An Optimal Wake-Up Scheduling Algorithm for Minimizing Energy Consumption while Limiting Maximum Delay in a Mesh Sensor Network

Reuven Cohen and Boris Kapchits
Department of Computer Science
Technion
Haifa Israel

Abstract— **This paper presents an algorithm for maximizing the lifetime of a sensor network while guaranteeing an upper bound on the end-to-end delay. We prove that the proposed algorithm is optimal, and that it requires simple computing operations that can be implemented by simple devices. To the best of our knowledge, this is the first paper to propose a sensor wake-up frequency that depends on the sensor's location in the routing paths. Using simulations, we show that the proposed algorithm significantly increases the lifetime of the network, while guaranteeing a maximum on the end-to-end delay.**

I. Introduction

A sensor network may contain a huge number of simple sensors that are densely deployed at some inspected site. In large areas, the sensor network is likely to have a mesh structure. In this case, the sensors also act as routers, forwarding packets from one of their neighbors to another.

The information gathered by the sensors should be delivered to a centralized node, usually referred to as a *gateway.* The gateway is assumed to have a much higher processing capability than the sensors and, in some cases, it has connectivity to a remote network as well. Sensors turn their communication hardware on and off to minimize energy consumption. Therefore, in order for two neighboring sensors to communicate, both must be in active mode. Two possible synchronization models can be implemented to this end: global synchronization [1] and local synchronization. With global synchronization, all sensors must wake up at the same time. In that case, a packet can be delivered from the source to the destination very rapidly, even if the two nodes are not within each other's transmission range. However, in large mesh networks, global synchronization is not only very difficult to achieve but also very inefficient [2]–[4].

In this paper we consider local synchronization, where each node need only select its active duty cycle and inform neighboring nodes about its selection. A node that needs to send a packet through a neighbor, must wake up and transmit it during the neighbor's duty cycle. This communication model imposes a clear trade-off between the delay encountered by a packet routed along the sensor network and the time during which the sensors along the route are in active mode. Solutions for addressing this trade-off depend, to a large extent, on the specific sensor network model [5], and in particular on the following aspects:

1) The data delivery model, i.e., whether data is delivered continuously by the sensor, or delivered only after an event of interest.
2) The expected amount of data to be delivered.
3) The routing scheme: whether, for example, a single route is used between each source and the gateway or multiple routes are concurrently employed, or whether routes are selected by the traversed nodes or by the sources ("source routing").
4) Whether the intermediate nodes process the packets they receive in order to merge similar observations from different sources or forward them as is.

With respect to (1) and (2), if the data is continuously delivered, or if there is a lot of it, an algorithm for determining the wake-up times of each intermediate sensor has to take into account the expected amount of traffic passing through the sensor and the expected receiving times. With respect to (3), if the routing algorithm is flexible enough, it can determine the route to be traversed by each packet according to the expected wake-up times of the intermediate nodes. Finally, with respect to (4), if an intermediate node can process the received data packets, it may choose to delay them until they can be merged with other packets.

In this paper we consider a sensor network for alarm events that are generated by the network's nodes and relayed to the network's gateway. For example, consider sensor networks for detecting smoke, or locally measured temperature that exceeds some threshold, or water level that exceeds some threshold. The most important property of such networks, with respect to this paper, is the existence of an upper bound on the time from when an alarm event is generated to the time the gateway must be informed about it. We present an algorithm for maximizing the lifetime of such networks while guaranteeing an upper bound on the end-to-end delay. We prove that the proposed algorithm is optimal, and that it requires simple computing operations that can be implemented by simple devices.

With respect to routing, we assume that a single route is used from each sensor to the gateway. This implies that the routes from the nodes to the gateway define a directed tree, rooted at the sensor network's gateway. This is a conventional approach in general data networks and sensor networks in particular, where selecting different routes for different flows originated at the same sensor is impractical. We make no assumptions regarding how the routes are selected. In particular, the schemes proposed in [6], [7] are all applicable. Moreover, in our simulation model (Section VI), one of the routing algorithms we consider routes only via nodes with sufficient energy. Following this assumption, each sensor node has a designated neighbor (parent) to which it forwards packets destined for the gateway. These packets are created either by the node itself or by some other, downstream, node. At any given time, the set of wireless links between every node and its preferred neighbor forms a virtual spanning tree. We make no assumption regarding this collection of links, and of course it may change dynamically, e.g., due to topological changes in the network.

The rest of this paper is organized as follows. In Section II we describe and justify the power consumption model considered throughout the paper. In Section III we present and discuss related work. Since the proposed algorithm is complex, we present it in two steps. In Section IV, we present the basic algorithm, which determines the wake-up time for every sensor such that the *total* energy consumed by the network is minimized and the end-to-end delay is upper-bounded. However, such an algorithm does not necessarily extend the lifetime of the network because the variance of the assigned frequencies can be too high. This basic algorithm is therefore used in Section V as a procedure for the ultimate algorithm, which minimizes the total energy consumed by the nodes while ensuring an upper bound on both the maximum delay *and* on the energy consumed by every individual sensor. In Section VI we present simulation results for the new algorithm, and show that the proposed algorithm can almost double the lifetime of a sensor network. Finally, Section VII concludes the paper.

II. Power Consumption Model

In order to minimize power consumption, nodes remain in a sleep mode most of the time, while adhering to the following two simple rules:

(R1) An inner node v in the virtual routing tree, i.e., a node that serves as a parent to at least one of its neighbors, *must* wake up periodically, in order to receive packets from its children. These children know the times when v switches from sleep to active. The node stays active as long as it receives packets from its neighbors. After a time-out period of not receiving any packet, the node returns to sleep mode.

(R2) Every node also wakes up when its parent wakes up if and only if it needs to forward a packet through the parent to the gateway.

Precise synchronization between neighboring nodes is unnecessary. If the nodes use a CSMA/CA-like MAC protocol in order to send data packets, a packet that is not ACKed due to non-perfect synchronization will be re-transmitted after a short time-out period. This energy expenditure model is formalized as follows. If a node v wakes up f_v times per second, its energy consumption is equal to $f_v \cdot c_v + b_v$ Watt, where

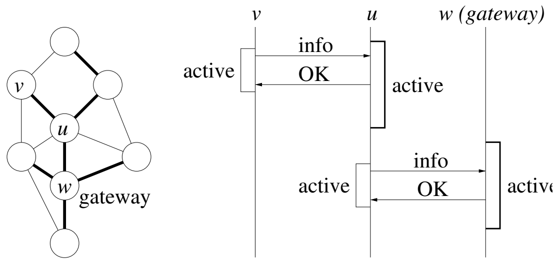
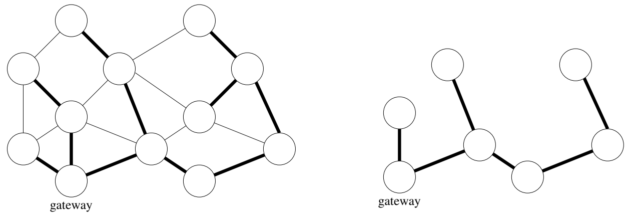


Fig. 1. A sensor network (the routing paths to the gateway are marked by thick lines) and a possible packet exchange for delivering data from v to w

- c_v is the average amount of energy (Watt) consumed during every wake-up period, including charging the registers, listening, receiving packets and transmitting ACKs.
- b_v is a constant amount of energy, expended on monitoring the environment, performing internal calculations, managing the clock, and forwarding packets to the next node along the route to the gateway; we say that b_v is constant for node v because it does *not* depend on the wake-up frequency f_v of v.

The justification for our energy consumption model comes from the major difference between the energy consumed by the sensor radio for transmitting/receiving/listening (i.e., active mode) and for sleeping (i.e., idle mode). For instance, the CC2420 device of Chipcon [8]–[10], which is 802.15.4-compatible, requires 52.2mW for transmitting, 56.4mW for receiving or listening, and only $3\mu W$ for sleeping. A typical duty cycle of such a sensor is 0.1%. For instance, this is the duty cycle of a sensor that wakes up once every 10 seconds (f_v=0.1) for 10 millisecond. The difference between the energy required for receiving/listening (56.4 mW) and transmitting (52.2 mW) is negligible compared to the difference between receiving/listening/transmitting (52-56 mW) and sleeping ($3\mu W$). Hence, while the actual energy consumed by the sensor is $0.001 \cdot 56 + 0.999 \cdot 0.003 + b_v$ mW, it can be very well approximated by $0.001 \cdot 56 + 0.003 + b_v$ mW. Thus, by incorporating the fixed part of the energy spent in the idle mode (0.003 mW) into b_v, we get that the average consumed energy is indeed $f_v \cdot c_v + b_v$ mW, where b_v does *not* depend on the wake-up frequency f_v of v.

