

Lifetime Optimization for Wireless Sensor Networks with Correlated Data Gathering

Nashat Abughalieh, Yann-Aël Le Borgne, Kris Steenhaut, Ann Nowé

► **To cite this version:**

Nashat Abughalieh, Yann-Aël Le Borgne, Kris Steenhaut, Ann Nowé. Lifetime Optimization for Wireless Sensor Networks with Correlated Data Gathering. WiOpt'10: Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, May 2010, Avignon, France. pp.252-258, 2010. <inria-00503914>

HAL Id: inria-00503914

<https://hal.inria.fr/inria-00503914>

Submitted on 19 Jul 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Lifetime Optimization for Wireless Sensor Networks with Correlated Data Gathering

Nashat Abughalieh¹, Yann-Aël Le Borgne^{1 2}, Kris Steenhaut^{1 3}, Ann Nowé²

¹Vrije Universiteit Brussel - ETRO

²Vrije Universiteit Brussel - COMO

³Erasmus hogeschool Brussel - IWT
Brussels, Belgium

Emails: {nabughal, yleborgn, ksteenha, ann.nowe}@vub.ac.be

Abstract—The nodes in wireless sensor networks often collect correlated measurements. Not taking into account this information redundancy is detrimental to the network lifetime, since communication is often the most energy consuming task for a sensor node. This paper tackles this issue by proposing an approach based on Distributed Source Coding (DSC), in which the rate assignments are adapted over time. The distinctive feature of the DSC technique is to make the compression independent of the routing. We rely on this feature to design two algorithms applicable to multi-hop routing trees to optimize the network lifetime. The first algorithm is the Updated CMAX (UCMAX) which improves the centralized CMAX routing algorithm, by considering the energy loss due to packet forwarding in multi-hop networks. The second algorithm is called Adaptive Compression Rate (ACR), and aims at maximizing the network lifetime by better balancing the energy losses in the network. Experimental results show that the proposed approach is easy to tune, and may significantly extend the network lifetime, particularly for dense, multi-hop networks.

Index Terms—Distributed Source Coding, Network Lifetime, Wireless Sensor Network.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are composed of low cost sensor nodes, which can be used for a wide range of applications. WSNs can be deployed for event detection, e.g. fire or intruders detection or for monitoring physical phenomena in an environment such as temperature, humidity, sound or light intensity. In particular, the technology of WSNs is envisioned to be of major importance interest in applications related to defense and disaster monitoring, such as battlefield monitoring and enemy detection.

The sensor nodes in a WSN have typically short battery life. Most WSN applications are however expected to run autonomously for months or years. Conserving battery energy is therefore of primary importance [1], and a large body of research has focused in the recent years on the design of techniques for reducing the energy consumption in WSNs.

In some applications, like environmental or battlefield monitoring, the nodes are typically tasked to periodically report their measurements to a centralized node referred to as base station (BS). The base station usually benefits from higher computational and energy resources (e.g. a desktop PC), and allows to monitor the variations of the measurements over the whole field. The sensor nodes transmit their data to the BS by

means of their on-board radio. The data of the nodes which are not in radio range of the BS is relayed by intermediary nodes in a multi-hop fashion. The role of the nodes in a sensor network is therefore twofold: they not only act as measurement systems, but also as routers for the measurements collected by the more distant nodes.

The sensor nodes in a WSN monitor a common phenomenon, and therefore the data of adjacent nodes is very often correlated. Since radio transmission is the most energy consuming component of a sensor nodes, a number of research efforts have recently been targeted at the design of compression techniques to suppress the redundancies in the correlated data. Compressing the data of adjacent nodes can save the sensor node's energy, reducing the amount of data transmitted in the network. The design of compression techniques for sensor networks is however challenging, as it requires optimization of both routing and coding of the data in the network.

Since routing and coding are difficult optimization problems, their optimization has been mainly studied separately in literature. The design of an optimal routing structure for maximizing network lifetime is shown to be NP-hard in [10]. The design of routing algorithms must therefore rely on heuristics, such as those included in CMAX [7]. CMAX is a centralized power aware routing algorithm which takes into account the residual energy of the sensing nodes while searching the optimal routing path.

For the source coding problem, two main approaches were introduced in literature: Explicit Communication (EC) and Distributed Source Coding (DSC) [2], [4]. In Explicit Communications [2], each node compresses its data using incoming flows as explicit side information, and can also compress incoming data from other nodes. For example, if a node receives data from two correlated nodes, it can use one of the node's flow as side information to encode the other node's data. The coding relies on side information, which must be available both at the encoder and the decoder. The use of EC techniques is however computationally challenging, particularly as the number of data sources increases. Besides the complexity of compression, finding the optimum solution for the joint problems of routing and compression has been shown to be NP-hard [2]. This moved the attention to an alternative coding approach called Distributed Source Coding DSC [4].

