Reliable Cooperative Information Storage in Wireless Sensor Networks

Tatiana K. Madsen, Anders Grauballe, Mikkel Gade Jensen, Achuthan Paramanathan, Janne Dahl Rasmussen, and Frank Fitzek
Aalborg University, Department of Electronic Systems
Niels Jernes vej 12 A, 9220 Aalborg East, Denmark
Email: (tatiana, agraubal, mgade, apku04, jannedr, ff)@es.aau.dk

Abstract—Reliable data retrieval from a sensor network can require a temporal distributed data storage of the measured data among all motes. This is the case when a sink of information, or a gateway, is only occasionally present in the network. We consider a network consisting of a set of motes that in order to save energy independently of each other go into offline mode, switching their radios off. We propose a cooperative mechanism for distributed data storage that allows complete data retrieval with high probability. Reliable data retrieval is crucial for data-critical applications. Reliability is achieved by introducing redundancy in the system by applying Reed-Solomon scheme for data encoding. This solution can tolerate the situation where a certain number of devices in the sensor cluster is unavailable at the moment of information retrieval. A probabilistic analytical model is developed to investigate the performance of the proposed algorithm. This model is verified by simulation and prototype testing. Prototyping is done on the OpenSensor platform developed at Aalborg University. Performance results indicate that the proposed cooperative data distribution method outperforms the non-cooperative one.

I. INTRODUCTION

Wireless sensor networks are gaining more and more attention and they are expected to be widely used in communication applications, such as intelligent home control systems or safety control in industries. A sensor in these systems could be a small device with the purpose of measuring environmental conditions such as humidity, pressure, temperature etc. and being able to communicate and relay raw or preprocessed data to a control unit or gateway. The role of the gateway is to transfer the collected information to an end user of this information. If the immediate transmission of the measured data from the measuring device to the gateway is not possible for some reason, the collected data should be stored in the network sensors. The data communication from the sensors to the gateway may not be possible at all times due to broken routes; this is the case in situations where the gateway is busy communicating with other nodes or in situations where the gateway is simply not always present in the network.

As the variety of applications of wireless sensor networks (WSN) increases, together with the amount of collected data to be archived, the problem of data storage is receiving more and more attention. Designing a proper data storage methods, a limited memory of sensor nodes should be taken into account. Another characteristic of WSN that impacts the design is the requirement on energy-efficient operation. Since the sensors are battery powered, the collected data is lost after the sensors are depleted of power. Additionally, one of the most popular techniques to reduce power consumption and prolong the network lifetime is the usage of power-sleeping modes, where devices alternate between active and sleep mode (see e.g. [1], [2]). Even though the collected data of a sleeping device is preserved, the availability of this data is reduced as it can not be retrieved when a device is offline. These examples emphasize the necessity of a distributed data storage, especially for data-critical applications.

When the retrieval of all collected information is required, a reliable in-network data storage should be implemented. The main mechanism for achieving data reliability is by introducing data redundancy. One of the possible approaches is to introduce dedicated storage nodes which main functionality is to back up the data collected in their proximity. For example, a storage architecture with a proxy tier, PRESTO [3], utilizes proxy nodes that can cache previously transmitted data and query responses. The LUSTER (Light Under Shrub Thicket for Environmental Research) system [4], developed for environmental applications, includes extra storage nodes. In [5] a thorough analysis of the optimal storage nodes placement is conducted. These research works have demonstrated the advantages of using extra devices e.g. in terms of alleviation of transmission load. In this paper we investigate methods for distributed data storage without deployment of any extra storage nodes. The question is if it is feasible to achieve cooperative data storage on sensor boards.

To improve the reliability of data storage, we advocate the cooperative approach in this paper. Recently, there has been great interest in cooperative communication [6]. Exploiting the cooperative diversity in WSN, a pool of common resources can be created that can be used e.g. to enhance reliability of the transmission link (see [7]). In this paper a pool of common resources is formed by aggregating the storage capacity of all cooperative nodes. We propose a data distribution method that utilizes the Reed-Solomon coding scheme and allows the complete data reconstruction even in case of partial data retrieval.

The reminder of the paper is organized as follows. In Section II the problem we are tackling is presented. Proposed solution is described in Section III. Analysis of the data storage algorithm is given in Section IV. Section V compares coopera-
tive and non-cooperative approaches in design of data storage algorithms. Section VI provides simulation and measurement results to complete the performance evaluation of the system. Finally, we give our conclusions in the last section.