Figure 1 depicts the considered communication model for the case where sensor v sends a packet to the gateway w via sensor u. Node v wakes up during the duty cycle of node u, and sends the packet to this node. The energy it expends for transmitting the packet to u is attributed to its b_v factor, whereas the energy expended by u is attributed to the $f_u \cdot c_u$ factor. Node u wakes up during the duty cycle of w and forwards the packet to w. The energy it expends for transmitting the packet is attributed to its b_u factor. Therefore,



(a) A network with its routing tree (b) The corresponding T−tree

Fig. 2. A network with the corresponding T-tree, which is subject to our optimizations

the end-to-end delay encountered by this packet is the time node v needs to wait until u wakes up, plus the time node u needs to wait until node w wakes up. This time is upper bounded by $1/f_u + 1/f_w$.

Consider the virtual routing tree from the nodes (sensors) to the gateway. Following R1 above, only an inner node u in this tree is assigned a constant wake-up frequency of f_u. A leaf v of this tree wakes up only when it needs to send a packet towards the gateway via its parent. Hence, for such a node, $f_v = 0$ holds. The energy expended by such a node is attributed to its b_u component only. Since we focus on determining the wake-up frequencies of the inner nodes, we shall ignore the leaves of the routing tree. Formally, Let $G = (V, E)$ be a network graph, where G is the set of sensors and the gateway, and E is the set of edges between nodes that can communicate directly. Let T be the routing tree that consists of all the routing paths from the sensors to the gateway without the leaves and their edges towards their parents. For the rest of the paper we shall address the allocation of wake-up frequencies to the nodes of T, assuming that for the other nodes the wake-up frequency is 0. Figure 2(a) gives an example for a network with its routing tree. Figure 2(b) shows the corresponding T-tree, which is subject to our optimizations. for a network with its routing tree.

Following our model, we conclude that energy depletion is *not* governed by the amount of transmitted data, but by the *time the sensors spend in active mode.* Therefore, the paper deals with the trade-off between this frequency and the maximum end-to-end delay in the network. Of course, a sensor node also uses its energy for other tasks, such as computation and sensing. However, it is well known that this energy is marginal compared to that spent in active mode [11].

III. Related Work

While many papers have been written on how to minimize energy consumption in sensor networks, very few have explicitly addressed the tradeoff between delay and energy. To the best of our knowledge, our work is the first to propose the assignment of different wake-up frequencies to nodes according to their role in the packet forwarding process. However, the energy-latency tradeoff has been thoroughly studied in sensor networks, as well as in other wireless networks. In this section we present related works, and compare their models and results with ours.

In the S-MAC protocol [12], packet latency caused by periodic sleeping of intermediate nodes is minimized by synchronizing the wake-up schedules of neighboring nodes. The duty cycles of all nodes are equal and predefined. The protocol is not intended to guarantee an upper bound on the end-to-end delay, but to minimize the energy consumption of the nodes.

Another paper [13] proposes to minimize the delay using special scheduling of the nodes' wake-up periods. This paper extends another work [14] by the same authors, where the nodes are organized in an unidirectional tree. However, in [13] the authors assumption of arbitrary communication patterns renders the problem NP-Complete. The authors propose algorithms that find an optimal solution for specific topologies, such as trees and rings. They also show that their algorithms can be used as heuristics for general graphs.

In [15], the authors address the trade-off between delay and energy in sensor networks from a different viewpoint. They search for an optimal routing path from a source node to the gateway, such that latency is minimized and energy cost is not "too big." In their network model, sensors randomly switch between sleep and active states. Two alternatives are studied: a centralized global optimization approach and a distributed approach.

Energy efficiency can be achieved in different ways. For example, energy aware routing finds a routing path while taking into account energy cost and the sensor's available energy. In [16] the number of hops along the forwarding path is considered, while keeping in mind that transmission between close nodes is more energy efficient, even if the resulting route is longer. The authors use a random network model to show the energy-latency-throughput dependency and to find the optimal transmission power for nodes in an ad-hoc network. As already indicated, we do not address the routing issues in our work. The scheme proposed in [16], as well as many others, can be used for this purpose.

In [17], the tradeoff between energy and latency is investigated using probabilistic computation. The authors consider a network of nodes that switch from passive to active mode independently, but with a predefined frequency. The packets are not forwarded on predefined routes, as they are in our model, but are sent instead to all neighbors in active mode. Therefore, the network density and the duty cycle should be high enough to ensure that each packet will finally reach its destination. A probabilistic analysis finds the portion of time each node is required to be in an active state in order to ensure that the packet is delivered to the gateway on time. This model differs from ours in that our model does not use flooding and assumes that nodes are aware of their neighbor's duty cycles.

The tradeoff between energy and latency in general wireless networks was also studied in a different context. For example, [18] and [19] investigate this tradeoff when a technique called "modulation scaling" is used. The authors base their work on the observation that, in many coding schemes, the transmission of a packet requires a smaller amount of energy if it lasts longer. They solve the problem of finding an optimal transmission schedule for a node, given that it has to forward a random number of packets whose arrival times follow the Poisson distribution. The optimization criterion is to minimize the overall energy consumption and bound the maximum delay. Two algorithms are proposed: an off-line algorithm that finds an optimal solution, and an on-line algorithm that approximates the optimal solution. This approach is taken further in [20]. The authors deal with more general setting, assuming that each packet may have a different deadline and number of bits.

In [21], this problem is generalized by considering an aggregation tree with packets routed along the tree to the root. As in [18], the energy cost of a packet transmission is a decreasing convex function of its transmission time. The cost is different for each node because of the different amounts of data to be forwarded. The packet should be delivered to the sink within a limited time period. The authors propose an off-line algorithm for an optimal solution whose running time complexity is unknown, and an approximation algorithm with pseudo-polynomial running time that needs to know the network topology. Although our model is different, the considered problem is similar to ours. In our work, by making some assumption on the energy-latency dependency, we propose an optimal algorithm with linear complexity.

The tradeoff between energy and latency appears also when data-aggregation techniques are applied. On the one hand, it is better to merge several packets reporting the same event into a single one. Doing so, on the other hand, increases the delay.

IV. Step 1: Minimizing the total consumed energy

Recall our energy expenditure model from Section I, where node v wakes up f_v times per second, and its energy consumption is $f_v \cdot c_v + b_v$. Since b_v does not depend on the wake-up frequency of v, it is *not* subject to optimization. Therefore, for the rest of the paper we consider only the first factor.

Definition 1: An optimal assignment of wake-up frequencies to nodes is an assignment that guarantees an upper bound D on the maximum delay while minimizing the total energy spent by the nodes in active mode. ■

We now formalize this definition as an optimization problem:

Problem 1: Let T be a tree that consists of all the paths from the sensors to the gateway without the leaves. Let f_v and c_v be the wake-up frequency and the average energy consumption during each wake-up for node v, respectively. Then, we want to

minimize: $\sum_{v \in T} f_v \cdot c_v$

subject to:

- for every directed path $v_1 \to \cdots \to v_N$ in T, where v_N is the gateway, $\sum_{v \in \{v_1 \cdots v_N\}} \frac{1}{f_v} \leq D$.
- for every node v in T: $f_v > 0$. ■

Note that we can ignore every non-maximal path, which is a sub-path of a longer path, and consider only the paths from the leaves of T to the gateway.

Although the target function is linear and the problem is convex, the constraints are nonlinear. However, we will make a few observations that allow us to find the optimal solution in linear time. We start by showing that under any optimal solution, the constraint for each maximal directed path is tight.

Claim 1:

(a) In an optimal assignment of wake-up frequencies to nodes, the delays on the paths from the leaves of T to its root are equal.

(b) In an optimal assignment, the delay on each path from the leaves to the root is equal to the maximum tolerated delay.