In DSC [4], the encoding takes place at the sensing node,

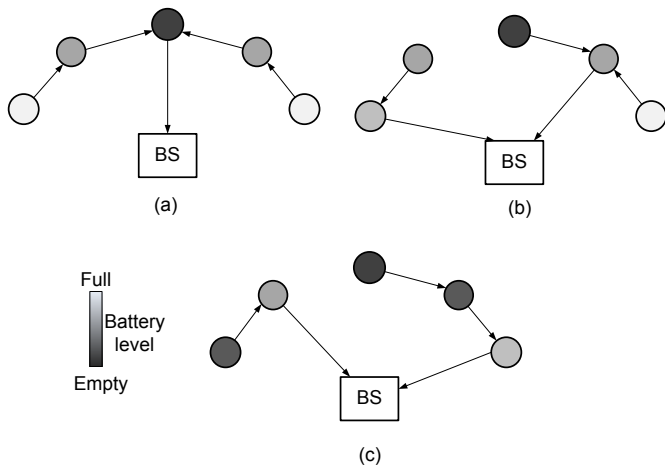


Figure 1. Evolution of routing. (a) A first routing tree is set up. Note that the central node sustains the highest energy consumption due to data forwarding. (b,c) Changing the routing tree over time allows to better balance the energy consumption among the set of nodes.

and side information needs only to be available at the decoder (the BS). The technique however requires the knowledge of the joint probability distribution of the data sources' measurements. If this assumption is met, the advantages of the DSC technique are twofold. Firstly, the computational cost at the encoders is very low. Secondly, and more importantly, the compression process is independent of the routing path, since the coding is done at the source node without knowing other nodes' data. Some previous works implemented DSC with channel coding techniques at the coding node to increase the robustness of the network against errors. Different coding techniques have been proposed, such as turbo codes [13], LDPC [14] or Cosset coding [6]. These techniques were shown to provide near optimal compression rates.

In this paper, we build on these works by considering the optimization of routing and coding in multi-hop networks. The routing structures and coding rates are adapted over time, which allows to better distribute the energy consumption among the sensor nodes. This is illustrated in Figure 1, where the routes are changed over time in order to balance energy consumption. Our algorithm relies on the CMAX routing algorithm and the DSC coding technique. More precisely, we optimize the network lifetime over two parameters: the routing path and the coding rates, for arbitrary sensor networks with one BS. Since the DSC is done at the source nodes, the two parameters can be optimized separately. In our solution, we divide the lifetime optimization into two optimization steps. First, we optimize the routing paths, using the centralized UCMAX algorithm. Second, we optimize the compression rates for the routing structure found with the first algorithm.

The paper is organized as follows. Section II discusses the network model. Section III and IV describe the UCMAX and ACR algorithms, respectively. Section V evaluates the performance of our algorithms on the basis of simulation results. We conclude in Section VI.

II. SYSTEM MODELING AND PROBLEM FORMULATION

Let $V_n = \{v_1, v_2, \dots, v_n\}$ be a set of n sensor nodes, and let v_0 denote the base station. The network is modeled as a directed graph $G(V, E)$, where $V = V_n \cup v_0$ are the network nodes and E is the set of edges. We further denote the initial energy of the node v_i by $IE(v_i)$, $1 \leq i \leq n$, and the current energy of the sensor v_i by $CE(v_i)$, $1 \leq i \leq n$. For each $(v_i, v_j) \in E$, let $e_{Tx}(v_i, v_j)$ denote the energy that node v_i requires to transmit one packet to node v_j , which depends on the channel quality between the two nodes. The channel quality depends mainly on the distance separating the nodes, but also on a number of other environmental characteristics such as channel fading, obstacles and interference. We consider that the channel quality between each node and its neighboring nodes is estimated at the nodes. Let e_{Rx} further denote the energy consumed for receiving one packet, which we assume to be the same for all nodes. Let $t \in \{1, 2, \dots, m\}$ denote the *update period*, at which the network optimizes the routing tree and the compression rates. m is the total number of *update periods* from the beginning of the data collection until the network is unable to deliver the network compressed data to the BS.