II. PROBLEM FORMULATION

We consider a data-critical application that requires a certain reliability of the data retrieval process. The scenario, considered in the paper, is a cluster of sensors, often called motes, which are within a communication proximity of each other. Our focus is on information storage and retrieval of data collected within the cluster; the cluster formation problem is outside the scope of this work.

The data measurements are occasionally collected by a gateway that is not always present as a part of the system. The gateway is staying in the network a limited period of time (e.g., needed to request and download the measured data from each mote in the cluster). The time of its next arrival is not predefined, but is random. Examples that motivates this scenario are a technician coming to collect measurements; an airplane collecting data from a sensor network distributed in a deserted and unreachable area and many more.

We assume that in order to save battery power the motes go periodically into a sleeping mode (turning both radio off and going into microcontroller power down mode). It means that if at the time of the gateway presence some motes are offline, they are not able to send their data to the gateway.

If each of the sensors in the cluster stores their own information, thus, there is no cooperation between the devices, the resulting system will be unreliable as the gateway will miss the measurements from inactive sensors. The goal of this study is to investigate if a cooperative distributed data storage can improve reliability of the data retrieval process.

III. PROPOSED ALGORITHM

To increase the probability to retrieve a data measurement one can redundantly store the data in the network. For example, a full copy of the measurement of a mote can be distributed to other motes in the cluster. The reliability of data retrieval process is improved: in order to extract the measurement, the node itself should not necessarily be active and participate in the communication with the gateway. The data will be relayed on his behalf by other nodes. However, this approach requires large storage capacity on the motes that might not be available. Instead of distributing the full copy of data, we consider distribution of the encoded data where one of the error-correction codes is used. A suitable method in relation to reliability and memory consumption is the Reed-Solomon (RS) coding scheme which is a technique used to ensure data reliability and to make systems fault-tolerant. Examples of the use of RS are in error detection/correction of CDs, DVDs, RAID storage and DVB systems [9].

In RS $l$ is defined to be the number of data devices and $m$ is the number of redundant checksum devices. The total number of devices in the system is $n = m + l$. The data on each devices is divided into words $w$ that is the word size in bits. The RS coding is performed as a linear combination of these words and checksum words on the checksum devices. In RS it is possible to recover the data if up to any $m$ devices in the system are missing. An example could be a system with 4 devices each holding 4 bits of information, so $l = 4$ and $w = 4$. If the goal is to recover the devices in the case where any 3 devices fail, then the number of checksum devices must be $m = 3$.

Since no extra checksum devices are assumed to be present in the network, the following data encoding and distribution is proposed. Each mote collects the data individually. When the data is collected, it is replicated and distributed to each of the motes in the cluster. The distribution of data from a sensor to other motes in the cluster is done in the following way applying RS coding in the system:

1) Data is split into $l = n - m$ parts
2) $m$ redundant parts are generated using RS on the data parts resulting in a total of $n$ parts
3) The parts are distributed to $n$ motes (including itself), one part for each mote

The steps in the algorithm are illustrated in Fig. 1.

Fig. 1. A message is split into three data parts ($d_1$ to $d_3$) and one redundant data part ($c_1$) is added. Three of the parts are distributed to other motes and one is kept on the mote itself.

One can note that if the number of nodes in the cluster is $n$, then the level of redundancy $m$ can be chosen between 0 and $n - 1$. The case $m = 0$ corresponds to the situation of non-cooperative devices in the cluster: everybody just keeps its own data. In order to retrieve all measurements, the gateway should contact and receive the information from each node individually. The case $m = n - 1$ presents another extreme: every node in the cluster receives a full copy of the measurements from all neighbors. In this case in order to retrieve all data, it is enough for the gateway to contact only one device.

The process of encoded data distribution among the nodes in the cluster will take some time. Its duration depends on the sleeping pattern of the nodes and the properties of the underlying MAC protocol. We assume that while distributing encoded data to its neighbors, a node persistently stays online until all $n - 1$ packets are sent. Only after that the node will delete its measurement from its memory keeping its own 1/($n$-th part of the encoded information. If the gateway starts data retrieval process before the data distribution phase is complete, the node just sends the whole measurement together with the stored encoded data from neighbors.
When the gateway arrives to collect data, it is done in the following way:

1) The gateway broadcast a signal to announce its presence.
2) Online motes stays online throughout the gateway session.
3) Gateway requests data from each mote. If a timeout occurs on request, the mote is considered offline.
4) Gateway receives data from each online mote.
5) With the gateway request each mote receives a new session number.
6) When the gateway has received data from all online motes, it leaves the cluster.
7) The online motes will then measure and distribute new data.
8) When offline motes become online, they learn about the new session receiving the new data being distributed by other nodes. Then they delete old data and initiate a new measurement round.

