TKN Telecommunication Networks Group

Technische Universität Berlin

Telecommunication Networks Group

Enabling Mutual Surveillance in Wireless Sensor Networks

Matthias Kühm, Andreas Willig, Adam Wolisz

{kuehm,willig,wolisz}@tkn.tu-berlin.de

Berlin, October 2009

TKN Technical Report TKN-09-007

TKN Technical Reports Series

Editor: Prof. Dr.-Ing. Adam Wolisz

Abstract

Keeping a group of persons or items geographically close together can be challenging task. In this report we consider the continuous self-surveillance of wireless sensor network (WSN) tags attached to individual group members. In such a *herding system* the tags check whether a given group is "together", i.e. whether the group members are close to each other. We introduce our specific framework to the design of future herding system in which the sensor nodes transmit regular beacons, and on this basis *cooperatively* check for the presence or absence of individual group members. To enable cooperation the tags are required to form monitoring groups to reliably monitor each group member over time. Especially in case of dense networks an individual tag is not able to monitor *all* of its neighbors but has to restrict to a subset. We present and evaluate a truly distributed approach to the problem of forming such monitoring groups called *Distributed Randomized Selection* (DRS). This approach is well able to adapt to different network densities, ensures that all tags are monitored by a sufficient number of neighbors and requires no extra control packets besides the regular beacons.

Contents

Chapter 1

Introduction

In real life it is often necessary to ensure that a group of persons or items remains geographically close together whereas individual members or the group as a whole are moving. Just think of a visitor group on a sightseeing tour or a shipment consisting of many different parts. While it seems natural to verify the *completeness* of a given group from time to time by comparing against a full list of its members or even plain counting it happens be a time-consuming and for humans often annoying task. As a result, the time intervals between consecutive checks are likely to become large and subsequently the time until the loss of a group entity is detected. From a technical point of view Radio Frequency Identification (RFID) offers a possibility to simplify the verification procedure but in general all items need to be captured separately at very short distances by a scanner device. This will still take plenty of time and thus happen rather rarely. Specialized solutions using a fixed scanner infrastructure on the other hand bound the togetherness checks to a pre-defined geographical area [1, 2].
To overcome these shortcomings we aim for a simple and low cost solution based on the technology of Wireless Sensor Networks (WSN) [3] supporting *continuous* monitoring of a group over a longer period of time. We envision that each member of a given group is tagged with a WSN node and each tag announces its presence by periodic transmission of *beacon* packets. The nodes observe each other's beacons and commonly decide on the presence of a particular node in spite of mobility and wireless link fluctuations. The continuous *self-surveillance* of the WSN nodes greatly reduces the effort for checking the togetherness of a group. The group is considered together as long as no single member is identified as missing, which can be verified quickly by querying the network for currently missing nodes. It is also conceivable to enable immediate notifications in case of a missing group member [1]. For the problem of checking the togetherness of a group with the help of WSN nodes we use the term *herding*. In preliminary work [4] we have shown that based on periodic beacons a fairly small group of observer nodes (we have used seven in [4]) can achieve very reliable judgements about the presence or absence of a node. The observers receiving the beacons work together in deciding whether or not the observed node is in proximity of at least some of them. This *cooperation* requires the exchange of information via radio packets which can also be provided by piggy-backing the information on the regular beacons. The results are valuable especially when looking at small groups (of nodes) in single-hop scenarios but provide an elementary step towards the development of a general framework for a herding system.

[1] Ideally, notifications are forwarded to a dedicated (leader) node for further processing. This is, however, beyond the scope of this report.

The usage of wireless sensor network technology provides several benefits: The nodes combine computational, communication, storage and sensing capabilities[2] in a single device. They can be built with small form factors, facilitating their usage as small tags attached to items or persons. Sensor nodes can be very cheap, allowing to equip larger groups at moderate costs. However, there are also drawbacks. For example, the limited energy budget available to a node mandates the usage of small transmit powers (typically in the range of $0\ \mathrm{dBm}$), which in turn translates into a limited communication range and therefore a bounded geographical extension of the group. Multi-hop communication can be used to overcome this limitation but again drains energy for packet forwarding and should therefore be avoided. Secondly, time-varying and sometimes rather poor link quality (amplified by the choice of cheap transceivers for WSN node designs) call for sophisticated adjacency measures that exploit cooperation of nearby nodes. However, tight limits on packet sizes (see for example the IEEE 802.15.4 standard [5] with its maximum packet size of 127 bytes) provide a natural limit on the amount of information that can be transmitted using a single radio packet.
Given the limiting constraints of WSN technology and some non-trivial group size there is a clear recognition that the participation of a single WSN node in the observation of other nodes is limited to some extend. From the results in [4] we can on the other hand conclude that a small number of observers per node is sufficient. Every node in the group should therefore only be monitored by a subset of nearby nodes that only directs the attention of the whole group to the verification of the node's current state if they jointly decide the node has left their proximity. However, it is not clear in advance how such a subgroup can be arranged and maintained over time in spite of mobility and varying wireless link properties. As the main contribution of this report we present a truly distributed approach to the problem of creating such subgroups, in which each node makes an autonomous decision which other nodes it is going to monitor. The *Distributed Neighbor Selection* (DRS) approach does not need extra control packets besides regular beacons and is applicable independent of group size and density.
In the next Chapter 2 we define our system model and present the specific framework of a herding system that we consider for this work. Chapter 3 presents the DRS approach and in Chapter 4 we describe the NS-2 based simulation setup that we have used for evalaution and discuss our results. Chapter 5 presents related work and Chapter 6 finally concludes this report.

