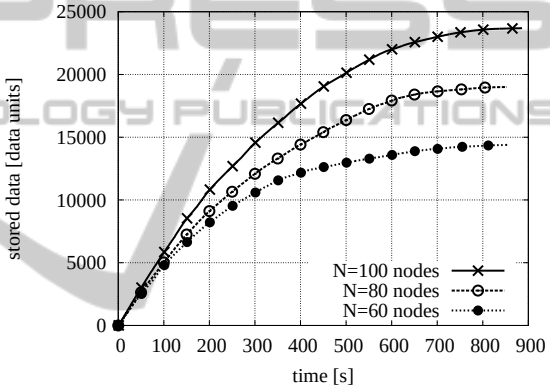


Figure 10: Data stored in the system varying the number of nodes N. No replication($R = 1$), $B = 250$ data units, $T_{adv} = 25$ s, transmit power $P_t = -15$ dBm. The experiment duration is 15 mins.

5 CONCLUSIONS

This paper has addressed the problem of redundant data distribution for WSN-based observation systems. We have proposed a low-complexity greedy mechanism to distribute and replicate measurements with a minimum signaling overhead. Through analytical results and experimentations on the SensLab large-scale testbed, we have shown how the performance is affected when different configurations of the main system parameters are used. To the best of our knowledge, this is one of the first works presenting experimental results at a really large-scale on SensLab. Further steps along these lines will consider the design of more sophisticated mechanisms to increase the spatial distribution of data. For this purpose, one goal is to add a routing scheme for an efficient placement of

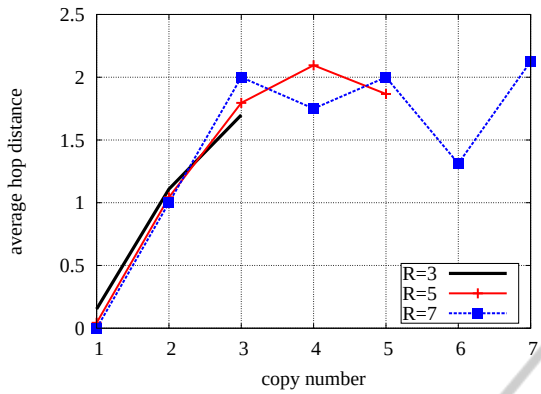


Figure 11: Average hop distance reached by the k-th replica versus k. k varies between 1 and 3 ($R = 3$), 1 and 5 ($R = 5$), 1 and 7 ($R = 7$), respectively. The average is computed on all the unique data present in the system with full redundancy, i.e., with exactly R copies.

data. Finally, we plan to extend our work to an IP-based scenario, using the 6LowPAN communication stack with optimized RPL routing strategy.

ACKNOWLEDGEMENTS

This work was funded by the European Community's Seventh Framework Programme, area "Internetconnected Objects", under Grant no. 288879, CALIPSO project - Connect All IP-based Smart Objects. The work reflects only the authors views; the European Community is not liable for any use that may be made of the information contained herein.

REFERENCES

Awad, A., Germany, R., and Dressler, F. (2009). Data-centric cooperative storage in wireless sensor network. In *2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies* (ISABEL), pages 1–6, Bratislava, Slovak Republic.

Beaufour, A., Leopold, M., and Bonnet, P. (2002). Smart-tag based data dissemination. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications* (WSNA'02), pages 68–77, Atlanta, GA, USA.

Ducrocq, T., Vandaele, J., Mitton, N., and Simplot-Ryl, D. (2010). Large scale geolocalization and routing experimentation with the senslab testbed. In *IEEE 7th International Conference on Mobile Adhoc and Sensor Systems* (MASS'10), pages 751–753, San Francisco, CA, USA.

Hamed Azimi, N., Gupta, H., Hou, X., and Gao, J. (2010). Data preservation under spatial failures in sensor networks. In *Proceedings of the eleventh ACM international symposium on Mobile ad hoc networking and computing* (MobiHoc'10), pages 171–180, Chicago, Illinois, USA.

Hongping, F. and Kangling, F. (2010). Overview of data dissemination strategy in wireless sensor networks. In *International Conference on E-Health Networking, Digital Ecosystems and Technologies* (EDT), pages 260–263, Shenzhen, China.

Luo, L., Huang, C., Abdelzaher, T., and Stankovic, J. (2007). Envirostore: A cooperative storage system for disconnected operation in sensor networks. In *26th IEEE International Conference on Computer Communications* (INFOCOM'07), pages 1802–1810, Anchorage, Alaska, USA.

Madden, S. R., Franklin, M. J., Hellerstein, J. M., and Hong, W. (2005). Tinydb: an acquisitional query processing system for sensor networks. *ACM Trans. Database Syst.*, pages 122–173.

Mathur, G., Desnoyers, P., Ganesan, D., and Shenoy, P. (2006). Ultra-low power data storage for sensor networks. In *The Fifth International Conference on Information Processing in Sensor Networks* (IPSN'06), pages 374–381, Nashville, TN, USA.

Neumann, J., Reinke, C., Hoeller, N., and Linnemann, V. (2009). Adaptive quality-aware replication in wireless sensor networks. In *International Workshop on Wireless Ad Hoc, Mesh and Sensor Networks* (WAMSNET'09), pages 413–420, Jeju Island, Korea.

Omotayo, A., Hammad, M., and Barker, K. (2007). A cost model for storing and retrieving data in wireless sensor networks. In *IEEE 23rd International Conference on Data Engineering Workshop* (ICDE), pages 29–38, Istanbul, Turkey.

Piotrowski, K., Langendoerfer, P., and Peter, S. (2009). tinyDSM: A highly reliable cooperative data storage for wireless sensor networks. In *International Symposium on Collaborative Technologies and Systems* (CTS'09), pages 225–232, Baltimore, Maryland, USA.

Ratnasamy, S., Karp, B., Shenker, S., Estrin, D., and Yin, L. (2003). Data-centric storage in sensornets with GHT, a geographic hash table. *ACM Mobile Networks and Applications*, pages 427–442.

SensLAB (2008). A very large scale open wireless sensor network testbed. Website: http://www.senslab.info/.