In the proposed solution any arbitrary mote can fail as long as the total number of offline motes does not exceed the number of checksum motes. This property makes RS an ideal coding scheme for the distribution model because it can provide a good trade-off between reliability and memory consumption and tolerate random failures in the system.

IV. ANALYSIS OF DATA STORAGE ALGORITHM

In order to evaluate the proposed solution a mathematical model of the system is presented in this section. First a scenario will be presented in detail, describing the parameters considered in the model. The reliability of the system is the main focus of the analysis.

A. System scenario

The cluster consists of \( n \) motes. The level of redundancy of the system is \( m \) \((m < n)\), that is, the system can tolerate \( m \) nodes failures and still the complete data retrieval can be performed. No motes will fail (uncontrolled breakdown) at any time, however a node can go into sleeping mode and become inactive.

After finishing measurements, a node encodes the data and starts the data distribution (DD) phase: one packet should be sent to each neighbor in the cluster. After the completion of the data distribution phase, the mote enters its idle phase during which it has a possibility to go offline. We assume that the time is slotted and at the beginning of each slot a node randomly chooses to stay online with probability \( p_{ON} \). Each node makes this decision independently. For simplicity of the model it is assumed that the probability to be online \( p_{ON} \) is the same for all nodes in the cluster and it does not change in time. Furthermore, it is assumed that while one node is distributing data, all other nodes are in the idle phase.

The time between two successive gateway arrivals is modelled as an exponential random variable with parameter \( \lambda \). The gateway arrival triggers the online motes to send their data so the gateway is able to regenerate the measurements from the received packets. When the gateway has left, the measurement and distribution phase will start again and the motes which were offline when the gateway was present will delete their packets and start measuring again. This scenario can be seen on a time line in Fig. 2 and 3.

\[
\text{Fig. 2. Gateway arrives after the data distribution phase is complete.}
\]

\[
\text{Fig. 3. Gateway arrives before the completion of the data distribution phase.}
\]

An ideal MAC protocol is assumed in our modelling. All packets sent are received correctly. Delays due to propagation delays and contention to the channel access are assumed to be small and they are not incorporated into the model.

B. Reliability

In this section a mathematical probabilistic system model of the sensor network is derived. The aim of this model is to describe the probability \( p_{R} \) that a gateway successfully will reconstruct the distributed message from the network.

By using the RS coding scheme, the system can tolerate loosing as many data parts as there are checksum parts. The number of online nodes is a binomial random variable with parameters \((k, p_{ON})\). Thus, the probability of a successful reconstruction of the message given a correct distribution can be found as

\[
p_{nk} = \Pr(\text{Number of online motes} \geq n - m) = \sum_{k=n-m}^{n} \binom{n}{k} p_{ON}^k (1 - p_{ON})^{n-k}
\]

Equation 1 can be used to find a lower bound on the probability \( p_{ON} \) which will satisfy the reliability requirements represented as the probability \( p_{nk} \). For example, if the Reed-Solomon RS(12,4) coding scheme is used (total number of motes is set to \( n = 12 \) and redundancy level is \( m = 4 \)) and it is required that each message must be reconstructed with a probability \( p_{nk} = 0.80 \), then \( p_{ON} \) can be found as

\[
\sum_{k=8}^{12} \binom{12}{k} p_{ON}^k (1 - p_{ON})^{12-k} = 0.8
\]

\[
\Leftrightarrow \quad p_{ON} = \begin{cases} 
-0.38 \\
0.73
\end{cases}
\]
In this example, each mote must be online with a probability of at least 0.73.