[2]The availability of sensor information like temperature, humidity and light intensity can be a valuable addition to a herding system that allows to monitor also the environmental conditions over time for each group item.

Chapter 2

System Model and Problem Formulation

In the following we assume a group of N uniquely identifiable WSN devices, called *entities*. These devices are attached to individual group items which should remain in a geographic proximity to each other and the group of entities constitutes our *herding system*. On the physical layer we consider a half-duplex transceiver in combination with an omnidirectional antenna. The transmission power of the transceiver is assumed to be adjustable to provide some rough tuning of the transmission range. On layer two we do not consider a specific scheme to coordinate the medium access of the entities. However, a carrier sense multiple access protocol (CSMA) is reasonable to reduce collisions to some extend. Each entity periodically broadcasts beacon packets with a common inter-beacon spacing T. There are no acknowledgements and no retransmissions. Furthermore, we do not require the entities to be highly synchronized in the time domain thus the effective beacon interval interval per entity is expected to be $T \pm \epsilon$ whereas $\epsilon \ll T$. The beacons are of fixed size and consist of a header field carrying the unique identification of each entity and a payload field that can be used to share collected monitoring information with adjacent entities.
The problem of monitoring the togetherness of a given group is best formulated in terms of the undirected connectivity graph $G = (V, E)$ imposed by the attached entities. The set of vertices V contains all the entities and the edges in E represent the (symmetric) radio connectivity of the entities according to some link quality measure. The graph is *connected* if there is a path from entity u to entity v for all entities $u, v \in V$. The graph is *fully* connected if there exists a link (edge) from entity u to entity v for all entities $u, v \in V$. A single entity w is connected to a subgraph $G' = (V', E')$ of G, with $V' \subseteq V$ and $E' \subseteq E$, if there exists an edge from w to at least on of the entities in V'. For the togetherness of a given group we require the connectivity graph to remain connected over time. In this sense, a violation occurs if at least one entity is disconnected from the group[1]. The task of a herding system is to reliably and immediately detect disconnected entities[2] and to make this information available to the outside world[3]. However, continuously monitoring the connectivity graph whose edges will be time varying due to mobility and fading is not a trivial task by itself. Even if the graph would be stable it is difficult to state how the graph is maintained, where it is to be stored and who is going to do the connectivity analysis. Therefore, we aim for a decentralised solution that

[1] The problem is in general more complex if, for example, the group splits into several components. For now, we are only interested in detecting a single entity that is disconnected from the remaining group.

[2] In fact, we even would like to detect situations in which a disconnection will happen with high probability. But this is beyond the scope of this report.

[3] For simplicity we just assume that disconnections are recorded locally and can be queried if required.

does not explicitly try to construct and analyse the connectivity graph, but will detect situations (with high probability) in which the graph would be disconnected.
In the next Section 2.1 we look back at preliminary work that analyses the joint decisions made by a fixed observer group regarding the presence of a single entity. This work provides the motivation for our specific approach to the herding problem that is presented in Section 2.2. Instead of deciding about the presence of individual links from a global point of view, we want the entities to cooperate locally in order to make a joint decision whether a given entity is connected to them. If this is true for all entities there is a strong indication that the group is together.

2.1 Cooperative Classification

In [4] we have presented a cooperative classification scheme that aims to classify the distance of a given entity i to a small group of monitoring entities. The classification is done into one of few pre-defined distance classes and a possible interpretation of such classes could be ”near”, ”far”, ”away” or ”i is present but exact classification failed” (”unknown”). Such a classification can be used to make a decision whether a given entity i is connected to a group of entities and is based on the periodic transmission of beacon packets by i. Please note that this is different from ranging where the geographical distance is measured with high precision.
The basic idea of the scheme can be described as follows: An entity j uses its own observations about entity i and possibly also observations that other entities have about i to assign to i one of the pre-defined distance classes. The classification is done individually by j based on all available information. Therefore, if the entities j and k share the same information about i they will assign the same distance class to i. The classification is done in three steps:

(1) In the reception step j uses information obtained from received beacons and also their absence (due to the assumed periodicity) to continuously update local statistics about i. The available information is preprocessed in order smooth out the expected noise in the observations because of channel fading.