At *update period* t , let r_i^t be the source coding rate of the measurements collected by sensor v_i , and let $r_{i,j}^{t,k}$ be the coding rate of the measurements collected by sensor v_k , and forwarded by v_i to v_j . The target is to maximize m (the total number of updates) by solving the following linear optimization problem:

$$\max_{r_i^t, r_{i,j}^{t,k}} \{m\} \quad (1)$$

$$\sum_{j \in V \setminus \{i\}} r_{i,j}^{t,k} - \sum_{j \in V \setminus \{i\}} r_{j,i}^{t,k} = \begin{cases} 0, & v_i \neq v_k, v_0 \\ r_i^t, & v_i = v_k \\ -r_i^t, & v_i = v_0 \end{cases} \quad (2)$$

$$\forall v_i \in V, v_k \in V \setminus \{d\}, \forall t \in \{1, 2, \dots, m\}$$

$$\sum_{t=1}^m \sum_{v_j \in V \setminus \{v_i\}} \sum_{v_k \in V \setminus \{v_i\}} (e_{Tx}(v_i, v_j) r_{i,j}^{t,k} + e_{Rx} r_{j,i}^{t,k}) \leq IE(v_i), \forall v_i \in V \setminus \{v_0\} \quad (3)$$

$$\sum_{v_i \in V_S} r_i^t \geq H(V_S/V_{Sc}), \forall V_S \subseteq V, t = 1, 2, \dots, m \quad (4)$$

Equation 1 shows that the optimization problem is twofold, and combines a routing optimization problem with a rate assignment problem. The routing optimization problem depends on the coding rates $r_{i,j}^{t,k}$ of the forwarded data, whereas the rate assignment problem depends on the source coding rates r_i^t . Equation 2 simply states that the amount of data generated by a node is the sum of its own amount of data and the amount of data it forwards. Equation 3 states that the energy dissipated in each node, in transmitting and receiving, is limited by the residual energy of the node's battery. Equation 4 correspond to the Slepian-Wolf rate region for correlated data sources [15], which states that the sum of the coding rates of a subset of nodes $V_S \subseteq V$ cannot be less than the entropy of the

measurements of V_S conditioned on the knowledge of the data generated by the complementary subset of nodes $V_{Sc} = V \setminus V_S$.

This optimization problem requires to search for the optimum routing tree and compression rates, and was shown to be NP-hard in [2] for Explicit Communication coding. However, choosing DSC for encoding the data allows separating the coding optimization problem from the routing optimization problem. Although this makes the problem easier to tackle, it must be noted that the routing optimization problem is NP-hard [8]. Different routing heuristics have therefore been designed. We relied in this article on the CMAX algorithm, which was introduced in [7] to approximate the optimum routing tree. The algorithm is centralized and runs at the BS. Every *update period* the BS adapts the routing structure, in such a way that the energy consumption in the network is balanced over time. A first contribution is adapting the algorithm by changing the weighting functions of the links between nodes. We present UCMAX in Section III.

For the coding problem, Cristescu *et al* in [4] searched for minimizing the total power losses in the network using DSC. They proved that the minimum total power loss in the network can be achieved using DSC when encoding rates are assigned according to the Equation 5. They considered that the optimum routing structure is the Shortest Path Tree (SPT) and the nodes are numbered according to the delivery cost of one packet from node v_i to the BS through the routing tree, where the delivery cost is the sum of all energy losses in the source and forwarding nodes in the routing path from node v_i to the BS.

For simplicity we will name this rate configuration as Minimum Total Power Consumption (MTPC). This rate assignment seeks to minimize the total network load, without taking into account the node's residual energy.

$$\begin{aligned} r_1 &\geq H(v_1) \\ r_2 &\geq H(v_2/v_1) \\ &\vdots \\ r_n &\geq H(v_n/v_{n-1}, \dots, v_1) \end{aligned} \quad (5)$$

Our second contribution is built on this strategy. It starts by assigning compression rates with MTPC, but adaptively updates the compression rates every *update period*. We avoid assigning high data rates for low residual energy nodes, while maintaining low total power consumption. This rate assignment algorithm works on top of the optimized routing tree generated by UCMAX algorithm. After each *update period* interval, the network first updates the routing tree using the UCMAX algorithm, then ACR compression rate algorithm searches for the optimum rate assignment. We present our compression rate algorithm in Section IV.

III. UCMAX ALGORITHM

In order to explain UCMAX we first introduce CMAX [7] on which our extension UCMAX is built. The CMAX algorithm runs at the BS after collecting nodes' batteries energy levels and the channels quality estimates. The CMAX algorithm runs in two steps:

Step 1: If all network nodes' batteries are full (i.e. $CE(v_i) = IE(v_i)$), jump to Step 2 without modifying the graph G . Else eliminate from G every edge (v_i, v_j) for which $CE(v_i) < e_{Tx}(v_i, v_j)$. Change the weight of every remaining edge (v_i, v_j) to $e(v_i, v_j) \times (\lambda^{\alpha(v_i)} - 1)$, where $\alpha(v_i) = 1 - CE(v_i)/IE(v_i)$ is the *energy utilization* of node v_i . λ is a configurable parameter, which quantify the penalty of using a link, according to the remaining energy in the transmitting node.

