Starting Up Multi-Gateway WSNs

Barbara Staehle and Dirk Staehle
University of Würzburg, Institute of Computer Science, Würzburg, Germany
{bstaehle,dstaehle}@informatik.uni-wuerzburg.de

ABSTRACT

The Internet connection of wireless sensor networks (WSNs) is in general assured by more than one gateway. Planning or optimizing such multi-gateway WSNs (MWSNs) has lately been the focus of the research community. For this purpose, most researchers assume that each sensor node knows about a path to each gateway node. The question how a MWSN can be efficiently started up, i.e. how the sensor nodes learn about the gateways has in contrast not yet been studied thoroughly. To close this gap, we formally describe the problem of starting up a MWSN and demonstrate the importance of optimizing the start-up phase by comparing the performance of an optimal solution to the performance of less efficient distributed algorithms.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication, Sensor networks

General Terms

Algorithms, Management, Performance

Keywords

Wireless Sensor Networks, Multi-Gateway, Start-Up

1. INTRODUCTION

For most people daily-life activities like administrative businesses, cinema evenings, or shopping are not imaginable without the Internet. So why should a geographer, not also search on the Internet if she is interested in the dynamics of glaciers? Thanks to Glacsweb [1], she can in fact connect to a WSN deployed on a glacier in Norway or access one of more than 50 other environmental monitoring WSNs via the Internet [2].

Most often, the connection between the Internet and a WSN is assured by several (Internet) gateways which have more energy resources and computing power, and are equipped with network interfaces with a higher data rate than the sensor nodes. If applications running on top of such a so-called MWSN shall take full advantage of the existence of several gateways, each sensor node has to know the next hops and distances to all gateways which enables the choice among several routes to the Internet. This is e.g. necessary to support applications where all sensor nodes periodically send data to the nearest gateway, but where a user may optionally trigger a request over a dedicated gateway. Another example are video sensor networks which could greatly benefit from the use of multiple description coding and and a mechanism sending each video layer to a different gateway. All in all, the situation where all sensor nodes know about all deployed gateways enables delay optimizations, QoS guarantees and resilience mechanisms.

Actually, most studies on MWSNs take it for granted that this knowledge is existing or assume that it can be created with a simple flooding mechanism. The results we present in this study do however demonstrate, that it is a non-trivial task to let all sensor nodes learn from all gateways. The procedure during which the gateways announce their presence to all sensor nodes is what we call *starting up* a MWSN or *MWSN start-up*. To the best of our knowledge it has never been considered how this start-up procedure can be successfully and efficiently carried out. We therefore establish an abstract framework which allows to formulate the MWSN start-up as an optimization problem. Thereby, we are able to point out to which degree simple distributed algorithms perform worse in terms of start-up success, time, and energy consumptions.

This work is structured as follows: In Section 2 we review related contributions. Section 3 introduces our analytical framework. We present an optimal start-up mechanism in Section 4 and two representative examples of distributed algorithms in Section 5. Section 6 contains a comparison of all start-up strategies. A conclusion and an outlook can be found in Section 7.

2. RELATED WORK

In the earth science domain WSNs are close to becoming a standard research tool. This is in any case the argumentation of Hart and Martinez [2] who reviewed over 50 examples of environmental monitoring WSNs. Most of those sensor networks are what we call MWSNs, i.e. are connected to the Internet by more than one gateway. Nearly as plentiful as deployed MWSNs are theoretical works on finding the optimal position of multiple gateways. Exemplary contributions are the works of Oyman et al. [3] and Vincze et al. [4]. Both papers propose to use distance-based clustering algo-

This work was funded by the Federal Ministry of Education and Research of the Federal Republic of Germany (Förderkennzeichen 01 BK 0800, GLab). The authors alone are responsible for the content of the paper.

rithms for finding lifetime-maximizing positions of a given number of sinks or gateways.

Another thoroughly investigated planning problem is the question how data is routed in MWSNs. Many authors consider the problem of optimizing the routing topology by balancing load among gateways, minimizing the delay, establishing redundant paths, etc. Egorova-Förster and Murphy [5] describe for instance a hop-count minimizing feedback enhanced learning protocol for a MWSN where all sensor nodes already know a path to each gateway. Another example is the work of Kalantari and Shayman [6] who formulate the communication load in a MWSN as a vector field in order to optimally balance the load among the sink nodes.

