Proportion based protocols for load balancing and lifetime maximization in wireless sensor networks

Andrei Dumbrava, Rahim Kacimi, Riadh Dhaou and André-Luc Beylot
University of Toulouse
IRIT-ENSEEIHT
Toulouse, France

Abstract— **This paper presents the problem of minimizing energy consumption and maximizing lifetime in a many-to-one sensor network. In such network pattern, all sensor nodes generate and send data to a single and fixed base station (BS), via multi-hop transmissions. When all the sensor data have to be forwarded to a single BS via multi-hop routing, the traffic pattern is highly non-uniform, putting a high burden on the sensor nodes close to the BS. Some strategies that balance the energy consumption of the nodes and ensure maximum network lifetime by balancing the load are proposed and analyzed. The key element of the research is the use of multiple transmission power levels. We studied an optimal solution for calculating the hop-by-hop traffic proportions for the particular case of nodes having just two transmission power levels, and compared the results given by the heuristics with those from the optimal analytical case. Another goal is to propose and implement a systematic approach for the construction of the sensor network based on real sensor nodes. The neighbor discovery phase, the way in which the BS finds out the network topology and than impose the strategy and decide whether the nodes to act locally or respect the instruction from the sink are part of the protocol that is described in the paper.**

Keywords-component; wireless sensor network, lifetime maximization, load-balancing, transmission power control, power consumption, simulated annealing, optimization

I. INTRODUCTION

The lifetime of Wireless Sensor Networks (WSN) is crucial. Network lifetime is the time span from the deployment to the instant when the network is considered non functional. This however is application-specific. It can be, for example, the instant when the first sensor dies, a percentage of sensor die, the network partitions or the loss of coverage occurs [1]. In this paper, we investigate the problem of energy consumption and lifetime maximization in a many-to-one sensor network. In such network pattern, all sensor nodes generate and send data to a single and fixed BS, via multi-hop transmissions. Although different definitions of lifetime exist in the literature [1,2], a sensor network that has a specifically defined grid topology, as we considered in our research, certainly has to be considered "dead" whenever it is no longer able to forward data to the BS. It is assumed that every sensor node in the network has an equal probability of generating data packets, that have to be forwarded to the single sink via multi-hop routing, using other sensor nodes as relays. It is obvious that the burden on the nodes close to the BS is considerably higher than on the nodes that are far away. They will die quickly, making the network useless, because those nodes should transport messages that originate on all the others sensors from the rest of the network, and also must transmit their own messages. We analyze and propose some strategies that balance the energy consumption of the nodes and ensure maximum network lifetime by balancing the load.

The research was developed in the framework of the "CAPTEURS" project founded by the National Telecommunications Research Network (RNRT) from France. The aim of the project was to design a solution [4] for monitoring the temperature on the whole cold chain, from the warehouses to the retailer, and being aware of the fact that minimizing energy consumption is a key goal in many multi-hop wireless networking systems, especially when the nodes of the network are battery powered. We focused on designing some new strategies to balance the overall energy consumption of the network and on maximizing the lifetime of sensor networks which are constructed on regular topologies and which have stationary nodes.

In this paper we propose some heuristics that extend the network lifetime by balancing the load between the network nodes, and the key element of our research is the use of multiple transmission power levels.

Moreover, we studied an optimal solution for calculating the hop-by hop traffic proportions for the particular case of nodes having just two transmission power levels, solution which is detailed in another paper [5]. In the present article we compared the results given by the heuristics with those from the optimal analytical case. We believe that the network overall lifetime may be increased when nodes send data not only to the closest neighbors towards the BS, using the minimum transmission power level, but also trying to reach farther neighbors or even directly the BS, by using a higher transmission power level. Although this approach may seem to consume more power locally, at every node, we proved that the overall network power consumption is reduced.

A systematic approach for the construction of the sensor network based on real sensor nodes was also implemented. The developed communication protocol involves the neighbor discovery phase, the way in which the BS finds out the network topology and than impose the strategy and decide whether the nodes to act locally or respect the instructions from the sink.

II. Load-Balancing Strategies for Lifetime Maximization in WSN

In this section there are designed and analyzed several load-balancing methods for a regular grid topology. The nodes are considered to be uniformly and stationary deployed. In order to achieve an extension in lifetime of the network, two transmission power levels are considered.

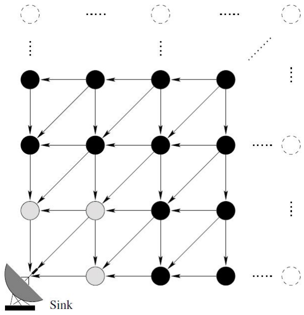

Figure 1. WSN with a grid topology and Sink in the corner. Sensors near the Sink will die faster than the other ones.