Step 2: Find the shortest path between each sensor node and the Base Station using Dijkstra's algorithm in the modified graph.

CMAX does not take into account the energy needed for receiving data when optimizing the routing tree, while the energy used in receiving can be as high as the energy used in transmitting. Therefore we updated the CMAX to include the reception costs while updating the weights of the graph's edges. Our modified UCMAX runs the following steps:

Step 1: If all the network nodes' batteries are full (i.e. $CE(v_i) = IE(v_i)$), jump to Step 2 without modifying the graph G . Else eliminate from G every edge (v_i, v_j) for which $CE(v_i) < e_{Tx}(v_i, v_j)$. Change the weight of every remaining edge to become

$$e(v_i, v_j) \rightarrow e(v_i, v_j) \times (\lambda^{\alpha(v_i)} - 1) + e_{Rx} \times (\gamma^{\alpha(v_j)} - 1) \quad (6)$$

λ and γ are configurable parameters, which quantify the penalty of using the link $e(v_i, v_j)$ according to the remaining energy of the transmitting node v_i and the remaining energy of the receiving node v_j .

Step 2: Find the shortest path between each sensor node and the BS using Dijkstra's algorithm in the modified graph.

The parameter γ allows to explicitly penalize the routing paths which involve low energy nodes. As will be shown in Section V, considering the energy used by receptions allows UCMAX to provide better performance than CMAX.

To understand our update on the CMAX algorithm, let us go through the example in Figure 2. The BS needs to find the optimum routing path for the two network nodes v_1, v_2 . In CMAX the nodes communicate their residual energy to the BS. Link weights $e_{Tx}(v_i, v_j)$ are known to the BS. CMAX changes network link's weights as shown in Step 1 of the CMAX algorithm. Then run Dijkstra's algorithm on the updated graph. If node v_1 routes its data through v_2 , v_2 will lose almost twice the energy lost in v_1 when forwarding v_1 's data. The energy lost in receiving at v_2 could be as high as the energy lost in transmitting the data to the BS. This problem motivated us to use the transceiver energy loss in receiving in changing the link's weights. Equation 6 shows our new weighting equation.

IV. ADAPTIVE COMPRESSION RATE ALGORITHM (ACR)

The minimum total power consumption rates with DSC compression in sensor networks are achieved with MTPC [4]. MTPC optimizes the network rates to minimize network total power, while not considering the per node residual energy or per node forwarding load. MTPC can indeed assign high transmitting rate for a node that has high data flow from other

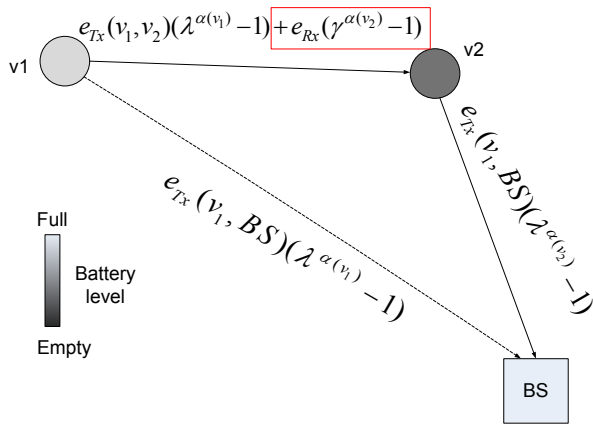


Figure 2. UCMAX optimum routing example, with two nodes and the BS.

nodes in the routing tree, which can cause a fast disconnectivity in the network. Our algorithm seeks the minimum total power consumption, but on the other hand protects low energy nodes by assigning low rates to them. Every time an update is required, the ACR algorithm is run at the BS, using the routing tree returned by UCMAX. ACR also uses the residual energy of the nodes for assigning low rates to low energy nodes.

ACR algorithm performs as follows. After optimizing the network routing tree with the UCMAX and learning the correlation structure between the nodes' data, the BS assigns the data compression rates according to MTPC rate assignment as derived in equation 5. After each *update period*, the UCMAX calculates the new routing tree and the ACR adaptively changes the data rates of the nodes such that nodes with higher residual energy and lower forwarding load send at higher rates than the nodes with less residual energy or high forwarding duty. The algorithm keeps the total network rate to be at least the joint entropy of all nodes.