In analogy to the studies on gateway position planning, which do not consider what is happening after the deployment, the works on routing optimization do neglect the question how the necessary information can be efficiently built up. Between two well analyzed periods in the life of a MWSN, there is thus a less thoroughly examined phase. The few works considering problems similar to the MWSN start-up we are aware of are reviewed in the following.

Mathew and Younis [7] propose a distributed protocol for energy-efficient *bootstrapping* of wireless sensor networks. During bootstrapping, the gateways take turn in broadcasting their presence to the sensor nodes. Upon the reception of an announcement, the sensor nodes send an identification message back to the gateways using an exponential backoff algorithm. This algorithm is however not usable for starting up a MWSN, as bootstrapping requires that the sensor nodes announce their presence to one of the gateway and not vice-versa. During start-up all sensor nodes have moreover to learn about all gateways. Additionally, is the proposed algorithm not suitable for a multi-hop scenario.

Training is thoroughly investigated by Wadaa et al. [8], Bertossi et al. [9], and Barsi et al. [10]. During the training process, the sink follows a precomputed schedule for broadcasting messages at different transmission output powers. The sensor nodes follow their own schedule to listen to the channel. By analyzing in which slot they receive or do not receive a transmission, they learn in which of k concentric circles around the sink node they are in. The protocol as described in [8] requires all sensor nodes to be synchronized to the sink node and to receive all messages of the sink. This requirement is not made by [9] which proposes to speed up the training process by reducing the number of messages the sink broadcasts and adding a second phase, where all sensor nodes which did not receive the message from the sink are forwarded this message by others. Another refinement allows a successful training process if not all sensor nodes are synchronized [10]. The problems that the sink node has to be able to reach all sensor nodes and that the algorithm is not suitable for the case of multiple sinks however persist.

Another algorithm for the network initialization, the *association* mechanism, is described by the IEEE 802.15.4 personal area network (PAN) standard [11]. Before becoming operational, each node of a PAN has to associate, i.e. needs to receive information from its PAN coordinator. In a previous study [12] we pointed out that the association procedure is not suitable for large multi-hop topologies and proposed optimization possibilities. Even with our extensions, the association procedure is however not suitable for MWSNs, where more than one PAN coordinator or gateway exists.

3. ANALYTICAL FRAMEWORK

In this section we introduce the notation which we use for abstracting a MWNS and the progress of the start-up phase.

3.1 Network Abstractions

A MWSN deployment consists of sensor and gateway nodes which we assume to be immobile. The latter have a wireless interface but are also connected to a wireless or wired backbone which assures the connection to the other gateways and to the Internet. We abstract this structure as a graph, where the set of *vertices* $\mathcal{V} = \mathcal{N} \cup \mathcal{G}$ is the union of the disjunct set of *sensor nodes* $\mathcal{N}$ and the set of *gateway nodes* $\mathcal{G}$. We define $V = |\mathcal{V}|$, $N = |\mathcal{N}|$ and $G = |\mathcal{G}|$.

The edges of this graph represent the wireless links. Link (x, y) exists if x and y can communicate in the optimal case, i.e. if the signal to noise ratio (SNR), $\gamma'_{x,y}$, is greater than a target signal to interference and noise ratio (SINR) γ^* which we assume to have the same value for all nodes in the network. Formally, for the existence of link (x, y) it is necessary that

$$\gamma'_{x,y} = \frac{R_{x,y}}{N_0} = \frac{T_x \cdot g_{x,y}}{N_0} \geq \gamma^*. \quad (1)$$

$R_{x,y}$ denotes the power received at node y when node x is transmitting, the thermal noise power is given by N_0. $R_{x,y}$ is obtained as a product of the transmit power of node x, T_x, and $g_{x,y}$, the path gain from node x to node y. We use Eq. (1) to define $\mathcal{E}$, the set of edges in the *connectivity graph* $\Gamma = (\mathcal{V}, \mathcal{E})$:

$$\mathcal{E} = \{(x, y) \in \mathcal{V} \times \mathcal{V} : \gamma_{x,y} \geq \gamma^*\}. \quad (2)$$