Proof: To prove (a), assume, by contradiction, that we have an optimal wake-up frequency assignment where there are two leaves, v_1 and v_2, such that the maximum delay on path P_1 from v_1 to the root is smaller than the maximum delay on path P_2 from v_2 to the root. By reducing the wake-up frequency of v_1, we can equate the delay on P_1 to the delay on P_2. The new assignment reduces the total energy consumption without affecting the maximum delay, contradicting our assumption. The proof of (b) is similar. ■

From Claim 1 follows that an optimal assignment must be unique. To see why, suppose there exist two optimal assignments O_1 and O_2. Among all the nodes whose assigned energy is different in O_1 and O_2, consider the one whose hop distance from the gateway is maximum. If there are two or more such nodes, choose one of them randomly. Let this node be v, and suppose that the energy assigned to v in O_1 is smaller than in O_2. However, by reducing the energy assigned to v in O_2 to the same value as in O_1 we can reduce the total energy of O_1 while still ensuring an end-to-end maximum delay of D.

Suppose that by assigning the values $a_1, a_2, \ldots, a_n$ to the variables $f_1,\ f_2, \ldots, f_n$ of Problem 1, we guarantee an upper bound D' on the delay. Then, by assigning the values $\alpha a_1, \alpha a_2, \ldots, \alpha a_n$, we guarantee an upper bound of $\frac{D'}{\alpha}$. Therefore, the problem can be solved in the following way. First, find the minimum delay for which $\sum f_v \cdot c_v = E$ holds. Let this delay be D'. Then, multiply the values assigned to $f_1,\ f_2, \ldots, f_n$ by α, where D'/α is equal to the target maximum delay D.

To solve the former problem, let $T(v)$ denote the sub-tree rooted at v, $D(v)$ denote the maximum delay on this sub-tree, and $E(v)$ denote the amount of energy assigned to $T(v)$. In addition, let $e(v)$ denote the amount of energy assigned to node v itself. Since c_v denotes the average energy cost of each wake-up of node v, $\frac{e(v)}{c_v}$ is the wake-up frequency of v.

We now present an algorithm that, when given a certain amount of energy E and a node v, divides this energy among the nodes in $T(v)$ such that the maximum delay encountered on this sub-tree, $D(v)$, is minimized. The algorithm starts at the leaves and propagates upstream towards the root in the following way. For a leaf v, the value of $D(v)$ as a function of $E(v)$ is simply $D(v) = c_v/E(v)$. Now, consider a node v whose children in the tree are $u_1, u_2 ... u_N$. Suppose that for every u_i we know the value of $D(u_i)$ as a function of $E(u_i)$.

From Claim 1(a) we know that in an optimal assignment all the paths from the leaves of $T(v)$ to v should have an equal delay. We therefore have the following set of N equations:

$$D(u_1) = D(u_2) = \ldots = D(u_N) \qquad (1)$$

$$e(v) + \textstyle\sum_{i=1}^{N} E(u_i) = E(v). \qquad (2)$$

The conditions expressed by these equations are necessary, but not sufficient, since we also have to minimize $D(v)$. This delay is equal to the delay introduced by v plus $D(u_i)$, for any $1 \leq i \leq N$. Therefore, and without loss of generality, we have $D(v) = \frac{c_v}{e(v)} + D(u_1)$. To find the value of e_v that minimizes $D(v)$ for a given value of $D(u_1)$, we differentiate $D(v)$ with respect to $e(v)$. This yields the following additional equation:

$$\frac{d(\frac{c_v}{e(v)} + D(u_1))}{d(e(v))} = 0. \qquad (3)$$

Since the value of $D(u_i)$ for every $2 \leq i \leq N$ is an explicit function of $E(u_i)$, we get $N+1$ equations with $N+1$ variables: $e(v), E(u_1), E(u_2), \ldots, E(u_N)$. This set of equations is solved in order to divide $E(u)$ among $u_1 \ldots u_N$, while guaranteeing the minimum delay.

When the algorithm finishes, and the equations are solved for each node, including the root r, we know the value of $D(r)$ as a function of $E(r)$. In order to solve Problem 1, we set $D(r)$ to be the maximum tolerated delay and find the minimum overall energy $E(r)$ that guarantees this maximum.

Claim 2: The relationship between $D(v)$ and $E(v)$ is of the following form:

$$D(v) = \frac{1}{B_v E(v)}. \qquad (4)$$

B_v is a constant that depends only on c_v and on $c_{u_1} \cdots c_{u_N}$, where $u_1 \cdots u_N$ are the children of v in T.

Proof: We prove this by induction on the height of the sub-tree of v. If the height is 1, i.e., v is a leaf, $D(v) = \frac{c_v}{E(v)}$. Consider now an inner node with N children, $u_1, u_2, \ldots, u_N$. By the induction assumption, the delay function for each child of v is $D(u_i) = \frac{1}{B_{u_i} E(u_i)}$. Therefore, the system of equations to be solved in order to determine how to divide $E(v)$ between v and $u_1 \ldots u_N$ is:

$$\frac{1}{B_{u_1} E(u_1)} = \frac{1}{B_{u_i} E(u_i)}, \text{for every i,} 2 \leq i \leq N, \qquad (5)$$

$$e(v) + \textstyle\sum_{i=1}^{N} E(u_i) = E(v), \qquad (6)$$

$$\frac{d(\frac{c_v}{e(v)} + \frac{1}{B_{u_1} E(u_1)})}{d(e(v))} = 0. \qquad (7)$$

Substituting Eq. 5 into Eq. 6 yields

$$e(v) + \sum_{i=1}^{N} \frac{B_{u_1} E(u_1)}{B_{u_i}} = E(v).$$

Denoting $B = \sum_{i=1}^{N} \frac{B_{u_1}}{B_{u_i}}$, we have

$$E(u_1) = \frac{E(v) - e(v)}{B}. \qquad (8)$$

Substituting Eq. 8 into Eq. 7 yields

$$\frac{d(\frac{c_v}{e(v)} + \frac{B}{(E(v)-e(v))B_{u_1}})}{d(e(v))} = 0 \qquad (9)$$

or equivalently

$$-\frac{c_v}{e(v)^2} + \frac{B}{(E(v) - e(v))^2 \cdot B_{u_1}} = 0. \qquad (10)$$

Therefore,

$$B \cdot e(v)^2 - (E(v) - e(v))^2 \cdot B_{u_1} c_v = 0,$$

and

$$(B - B_{u_1} c_v) e(v)^2 + \\ +2 B_{u_1} c_v E(v) e(v) - B_{u_1} c_v E(v)^2 = 0.$$

In those cases where $B = B_{u_1} c_v$, we get

$$e(v) = \frac{B_{u_1} c_v E(v)^2}{2 B_{u_1} c_v E(v)} = \frac{1}{2} E(v). \qquad (11)$$

Otherwise,

$$e(v) = E(v) \frac{-B_{u_1} c_v \pm \sqrt{(B_{u_1}^2 c_v^2 + (B - B_{u_1} c_v) B_{u_1} c_v}}{B - B_{u_1} c_v},$$

$$\text{thus,} \quad e(v) = E(v) \frac{-B_{u_1} c_v \pm \sqrt{B \cdot B_{u_1} c_v}}{B - B_{u_1} c_v}.$$

Substituting the definition of B,

$$e(v) = E(v) \frac{-B_{u_1} c_v \pm \sqrt{\sum_{i=1}^{N} \frac{B_{u_1}}{B_{u_i}} B_{u_1} c_v}}{\sum_{i=1}^{N} \frac{B_{u_1}}{B_{u_i}} - B_{u_1} c_v},$$

$$e(v) = E(v) \frac{-c_v \pm \sqrt{\sum_{i=1}^{N} \frac{c_v}{B_{u_i}}}}{\sum_{i=1}^{N} \frac{1}{B_{u_i}} - c_v}.$$

Since $e(v)$ must be positive and smaller than $E(v)$, the final solution will be:

$$e(v) = E(v) \frac{-c_v + \sqrt{\sum_{i=1}^{N} \frac{c_v}{B_{u_i}}}}{\sum_{i=1}^{N} \frac{1}{B_{u_i}} - c_v}. \qquad (12)$$