The Rate Optimization Algorithm executes the following steps:

Step 1: Assign compressing rates according to equation 5 where nodes are numbered according to the nodes weights, so that $w(v_1) \leq w(v_2) \leq \dots \leq w(v_n)$, where $w(v_i)$ is the total energy to transmit one bit of data from node i to the BS through the UCMAX optimum route tree. Let the total delivery power parameter be $P_L = \sum_{i=1}^n w(v_i)r_i$.

Step 2: Find the node with minimum residual energy v_{min} , and search in all nodes for one with lower data rate, lower forwarding rate and higher residual energy. This node should not be one of v_{min} 's children on the routing tree. Let the elected node be v_x .

Step 3: Swap v_x and v_{min} 's compression rates so that if node v_{min} was transmitting at rate r_{min} and v_x was transmitting at rate $r_x < r_{min}$, the new rate for v_{min} will be $r_{min} \geq H(v_{min}/v_1, v_2, \dots, v_x, \dots)$ while node v_x will transmit at rate $r_x \geq H(v_x/v_1, v_2, \dots)$.

Step 4: Calculate the total delivery power, let it be P_{new} . If all rate switching possibilities are tested, choose the rate switching that leads to the minimum total delivery power P_{new}

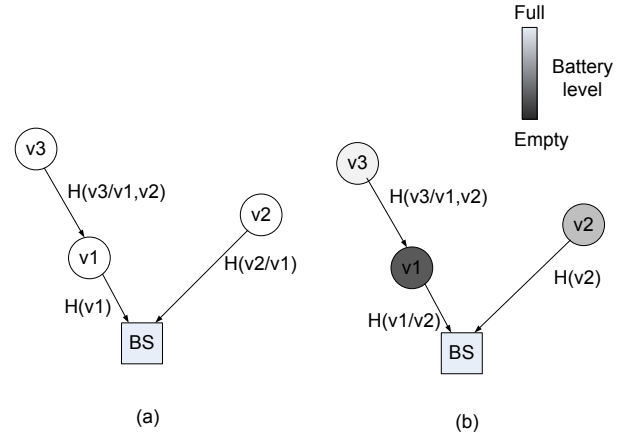


Figure 3. ACR Algorithm. (a) Rates at start-up of the network. (b) Rates obtained with the ACR Algorithm after *update period*

and let it be P_{best} , else return to step 3.

Step 5: If all minimum residual energy nodes are tested move to step 6. Else back to Step 2 to search for other minimum residual energy nodes.

Step 6: If P_{best} is less than zP_L , where $z \geq 1$ is a constant, accept the new rates. Else, do not change the compression rates.

The worst case happens when each node in the network is selected in Step 2 as minimum residual energy node and it checks all other nodes for possible rate switching between them. The ACR algorithm complexity is bound by $O(|V|^2)$.

An example for ACR algorithm is shown in Figure 3, where three nodes network is depicted. The network starts up with UCMAX route tree and MTPC rate assignment as shown in Figure 3 (a) (for simplicity we neglect the training period). After the network runs for one *update period* delivering packets to the BS, nodes transmit their residual energy to the BS. The new optimum route is calculated with UCMAX. The ACR first calculates the MTPC rates and then searches for the node with minimum energy. It will be v_1 , since it was sending at full entropy rate and forwarding v_3 's compressed data for the previous *update period*. The algorithm searches for a node that has higher battery level and less data flow through it, which is v_2 . If r_2 is less than r_1 the algorithm swaps the rates so that v_2 will transmit at rate $r_2 \geq H(v_2)$ while v_1 will transmit at rate $r_1 \geq H(v_1/v_2)$.

V. EVALUATION

The main goal of our experiments is to evaluate the gains which can be obtained in terms of network lifetime. After presenting in Section V-A the setup used for the experimental evaluation, we compare in Section V-B the UCMAX to the CMAX algorithm. Section V-C then provides an analysis of ACR algorithm, combined with UCMAX. Finally, Section V-D discusses the network overhead caused by the compression rate algorithm.

A. Experimental setup

In order to evaluate the performance of the UCMAX and ACR algorithms, we rely on a simulated environment of $100 \times 100 m^2$ area, and consider two network configurations. In the first one, nine nodes are located on a 3×3 grid, and in the second one, the set of nodes is extended to twenty-five, arranged in a 5×5 grid. In both cases, the BS is located at the center of the area. These configurations allow to compare the performance of the algorithms in function of node density.