In the presence of interference, a transmission on $(x, y) \in \mathcal{E}$ may fail, as Eq. (1) is only necessary for the transmission success. A necessary *and* sufficient condition for a successful communication between x and y at time t is that the SINR between them, $\gamma_{x,y}(t)$, is larger or equal than the target SINR:

$$\gamma_{x,y}(t) = \frac{R_{x,y}}{N_0 + I_{x,y}(t)} = \frac{R_{x,y}}{N_0 + \sum\limits_{z \in \alpha(t)} R_{z,y}} \geq \gamma^*. \quad (3)$$

$I_{x,y}(t)$ denotes the interference at node y for a transmission of node x at time t. It is computed as the sum of the transmission powers received at y from all nodes in the set $\alpha(t)$ which are transmitting at the same time as x.

To determine which transmissions may be successfully carried out at the same time, we use *collision domains*. The collision domain $\mathcal{C}_{x,y} \subseteq \mathcal{V}$ of link $(x, y) \in \mathcal{E}$, contains all nodes which must not transmit at the same time as x for guaranteeing the transmission success on (x, y). Often the collision domain is derived as the set of all one-hop neighbors of y in Γ. This is however a too rough abstraction of the reality where a transmission can also fail because one or more remote nodes cause a too large amount of interference. To incorporate the effect of interference, while keeping the complexity of our model at an acceptable level, we consider only one potential interferer and derive $\mathcal{C}_{x,y}$ for the non-MIMO case

$$\mathcal{C}_{x,y} = \{v \in \mathcal{V} \setminus \{x, y\} : \frac{R_{x,y}}{N_0 + R_{v,y}} < \gamma^*\}. \quad (4)$$

3.2 Start-up Mechanism Abstraction

We assume idealistic conditions, i.e. a slotted channel access is possible and all nodes are perfectly synchronized which allows them to sleep unless they are transmitting or receiving. For abstracting the problem of starting up a MWSN within at most T time slots, we use the following four vectors:

- $k \in \{0, 1\}^{G,V,T+1}$: $k(g, x, t) = 1$, if node x knows from gateway g in slot t and 0 otherwise,

- $s \in \{0, 1\}^{G,V,T+1}$: $s(g, x, t) = 1$, if node x transmits the gateway announcement of g in slot t and 0 otherwise,

Table 1: Binary integer linear program to determine an optimal sending and receiving schedule for the MWSN start-up

Maximize

$$f(s,r,k) = \sum_{g\in\mathcal{G}} \sum_{x\in\mathcal{N}} \sum_{t=1}^{T} \left[\kappa k(g,x,t) - r(g,x,t) - s(g,x,t)\right] \quad (5)$$

Subject to

$$\sum_{g\in\mathcal{G}} k(g,g,0) = G \quad (6)$$

$$\sum_{g\in\mathcal{G}} \sum_{x\in\mathcal{V}\setminus\{g\}} k(g,x,0) = 0 \quad (7)$$

$$\sum_{g\in\mathcal{G}} \sum_{x\in\mathcal{V}} s(g,x,0) = 0 \quad (8)$$

$$\forall_{g\in\mathcal{G}}\forall_{1\leq t\leq T}\ \forall_{x\in\mathcal{V}} \quad k(g,x,t) \leq r(g,x,t) + k(g,x,t-1) \quad (9)$$

$$\forall_{g\in\mathcal{G}}\ \forall_{1\leq t\leq T}\ \forall_{x\in\mathcal{V}} \quad s(g,x,t) \leq k(g,x,t-1) \quad (10)$$

$$\forall_{t\leq T}\ \forall_{x\in\mathcal{V}} \quad \sum_{g\in\mathcal{G}} (s(g,x,t) + r(g,x,t)) \leq 1 \quad (11)$$

$$\forall_{g\in\mathcal{G}}\ \forall_{t\leq T}\ \forall_{x\in\mathcal{V}} \quad r(g,x,t) \leq \sum_{(y,x)\in\mathcal{E}} \tilde{r}(g,x,y,t) \quad (12)$$

$$\forall_{g\in\mathcal{G}}\ \forall_{t\leq T}\ \forall_{x\in\mathcal{V}}\ \forall_{y:\in(y,x)\in\mathcal{E}} \quad 2\tilde{r}(g,x,y,t) \leq 1 + s(g,y,t) - \frac{\left[\sum_{h\in\mathcal{G}}\sum_{z\in\mathcal{C}_{y,x}} s(h,z,t)\right] + \sum_{h\in\mathcal{G}\setminus\{g\}} s(h,y,t)}{G(|\mathcal{C}_{y,x}|+1)-1} \quad (13)$$