Some assumptions should be made in the case of a grid network with all-to-sink traffic pattern:

- Nodes are uniformly distributed in a grid topology with size N. Moreover, the density is uniform throughout the entire network (Fig. 1).
- Each node continuously generates constant bit rate data and sends it to the BS through multi-hop routes.
- The hop by hop routing and load-sharing is made between the accessible nodes. Indeed, load sharing becomes possible without any signalling protocol, which could load the network traffic considerably. Basically, the calculations can be made earlier in the life of the network (calculations may be computed by the BS) and only the proportions should be transmitted to the different sensors.
- "Mostly-off" network pattern is better than "Mostly-on" one and that is why is more accurate to talk about proportions than probabilities because the load-sharing by probabilistic routing is costly and requires "Mostly-on" nodes.
- It is supposed that a Medium Access Control (MAC) protocol exists and it is based on perfect scheduling, so that no collision or retransmission occurs.
- Sensors nodes have two different transmission ranges of d and $\sqrt{2}\, d$ meters with respective power levels TPL_1 and TPL_2.
- Since energy consumption is proportional to transmission power, it is assumed that $E(TPL_2) \approx 2E(TPL_1)$.

A. Equal-Probability Heuristic

This heuristic is the less complex one. A straightforward way to make load balancing is to take a local decision and every node should distribute the load equally between the different paths identified to reach the BS. Whenever a packet arrives to a node and it should be forwarded through the network with the BS as the final destination, the node chooses randomly the neighbor to send the message to. The neighbors are chosen with equal probability, no matter if they can be reached with TPL_1 or TPL_2.

Although this heuristic may seem like a very simple approach, it can improve a simple routing protocol, in order to vary the static paths calculated to route the packets to the BS. In this way, it would not exist anymore just one active neighbor, which is supposed to carry the whole load. The energy consumed with forwarding the packets would be split and shared between all the neighbors.

B. Shortest-Path Heuristic

A common method to prevent neighbors unnecessarily consume energy is to choose the shortest path or share the load between the shortest paths to the BS, when the node has several. In the context of energy balancing the shortest-paths are the paths that have the lowest cost in terms of energy consumption.

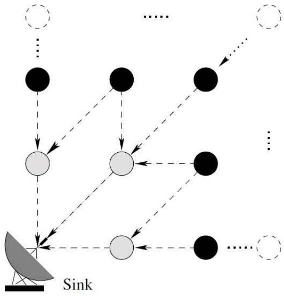

Figure 2. Shortest-Path strategy.

It has been considered that the cost of every link is made of two components: the receiving cost, which is constant and independent of the distance and the transmission cost, which varies with the distance, as stated in the assumptions made above. In order to choose the shortest path to the BS, every node should sum the link costs from itself to the BS and chose one or more, if it is the case, with the minimum cost. It has also been considered that in the case of a node having multiple shortest paths, before every transmission it will randomly choose one of them.

In the case of the grid topology shown in Fig. 2, the nodes on the two border lines leading to the BS transmit their data always in the direction of their boundary line, due to the fact that sending in diagonally, to a neighbor that could be reached with TPL_2 would be too expensive for the node's energy consumption (the cost would be greater). While the shortest paths of the rest, is often to take the diagonal link or go to the main diagonal of the grid. We can conclude that the most critical node will be the one on the main diagonal, the closest to the BS.

C. Contribution-Based Heuristic

Here it is proposed a heuristic which attempts to improve the load balancing and increase the network lifetime by distributing contributions from the BS to the network, depending on the number of neighbors and corresponding power level. The heuristic illustrated by Fig. 3, works in the following steps:

1. Starting with the BS, each node calculates the contribution of each of its downstream neighbors, taking into account the power of these neighbors to reach it. For example, the BS has 3 downstream neighbors: "1", "3", and "2" (with transmission power level 2) so the contributions are respectively 2/5, 2/5 and 1/5 (Fig. 3).

2. Thereafter, each node communicates each contribution to the corresponding downstream neighbor. Before distributing the contributions to downstream neighbors, a node must first sum the contributions it receives from each upstream neighbors. *For instance, the node "2" gets 3 contributions from BS, node "1", and node "3" in Figure 3(a). Let S be the sum of the contributions* $S = C_{BS,2} + C_{1,2} + C_{3,2}$. *So, "2" distributes* $SxC_{2,i}$ *contribution for each downstream neighbor "i".*

3. When each sensor node receives all its link contributions, then it calculates the link proportions. *For example node "2" has 3 contributions (BS, "1", and "3") so for each upstream neighbors "j" the traffic proportion is* $C_{j,2}/S$.

4. At the end of these four steps, each node in the network has the exact traffic proportions to send to each upstream neighbour.