(2) In the classification step j tries to classify the distance to i into one of the distance classes based on the local statistics about i. Entity i is considered lost if no beacon is received for a given pre-determined amount of time. If no classification is possible based on the available information the result is undecided.

(3) In the cooperation step j uses either the classification results or the observations of other entities regarding i in order to create a refined classification. A classification based on the exchange of local observations or intermediate results is called *soft-decision cooperation* and a classification based on the exchange of local classification results *hard-decision cooperation*.

We have applied this scheme in an experimental study using an IEEE 802.15.4-compliant physical layer and a specific setting [4]. For the reception step we consider *received signal strength indication* (RSSI), *link quality indication* (LQI) and beacon/packet reception rate (PRR). The observables are filtered using an exponential moving average and used as the input vector for an *artificial neural network* (ANN) that is considered for the classification step. The dimension of the output vector is defined by the number distinct distance classes and contains values from the interval $[0, 1]$. The classification results for each entity are then based on the output vector of the ANN. For the refinement

of the classifications in the cooperation step we either consider the classification results of all entities (hard decision) or the respective output vectors (soft-decision).
The results for this setup show that a (cooperative) classification into few distance classes is indeed possible with high quality. However, the size and the specific selection of the distance set have a big influence on the quality of the classification. The best results are achieved for only two distances which have a relatively large separation in their average LQI and RSSI values and, perhaps even more important, the overlap in the histograms of the (raw) values is minimal. This is especially the case when the difference in the selected distances is large. It is therefore possible to do a classification into ”near” and ”far” which in combination with ”away” provides a reasonable basis to decide about the connectivity of an entity to a set of observers. Compared to individual classification, cooperation clearly corrects entities when their classification result is wrongly ”away”. This is an important result as it indicates that cooperation can reduce the rate of decisions in which an entity is wrongly considered disconnected. The hard-decision cooperation scheme appears to perform slightly better than soft-decision cooperation scheme.

2.2 Herding Framework

From our perspective the cooperative classification scheme provides the basis upon which a reliable herding system should be built. For scenarios with only a small number of entities that form a fully connected communication graph it is reasonable to apply such a scheme directly and let the entities mutually monitor each other. That means each entity has complete knowledge on what other entities belong to the group and continuously monitors their connectivity to the group. However, for a growing number of entities scalability becomes a serious problem for several reasons: First of all, the monitoring capabilities of a single entity are limited to some extend due to the hardware limitations of WSN devices (memory, computation). Secondly, a cooperative classification scheme requires the exchange of information using radio packets. With a growing number of entities the amount of information is also growing. This is costly in terms of energy consumption for packet transmissions and clearly limited because of typically tight bounds on packet sizes. These limitation can hardly be expressed in numbers and therefore we aim for a design that from the very beginning puts a fixed upper bound on the workload for each entity independent of the group size. Even if we neglect the scalability concerns we can not always assume for large groups that the communication graph is fully connected. Thus there is an inherent need to distribute the herding task among the entities without requiring the mutual surveillance of all entities.
The specific framework of a herding system that is considered in this work tries to incorporate the cooperative classification scheme but takes also the scalability issues into account. In general we consider a rather small number of observers in the local neighborhood of each entity and also a limitation on the participation in these groups for each entity. The idea is to continuously check for the connectivity of the entities from this local point of view only based on the transmission of periodic beacons. Just in case of a detected disconnection from an observer group a verification that involves all group entities is performed. We consider three major building blocks for our framework:

(1) *Formation:* For each entity i of the group a small group of adjacent entities (single-hop) is formed that is responsible for the local surveillance of i. We refer to this group as i's Circle of Friends, short $CoF(i)$. The configuration of $CoF(i)$ may change over time to account to mobility and connectivity changes in the group but its size should not fall below a given minimum size to

ensure that i can be reliably monitored. The participation of each entity in monitoring other entities is clearly limited to put an upper bound on the expected workload.

(2) *Surveillance:* The members of $CoF(i)$ continuously monitor the presence (connectivity) of entity i in their local proximity. This done by applying a cooperative classification scheme. The classification results are also be used to adapt the configuration of $CoF(i)$ in case of mobility.

(3) *Verification:* If from their local point of view the members of $CoF(i)$ decide that entity i is no longer connected to them the attention of the whole group is directed to the connectivity of i. If the connectivity of entity i can not be verified by the whole group i is considered disconnected. If entity i is still connected to the group $CoF(i)$ needs to be reformed to account to i's new location within the group.