- $r \in \{0,1\}^{G,V,T+1}$: $r(g,x,t) = 1$, if node x receives the gateway announcement of g in slot t and 0 otherwise,
- $\tilde{r} \in \{0,1\}^{G,V,V,T+1}$: $\tilde{r}(g,x,y,t) = 1$, if node x receives the gateway announcement of g from node y in slot t and 0 otherwise.

Finding a successful and efficient start-up mechanism now simply translates to finding a transmitting and receiving schedule, which allows all nodes to know about the gateways within T. This is done by assigning non-zero values to s and r and thereby fixing what happens in the time slots 0 to T. $\tilde{r}$ is an auxiliary variable, which is used by the binary integer program (BIP) which allows to find an optimal schedule. It is introduced in the next section. Section Section 5 contains two algorithms for a non-optimal start-up phase.

4. THE OPTIMAL STARTING STRATEGY

In this section we introduce the BIP formulation for an optimal MWSN start-up which is depicted in Table 1. The first equation of the BIP, Eq. (5), shows the objective function f. It expresses the bi-objective goal of the start-up phase. On the one hand, all sensor nodes shall know about all gateways as fast as possible, i.e. $\sum_{g\in\mathcal{G}}\sum_{x\in\mathcal{N}}\sum_{t=1}^{T}\ k(g,x,t)$ shall be maximized. On the other hand, energy consumptions, i.e. the number of transmissions and receptions expressed by $\sum_{g\in\mathcal{G}}\sum_{x\in\mathcal{N}}\sum_{t=1}^{T}\ s(g,x,t)+r(g,x,t)$ shall be minimized. Note that if T is chosen too small, e.g. $T < G$ then the result of the BIP will be a transmission and reception schedule which can not guarantee that all nodes have learned about the gateways. The variable κ sets the value of knowledge about gateways in relation to energy expenses. $\kappa = 1$ could result in a schedule with time slots where no node transmits, as a transmission where only one new node learns from a gateway would not contribute to maximizing f. Finding a combination of T and κ which minimizes the runtime of the BIP solution would be an optimization problem of its own, we therefore postpone this problem to later works and use $\kappa = N$ and $T = GN$ in the following. This guarantees schedules which allow all nodes to learn about all gateways.

Equations (6-8) describe the initial state of the MWSN, i.e. $t = 0$: only the gateways are aware of their own presence and no node transmits. The functionality of the gateway announcement protocol is modeled by equations (9-11): At time t, a node knows from a gateway because it just receives the corresponding announcement or because it already knew from it in slot $t-1$. Any node can only forward the announcement of a gateway in slot t, if it already knew from its existence in slot $t-1$. We consider non-MIMO nodes and assume that messages are not aggregated. Hence in each time slot a node can either transmit or receive the announcement of one gateway node.

The remaining two equations describes how messages are received: Eq. (12) implies that a node receives the gateway announcement if it receives it from any of its neighbors. Eq. (13) is the most important constraint of the BIP. It expresses that a transmission on link (x,y) can only be successful if Eq. (3) does hold. This is achieved in the following way: $\tilde{r}(g,x,y,t)$ can only become 1, i.e. the gateway announcement can be successfully received over (x,y) if $s(g,x,t) = 1$, i.e. if x sends the announcement and if the fraction on the right hand side equals to zero. This does only happen if none of the nodes in $\mathcal{C}_{x,y}$ sends and if y is sending no other gateway announcement at time t. The denominator is responsible for making the entire fraction always smaller than 1 in order to prevent the right hand side of the constraint from becoming smaller than 0.

5. NON-OPTIMAL STARTING STRATEGIES

Two algorithms for the MWSN start-up for the situation where only each gateway knows about its own presence are introduced in this section. The *Smart Flooding* (SF) and the *Association Inspired* (AI) algorithm abstract mechanisms which are included in most existing routing algorithms. Therefore, they are representative examples for imaginable simple proactive and reactive start-up algorithms and allow to illustrate the inherent optimization potential of the start-up phase. More efficient start-up algorithms are the scope of our future studies.