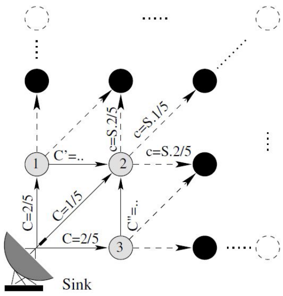

Figure 3. Contribution-Based Heuristic

By implementing this heuristic, the network lifetime would be increased due to the fact that the traffic would be well proportioned, according to the number of neighbors that each node has and the different power levels needed to reach these neighbors.

By comparison with the Equal-Probability Heuristic, where the data are sent to each of the neighbors with equal probability, in the case of Contribution-Based Heuristic, the probability to send to a far neighbor that can be reached with a higher power level is lower than the probability to send to a near one. In this way it is achieved a power consumption reduction and also an energy balance, not only a load distribution between neighbors.

III. SIMULATION RESULTS

A. Simulation Scenario

The simulations that were done in order to compare the various heuristics proposed in the previous part do not take into account any issue regarding the lower level protocols that should be taken care of in the real implementation and deployment of a sensor network. What should our concern be, actually, is how the packets load is balanced between nodes and also the evolution of the lifetime of the network. In the following simulations it is assumed that the medium access protocol is based on a mechanism that allows the nodes to exchange packets without interfering or collisions. Thus, no retransmissions of messages appear in the simulations, nor any packets are lost during the transmissions between the nodes.

Another assumption that is made for the simulations is that every node knows the address of all of its neighbors and that every node has different neighbors according to different transmission power levels. The nodes form a matrix with equal distance (*d*) between lines and columns and the BS is situated in the corner of the grid, in one scenario, or in the center of the grid, in the second scenario.

B. Comparison between Heuristics

For different grid dimensions, although all the three heuristics contribute to the maximization of the network lifetime by way of load balancing between nodes, there are some notable differences between them, which should be mentioned:

- By using the equal-probability strategy, no calculation is done by the nodes, and the traffic is balanced by sending each time to a random neighbor, with an equal probability. A weighted round robin mechanism may also be implemented. The results of the simulation showed that the nodes that are mostly on demand are the nearest neighbors of the BS, and also the nodes that come after these neighbors, on the first line and the last column of the matrix.
- In the case of the shortest-path heuristic, the nodes tend to follow their diagonal when sending towards the BS, or try to reach the main diagonal. This choice is made by every node with equal probability if it has more then one neighbor. The scope of this strategy is to split the load between neighbors, but still keep the shortest and less costly way to reach the BS. The results pointed out that the critical nodes are the ones near the sink and also the ones from the main diagonal, but the biggest pressure is put on the node from the main diagonal, nearest to the BS.
- The contribution-based strategy is clearly superior to the other ones, because using by it, every node will share the load with its neighbors, depending on the number of them and the transmission power needed to reach them. The load is equitably shared between critical nodes and the lifetime of the network is longer than in the case of applying other strategy.

If the sink is situated in the center, the burden created by critical nodes which consume more energy than the others is positioned near the BS, symmetrical to all the axes.

The results are a bit different from the case in which the BS was positioned in the corner of the grid. Now, using the equal-probability heuristic, the pressure falls on the nodes from the base station's line and column. By using the shortest-path strategy, the load is concentrated on the sink's neighbors but mostly on the nodes belonging to the two main diagonals of the grid that intersects in the center, where the BS is located. The contribution-based heuristic is also the best choice, even in this scenario, because it balances the energy consumption between the neighboring nodes of the sink and achieves the maximum lifetime compared to the other two strategies.

In order to clearly present how effective in terms of lifetime maximization each of the strategy is, and also compare the results with the analytical optimum [5], the evolution of the remaining energy of the most critical node, that firstly dies from the whole network, is plotted in Fig. 4. For obtaining an optimal load balance between the nodes that are neighbors of the BS, the node situated on the main diagonal and much close to the sink should send all the messages with a probability equal to 1 directly to the BS and none of the messages should be sent to other neighbors.

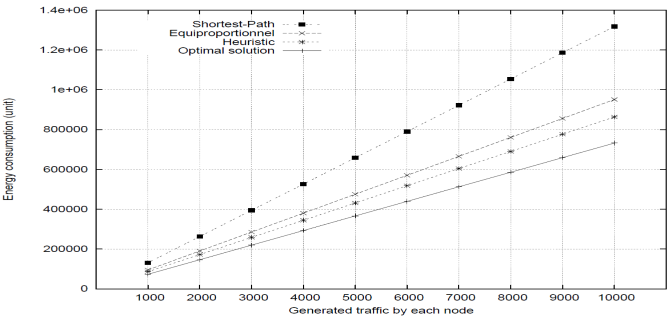

Figure 4. Maximum energy consumption (BS in the corner of a 10x10 grid)

The simulation supposed that each node from a 10x10 grid, initially had a total energy of 30000 units, and the network lifetime is expressed by the number of packets that are sent from the critical node before it dies.