We can rewrite the solution such that $e(v) = B'_v E(v)$, where B'_v is a constant number. Therefore $E(u_1) = \frac{E(v) - B'_v E(v)}{B} = B''_v E(v)$ (from Eq. 8), and the delay on the sub-tree is indeed

$$\frac{1}{e(v)} + \frac{1}{B_{u_1} E(u_1)} = \frac{1}{B'_v E(v)} + \frac{1}{B_{u_1} B''_v E(v)} = \\ = \frac{B_{u_1} B''_v + B'_v}{B'_v B''_v B_{u_1} E(v)} = \frac{1}{B_v E(v)}. \qquad (13)$$

■

A formal description of the algorithm to be executed is presented in the following (Algorithm 1). The algorithm consists of two functions: *Calculate-Frequency-Division(v)* and *Assign-Frequencies(v,energy)*. The first function receives a node identifier v and returns the delay $D(v)$ on the sub-tree of v as a function of the amount of energy $E(v)$ assigned to this sub-tree. This function performs a recursive call for each node in v's sub-tree, in order to find its sub-tree delay. Since the energy-delay dependency has a well-defined form $D(v) = \frac{1}{B_v E(v)}$, this function returns only the coefficient B_v. In addition, this function computes the partition of the energy assigned to the sub-tree of v between node v and each of its children's sub-trees: $e(v), E(u_1), \ldots, E(u_N)$. Afterwards, the overall tree energy, $E(root)$, is computed. In order to solve Problem 1, the algorithm sets $D(root)$ to be the maximum tolerated delay, and computes $E(root)$ using the equation $E(root) = \frac{1}{B_{root} D(root)}$. The second function, *Assign-Frequencies(v,energy)*, performs the actual energy assignment. This function receives a node identifier and the amount of energy available to the sub-tree of this node, and calculates the node's wake-up frequency using the value of $e(v)$ as found by *Calculate-Frequency-Division*. It then performs a recursive call for the node's children, using the values $E(u_1), \ldots, E(u_N)$.

Algorithm 1: For a given value of maximum delay, this algorithm determines the wake-up frequency for every node such that the overall energy is minimized.

1. Calculate the optimal wake-up frequency assignment by executing function *Calculate-Frequency-Division(root).*
2. Find $E(root)$
3. Calculate the precise wake-up frequencies of all the nodes by calling *Assign-Frequencies(root,E(root)).*

Function **Calculate-Frequency-Division(v)**

The function receives a node identifier v. It considers one unit of energy and returns the value of B_v in Eq. 4. This value determines the ratio of $E(v)$ to $D(v)$. As a secondary result, the function calculates the energy division between the node and its children sub-trees, $e(v), E(u_i)$.

```
if $v$ is a leaf
    $e(v) = 1$        /* assign the whole unit of energy */
    return $(\frac{1}{c_v})$        /* for a leaf: $D(v) = \frac{c_v}{E(v)}$ */
else
    for every  $u_i \in children(v)$
        $B_{u_i} = Calculate\text{-}Frequency\text{-}Division(u_i)$
    if $\sum_{i=1}^{n} \frac{1}{B_{u_i}} = 1$        /* this means $B = B_{u_1}$ */
        $e(v) = \frac{1}{2}$        /* from Eq. 11 */
    else
        $e(v) = \frac{-c_v + \sqrt{\sum_{i=1}^{n} \frac{c_v}{B_{u_i}}}}{\sum_{i=1}^{n} \frac{1}{B_{u_i}} - c_v}$        /* from Eq. 12 */
    $E(u_1) = \frac{1 - e(v)}{\sum_{i=1}^{n} \frac{B_{u_1}}{B_{u_i}}}$        /* from Eq. 8 */
    for every  $u_i \in children(v)$
        $E(u_i) = \frac{Bu_1}{Bu_i} E(u_1)$        /* from Eq. 5 */
    return $\frac{e(v) B_{u_1} E(u_1)}{B_{u_1} E(u_1) + e(v)}$        /* from Eq. 13 */
```

Function **Assign-Frequencies(v, energy)**

This function receives a node identifier and the energy assigned to its sub-tree and divides the energy between the node and its children sub-trees:

```
$f(v) = energy \cdot e(v)$
for every  $u_i \in children(v)$
    $Assign\ Frequencies(u_i, energy \cdot E(u_i))$
```

■

Theorem 1: Algorithm 1 finds the optimal energy assignment.

Proof: Assume, by contradiction, that there exists an optimal assignment A^*, which achieves smaller delay than assignment A as calculated by Algorithm 1, with the same overall energy. Let $e(v)$ and $e^*(v)$ denote the wake-up frequency of node v under A and A^* respectively. Let $E(v)$ and $E^*(v)$ be the overall energy of the sub-tree rooted at v under A and A^* respectively.

Since $A \neq A^*$, there must exist a node u such that $\frac{e(u)}{E(u)} \neq \frac{e^*(u)}{E^*(u)}$. Because for every leaf v $\frac{e(v)}{E(v)} = 1 = \frac{e^*(v)}{E^*(v)}$ (all the energy assigned to the sub-tree of a leaf is used only by the leaf), there must exist a node u such that $\frac{e(u)}{E(u)} \neq \frac{e^*(u)}{E^*(u)}$, while for all the descendants of u the two ratios are equal. Let us consider the delay on the sub-trees of the children of u: $w_1 \ldots w_N$. For each w_i, the delay has the same form as in Eq. 4: $D^*(w_i) = \frac{1}{B_{w_i} E^*(w_i)}$ under assignment A^* and $D(w_i) = \frac{1}{B_{w_i} E(w_i)}$ under assignment A. Note that the constant B_{w_i} is the same for both assignments, because it depends on the division of energy between the nodes in $T(u)$ and not on the actual amount of energy assigned to each node. Recall that u was chosen such that $\frac{e(v)}{E(u)} = \frac{e^*(v)}{E^*(u)}$ holds for every descendant v of u.

According to Claim 1(a), for any two children of u, w_i and w_j, the delays on their sub-trees are equal: $D^*(w_i) = D^*(w_j)$. Therefore, the delay on the sub-tree of u under A^* is $D^*(u) = \frac{c_u}{e^*(u)} + \frac{1}{B_{w_i} E^*(w_i)}$ for every i. Hence, using the equality from Eq. 8, we can represent the delay on $T(u)$ under A^* as a function of $e^*(u)$, when $E^*(u)$ is given as a parameter. But this is exactly the dependency between the delay under A and the wake-up frequency $e(u)$ when $E(u)$ is given as a parameter. As shown in the proof of Claim 2, the delay is minimal for $e(u) = E(u) \cdot B'_u$. Therefore, when the sub-tree energy is equal to $E^*(u)$, the delay is minimal if the wake-up frequency of u is $e^*(u) = E^*(u) \cdot B'_u = E^*(u) \frac{e(u)}{E(u)}$. Any other energy division between node u and its children sub-trees will cause a greater delay. Therefore, assigning node u this wake-up frequency and adjusting the energies of its children sub-trees respectively will cause the delay on the sub-tree of u to decrease below $D^*(u)$, while the overall sub-tree energy remains the same. Consequently, the maximum delay of the whole network can be reduced, in contradiction to the optimality of the assignment A^*. ■

We have implemented the algorithm above in order to calculate the optimal assignment of 1000 units of energy on different topologies. For simplicity, each node v is assigned the same value of c_v. Figure 3(a) shows a case where the collection of routes forms a full binary tree. As expected, the root receives the highest wake-up frequency and the nodes' wake-up frequencies decrease exponentially as we go down the tree. The next topology, shown in Figure 3(b), is a star with n leaves as the root children. In this case the optimal wake-up frequency assignment requires that the root wake-up frequency be $\sqrt{n}$ times larger than that of the leaves. The third studied topology, in which every inner node has a single child, is presented in Figure 3(c). Since in this network there is only one path from the leaves to the root, the optimal energy assignment is to give an equal share to every node. The last topology is a truncated binary tree, presented in Figure 3(d). This example reveals that two factors determine the wake-up frequency distribution: the distance of a node to the root and the size of the node's sub-tree. For an inner node, the size of the subtree is the dominating factor in determining the wake-up frequency distribution ($a > b_2 > c_2 > d_2 > e_2$), while for a leaf node, the distance of the node to the root is the dominating factor ($b_1 < c_1 < d_1 < e_1$).