5.1 Smart Flooding Algorithm

Each node x who knows about gateway g broadcasts the announcement of g. This message is sent in a time slot succeeding the one when x has learned about g which is chosen with a probability of b. We do not assume the presence of a MAC protocol, the variable b hence realizes a CSMA/CA-like collision avoidance mechanism. As collisions can not totally be avoided, each node transmits each announcement up to a times. Each node which has not yet learned about all gateways and which does not transmit, listens to the channel. If a node does neither transmit, listen nor receive, its transceiver is in the idle mode.

5.2 Association Inspired Algorithm

Each sensor node wishing to join a non-beacon enabled 802.15.4 PAN (which would be the technique enabling multi-hop WSNs) sends out a beacon request [11]. The PAN coordinator or each already associated node which receive this request sends the information which allows the node to join the network. Based on this principle, we implemented the AI algorithm. As long as a node does not have information about all gateways and has not yet sent a requests, it sends out a request message for one of the gateways it does not yet know of in a time slot which is chosen with a probability of b. Afterwards it listens to the channel for c slots in order to receive the answer. A node which knows about at least one gateway will also listen to the channel. As soon as it has received a request for a gateway it knows of, it transmits the corresponding answer in one of the following slots which is chosen with an increased probability of $d \cdot b$. Again, if a node does neither transmit, listen nor receive, its transceiver is in the idle mode.

6. EVALUATION

In this section we demonstrate the performance differences between the optimal and the non-optimal starting strategies. In the following we introduce our simulation and evaluation environment before we show our results.

6.1 Simulation Setup

The optimal announcement schedules are computed with lpsolve. The performance of SF and AI clearly depends on the parameterizing, we therefore analyze the performance of combinations of $a = \{1, 2, 5, 10, 20, 50\}$ together with $b = \{0.01, 0.05, 0.1, 0.2, ..., 1\}$. Additionally, we use $c = \{5, 10, 20\}$ and $d = \{1, 5\}$ for AI. The performance of each parameter combination is tested in 20 instances of 16 different scenarios. Under a *scenario* we understand a combination of node number and deployment area size which results in MWSN topologies with varying node density and hop count. On the x-axis of Figure 1 and 3 the 16 different considered deployment scenarios are depicted. The combinations of node number N and square length l are chosen out of $N = \{5, 7, 15, 25, 50, 100\}$ and $l = \{50, 75, 85, 100, 150, 100, 200\}$. In each scenario, 2 gateway nodes are placed in the lower right and upper left corner.

The initial parameter screening enables us to find out two efficient sets of parameters for each heuristic which we investigate more closely using 50 instances of topologies for each scenario. All results shown in this section are averaged over 20 or 50 topology snapshots respectively and are represented with the corresponding 95% confidence interval.

The simulations are done with an abstract MATLAB simulation framework. This enables us to examine a large range of parameters in MWSNs with different topological characteristics. To create simulation results which show the same trends as results obtained with real hardware, while still maintaining a high degree of abstraction, we use a simplified 802.15.4 channel model and an energy consumptions model corresponding nodes with the 802.15.4 compliant TI CC2420 module [13]. We detail on the energy model in the next section, the channel model is described in the following. For deciding upon the transmission success of a packet sent from x to y, we implement Eq. (3). The power received at y is computed using CC2420's maximal transmission output power of 0 dBm for T_x. The path gain is taken from the 802.15.4 channel model [11]. The current interference is calculated as the power received by y from all nodes which are transmitting at the same time as x. We use $\gamma^* = 5$ dB, a value which has been experimentally obtained by Maheshwari et al. [14]. The noise floor is set to $N_0 = -101.1$ dBm which corresponds to the thermal noise for the 2.45 GHz band of 802.15.4 plus a noise figure of 6 dBm. As we could not find a noise figure value for CC2420, we extracted it from the data sheet of the similar Atmel AT86RF230 [15].