One should note that \( p_{nk} \) presents a conditional reliability of the system under the assumption that the gateway arrives after the data distribution phase is complete. The unconditional system reliability is higher since in situations when the gateway arrives early, the node still has its message and it can be retrieved. Therefore, the system reliability can be found as:

\[
p_r = \Pr(\text{Message successfully reconstructed}) = \Pr(\text{DD phase is complete}) \cdot \Pr(k \text{ motes online}) + \Pr(\text{DD uncomplete}) \cdot 1 = p_d p_{nk} + (1 - p_d) \tag{2}
\]

where \( p_d \) denotes probability that DD phase is complete at the time of the gateway arrival.

Let us assume that the gateway has arrived at the \( N \)-th time slot (that is between time \((N-1)T\) and \(NT\)); \( T \) is a slot duration; it can be normalized to 1. The DD phase will be completed before the \( N \)-th time slot, if each of the neighbors is online at least during one time slot within the \( N \) slots (see Fig. 4). The probability \( p_d \) can be calculated based on the binomial distribution and is given by the following formula as a function of \( N \):

\[
p_d(N) = \Pr(\text{Message distributed in time } N \cdot T) = \Pr(\text{All motes online somewhere within time } N \cdot T) = 1 - \Pr(\text{One or more motes still offline at time } N \cdot T) = 1 - \sum_{i=1}^{n-1} \binom{n-1}{i} (p_{OFF})^i (1 - p_{OFF})^{n-i-1},
\]

where \( p_{OFF} = 1 - p_{ON} \).

The probability that the gateway appears during the \( N \)-th time slot is given by:

\[
p_G = e^{-\lambda N T} (e^{\lambda T} - 1)
\]

where \( \lambda \) is a parameter of the exponential distribution that can be interpreted as the arrival rate. Since probability \( p_d \) depends on the parameter \( N \), in order to calculate the total system reliability we have to take an average over the whole range of \( N \). Thus, the final formula for \( p_r \) can be found as:

\[
p_r = \sum_{N=1}^{\infty} [p_d(N) \cdot p_{nk} + (1 - p_d(N))] \cdot p_G \tag{3}
\]

Fig. 5 shows the graph of the system reliability calculated using Equation 3 compared with the conditional reliability \( p_{nk} \) calculated by Equation 1. It can be seen that for large values of \( p_{ON} \) the system reliability \( p_r \) converges to \( p_{nk} \), since the larger \( p_{ON} \) is, the sooner the data distribution phase is completed and higher is the probability that the gateway arrives while the node is in the idle phase. If \( p_{ON} \) tends to 0, it means that the neighboring nodes are constantly offline. The data can not be distributed and it is kept in the node’s memory until gateway arrival. This explains why \( p_r = 1 \) when \( p_{ON} = 0 \). The higher the gateway arrival rate \( \lambda \) is, the higher is the reliability \( p_r \), since the node simply does not have time to distribute the data. If the message has not been distributed, it can be successfully retrieved by the gateway (we assume no transmission errors).

One can see that when values for \( \lambda \) decreases, the curve for \( p_r \) follows very closely the curve for \( p_{nk} \). For such \( \lambda \) as \( p_{nk} \) can be used as a good approximation of the system reliability.

C. Storage requirement

Assuming that the cluster consists of \( n \) motes in total and the redundancy level is \( m \), it leaves \( n - m \) data motes. Each data mote and checksum mote hold \( \frac{1}{n-m} \) of the message. Thus the total memory consumption for one message in the cluster \( c_{total} \) is:

\[
c_{total} = \frac{n - m}{n - m} + \frac{m}{n - m} = \frac{n}{n - m}
\]

This ratio also represents the total memory consumption in each mote when holding messages for all other motes (including own message). Figure 6 illustrates the memory consumption, that we also refer to as the storage requirement, for the cluster of \( n = 21 \) nodes as a function of \( m \). Equation 4 can be used to decide the number of checksum motes, if there is a requirement to the maximum memory consumption and the total number of motes is known.
V. COMPARISON OF COOPERATIVE AND NON-COOPERATIVE APPROACHES

Figure 7 shows the dependency of the system reliability on the redundancy level $m$ for the cluster of 12 devices. Recall that $m = 0$ corresponds to the non-cooperative behavior, while $m = 20$ means cooperative behavior but with no data encoding. Values $1 \leq m \leq 19$ shows cooperative behavior with different parameters of the RS scheme. From Equation 1 the formula for the non-cooperative case can be obtained by letting $m = 0$:

$$p_{nk} = p_{ON}^m$$

One can observe that non-cooperative behavior results in very poor system reliability and in order to achieve some reliability, the values for $p_{ON}$ should be chosen very high, thus minimizing the energy savings. Choosing a high redundancy level $m$, a high reliability can be met even when nodes are sleeping a significant amount of time. However, the price to pay for it is the high memory requirement as it can be seen from Fig. 6. There is a clear trade off between system reliability, storage requirements and energy efficiency.