V. Step 2: Limiting the Energy Consumed by a Single Node

Network lifetime is not affected only by the overall consumed energy, but also by the way this energy is divided among the network nodes. As shown in the examples of Figure 3, the variance of the frequencies assigned by Algorithm 1 might be very high. We now present Step 2 of the algorithm, which addresses this problem. We start with the following definition:

Definition 2: A τ-optimal assignment of wake-up frequencies to nodes is an assignment that guarantees an upper bound τ on the energy assigned to every node and an upper bound D on the maximum delay, while minimizing the total consumed energy.

We now formalize this definition as an optimization problem.

Problem 2: Let T be a tree that consists of all the paths from the sensors to the gateway without the leaves. Let f_v and c_v be the wake-up frequency and the average energy consumption during each wake-up for node v, respectively. Then, we want to

minimize: $\sum_{v \in T} f_v \cdot c_v$

subject to:

- for every directed path $v_1 \rightarrow \cdots \rightarrow v_N$ in T, where v_N is the gateway, $\sum_{v \in \{v_1 \cdots v_N\}} \frac{1}{f_v} \leq D$.
- for every node v in T: $0 < f_v \cdot c_v \leq \tau$. ■

The difference from Problem 1 is the second condition, where for every node v, $f_v \cdot c_v \leq \tau$ must also hold.

We now show how to solve Problem 2 in the special case where the average amount of energy expended by a node u during each wake-up is equal to or smaller than the average amount of energy expended by its parent v. That is, $c_u \leq c_v$. We believe that this is a realistic assumption for the following reasons:

- If the traffic generated in the network is sparse, as often is the case in sensor networks for alarm events [22], we can assume that when a node wakes up and stays in active mode for a short period (which is still long enough to determine if there exists a pending packet from a neighboring node), it is able to receive from its downstream neighbors all their awaiting packets. In some busy times, a node will have to extend its active period in order to allow all the packets to be received. However, on the average we could assume that for every two nodes v and u, $c_v \approx c_u$.
- If the traffic generated in the network is not sparse, nodes that are located closer to the gateway are expected to be more loaded. Hence, these nodes will have to expand their active periods much more often than their children. Hence, in such networks $c_u < c_v$ holds if v is the parent of u.

Claim 3: If node v is a parent of node u, then under optimal assignment the energy expended by v is greater than or equal to the energy expended by u (i.e., $c_v \cdot f_v \geq c_u \cdot f_u$).

Proof: Assume the claim does not hold, namely, $c_v \cdot f_v < c_u \cdot f_u$. By our assumption, $c_v \geq c_u$. Hence, $f_v < f_u$. Now, assume that we increase f_v and decrease f_u simultaneously, such that $c_v \cdot f_v + c_u \cdot f_u$ does not change. Clearly, the overall energy expended by the tree's nodes does not change, while the delay $D(v)$ on the sub-tree of v decreases, in contradiction to the optimality of the original assignment. ■

Claim 4: Let the optimal assignment found by Algorithm 1 be A. Suppose there exists an assignment A^{τ} that achieves the same upper bound on the maximum delay while limiting the energy expended by each sensor to τ. If under assignment A the energy expended by a node is greater than τ, then under assignment A^{τ} the energy expended by this node is exactly τ.

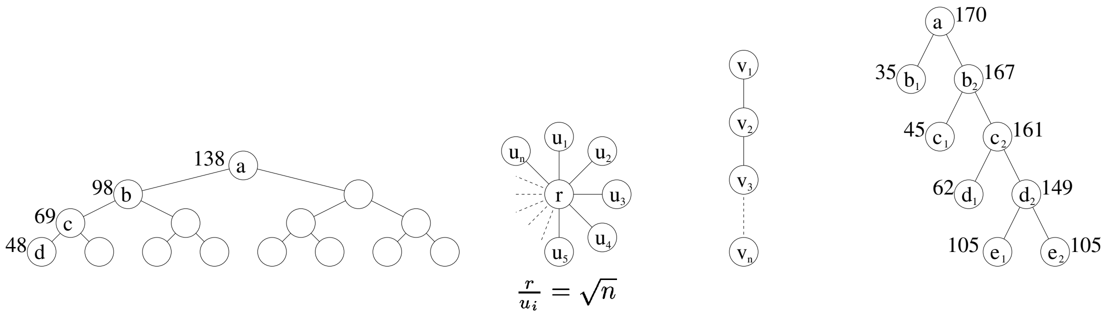
Proof: Assume the claim does not hold. Namely, there is a non-empty set $\overline{V}$ of nodes whose energy consumption under A is higher than τ while their energy consumption under A^{τ} is strictly smaller than τ.

We first prove that $\overline{V}$ cannot contain a leaf node. Suppose that v is a leaf node in $\overline{V}$. Consider the path P from v to the root. By Claim 3, the energy expended by all the nodes along this path under A is higher than τ. By Claim 1(b), the delay from v to the root under A is equal to the maximum tolerated delay D. Hence, under A^{τ}, the energy expended by at least one of the nodes along P is greater than τ, which yields a contradiction.

Suppose now that $\overline{V}$ contains a non-leaf node v. By Claim 1(b), under both A and A^{τ} the delay from the leaves to the root is equal to the maximum tolerated delay D. However, the delay from node v to the root under A^{τ} is higher than the delay from node v to the root under A. This is because, by Claim 3, the energy expended by every ancestor of v is greater than τ under A, but is not greater than τ under A^{τ}. Therefore, in the sub-tree rooted at v, the delay from the leaves to node v under A^{τ} must be shorter than the delay from the leaves to node v under A.

Let $E^{A}(w)$ be the energy assigned by A to the sub-tree of a node w under A and $E^{A^{\tau}}(w)$ be the energy assigned to this sub-tree by A^{τ}. Let $e^{A}(w)$ be the energy expended by node w under A, and $e^{A^{\tau}}(w)$ be the energy expended by this node under A^{τ}.

We now show that every node in the sub-tree of v consumes the same amount of the sub-tree's overall energy under both assignments. If this claim is incorrect, there must exist a



(a) a full binary tree (b) a star tree (c) a cascade (d) a truncated binary tree

Fig. 3. The considered topologies and the results of Step 1

non-leaf node u such that $\frac{e^A(u)}{E^A(u)} \neq \frac{e^{A^\tau}(u)}{E^{A^\tau}(u)}$, while for any descendant w of u, $\frac{e^A(w)}{E^A(w)} = \frac{e^{A^\tau}(w)}{E^{A^\tau}(w)}$ holds. Note that such a u exists, because, for every leaf y, $E(y) = e(y)$ under any assignment. Since the wake-up frequency of every descendant of u is smaller than τ, the delay function of every child u_i of node u under A^τ is $D^{A^\tau}(u_i) = \frac{1}{B_{u_i} E^{A^\tau}(u_i)}$, the same function as under A with the same constant B_{u_i}. Therefore, the delay function of the sub-tree of u under A^τ is $\frac{1}{e^{A^\tau}(u)} + \frac{1}{B_{u_i} E^{A^\tau}(u_i)}$. The derivation of this function has the same form as in Eq. 7, and therefore has only one solution. Hence, the delay function has no local minimum, and if the energy division is not optimal, it can always be improved.

All the nodes in the sub-tree of u have wake-up frequencies smaller than τ, as follows from Claim 3. Therefore, if the delay under A^τ is $D^{A^\tau}(u) = \frac{1}{e^{A^\tau}(u)} + \frac{B}{E^{A^\tau}(u) - e^{A^\tau}(u)}$, as in Eq. 9, there must exist an ε (positive or negative), such that by defining the wake-up frequency of node u to be $e^{A^\tau}(u) + \varepsilon$, and re-computing the energy assigned to every sub-tree of node u's children without changing the overall energy of u's sub-tree, the delay on the sub-tree of u is reduced to $D'(u) = \frac{1}{e^{A^\tau}(u)+\varepsilon} + \frac{B}{E^{A^\tau}(u) - (e^{A^\tau}(u)+\varepsilon)}$. Hence, a smaller maximum delay is achieved with the same sub-tree energy or, optionally, the same maximum delay is achieved with less energy, in contradiction to the fact that A^τ is a best energy assignment under the given constraint. We conclude that for every node in the sub-tree of v, its assigned fraction of energy is equal under both assignments.