For computing energy and time consumptions, we need to know the length of a gateway announcement. We consider such a packet to contain the gateway's full address and a hop count field. If we assume that each address has a length of 4 Byte and that the hop count is encoded within 1 Byte, this results in 5 Byte of payload. Including the 802.15.4 PHY and MAC overhead, the size of such a packet would thus be 20 Byte. The raw bit rate of CC2420 is 250 kbps, a packet is hence transmitted within 0.64 ms. For sakes of simplicity, we use the same numbers for an announcement request packet used by AI. We set the length of a slot to be 50 % longer than the packet transmission time, i.e. 0.96 ms. This choice is made to model a mechanism which avoids clock synchronization problems and considers hardware delays.

6.2 Evaluation Methodology

To compare the performance of the protocols, we evaluate each executed simulation run for the heuristics and the optimal schedule obtained by the BIP with the three following metrics: $0 \leq \rho_S \leq 1$ indicates the start-up success which we define to be the percentage of sensor nodes which have heard from all gateways. The start-up time consumption $1 \leq t_S \leq T$ give the number of slots which pass until all nodes have heard from all gateways. The start-up energy consumption E_S is computed as the energy which is consumed for the start-up averaged over all nodes.

ρ_S and t_S may be simply derived from the variables k, s, r which are found by the BIP or during the simulations. To obtain E_S, the simulation framework uses a method similar to the one used in an earlier study [12]. E_S required by one parametrization of the heuristic is estimated as the energy consumptions of the transceiver plus the 2 mA a typical sensor node MCU which is e.g. included in the Tmote Sky node [17] requires during the start-up phase. We use the state machine model proposed by Wang and Yang [16], who extract values for current consumptions from transceiver data sheets and assume a typical voltage of $V = 1.8$ V. Experiments with the CC2420 showed that this approach together with an exact model of the communication behavior closely estimates the current consumptions. As a simplification, we neglect the small energy consumptions for transitions between the states of the transceiver. To obtain E_S for the optimal schedule, we map the vectors s and r to the times spent in the different states and use the same state machine model to derive the energy consumptions.

6.3 Numerical Results

The results from the parameter screening study are depicted in Figure 1. The x-axis shows the considered scenarios, ordered first

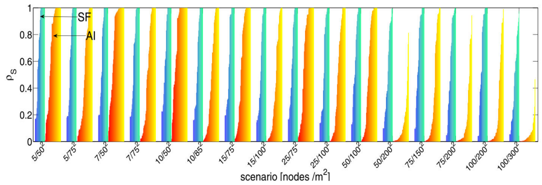
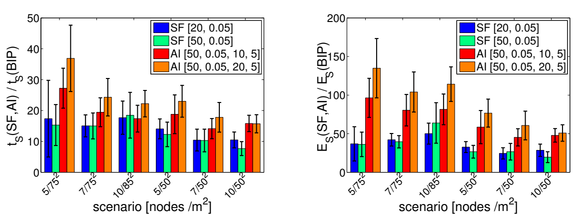


Figure 1: Start-up success of all considered parameters

by increasing node number, then by deployment area size. Hence, dense and sparse topologies alternate. For each scenario two groups of bars are shown. They represent the average starting success achieved within a time limit of $T = 1000$ slots by the considered 28 different parameter combinations for SF and the 168 different combinations for AI. To clarify the presentation we do not label the bars one by one, omit the confidence interval, and order the bars by increasing size. This allows to see that the performance of the heuristics significantly depends on the used parameters: An unlucky parameter choice for AI like $a = 50, b = 1, c = 20, d = 5$, for which we use the shorthand $[50, 1, 20, 5]$ in the following, or likewise $[50, 1]$ for SF would result in $\rho_s < 0.25$ for SF and ρ_S even be close to 0 for AI. With such a protocol, only a fraction of the nodes are able to learn about the gateways within T. Observe moreover that for the larger topologies, T is chosen too small, so that only a few parameterizations or even none for AI are successful. Clearly this is not acceptable, as any MWSN application would hence not run properly. The time and energy consumptions for this experiment show variations in a similar manner. Inefficient parameter choices do not function properly and waste time and energy. These results hence demonstrate, that without a good start-up algorithm any optimization for the subsequent operational phase is useless as it is not guaranteed that the MWSN works properly.

On the other hand, there exist parameter combinations which are able to successfully start-up the MWSN given a longer amount of time. Out of those parameters, we choose 2 parameterizations for SF and AI respectively which had the smallest time and energy consumptions and compare them to the performance of the optimal schedule computed by the BIP. With a time limit of 5 hours per network instance, lpsolve was only able to generate perfect schedules for the six smallest scenarios we considered.