The first two building blocks are solely based on the periodic transmission of beacon packets. All required information is transmitted in the payload of the beacons. The last block requires network-wide communication, as a last resort by using a flooding approach, and should therefore only be applied infrequently (especially in case of a connected communication graph).
For the success and the efficiency of the whole approach the proper formation and adaptive refinement of observer groups is essential. For initialization it is reasonable to assume that the group is rather static and geographically close together so that the entities form a connected communication graph. Just think of goods waiting for their shipment or persons waiting at a meeting point. When an entity is activated it only knows its own identification and starts to broadcast beacon packets while listening to the beacons from other entities to discover its neighborhood. For the formation of CoF configurations only the best neighbors, according to some pre-defined criteria considering link quality or existing configurations, should be selected. This can be formulated in terms of a *join rule*. For balancing the configurations it is also reasonable to exchange an already selected entity by a previously unselected entity to ensure a minimum CoF size for all adjacent entities (*exchange rule*). If later on an entity i is subjected to mobility but remains within the group it must also be possible for the members of $CoF(i)$ to revoke their selection of i. Such a *revoke rule* must nevertheless ensure that i is still monitored by the group and not accidentally lost.

2.3 Scope and Performance Metrics

In this report we investigate the first building block of the herding framework, the formation of CoF configurations for each entity of a given group. We will not consider possible implications of the other blocks and only look at the initialization of a static herding system. We present the truly distributed DRS approach that specifies a join and an exchange rule and analyse its performance by simulation.
The major measure of performance we want to investigate in this context is the probability of forming valid CoF configurations for all entities in a given period of time starting from a clean state with no existing selections. A CoF configuration for an entity i is valid if i is selected for monitoring by a minimum number of adjacent entities. For the DRS approach this minimum number is explicitly given as a parameter. We call an entity with at least the minimum number of monitoring entities *covered* and thus we investigate the coverage probability. For a complete picture it is also reasonable to look at the number operations that are required to establish the desired coverage and result from the join and exchange rules.

For the evaluation we want to consider two causes for performance loss and provide a *ground truth* for comparison. First of all, we define an upper bound on the workload for each entity by giving a maximum number of CoF configurations each entity can participate. If we encounter a high probability of invalid configurations we need to make sure that this is really caused by our approach and not already generated by the (random) entity deployment. Thus it is necessary to also look at the case without this limitation to determine the best possible performance for a given deployment. Another cause for performance loss can be the estimation of CoF sizes. As we consider wireless transmissions (beacons) to distribute information on CoF configurations it is reasonable to expect some error in the estimation of the size $CoF(i)$ made by entity i. To see the effect of the error on the performance it is necessary to also consider the case without error.

Chapter 3

Distributed Randomized Selection

In this chapter we present a truly distributed approach to the problem of forming CoF configurations for a given group of entities. The basic idea of this approach is that an entity j independently selects neighbors to participate in their CoF configurations based on the reception of periodic beacons and the information derived from the beacons. In the next Section 3.1 we start by introducing basic mechanisms that are relevant for the description of the selection rules and operations presented in Section 3.2. The final Section 3.3 is devoted to an important aspect of the approach: the usage of a *selection probability*.

3.1 Basic Mechanisms

From our system model we know that an entity j transmits beacons with a common beacon period T. Suppose that entity j has selected a set of adjacent entities including entity i, thus $j \in CoF(i)$. In its beacons entity j includes the following information:

(i) its own identification;

(ii) its own estimate $K(j)$ (see below); and

(iii) for each selected entity i it includes i's identification.

Entity j therefore not only indicates its presence using the beacon packets, it also distributes information on its selections. Please note that any technology-dependent maximum on the allowable packet size s_{max} puts a limit on the number of entities that can be selected (monitored) by j.
Besides transmission of own beacons entity j also listens to the beacons from other entities. From the beacons entity j extracts two different kinds of information. First, it estimates the number of its friends (i.e. which have j included in their list of selected entities). In-between sending own beacons entity j counts the beacons from entities $k \in CoF(j)$. Based on this counter the size of $CoF(j)$, denoted as $K(j)$, is maintained as an exponentially weighted moving average of the form

$$\bar{x}_n = \alpha \cdot x_n + (1 - \alpha) \cdot \bar{x}_{n-1} \tag{3.1}$$

where $\bar{x}_n$ represents the new estimate of $K(j)$, x_n the new counter value after n, $n > 0$, beacon periods and $\bar{x}_{n-1}$ the old estimate. The weight α is a tuning parameter that allows a tradeoff between stability and agility of the estimate.

The second information that is taken from received beacons is $K(i)$, the number of friends for a given entity i. This can directly be read from i's beacons and is used to decide whether i is in need of a friend.