Since the wake-up frequency of v under A^τ is smaller than under A ($e^{A^\tau}(v) < e^A(v)$), the overall amount of energy consumed by the sub-tree is smaller as well, i.e., $E^{A^\tau}(v) < E^A(v)$. Therefore, the delay of the sub-tree of v under A^τ is shorter than under A, while the overall amount of energy consumed by the sub-tree is also smaller. This is impossible of course, because the energy division was shown to be the same under both assignments. ■

The following algorithm is developed, from Claims 3 and 4, in order to find a τ-optimal wake-up frequency assignment to the nodes.

Algorithm 2:

1. If by assigning the maximum energy τ to every node there is a routing path for which the delay is greater than the maximum tolerated delay D, then no solution exists. Otherwise, perform the following steps:
2. Calculate the optimal assignment using Algorithm 1. Recall that this assignment does not guarantee an upper bound on the energy assigned to a single node.
3. For every node v whose assigned energy is greater than τ, reduce it to τ.
4. For every node v whose assigned energy is equal to τ but has a child u whose assigned energy is smaller than τ, recalculate the child sub-tree's energy division using Algorithm 1 in order to reduce the maximum delay on this sub-tree such that it will be equal to the maximum tolerated delay D minus the maximum delay from the root to v.
5. Following step 4, the energy assigned to some nodes may exceed the threshold τ. If there are such nodes, return to step 3.

Theorem 2: If there exist feasible schedules that meet the constraints on the maximum delay D and the maximum energy assigned to every node τ, Algorithm 2 will find the one with the minimum total energy.

Proof: First, we prove that the algorithm stops. If no feasible assignment exists, the algorithm discovers this at step 1, because such an assignment exists if and only if the maximum delay is not larger than D when all the nodes are assigned energy τ. If the algorithm passes step 1, it must stop unless it enters an infinite loop in steps 3–5. However, such an infinite loop is not possible, because after the energy assigned to a specific node is reduced to τ in step 3, it cannot be changed. Therefore, each time the algorithm returns to step 3, the energy of at least one additional node becomes fixed, and after at most $|V|$ iterations, where $|V|$ is the number of nodes, the algorithm must stop.

Next, we prove that the energy assignment found by the algorithm is feasible, i.e., that both requirements are met. When the algorithm stops, no node is assigned energy greater than τ, thereby meeting the first requirement. The second requirement, namely an upper bound on the maximum delay, is fulfilled due to the correctness of Algorithm 1: every time the energy of a node is reduced in step 3, the energy assignment on every path through this node is recalculated, such that the

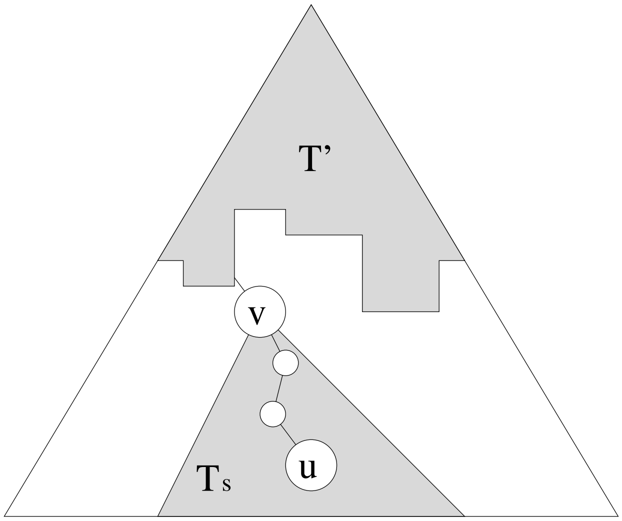


Fig. 4. Algorithm 2 after $k-1$ iterations

maximum delay D is guaranteed for this path.

Finally, we show that the energy assignment determined by the algorithm is indeed optimal. We start by proving that after each iteration of step 3 of the algorithm, the set of nodes whose energy is equal to τ is either empty or it forms a connected sub-tree containing the root. We prove the claim by induction on the iterations. From Claim 3 we know that nodes located closer to the root are assigned greater energy. Therefore, if the energy assigned to a node in step 2 is greater than τ, then all the ascendants of this node towards the root are also assigned energy greater than τ. Hence, the set of nodes whose energy is reduced to τ during the first iteration of step 3 forms a sub-tree rooted at the root. Let us consider now the k^{th} iteration of step 3. From the induction assumption we know that the nodes whose energy is τ following the $(k-1)^{th}$ iteration of step 3 form a sub-tree rooted at the gateway. Let this tree be denoted as T' (see Figure 4). For every sub-tree T_s whose energy increases in step 4 of the $(k-1)^{th}$ iteration, Claim 3 also holds. Therefore, if a node u in such a sub-tree is assigned energy greater than or equal to τ, all its ascendants in the same sub-tree T_s are also assigned energy greater than or equal to τ. This implies that if the energy of a node u is reduced in step 3 of the k^{th} iteration, the energy of the sub-tree root v is also reduced, and the same holds for all the nodes on the path from u to v (see Figure 4). By the definition of step 4, when the sub-tree root is chosen in step 4 of the $(k-1)^{th}$ iteration, its parent has energy of τ. By the induction assumption, the parent of a sub-tree root belongs to the sub-tree T', formed by the nodes with energy of τ. Hence, after the k^{th} iteration, all the nodes whose energy is equal to τ indeed form a connected sub-tree.

We now show that every node u whose assigned energy is τ will be assigned that energy under any τ-optimal assignment. The proof is by induction on the iteration of step 3, during which node u is assigned τ energy. After the first iteration, the correctness of the claim follows from Claim 4. Assume that for the first $k-1$ iterations the claim holds. If the energy of node u is reduced to τ in the k^{th} iteration, then its energy was recalculated in step 4 of the $(k-1)^{th}$ iteration. Therefore, u is a member of a sub-tree T_s rooted at node v. Since v's parent belongs to T', v was assigned energy greater than τ in step 4 of the $(k-1)^{th}$ iteration. By the previous claim, all the ascendants of v are assigned energy of τ at this moment. Therefore, by the induction assumption, all the ascendants of v are assigned energy of τ under any τ-optimal assignment. This implies that under any τ-optimal assignment the delay on the sub-tree T_s of v is equal to D minus the delay from v to the root (see Figure 4). The latter delay is equal to $\frac{1}{\tau}$ multiplied by the number of nodes along this route. From the correctness of Algorithm 1 we know that in step 4 of the k^{th} iteration, an optimal assignment for the sub-tree of v is determined. Therefore, by Claim 4, every node that receives energy greater than τ under an optimal assignment will receive energy of τ under any τ-optimal assignment.

To summarize, when the algorithm stops, every node whose assigned energy is τ will have the same energy under any τ-optimal assignment, while the energy assigned to each of the other nodes is calculated in the last iteration of step 4 using Algorithm 1, and has already been proven to be optimal. ■

Algorithm 1 performs two passes along the tree, and for every node the calculations require $O(1)$ operations. Therefore, the complexity of the algorithm is $O(2|V|)$, where $|V|$ is the number of nodes. Algorithm 2, in its naive implementation, requires $O(|V|^2)$ operations, because the energy division can be recalculated for each node sub-tree. Note however, that as shown in the correctness proof of Algorithm 1, the optimal energy division in a sub-tree is unique and does not depend on the available energy. Since Algorithm 1 is implemented in the first iteration, the optimal energy division for every sub-tree is already determined. This assignment can be used while allocating energy to nodes in step 3 of the algorithm. Therefore, the reallocation process can be performed in $O(1)$ operations at each node, and the complexity of Algorithm 2 is reduced to $O(3|V|)$: two passes on the tree in step 2 and an additional pass for the energy reallocation in steps 3, 4 and 5.

VI. Simulation Results

We compare now the algorithms presented in the paper with an equal wake-up frequency that does not depend on the location of the nodes. The latter approach is referred to as "equal assignment." The comparison is conducted by building routing trees on random graph [23]–[25]. One thousand nodes, representing the sensors, are randomly placed over a 100X100 grid. The transmission range varies between 15 and 60. For each tested range r, any two nodes whose Euclidean distance is not greater than r are considered to be connected. If the created graph is not connected, this instance is ignored by the simulator. If it is connected, we select a random node to function as the gateway, and assume that the routing from each sensor to the gateway is performed over the shortest path.

We first compare the overall energy consumption of the network under equal assignment and under Algorithm 2.