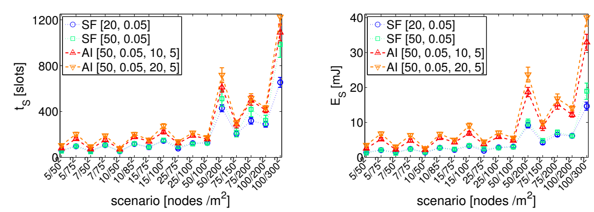
(a) Time (b) Energy

Figure 2: Relative start-up resource consumptions

A comparison between optimal and non-optimal starting strategies is therefore only possible for 10 instances of the six smallest scenarios. Results from this comparison are illustrated in Figure 2 where the scenarios are represented on the x-axis and ordered by increasing node density. The y-axis shows the time and energy consumptions required by the heuristics divided by the ones of the optimal solution, i.e. the optimization potential of the heuristics. The fact that in both sub-figures the bars show a tendency to decrease from left to right demonstrates that the relative difference is larger for sparse topologies. This effect is especially noticeable for the energy consumptions of AI shown in Figure 2(b) and visualizes that sparse topologies, which are due to high hardware costs still more likely to occur in reality, have to be started with special care. This phenomenon is also observable in Figure 1. For each first scenario out of two with the same number of nodes, which is the denser one, the bars are somewhat fatter which represents a higher share of successful start-up parameter combinations.

Figure 2 illustrates, that the parameterizations of the heuristics which we identified to be the most efficient ones, do require at least 10 times more time and 40 times more energy than the optimal solution. Less efficient parameterizations consume up to 40 times more time and over 100 times more energy. Clearly, the optimal schedule represents a real lower bound for energy consumptions. The assumption that the radio unit can go to sleep mode if it is neither receiving nor transmitting makes the optimal solution a good benchmark for the design of any start-up algorithm.

A comparison of the two types of heuristics shows that the resource consumptions of AI are even higher than those of SF. Hence, the type of algorithm used for the MWSN start-up is also important. A reactive approach like the one used by AI would be still less efficient than the proactive approach used by SF. Observe also that the differences between the two parameterizations for AI and SF respectively are for some scenarios larger than for others. The start-up performance is hence strongly topology dependent. An ideal start-up protocol will thus be not the same for all MWSN deployments, but has to be adapted to the network characteristics.



(a) Time (b) Energy

Figure 3: Absolute start-up resource consumptions

In Figure 3 we show the absolute start-up resource consumptions of the heuristics for all considered scenarios. To increase the percentage of successful start-ups in the large scenarios, we use $T = 10000$ slots. In the sparse topologies with 50, 75 and 100 nodes, the start-up success averages to values between 0.85 and 0.96, in all other scenarios, the heuristics were always able to carry out a successful network start-up. The time consumptions are shown in slots, the absolute values vary between 100 and 600 ms. Accordingly are the current consumptions far below one percent of typical sensor node battery capacities. Remind however that our numbers are the result of an idealistic simulation and would be much larger in a real-world deployment.

The zig-zag shape of the curves is again due to the fact that on the x-axis one dense scenario is followed by a sparse one where the MWSNs are less easy to start-up. Figure 3 shows moreover again that the time and especially the energy consumptions of the AI heuristic are higher than the ones of SF. It also illustrates that the performance of two parameterizations of one algorithm are rather similar. A close comparison reveals however that it is more efficient to send out more messages and to listen longer to the channel, as the increased energy consumptions are compensated by the increased message reception probability. Finally we see that the resource consumptions increase dramatically for topologies with more nodes. Together with our observation that the start-up success decreases for larger topologies and more gateways, this demonstrates again the need for optimized start-up scenarios in order to guarantee an efficient MWSN operation.

7. CONCLUSION AND OUTLOOK

In this work we introduced the MWSN start-up problem. By starting up a MWSN we define the process during which all sensor nodes get to know about all gateways. We established an analytical framework which allows to formalize the successful, fast and energy efficient start-up procedure as an optimization problem. The resulting optimal start-up schedules serve as benchmark for a perfectly efficient carried out start-up phase. To analyze the efficiency distributed algorithms, we designed two abstract, highly configurable algorithms as representative examples for simple reactive or proactive start-up approaches.