The energy consumption of the nodes is estimated by the energy consumed by the radio transceiver, which is widely considered as the dominant source of energy consumption in the wireless node [12]. For the sake of our analysis, we consider that the network nodes are time synchronized, and therefore only consider the energy required to transmit or receive a packet.

As stated in Section II, the receiving energy e_{Rx} is the same for all nodes. The transmitting energy $e_{Tx}(v_i, v_j)$ depends on the distance separating the two nodes v_i and v_j , we rely on the common model

$$e_{Tx}(v_i, v_j) = [\alpha d(v_i, v_j)^\kappa + \rho] \times T_{Tx}$$

where α , κ , and ρ are constants that depend on the radio chip characteristics and the environmental conditions. T_{Tx} is the packet transmission period. For our experiments we depend on [12] to calculate these parameters. Table I shows the parameter's values used in our experiment.

The correlation between the sensor's measurements is represented by a Gaussian model

$$f(x) = \frac{1}{\sqrt{2\pi} \det(K)^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)K^{-1}(x-\mu)} \quad (7)$$

where K is the covariance matrix and μ is the vector of the measurement's average. The matrix K actually represents the spatial correlations between the measurements, and is created by assuming that these correlations are changing according to distance between the nodes. More precisely, the following model

$$K_{i,j} = \sigma^2 \exp(-c|d(v_i, v_j)|^2)$$

is used to define this relationship. $K_{i,j}$ and $d(v_i, v_j)$ respectively represent the correlation and the distance between node v_i and v_j . σ^2 is the variance of the nodes' measurements (we consider all nodes' measurements have the same variance) and c is the attenuation factor of the correlation [4]. The output of each node is quantized with a uniform quantizer whose step level $\Delta = 0.01$.

In our simulations, a training period is first carried out in order to learn the correlation model and the entropy of the sources. During this training period, only the UCMAX algorithm is running, and the packets are sent uncompressed to the BS. In a second stage, the BS starts running the joint optimization algorithms by broadcasting the compression rates with the routing tree structure to all the nodes. We consider the correlation structure is being constant.

Table I
EXPERIMENT PARAMETERS

$\alpha = 5.219 \times 10^{-4}$	$\sigma = 1$
$\kappa = 3.5$	$c = 0.001$
$\rho = 1.2 \times 10^{-5}$	Bit rate = 250 kb/s
Transmission range = 100 m	maximum packet size = 128 bytes
$IE(v_i) = 10$ Joule	maximum $T_{Tx} = 4.1$ ms
$e_{Rx} = 59.1 \times 10^{-3} \times T_{Tx}$ Joule	

B. CMAX versus UCMAX

The proposed UCMAX algorithm extends CMAX by including, in the weights of the network graph, the energy cost related to packet reception (Equation 6). This cost is parameterized by the user defined constant γ , and is the counterpart of the constant λ used for penalizing the transmission costs in both CMAX and UCMAX.

The role of λ was studied in [7], and we summarize here their analysis. From the weight update equation in CMAX (see Section III), we see that link weights are updated with λ to the power of the *energy utilization*. Therefore, increasing the value of λ allows to avoid the use of nodes whose remaining energy is low. Given that the relationship is of a power type, the routes algorithm becomes insensitive to changes in λ when it is larger than or equal to 100. We found similar results, and therefore λ is fixed to 100 in the following.

In Figure 4, we evaluate how γ , the new parameter introduced by UCMAX, can extend the network lifetime. The figure shows the lifetime of the network configurations, in number of *update periods*, as a function of γ , for values ranging from 0 to 30. Note that if $\gamma = 0$, UCMAX behaves as CMAX, since the energy consumption related to packet reception is not taken into account. As γ increases however, we see that UCMAX allows to significantly extend the lifetime, particularly for the 5×5 network. The figure shows that the optimum value for γ is 10 for both the 3x3 and 5x5 grid networks.

Interestingly, we also observe that with UCMAX, the network lifetime of the 5×5 network gets even longer than the 3×3 network. This happened because in the 3×3 network the distances between the nodes are high, so that the energy lost in transmission is higher than the energy lost in reception. Because of that UCMAX did not give much improvement in CMAX performance. In the 5×5 grid the network is more condense, then the energy loss in reception is comparable to the energy loss in transmission. The aggregating nodes in the routing tree will lose a considerable amount of their energy in receiving data from the children nodes, which is not taken into account in CMAX. Figure 4 shows the improved performance of UCMAX over CMAX.

In these results, the compression rates were set by relying on the MTPC as in [4], and the routing tree was adapted every *update period* which equals to 50 packets delivery from each node to the BS. We consider the network is time synchronized. The influence of these parameters is studied in the following experiments.