To this end, we assign an equal wake-up frequency e to every node and then measure the maximum delay resulting from this assignment. This value is considered as the target maximum delay D. Next, we execute Algorithm 2 with D as the maximum tolerated delay and $\tau = e \cdot$*Limit-Factor* as the maximum wake-up frequency assigned to a single node. When *Limit-Factor* is 1, this maximum is equal to the wake-up frequency of equal assignment. When *Limit-Factor*$\rightarrow \infty$, there is no limit on the difference between the maximum and minimum assigned wake-up frequencies. Therefore, in this case Algorithm 2 is reduced to Algorithm 1.

Figure 5 shows the overall energy ratio between Algorithm 2 and equal assignment for shortest-path routing. We start with the case where *Limit-Factor*=1. Despite the fact that the maximum wake-up frequency a single node can be assigned is exactly as in equal assignment, the total consumed energy is reduced by 35%-65%. The reason for this is that under equal assignment the maximum delay from some of the leaves is smaller than D, while under our algorithm the maximum delay from all the leaves to the root is equal to D. Consequently, leaves closer to the root, as well as nodes on the path from these leaves to the root, can be assigned a smaller wake-up frequency than nodes on the longer paths. If all the leaves had the same depth, the overall energy consumption of the two algorithms would be equal for this *Limit-Factor*, but the probability that a random tree will have this property is negligible.

When *Limit-Factor* grows, Algorithm 2 has the flexibility to assign higher wake-up frequencies to some of the nodes, thereby approaching the optimal solution of Algorithm 1. Apparently, *Limit-Factor*=5 is sufficient for Algorithm 2 to find almost the same wake-up frequency assignment as Algorithm 1. This is evident from the convergence of the curves with *Limit-Factor*=5 and *Limit-Factor*=50.

The improvement achieved by our algorithm is almost independent of the chosen routing algorithm. To show this, we repeat the experiment, this time considering a routing algorithm that minimizes the number of internal nodes (that is, the number of nodes that have to wake up periodically) [26]. As shown at Figure 6, the improvement achieved by our scheme is similar to the improvement in Figure 5 for the shortest-path routing.

As shown earlier, in some networks the maximum wake-up frequency (energy) assigned to some of the nodes is much greater than the average wake-up frequency. This may results in a loss of these nodes and possible loss of network connectivity. We performed additional simulations in order to determine how the standard deviation of the assigned wake-up frequency is influenced by the limit imposed on a single node. The transmission range is 15 in one instance and 40 in another. The value of *Limit-Factor* ranges between 1 to 5. The results are presented in Figure 7, where the standard deviation of the wake-up frequency is compared to the average wake-up frequency.

We can see that when *Limit-Factor* grows, the variance of the assigned wake-up frequencies grows as well, while the average wake-up frequency decreases. The differences in the two graphs can be explained by the different routing tree structures. When the range is equal to 40, the routing tree resembles a star tree. Therefore, a significant portion of its nodes are leaves adjacent to the root. All these nodes are assigned the same low wake-up frequency. They reduce the wake-up frequency average, but increase the wake-up frequency variance.

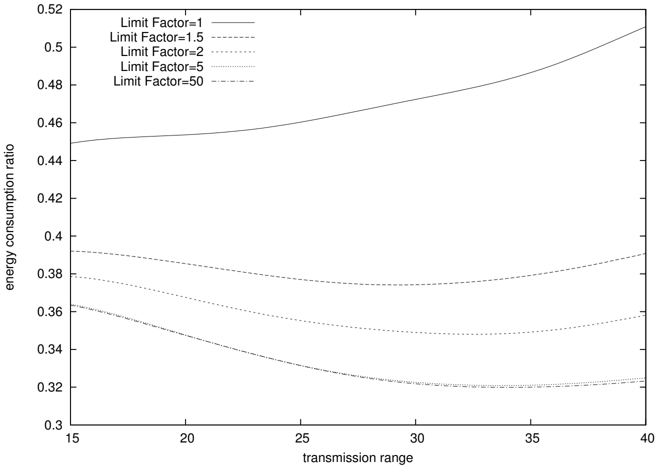


Fig. 5. The energy consumption ratio for a shortest-path routing as a function of the transmission range for different wake-up frequency upper bounds

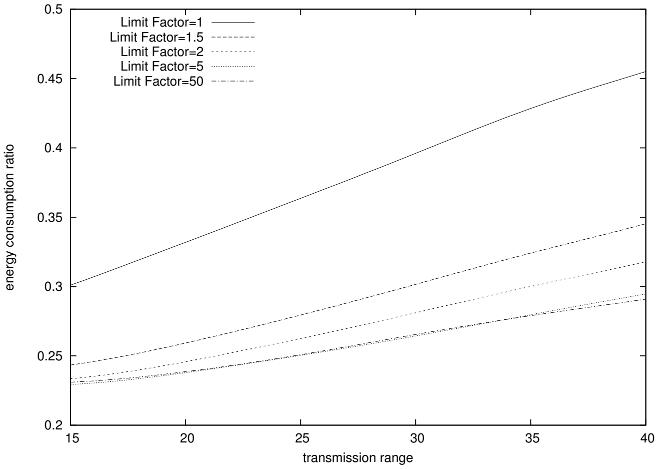
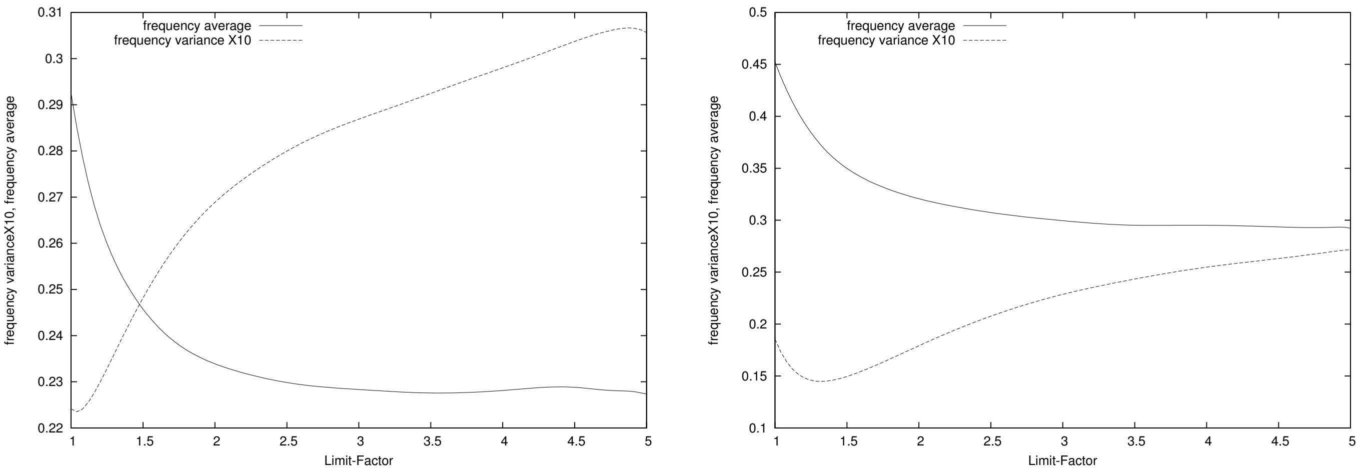


Fig. 6. The energy consumption ratio for the case where the routing algorithm minimizes the number of internal nodes

We now show that the new algorithm significantly extends the sensor network lifetime. For this study, we use an energy-constrained shortest path routing algorithm that takes into account the energy available at every node, as explained below. In each simulation's instance, the energy consumed by an inner node during a time unit is set to be equal to the node's wake-up frequency, while the energy consumed by a leaf node is fixed and small. When the energy of an inner node falls below a threshold, whose value is set to 20% of the initial energy, routing through this node is not allowed any more. Hence, shortest paths are computed without taking this node



(a) transmission range = 15

(b) transmission range = 40

Fig. 7. Assigned wake-up frequency average and variance as a function of *Limit-Factor*

into account. Consequently, the energy consumed by such a node is minimized. Each simulation instance is stopped when the routing from some sensor to the gateway is not possible any more. The lifetime of the network for this instance is then compared to the case where wake-up frequencies are equal for all nodes. Figure 8 shows the ratio between the average lifetime of the network in these two cases, for a 100x100 grid and two *Limit-Factor* values.