The chart from Fig. 4 shows us that the contribution-based heuristic is the most close to the optimal and it is superior to the other two strategies. The equal-probability strategy overcomes the shortest-path in the case in which BS is in the corner.

In the second scenario, the comparison between the three strategies is plotted in Fig. 5. The figure shows that the best strategy to implement is also the contribution-based heuristic, but in this scenario, the shortest-path maximizes the lifetime almost as much as the contribution-based strategy.

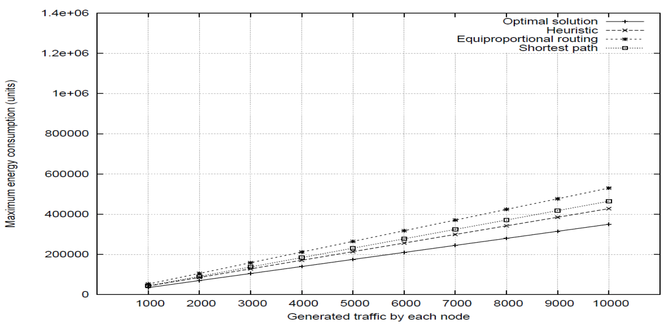

Figure 5. Maximum energy consumption (BS in the center of a 10x10 grid)

C. Simulated Annealing Meta-heuristic Approach

The previous simulations results were obtained under the assumption that the nodes from the grid have only two transmission power levels, and that they are able to send messages only to the nearest neighbors from their line or column with TPL_1 or to the nearest neighbor from their diagonal with TPL_2. For this particular case, it was found the analytical optimum and several heuristics were proposed to maximize the network lifetime by way of minimizing the overall power consumption.

Further on, the previous assumption must be generalized and we try to find the maximum lifetime of the network in the case in which nodes have multiple transmission power levels and can directly send messages to higher distances than *d* or *d* meters. According to previous statements, the power of transmission should be proportional to the square of the distance between nodes. This assumption also remains valid for the following simulations.

The problem can be formulated as follows: Let N be the total number of nodes $\Lambda = (\Lambda_1, \Lambda_2, \ldots, \Lambda_N)$ the vector of output traffic rates of all nodes in the network. The load Λ_i of the node i can be written as follows: $\Lambda_i = \lambda_i + \sum_j \Lambda_j p_{ji}$, with λ_i as the traffic generated by i itself (we assume that each node generates the same volume λ of traffic from its measures) and p_{ji} is the traffic proportion of traffic sent by node j to i. thus we can write: $\Lambda = \lambda \mathbb{1} + \Lambda P$

$\mathbb{1}$ is the identity vector and P is the stochastic matrix of traffic proportions between the nodes.

Let q_{ij} be the transmission power between node *i* and node *j*. To maximize network lifetime we must minimize the energy consumption of the critical nodes. These are those consuming more energy in the network.

Let E_i be the energy consumed by sensor node *i* in the network and $E(P) = (E_1(P), \ldots, E_N(P))$ the corresponding vector.

We assume that energy consumption for one packet receiving is 1 unit.

$$E_i(P) = \lambda_i + \sum_j \lambda_i p_{ij} q_{ij}$$

Then the problem is defined as follows: $E^* = \min_P \| E(P) \|_\infty$

This problem is nonlinear under linear constraints. This problem may be solved when the sensors are placed according to a grid topology and when the maximal user power q_{ij} is equal to 2. The optimal case is obtained when the three neighbours of the BS (the BS is placed in a corner) consume identical energy. The expression of optimal energy consumption is derived [5]: $E^* = \frac{3}{4}(N-4) + \frac{5}{4}$

For the case when the maximal transmission power q_{ij} is greater than 2, and nodes are able to send directly even to the BS, a meta-heuristic based on simulated annealing was used in the will of trying to solve this complex problem of maximizing the network lifetime by minimizing the global energy consumption.

Simulated annealing (SA) is a generic probabilistic meta-heuristic for the global optimization problem of applied mathematics, namely locating a good approximation to the global minimum of a given function in a large search space. It is often used when the search space is discrete (e.g., all tours that visit a given set of cities). For certain problems, simulated annealing may be more effective than exhaustive numeration, provided that the goal is merely to find an acceptably good solution in a fixed amount of time, rather than the best possible solution.

Firstly, the scope of running the previously mentioned algorithm is to discover other network topologies in the case of a maximum TPL_2, other than the one demonstrated by the analytical model that was previously mentioned in [5].