Cooperative and non-cooperative approaches are furthermore compared in Fig. 8 for different parameters of the RS coding scheme. The probability that the gateway is able to reconstruct a message if the cooperative data distribution algorithm is applied is shown with solid lines and the case of no cooperation among the motes in the cluster is given by the dashed line. These results again confirm our intuition that using the cooperative method applying Reed-Solomon coding is better than the non-cooperative case. For example in Reed-Solomon a reliability at 50% can be guaranteed with only 0.50 online probability where the non-cooperative case needs an online probability at approximately 0.80 to ensure the same reliability. A online probability in both methods at 0.70 gives a reliability in Reed-Solomon at 78.5% and in the non-cooperative case at 35%.

VI. SIMULATION AND MEASUREMENT RESULTS

In order to validate the presented analytical model a simulation of the system has been implemented in Java. To ensure that the motes are independent regarding a state shift, each mote runs in a separate thread. Furthermore, an ideal MAC protocol is assumed, that is, each sent packet is received without any errors.

Moreover, to have an indication of how the system will perform in a real life conditions (e.g. taking into account packet losses), a prototype has been designed and implemented. The prototype is based on the OpenSensor platform developed by Aalborg University [8]. The OpenSensor v2.0 consists of two different boards stacked on each other which gives the flexibility of adding more features in a later design. The board has the following features:

**Main board**
- Li-Poly battery interface which can be used for power supply to the mote
- Mini USB interface which can be used for both serial RS-232 interface to the mote and external power supply. It is also possible to charge the Li-Poly battery by changing the jumper settings on the board.
• dsPIC33FJ256GP710 (Microchip) microprocessor for controlling the mote
• 22.1 MHz oscillator as external clock source for the dsPIC
• ICD 2 debugger/programmer interface

**Wireless board**

• Bluetooth module (Amber Wireless AMB2300) for wireless communication with mobile phones or PCs, connected with serial RS-232 connection to the microprocessor
• nRF905 (Nordic Semiconductor) transceiver for communicating via the ISM band
• Loop antenna for the RF transceiver

The testbed consists of four devices, where one is used as a gateway and three as motes. Each of the distinct prototype motes has a routine which includes executing several tasks during a given period of time. A state diagram for one mote in the system can be seen in Fig. 9.

![State diagram for one mote.](image)

Three different test cases with distinct online probabilities {0.5, 0.8, 0.9} are tested. In each test case 200 samples are collected. Due to the limitation in the number of motes in the prototype, it was possible to test the performance of the proposed protocol only with the RS(3,1) scheme.

Figure 10 summarizes the results from prototype testing and simulations. For reference the system reliability calculated using the analytical model is plotted as well. One can observe that the simulation results follows closely the analytical results. However, the results from the prototype test shows a deviation from the analytical and simulation results. The measured probability to retrieve a message is lower. This deviation can be explained by the fact that ideal MAC protocol is used for the analytical and simulation models. In the implemented testbed a packet loss is not negligible. It is worth mentioning that the implemented MAC incorporates a collision avoidance mechanism based on the carrier sensing. We can conclude that the implemented MAC protocol still needs fine tuning and optimization for the usage in sensor networks. In parallel we are working on describing quantitatively the observed packet loss pattern and incorporating it in our analytical model.

VII. CONCLUSION

This paper deals with the problem of reliable data storage in wireless sensor networks, the proposed solution is relying on the cooperative behavior of the motes within one cluster, where the information collected by one node is cooperatively stored by all devices in the cluster. The measured data is encoded using the Reed-Solomon coding scheme, thus, redundancy is introduced into the system. In order to retrieve the complete data, the stored information from \( n - m \) devices should be available, where \( n \) is the number of devices in the cluster and \( m \) is redundancy level.

An analytic probabilistic model has been developed to evaluate the system performance in terms of the probability to retrieve the measured information and the required storage capacity. The trade-off between system reliability, storage requirements and energy efficiency has been illustrated. Furthermore, the analytical results are compared with simulation results and the results from the prototype testing.

**REFERENCES**