Consider first the upper curve, for *Limit-Factor*$= 3$. We can see that for reasonable values of transmission range (< 25), our algorithm increases the network lifetime by 40%-80%. The improvement of our algorithm decreases when the network is dense. This is because in such a case the routes from the sensors to the gateway become shorter. Therefore, the wake-up frequency assigned by our algorithm to the nodes that are close to the gateway is greater, these nodes are likely to run out of energy faster. When the network is not so dense, the routing paths are longer, and the wake-up frequency assigned by our algorithm to these key nodes is smaller. Consequently, routing through these nodes is possible for longer time, and the total network's lifetime increases.

When we increase the value of *Limit-Factor* from 3 to 5, the improvement in the network lifetime is smaller. At first glance this result seems to contradict the results shown in the previous graphs, because we saw there that greater values of *Limit-Factor* improve the performance of our algorithm. Indeed, when the value of *Limit-Factor* increases, the total consumed energy is reduced, but the likelihood that some popular nodes will run out of their energy sooner is greater. Therefore, when selecting the value of *Limit-Factor* to work with, one needs to take into account both the network lifetime and the total consumed energy. If only the latter is important, as is the case when energy cannot be re-charged, the value of *Limit-Factor* should be close to 1.

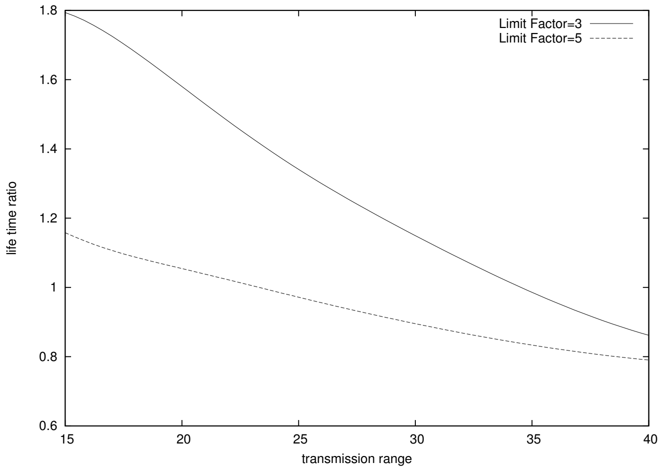


Fig. 8. The network life time ratio as a function of the transmission range for different values of *Limit-Factor*

VII. Conclusions

We presented an algorithm for determining the wake-up frequency of the nodes in a sensor network. This algorithm minimizes the energy consumption of the nodes and bounds the maximum delay on the routes from the nodes to the gateway. We simulated the algorithm over random sensor networks with different topologies and studied its impact on network energy consumption. This study revealed that the algorithm reduces the total energy consumption by 60-70% compared to energy consumption under equal assignment. When the proposed algorithm is used along with an energy-aware routing algorithm, the network lifetime was shown to increase by 40%-80%. Future work is needed in order to find closed formulas for the relationship between the upper bound τ on the frequency assigned to every node and the network lifetime. This relationship depends on several important factors such as network topology, the algorithm for building the routing tree T and the algorithm for deciding when a node with limited energy should be excluded from the routing paths.

References

[1] Q. Li and D. Rus, “Global clock synchronization in sensor networks,” in *Infocom 2004*, 2004.

[2] J. Elson and K. Römer, “Wireless sensor networks: a new regime for time synchronization,” *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 149–154, 2003.

[3] Y. Li, W. Ye, and J. Heidemann, “Energy and latency control in low duty cycle MAC protocols,” in *Proceedings of the IEEE Wireless Communications and Networking Conference*, New Orleans, LA, USA, March 2005.

[4] J. Elson, L. Girod, and D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 147–163, 2002.

[5] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, “A taxonomy of wireless micro-sensor network models,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 2, pp. 28–36, 2002.

[6] V. Annamalai, S. Gupta, and L. Schwiebert, “On tree-based convergecasting in wireless sensor networks,” in *IEEE Wireless Communications and Networking Conference*, vol. 3, 16-20 March 2003, pp. 1942 – 1947.

[7] Y. Xue and B. Li, “A location-aided power-aware routing protocol in mobile ad hoc networks,” in *Proc. of IEEE Globecom*, 2001.

[8] C. Inc., “Cc2420 data sheet.”

[9] H. Cao, K. Parker, and A. Arora, “O-MAC: a receiver centric power management protocol,” in *ICNP*, 2006.

[10] Q. Ye, F. Silva, and J. Heidemann, “Ultra-low duty cycle mac with scheduled channel polling,” in *SenSys*, 2006.

[11] C. B. Margi and K. Obraczka, “Instrumenting network simulators for evaluating energy consumption in power-aware ad-hoc network protocols,” in *MASCOTS ’04*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 337–346.

[12] W. Ye, J. S. Heidemann, and D. Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks.” *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, 2004.

[13] G. Lu, N. Sadagopan, B. Krishnamachari, and A. Goel, “Delay efficient sleep scheduling in wireless sensor networks.” in *INFOCOM*, 2005, pp. 2470–2481.

[14] G. Lu, B. Krishnamachari, and C. S. Raghavendra, “An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks.” in *IPDPS*, 2004.

[15] W. Lai and I. C. Paschalidis, “Routing through noise and sleeping nodes in sensor networks: latency vs. energy trade-offs,” in *Proceedings of the 45th IEEE Conference on Decision and Control*, December 2006, pp. 2716–2721.

[16] A. E. Gamal and J. Mammen, “Optimal hopping in ad hoc wireless networks,” in *INFOCOM*, 2006.

[17] O. Dousse, P. Mannersalo, and P. Thiran, “Latency of wireless sensor networks with uncoordinated power saving mechanisms.” in *MobiHoc*, 2004, pp. 109–120.

[18] B. Prabhakar, E. Uysal-Biyikoglu, and A. E. Gamal, “Energy-efficient transmission over a wireless link via lazy packet scheduling.” in *INFOCOM*, 2001, pp. 386–394.

[19] C. Nair, A. E. Gamal, B. Prabhakar, E. Uysal-Biyikoglu, and S. Zahedi, “Energy-efficient scheduling of packet transmissions over wireless networks.” in *INFOCOM*, 2002.

[20] L. Miao and C. G. Cassandras, “Optimal transmission scheduling for energy-efficient wireless networks.” in *INFOCOM*, 2006.

[21] Y. Yu, B. Krishnamachari, and V. K. Prasanna, “Energy-latency tradeoffs for data gathering in wireless sensor networks.” in *INFOCOM*, 2004.

[22] E. H. Callaway, *Wireless Sensor Networks: Architectures and Protocols*. Boca Raton, Florida 33431: Auerbach Publications, 2004.

[23] Y. Xu, J. S. Heidemann, and D. Estrin, “Geography-informed energy conservation for ad hoc routing,” in *Mobile Computing and Networking*, 2001, pp. 70–84.

[24] J. van Greunen and J. Rabaey, “Lightweight time synchronization for sensor networks,” in *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*. ACM Press, 2003, pp. 11–19.

[25] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, “Topology management for sensor networks: exploiting latency and density,” in *MobiHoc ’02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM Press, 2002, pp. 135–145.

[26] R. Solis-Oba, “2-approximation algorithm for finding a spanning tree with maximum number of leaves,” in *European Symposium on Algorithms*, 1998, pp. 441–452.

PLACE
PHOTO
HERE

Reuven Cohen (M’93, SM’99) received the B.Sc., M.Sc. and Ph.D. degrees in Computer Science from the Technion - Israel Institute of Technology, completing his Ph.D. studies in 1991. From 1991 to 1993, he was with the IBM T.J. Watson Research Center, working on protocols for high speed networks. Since 1993, he has been a professor in the Department of Computer Science at the Technion. He has also been a consultant for numerous companies, mainly in the context of protocols and architectures for broadband access networks. Dr. Cohen has served as an editor of the IEEE/ACM Transactions on Networking, and the ACM/Kluwer Journal on Wireless Networks (WINET). Dr. Cohen is a senior member of the IEEE and heads the Israeli chapter of the IEEE Communications Society.

PLACE
PHOTO
HERE

Boris Kapchits received the B.Sc. in and M.Sc. in Computer Science from the Technion - Israel Institute of Technology, Haifa, Israel, in 2003 and 2005, respectively. Since 2005, he has been a Ph.D. student in Computer Science Department in the Technion, working on mesh sensor networks.