The implementation of the previously presented simulated annealing strategy helped us to obtain the following results, revealed in Tab. 1. In the table, the obtained simulated annealing results are compared with the optimal case. It should be mentioned that the maximum transmission power of nodes is TPL_2, which means that the nodes would only reach at most the neighbors from the same diagonal.

TABLE I. COMPARISON BETWEEN OPTIMAL CASE AND SIMULATED ANNEALING FOR A MAXIMUM TPL EQUAL WITH 2

No. of Nodes	Optimal Case	Simulated Annealing	No. of Nodes	Optimal Case	Simulated Annealing
9	5.00	5.00	**49**	35.00	35.01
16	10.25	10.25	**64**	46.25	46.26
25	17.00	17.00	**81**	59	59.02
36	25.25	25.25	**100**	73.25	73.27

The aim of the simulation was to find the stochastic matrix with sending probabilities that minimize the overall energy consumption of the network. In order to achieve this, we found in the table the maximum of vector T, where T is the energy consumed by the sensor nodes of the network:

It can be seen from Tab. 1 that the load of the most critical node obtained by simulated annealing is equal, for a low number of nodes, with the maximal load obtained in the analytically. As the number of nodes increases, the simulated annealing performance becomes weaker and the values obtained are a little bigger than the ones from the analytical optimum.

But the main idea that has to be underlined here, is that the simulated annealing gives us a different stochastic matrix that the one given by the optimal case. Although the critical nodes send with probability equal to 1 towards the BS, in the rest of the network each node splits its load between all of its neighbors, despite the TPL needed to reach them. This is not the case of the analytical approach, where the sending probabilities are equal to 0 for the diagonal neighbors, except for the critical node from the main diagonal. To be more specific, in the analytical model nodes send only to the TPL_1 neighbors (except from the critical node from the main diagonal), and in the simulated annealing approach all the nodes send to TPL_1 and TPL_2 neighbors.

In this way it was demonstrated that exists more than one way to reach the optimal load balancing in order to achieve the minimum overall energy consumption in a sensor network.

Moreover, in the following we intend to generalize the problem and to consider more than two transmission power levels. In this way, each node could reach more neighbors and send packets towards the BS with fewer intermediate hops. We ran many simulations, considering grids with different dimensions and different base station's positions. The results are shown in tables 2 and 3.

Analyzing the obtained results for numerous simulations, with various transmission power levels and different grid dimensions, we can conclude that the critical load decrease when the transmission power level increases. We indeed can conclude the fact that it is more convenient for the overall power consumption for the nodes to send as far as they can towards the BS to maximize the lifetime of the network.

TABLE II. SIMULATED ANNEALING RESULTS FOR VARIOUS DIFFERENT GRID DIMENSIONS AND DIFFERENT MAXIMUM TPL (BS IN THE CORNER)

Transmission Power Level	Number of Nodes				
	9	**16**	**25**	**36**	**100**
2	5	10.25	17	25.25	73.27
4	3.79	8	13.51	20.31	**60.41**
5	**3.59**	7.82	13.13	20.49	60.86
8	3.60	7.77	13.82	20.53	64.15
18	*	**7.36**	*	*	*
32	*	*	**12.60**	*	*
50	*	*	*	**19.17**	*

TABLE III. SIMULATED ANNEALING RESULTS FOR VARIOUS DIFFERENT GRID DIMENSIONS AND DIFFERENT MAXIMUM TPL (BS IN THE CENTER)

Transmission Power Level	Number of Nodes			
	25	**49**	**81**	**121**
2	6.36	13.76	23.26	35.56
4	**5.42**	**11.81**	**20.44**	**31.95**
5	5.84	12.57	21.97	33.62
8	5.71	12.38	21.37	32.38

As we do not have an analytical optimum for the case in which nodes can communicate with transmission power more than 2, but for maximum TPL equal to 2 the results given by the simulated annealing are close to the analytical results, we can assume that for a reasonable number of nodes (limited number of sensors), the results can be acceptable.

Due to lack of space, we did not introduce in this paper the stochastic matrix with the probabilities generated by the simulations, but they can be used by the BS to send them to every node from the network.

IV. PROTOCOL DESIGN AND IMPLEMENTATION

In this section, a description of the design and implementation of the previous discussed strategies to maximize the lifetime of a WSN is presented. In the first two subsections, a description of the hardware and software technologies is made. Eventually the protocol steps and its correct behavior are described.

A. Hardware and Software Technologies

In our implementation we used the Tmote Sky platform. Using TinyOS as an embedded operating system expressly designed for WSNs. The benchmark is the IEEE 802.15.4 radio standard, with low data rate (around 250 kbps). A node is equipped with a microcontroller (8-16 bit) and low storage memory.