C. Adaptive Compression Rates

We study in the section how the proposed ACR allows to improve the network lifetime. The main parameter of the

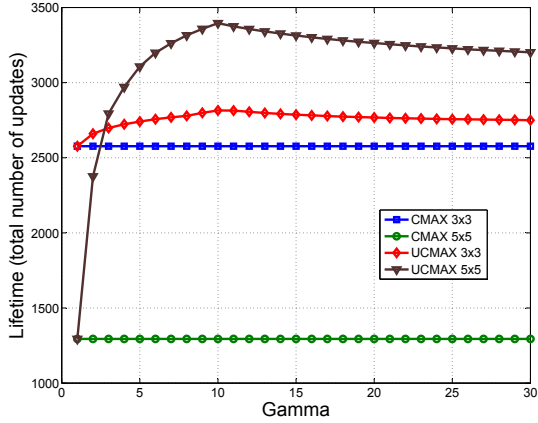


Figure 4. Network lifetime achieved of CMAX and UCMAX on 3×3 and 5×5 grid networks, with different values of γ

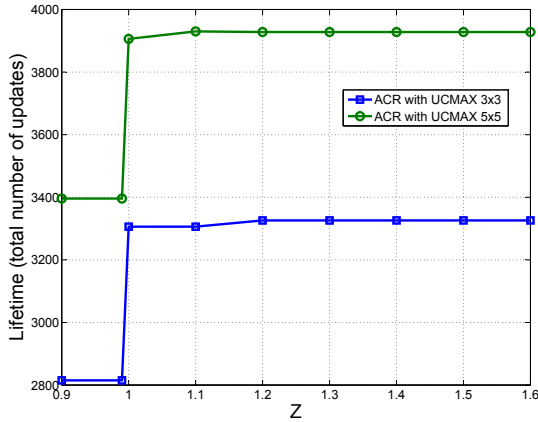


Figure 5. Network lifetime as a function of the z for the 3×3 and 5×5 networks

algorithm is the constant z , which determines the threshold from which the new rates are accepted or rejected. The lifetime of the network as a function of z is reported in Figure 5, where z is varied from 0.9 to 1.6, for the 3×3 and 5×5 network configurations.

If z , is less than 1, the new rates are in all cases rejected, and therefore the algorithm uses the compression rates as in MTPC algorithm. If the network nodes are distributed uniformly around the BS (nodes have equal delivery power to the BS), small value of z (with $z \geq 1$) or even $z = 1$ can lead to optimum lifetime. The ACR algorithm can swap compression rates between the nodes with equal deliver power to the BS without changing the total deliver power. For random distributed nodes, the values of z need to be more than 1, to give the algorithm the flexibility to switch the rates between the nodes with different delivery power.

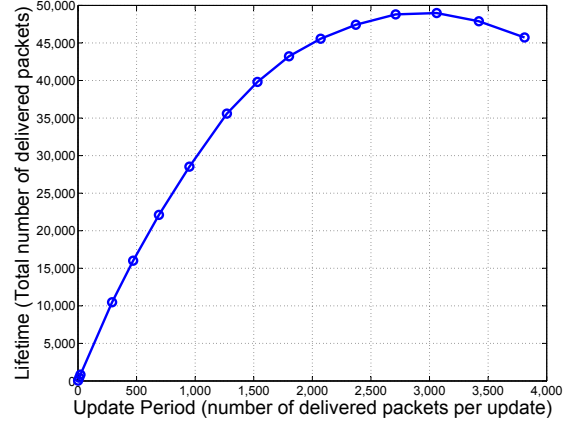


Figure 6. Network lifetime as a function of *update period*

D. Update rate trade-offs

Updating the network routing tree and compression rates allows to better balance the load. It however comes at a communication cost. The updating of the network routing tree and rates requires that the nodes transmit their residual energy to the BS. Then, the BS must broadcast the new rates and routing tree through the network. As the network size increases, updating the routes and the compression rates may incur significant communication overhead. The communication overhead caused by these updates is mainly dependent on the data sent from the BS to the nodes. The nodes' residual energy is indeed a small piece of information, which can be communicated to the BS via the data packets. When the BS broadcasts the data related to the new routing tree and compression rates, the communication overhead is however not negligible, as the nodes which receive the broadcast packet must retransmit it to their neighbors until the packet is communicated throughout the whole network. For this broadcast, we assume that every node receives and transmit the packet once. Each node therefore loses an amount of energy equivalent to that of receiving and transmitting one packet at maximum transmission power. We run the algorithm with different *update periods*. The *update period* is changed by changing the number of packets delivered from each node before each update. We computed lifetime which is represented by the total number of packets delivered from each node until one of the nodes runs out of energy. The results are reported in Figure 6, for the 5×5 network configuration. If the period of the updates is small, the communication overhead caused by these updates considerably reduces the network lifetime. At the other extreme, if this period is too large, the lifetime also decreases, as some of the nodes run out of energy although another routing tree or rate assignment could have saved their energy. In this experiment, the optimal period for updating the network is of about 2900 packets delivery from each node per *update period*.