3.2 Selection Rules and Operations

The DSR approach specifies a join and exchange rule for the formation of CoF configurations. A revoke rule as motivated in 2.2 is postponed to future work. To better explain the instantiation of the rules, we first introduce two different threshold values:

- The number C_{max} denotes the maximum number of entities that can be selected by one entity. This threshold accounts to the upper bound of the workload for each entity.
- The number K_{min} denotes the minimum required number of entities to cover entity i, i.e. the minimum required size of its $CoF(i)$. With at least K_{min} entities in $CoF(i)$, we consider i as sufficiently covered.

We denote by $C(j)$ the number of selections of entity j. Now suppose that entity j receives a beacon from entity i and j has not already selected i. In this moment entity j has to decide whether it will select i or not (we say: if it declares itself a *friend* of i or not). This is done by using either the join or exchange rule depending on $C(j)$:

- Join rule: When $C(j) < C_{max}$ then entity j performs an independent Bernoulli experiment with success probability p (we call this the *selection probability*, see below). If the experiment is successful, entity j becomes a friend of i.
- Exchange rule: When $C(j) = C_{max}$ then entity i is selected as a friend when all of the following conditions hold: (i) i's own estimate $K(i)$ of $|CoF(i)|$ is smaller than K_{min} (i.e. i is not sufficiently covered); (ii) entity j has selected another entity k with $K(k) > K_{min}$; and (iii) the result of an independent Bernoulli experiment with success probability p is positive.

The specified rules have some intended properties: they do not bound the number of entities that can select an entity i and they also push entities to participate in the CoF configurations as long as there is capacity for further selections ($C(j) < C_{max}$) or an adjacent entity i is not sufficiently covered ($K(i) < K_{min}$). In the following we will denote the application of the join and exchange rule by an entity j as join and exchange operation, respectively. For the exchange operation we consider entity k with the largest value $K(k)$ for exchange if more than one candidate exists. Please note that both operations are carried out independently by individual entities and do not create extra packets. It should also be mentioned that an entity j can apply the operations whenever it receives a beacon from a previously unselected entity i. Therefore, they are not limited to the initialization of a herding system.

3.3 Selection Probability

In the previous section we have considered the selection probability p that an entity j shall use in join and exchange operations. The rationale for this is to provide a simple mechanism to avoid

oscillations: when a new entity i has just been switched on, its estimate $K(i)$ of $|CoF(i)|$ is zero. Without the down-sampling of the selections provided by a selection probability smaller than one, all neighbors of i could decide at the same time to select entity i, which in one step can create an exorbitant coverage of i but at the same time an insufficient coverage of another entity k that has been selected so far. With the down-sampling of the join and exchange operations $K(i)$ increases more smoothly and there is enough time to obtain feedback from k's and i's beacons, avoiding a heavily unbalanced entity coverage.
For this study we consider the selection probability to be either *static* or *adaptive*. When static, all entities use a common (and constant) value of p. In case of an adaptive selection probability an entity j will start with a common value of p and adapt the selection probability to the ratio of the maximum selectable number of neighbors C_{max} and the total number of distinct neighbors M_j. Thus, the adapted selection probability $\tilde{p_j}$ for entity j is given as follows:

$$\tilde{p_j} = \begin{cases} \frac{C_{max}}{M_j}, & M_j > C_{max} \\ 1, & \text{otherwise} \end{cases} \tag{3.2}$$

The number of neighbors is not known in advance but can be estimated by an entity j by counting the distinct senders of beacons received in-between the transmission of own beacons. To account for fluctuations in the estimate the moving average from Equation 3.1 is used.

Chapter 4

Performance Evaluation

We evaluate the performance of the DRS approach using the *Network Simulator 2 (NS-2)* [6]. N2-2 is widely used in research and supports several popular network protocols for wired and also wireless network architectures. The focus of our investigation is on the coverage probability. More precisely, we want to show that for an arbitrary and static entity deployment a minimum number of friends for each entity can be established when using DRS. This is especially relevant for the initialization of a herding system.
We start by describing our simulation setup and specify the fixed DRS parameters in Section 4.1. In Section 4.2 we give the ground truth for comparison and in Section 4.3 we finally discuss the results of our study