B. Protocol Description

In the simulations previously presented it was assumed that every node from the wireless sensor network is aware of its own position in the grid and also its neighbors' positions. In a real world scenario, when they are firstly deployed, sensor nodes have no idea about there location and they do not know their neighbors or the power needed to reach them. The Tmote node does not have a GPS receiver or any other additional device that could inform the node about its current location.

In order to create the network topology and to construct a correct communication protocol between the sensor nodes, the proposed implementation should have an initial phase, known as a neighbor discovery process. After making each node aware of its neighbors towards the BS, and the power that it needs to reach them, the proposed load-balancing techniques could be implemented.

We propose a new self-organization protocol to extend the system life by solving the drawbacks of the conventional protocols and reducing the power consumption of all sensors in the network system. Our protocol makes use of some command messages to create the network architecture and to safely send the reading data to the BS. As all sensors send data destined for the sink by dissipating uniformly the power of all sensors, the lifetime of sensor network is extended.

For analyzing and evaluating the fundamental performance of our protocol, we first describe the power conserving behavior of the protocol. In wireless sensor networks, all sensing nodes have the maximum transmission range, and they send data to the sink directly or through intermediate nodes in this range. In our protocol, we propose to use the Tmote node capability to have multiple transmission power levels and maximize the lifetime of the network by varying this power and share the load between different neighbors. The sensor nodes can send data to the sink through intermediate nodes closest to the sink or directly to the BS, by applying one heuristic or another.

Our implementation is based on a set of command messages that originate at the BS and are flooded in the network so that any single node can receive the command, interpret it and act consequently.

There are eight command messages involved by the network: `(DISCOVERY_TPL1, DISCOVERY_TPL2, GATHER_TABLES, SEND_TO_BASE, READING, EQUAL_PROBABILITY, CONTRIBUTION_BASED)`.

We first define notations used in our protocol before describing the detail operation of our protocol. The BS and all the nodes have a depth level. The depth level of the BS is initially set to zero and that of all the other nodes is set by an infinite value. To distinguish a node with others, each node has a unique identification (ID).

In the following, it is shown a detailed presentation of how the network is created and how it works after the neighbor discovery process is completed.

Initially (STEP 1), the BS broadcasts with transmission power level set at the minimum (TPL = 1) the command message `DISCOVERY_TPL1`. This message has the following fields: `msg.command=DISCOVERY_TPL1; msg.source=BS.address(ID)` and `msg.depth=0`.

Every node in the network has the depth initially set at 100 (infinite), is in idle state and always listen the media for receiving packets. When a packet arrives, the node processes the command.

If msg.depth<node.depth, the node saves the sender address in the neighboring table with nodes close to the BS than itself (if not already in there), with the power needed to reach it equal with TPL_1, modify its own depth by increasing the received msg.depth with 1 and forwards the message by broadcast, but only after it updated the msg.source with its own source and msg.depth with its own new depth. The node will broadcast the received and updated packet only once, in order to minimize the number of packets that are transmitted in the network. That packet will be sent with TPL_1 after a random period to avoid the collisions.

In the case in which msg.depth>node.depth, the node saves the sender address, but this time in the neighboring table with nodes far from the BS than itself (if not already in there), with the power needed to reach it equal with TPL_1. Every node should also be aware of the nodes far from the BS because when the contribution based heuristic is chosen, all the nodes should distribute the contributions received from the BS to all of the far neighbors. In this last case, the node will not broadcast the received packet, just save the far neighbors' addresses.

After STEP 1, every node has the appropriate depth in the network, and the corresponding TPL_1 neighbors saved in the neighboring tables (far from and close to the BS).

In STEP 2 of the protocol, the BS broadcasts with transmission power level set at the maximum (TPL = 31) the command message `DISCOVERY_TPL2`. This message has the following fields:

`msg.command=DISCOVERY_TPL2; msg.source=BS.address (ID)` and `msg.depth=0`.

Every node in the network has the depth set at the proper value from STEP 1, is in idle state and always listen the media for receiving packets. When a packet arrives, the node processes the command.

If msg.depth<node.depth, the node saves the sender address in the neighboring table with nodes close to the BS than itself (if not already in there), with the power needed to reach it equal with TPL_2 and forwards the message by broadcast, after updating the msg.source with its own source and msg.depth with its own depth. At this step, the depth level of the nodes

does not suffer any modifications. Just the depth field of the message is set with the node's existing value (msg.depth = node.depth).

The node will broadcast the received and updated packet only once, in order to minimize the number of packets that are transmitted in the network. That packet will be sent with TPL_2 at this step, also after a random period to avoid the collisions. In the case in which msg.depth>node.depth, the node saves the sender address, but this time in the neighboring table with nodes far from the BS than itself (if not already in there), with the power needed to reach it equal with TPL_2. Every node should also be aware of the nodes far from the BS, because when the contribution-based heuristic is chosen all the nodes should distribute the contributions received from the BS to all of the far neighbors. In this last case, the node will not broadcast the received packet, just save the far neighbors' addresses.