Our results demonstrate that similar algorithms have to be designed with care: Out of the wide range of parameters, representing imaginable start-up strategies, we considered, the most were not able to guarantee the prerequisite of a flawless MWSN operation, namely the successful star-up. Moreover did even the best parameterizations of the heuristics, which were able to successfully carry out the start-up, perform significantly worse than the optimal solution. This additionally demonstrates that a carelessly performed MWSN start-up may not only decrease the application performance, but also waste a large amount of resources.

Our future works will be dedicated to a further refinement of the BIP for producing lower bound results for larger topologies and to include the effects of PHY and MAC layer in the simulation framework. We intend to use metrics like the shape of the resulting multi-gateway routing topology or the performance of MWSN applications for developing better start-up algorithms. This in turn will allow an optimal MWSN operation and thereby a more seamless integration of sensor networks in the Internet.

Acknowledgment

The authors would like to thank Phuoc Tran-Gia and Rastin Pries for their helpful comments and inspiring discussions.

8. REFERENCES

[1] University of Southampton, “GlacsWeb,” http://envisense.org/glacsweb/index.html, last accessed 04/2010.

[2] J. Hart and K. Martinez, “Environmental Sensor Networks: A Revolution in the Earth System Science?” *Earth-Science Reviews*, vol. 78, July 2006.

[3] E. Oyman and C. Ersoy, “Multiple Sink Network Design Problem in Large Scale Wireless Sensor Networks,” in *ICC’04*, Paris, France, June 2004.

[4] Z. Vincze, R. Vida, and A. Vidacs, “Deploying Multiple Sinks in Multi-hop Wireless Sensor Networks,” in *ICPS’07*, Istanbul, Turkey, July 2007.

[5] A. Egorova-Förster and A. Murphy, “A Feedback-Enhanced Learning Approach for Routing in WSN,” in *WMAN’07*, Bern, Switzerland, February 2007.

[6] M. Kalantari and M. Shayman, “Design Optimization of Multi-Sink Sensor Networks by Analogy to Electrostatic Theory,” in *WCNC’06, 2006*, Las Vegas, NV, USA, April 2006.

[7] R. Mathew and M. Younis, “Energy-Efficient Bootstrapping Protocol for Sensor Networks,” in *ICWN’02*, Las Vegas, NV, USA, June 2003.

[8] A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, and K. Jones, “Training a Wireless Sensor Network,” *Mobile Networks and Applications*, vol. 10, no. 1, 2005.

[9] A. Bertossi, S. Olariu, and C. Pinotti, “Efficient Corona Training Protocols for Sensor Networks,” *Theoretical Computer Science*, vol. 402, no. 1, 2008.

[10] F. Barsi, A. Bertossi, F. Sorbelli, R. Ciotti, S. Olariu, and C. Pinotti, “Asynchronous Training in Wireless Sensor Networks,” *Lecture Notes in Computer Science*, vol. 4837, 2008.

[11] IEEE Computer Society, “802.15.4 Standard: Wireless MAC and PHY Specifications for Low-Rate WPANs,” September 2006.

[12] B. Staehle, “Optimizing the Association Procedure for Low-Power 802.15.4 Nonbeacon Sensor Networks,” in *Networking’09*, Aachen, Germany, May 2009.

[13] Texas Instruments, “2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver,” April 2006.

[14] R. Maheshwari, S. Jain, and S. Das, “A Measurement Study of Interference Modeling and Scheduling in Low-Power Wireless Networks,” in *SenSys’08*, Raleigh, NC, USA, November 2008.

[15] ATMEL, “AVR Low Power 2.4 GHz Transceiver for ZigBee, IEEE 802.15.4, 6LoWPAN, RF4CE and ISM Applications - AT86RF230,” February 2009.

[16] Q. Wang and W. Yang, “Energy Consumption Model for Power Management in Wireless Sensor Networks,” in *SECON’07*, San Diego, CA, USA, June 2007.

[17] Moteiv Corporation, “Tmote Sky: Ultra Low Power IEEE 802.15.4 Compliant Wireless Sensor Module,” February 2009.