4.1 Simulation Setup

We consider a system area of $30 \times 30\ \mathrm{m}^2$ for our simulations. The number of wireless entities is Poisson distributed with parameter (mean) $\lambda = 100$. The positions of the entities are uniformly distributed over the whole system area. For each simulation run we first generate the number of entities and then the position for each entity to define a static entity deployment. The entities are activated at random times within the first second of simulated time and immediately start to transmit and receive beacons.
For the entities we use the implementation of the IEEE 802.15.4 (LR-WPAN) standard [7] available in NS-2 to operate the physical and the MAC layer. We consider a receive and carrier-sense sensitivity threshold of $-90\,\mathrm{dBm}$. When operating in the nonbeaconed mode the LR-WPAN standard defines a CSMA-CA scheme for comcurrent medium access. Beacons are transmitted every $T = 1\,\mathrm{s}$. The simulated packet (beacon) size is set to $80\,\mathrm{Bytes}$ thus being compliant with the limitations of the standard. The Tmote Sky sensor node platform [8] is compliant to the IEEE 802.15.4 standard and one of the platforms used at TKN for experimental research.
For the radio propagation we exploit the shadowing model available in NS-2 [9, Chap. 18.3] that implements a log-normal fading. More specifically, for each transmitter-receiver pair and each beacon a new shadowing coefficient is generated for the respective link. We assume an unobstructed outdoor environment with path loss exponent $\beta = 2$ and a shadowing deviation $\sigma_{\mathrm{dB}} = 4$. In Figure 4.1 the packet yields for different transmit power levels and distances up to $100\ \mathrm{m}$ are given. The results are taken from a single transmitter scenario with 5000 transmitted beacons. For performance evaluation

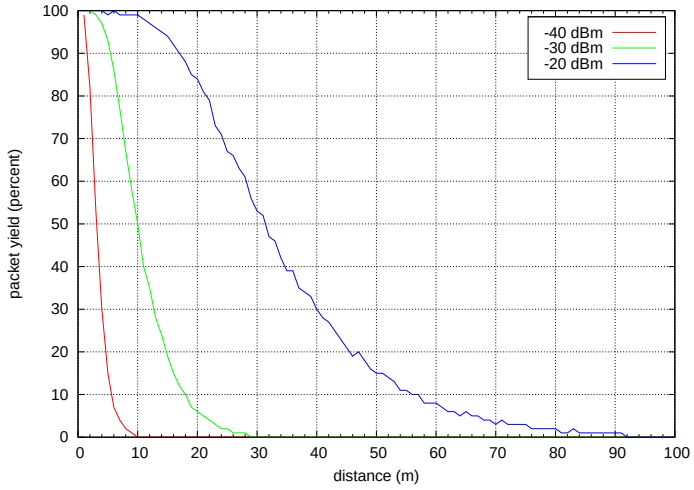


Figure 4.1: Packet yield for different transmit power levels

we will look at high and low entity densities by setting the entities' transmit power to $-20\,\mathrm{dBm}$ and $-40\,\mathrm{dBm}$, respectively.
To evaluate the performance of DRS as described in Chapter 3 we consider the fixed parameters $K_{min} = 5$, $C_{max} = 10$ and $\alpha = 0.25$ and vary the selection probability scheme. We will apply the static selection probability with $p = 1.0$ and $p = 0.25$ as well as the adaptive selection probability. The static case with $p = 1.0$ is a special case without using the down-sampling of selections at all because the result of the independent Bernoulli experiment is always positive.

4.2 Ground Truth

For comparison we establish a ground truth for the pure DRS (DRS-pure) approach in three different ways: First of all, we do not limit C_{max} thus consider $C_{max} = \infty$. This will be denoted as *DRS-nolimit* and allows us to identifiy effects of the random deployments. If it is not possible to establish a sufficient coverage for all entities using DRS-nolimit it will also not be possible for the pure DRS because there are simply not enough entities to establish K_{min} friends for each entity. The reverse is, of course, not true.
The second ground truth, denoted as *DRS-direct*, uses the fixed value of C_{max} as for the pure DRS but does not use the estimation of CoF sizes based on the received beacons. The value of $K(j)$ for an entity j that is used in the exchange rule is calculated by looking directly at the current selections made by all entities. In this way the possibly errorneous estimates of the CoF sizes transmitted in the beacons are replaced by the real number of friends for each entity. This shows the effect of the error in the estimation of $K(j)$ made by each entity j.
Besides looking at the best possible performance of DRS we also want to establish a third ground truth for the lower bound of the performance. The problem of forming CoF configuations is in some way related to the problem of building up a neighborhood table. In [10] several approaches to the maintenance of *good* neighbors are investigated and insertion, eviction and reinforcement policies are