After STEP 2, every node will have the appropriate neighbor tables constructed, with neighbors far from and near to the BS.

By this time our protocol should have completed the neighbor discovery phase, and now every node knows its neighbors towards the BS or far from it. Afterwards, each node should send its previously constructed tables to the BS, in order to have a complete view over the network topology. We remind here that the BS is usually connected to a computer with high computing capabilities and no power consumption constraints.

In STEP 3, the BS broadcasts with transmission power level changed again at minimum (TPL = 1) the command message GATHER_TABLES. This message has the following fields: msg.command=GATHER_TABLES; msg.source = BS.address(ID); and msg.depth=0.

Every node in the network has the depth set at the proper value from STEP 1, is in idle state and always listen the media for receiving packets. When a packet arrives, the node processes the command.

If msg.depth<node.depth, the node forwards the message by broadcast, after updating the msg.source with its own source and msg.depth with its own depth. The node will broadcast the received and updated packet only once, in order to minimize the number of packets that are transmitted in the network. That packet will be sent with TPL_1 after a random period to avoid the collisions.

After completing this task, when receiving this type of command a node should also do another task. It has to construct a new message with its two neighboring tables and send it by unicast to one of his TPL_1 neighbors (if it has one) or to one TPL_2 neighbors (in the opposite case). However, this randomly chosen neighbor must be in the neighboring table towards the BS (must have a lower depth level than its own).

This new command message has the following fields:

```
msg.command=SEND_TO_BASE;
msg.source=node.address(ID);
msg.depth=node.depth;
msg.neighbors=node.neighbors.
```

In order this message to arrive to the BS it should be forwarded by all the intermediate nodes towards the sink. So, again every node will be in idle state and will listen to the media for receiving packets. When a packet arrives, the node processes the command and if the command is "SEND_TO_BASE" and msg.depth>node.depth, the node forwards the message by unicast to one of his TPL_1 neighbors (if it has one) or to one TPL_2 neighbors (in the opposite case). However, this randomly chosen neighbor must be in the neighboring table towards the BS (must have a lower depth level than its own). In this last case of forwarding the message towards the BS, no modifications are made to the packet, not even to the source or to the depth fields.

By this time the protocol should have completed the neighbor tables gathering phase, and by now the BS has created the whole network topology, having the complete information about all the nodes and all of their neighbors, reached with different transmission power levels. In our case the neighbor tables gathering phase is initiated by the BS, but it can also be easily implemented if this phase starts from the farthest node. We can identify the farthest node as being the one that does not receive any messages in a fixed amount of time from a node with the depth level greater than its own.

As the network complete topology is known by the BS, it can propose a strategy to maximize the lifetime of the network by way of minimizing the overall power consumption. It can directly send the stochastic matrix and flood it throughout the network, or individually unicast the sending probabilities to each node. This choice can be made by the user of the sensor application or by the developer.

In our particular case, we chose to implement the equal-probability heuristic and the contribution-based heuristic. In the following, the BS will just dictate the strategy and will broadcast the command message throughout the network.

Knowing the network topology, for the BS to calculate and then transmit the optimal sending probabilities to all the nodes is a very easy task to accomplish. At this step, the probabilities that will minimize the power consumption and implicitly maximize the network lifetime could be calculated by the BS, or by the computer wired through the USB to the BS. It can also be used the Simulated Annealing meta-heuristic proposed in Section III.C or the optimal solution (from [5]).

C. *Protocol Implementation and Specific Tools Used*

The communication between the Tmote node and the PC was established over the COM port, using a modified application from the TinyOS environment called TOSBase. This application was modified to run on the BS mote, to echo UART messages to the radio and radio messages to the UART and maintain UART and radio message buffers.

The TOSBase application was modified in order to save the addresses of the neighboring motes, to compute the contributions needed for the contribution-based heuristic and also to transmit command messages with different transmission power levels.

The Java application net.tinyos.sf.SerialForwarder is a Java-based applet that creates a TCP socket to allow data to be shared with other PC applications. It takes the incoming messages from the UART, and forwards it over an internet connection so that PC based applications can use the data. SerialForwarder also works in another direction. It forwards the request packet to the UART, and the attached mote forwards the message over the radio.

Another Java application used was the net.tinyos.tools.BcastInject, which allows a user to interact with the connected mote in a simple way, by injecting commands messages into the network. This application was modified in order to allow us to transmit the command messages that we needed for the protocol implementation.