VI. CONCLUSIONS

This paper presented two algorithms for WSN lifetime maximization. The first one is the Updated CMAX algorithm

(UCMAX), which improves CMAX by taking into account the radio reception costs for determining routing paths. Our experiments showed that UCMAX provides important improvements in terms of lifetime. This was particularly remarkable in the case of dense networks, where UCMAX increased by up to 160% the network lifetime. The second proposed algorithm, namely the Adaptive Compression Rates (ACR), aims at balancing the energy consumption by adapting the DSC rate assignments over time. In our experiments, the ACR further improved the lifetime by 17%. The two algorithms form a joint optimization solution for maximizing the lifetime of networks collecting correlated data.

ACKNOWLEDGMENTS

Part of the work presented in this article was supported by the FWO (Fonds Wetenschappelijk Onderzoek), Flanders, Belgium, project G.0219.09N

REFERENCES

- [1] Chang, J.-H., Tassiulas, L.: Energy Conserving Routing in Wireless Ad-hoc Networks. *Proceeding of INFOCOM*, Vol. 1. IEEE (2000) 22-31
- [2] R. Cristescu, B. Beferull-Lozano, M. Vetterli and R. Wattenhofer, "Network Correlated Data Gathering with Explicit Communication: NP-Completeness and Algorithms," *IEEE/ACM Transactions on Networking*, vol. 14, pp. 41 – 54, 2006
- [3] L. Huiyu and G. Pottie, "Designing Routes for Source Coding With Explicit Side Information in Sensor Network," *IEEE/ACM Transactions on Networking*, vol. 15, no 6, Dec 2007
- [4] R. Cristescu, B. Beferull-Lozano, M. Vetterli "Networked Slepian-Wolf: Theory, Algorithms and Scaling Laws," *IEEE Trans. Inf. Theory*, Vol 51, no.12,pp 4057-4073, Dec.2005
- [5] Slepian, D., Wolf, J.K. "Noiseless Coding of Correlated Information Sources," *IEEE Trans. Information Theory*, IT-19, 1973, pp. 471-480
- [6] J. Chou, D. Petrovic, K. Ramchandran, "A distributed and adaptive signal processing approach to exploiting correlation in sensor networks", *Elsevier, Ad Hoc Networks Volume 2, Issue 4*, October 2004, Pages 387-403
- [7] K. Kar, M. Kodialam, T. V. Lakshman, L. Tassiulas, "Routing for Network Capacity Maximization in Energy-constraints Ad-hoc Networks", *Proc. of Infocom 2003*.
- [8] Aslam J., Li Q., Rus D "Three Power-aware Routing Algorithms for Sensor Networks. *Wireless Communications and Mobile Computing*," Vol. 3. *John Wiley & Sons* (2003) pp. 187-208
- [9] MicaZ and Mica2, both developed by *Crossbow Technology Inc.* (www.xbow.com)
- [10] X. Su, "A combinatorial algorithmic approach to energy efficient information collection in wireless sensor networks," *ACM Trans. on Sens. Netw.* vol. 3, no. 1, Mar. 2007.
- [11] A. Swami, Q. Zhao, *Wireless Sensor Networks: Signal Processing and Communications*, John Wiley & Sons (2007), pp.95-100
- [12] Cui, S., Goldsmith, A. J., & Bahai, A. (2005). "Energy-constrained modulation optimization". *IEEE Transaction on Wireless Communications*, 4, 2349-2360.
- [13] J. Bajcsy et al. "Coding for the Slepian-Wolf Problem with Turbo Codes," *Proc. IEEE Global Communications Symposium*, pp. 1400-1404, San Antonio, TX, Nov. 2001
- [14] M. Sartipi et al. "Distributed Source Coding using LDPC Codes: Lossy and Lossless Cases with Unknown Correlation Parameter," *Forty-Third Annual Allerton Conference on Communication, Control and Computing*, October 2005
- [15] Cover, T. and Thomas, J. *Elements of Information Theory*, John Wiley & Sons 1991