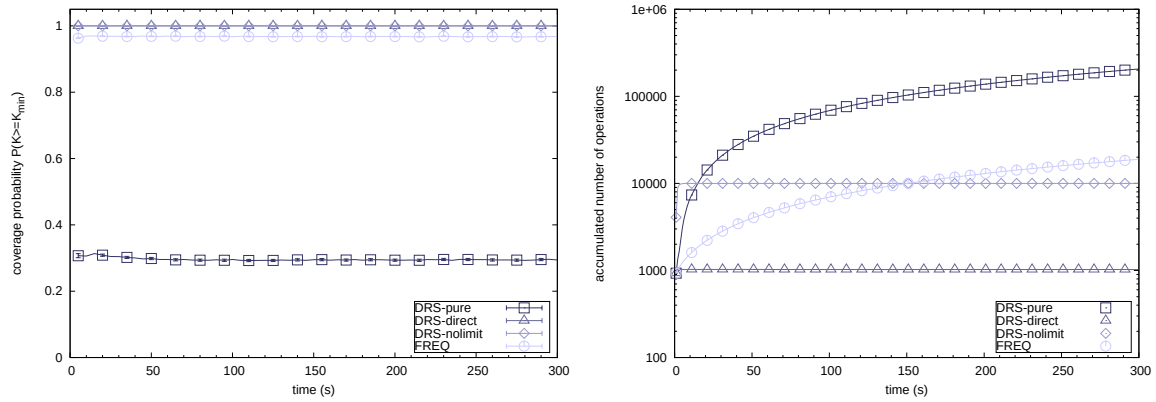
given. The insertion policy is comparable to the join rule of DRS, the proposed solution in [10] uses the same adaptive down-sampling scheme as given in Section 3.3. For eviction and reinforcement several approaches are discussed. We will consider a slightly modified version of the FREQUENCY algorithm investigated in [10]. On reception of a beacon an existing table entry for the sender is reinforced by setting a *staleness* counter to zero. After each beacon period (just before sending the next beacon) the staleness counters for all table entries are incremented. A new entity is inserted into the table (selected) if there is free capacity or there is an entity with a counter value greater than 1. In the latter case the first entry found in the table is evicted. This policy defines an exchange rule that does not take knowledge on current CoF sizes into account and therefore does not enforce the minimum number of friends K_{min} for each entity as DRS. The question is, how much better in terms of coverage DRS performs compared to this neighborhood table management if we assume the same maximum number of selections given by $C_{max} = 10$. With regard to the FREQUENCY algorithm we denote this ground truth as *FREQ* in the following.

4.3 Coverage Probability

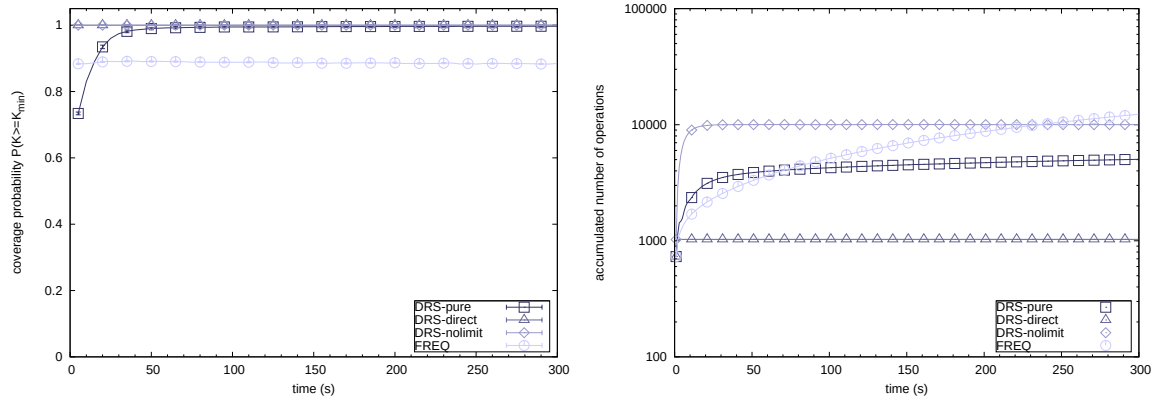
For the evaluation of the coverage probability we consider 1024 simulation runs of length $300\,\mathrm{s}$ simulated time for each setup. In steps of $5\,\mathrm{s}$ we count the number of entities that have been selected by at least K_{min} other entities, thus $K \geq K_{min}$. Compared to the total number of entities N for a run we get the percentage of sufficiently covered entities at time t_l, $l \in \{1, .., 60\}$. Averaged over all runs we get the probability that at time t_l an arbitrary entity of the deployment is covered by a CoF of size K_{min} or larger. For better understanding of the results we also give the accumulated number of join and exchange operations over time that is averaged over all runs for a setup.