The data received from the network, including the reading data from the sensors and the neighboring tables were captured on the screen using another Java application called the net.tinyos.tools.Listen. Modifying this application allowed us to parse the incoming messages and print on the screen the information useful to our protocol design.

Fig. 6 shows the communication scheme between the BS node and the PC. The sensor network that is used for our protocol has two types of nodes: one BS node, which is connected to a PC as previous shown and many remote nodes. The BS node is responsible for transferring between USB message and radio message and initiating Command Messages to allow the remote nodes to initiate the neighbor discovery phase and dictate the load balancing strategy for the whole network. It provides the interface between the PC and the display software.

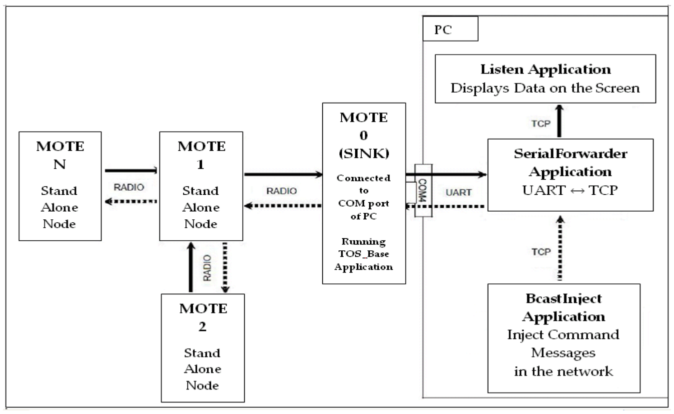

Figure 6. Communication Scheme

The functions of the remote motes include the listening of the media and processing the commands received from the BS. They should also transmit data towards the BS, relay the data packets from other remote nodes towards the sink or backwards, to the farthest nodes (the command messages). They also calculate the contributions in the case of the contribution-based heuristic maintain the mote ID and also compute the node depth level according to our protocol neighbors' discovery phase.

V. CONCLUSION

We focused the attention on lifetime issues of WSN and we designed, analyzed and compared several load-balancing methods for a regular grid topology (Equal-Probability Heuristic, Shortest-Path Heuristic and Contribution-Based Heuristic), using multiple transmission power levels.

We firstly realized a comparison, by simulation, between the proposed load-balancing heuristics. For different grid dimensions, with the BS situated in the upper-right corner, although all the three heuristics contribute to the maximization of the network lifetime by way of load balancing between nodes. The best solution depends on the considered scenario. The simulated annealing algorithm has consequently been implemented in order to study larger scenarios for which exact solutions can not be derived. It gives us a different stochastic matrix that the one given by the optimal case in small configurations. In this way it was demonstrated that there are more than one way to achieve the optimal load balancing in order to achieve the minimum overall energy consumption in a sensor network using *two transmission power levels*.

Analyzing the obtained results for numerous simulations, *with more than two transmission power levels* and different grid dimensions, we can conclude that the critical load decrease visible when the transmission power level increases. We indeed can conclude the fact that it is more convenient for the overall power consumption for the nodes to send as far as they can towards the BS to maximize the lifetime of the network.

Concerning the ***implementation***, we proposed a new self-organization protocol to extend the system life by solving the drawbacks of the conventional protocols and reducing the power consumption of all sensors in the network system. Our protocol makes use of some command messages to create the network architecture and to safely send the reading data to the BS. As all sensors send data destined for the sink by dissipating uniformly the power of all sensors, the lifetime of sensor network is extended. Our implementation is based on a set of command messages that originate at the BS and are flooded in the network so that any single node can receive the command, interpret it and act consequently.

Knowing the network topology, for the BS to calculate and then transmit the optimal sending probabilities to all the nodes is a very easy task to accomplish. The probabilities that will minimize the power consumption and implicitly maximize the network lifetime could be calculated by the BS, or by the computer wired through the USB to the BS.

REFERENCES

[1] J. Champ, C. Saad, A. E. Baert, "*Lifetime in Wireless Sensor Networks"*, International Conference on Complex, Intelligent and Software Intensive Systems, 2009

[2] Y. Chen, Q. Zhao, *"On the Lifetime of Wireless Sensor Networks"*, Communications Letters, IEEE, 2005

[3] Tmote-Sky Data-sheet, *http://www.snm.ethz.ch/pub/uploads/Projects/tmote_sky_datasheet.pdf*

[4] R. Kacimi, R. Dhaou, A.-L. Beylot, "Energy-aware self-organization algorithms for wireless sensor networks". In Proceedings IEEE GLOBECOM 2008, december 2008;

[5] R. Kacimi, R. Dhaou, A.L. Beylot, "Energy-balancing Strategies for Lifetime Maximizing in Wireless Sensor Networks, in Proceedings IEEE ICC 2010, may 2010.