High Entity Density

We first look at the case of static selection probability schemes and high entity densities. With this setup the entities can basically transmit beacons across the whole system area. Thus each entity will receive beacons from more than C_{max} entities and has to select (at most) C_{max} to become their friend. On the left side of Figure 4.2 the progress of the coverage probability over time is given for the different static selection probability schemes. The highlighted points show the 95 percent confidence intervals for selected points in time. On the right side of Figure 4.2 the accumulated number of operations over time using a logarithmic scale on the y-axis is given. The arrangement of the results will be the same for the rest of this Chapter.
For the static case with $p = 1.0$ in Figure 4.2(a) DRS performs very poor with regard to the coverage probability. The value resides around 0.3 over the whole investigated period of time. This means that more than two third of the entities are not sufficiently covered. In contrast to that DRS-direct and DRS-nolimit perform very well with a coverage probability close or equal to one. Also FREQ, not even enforcing a high coverage probability, performs clearly better than DRS. So, how to interpret these results? The performance of DRS-nolimit shows that it is well possible to sufficiently cover all entities for this scenario. The performance of DRS-direct shows that this does not require unlimited selection capacity but can also be achieved by minimizing the error in the estimation of CoF sizes for each entity. It seems that without the down-sampling of the operations the CoF size estimates become invalid very quickly and at the same time the entities exchange their selections to provide a sufficient coverage for adjacent entities. As a result the selections change very rapidly without any



(a) p=1.0

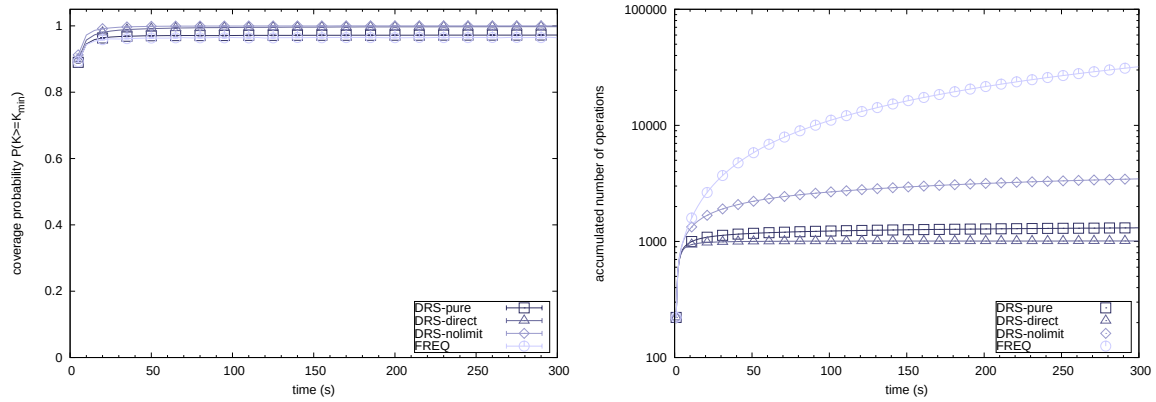


(b) p=0.25

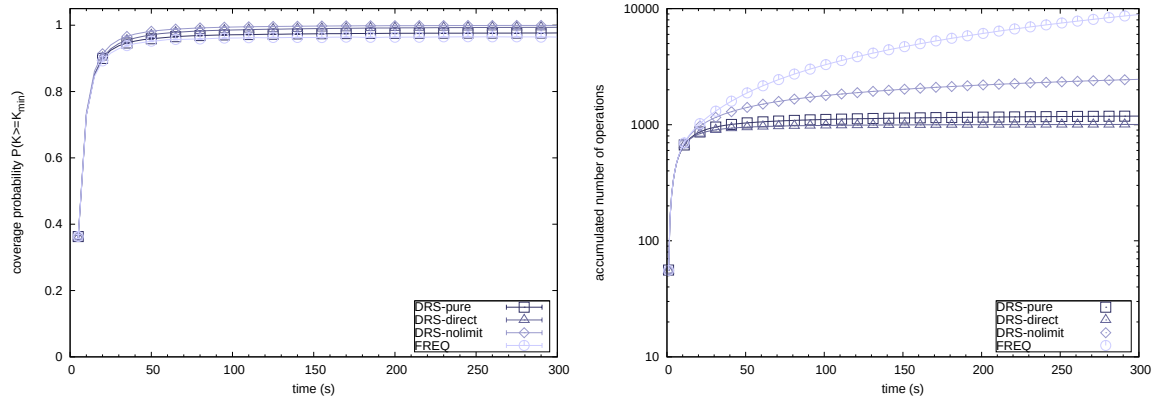
Figure 4.2: High entity density, static selection probability

convergence. This becomes clear if if we look at the accumulated number of operations. Above 1000 (except for DRS-nolimit) the operations are basically exchange operations and for DRS-pure we see a linear increase over time (recall the logarithmic scale) that is way beyond the total numbers for DRS-nolimit, DRS-direct and also FREQ. For DRS-direct and DRS-nolimit a convergence is reached very quickly and the total numbers reside around 1000 and 10000, respectively. This is a result of the setup with an average number of 100 entities per run and, for DRS-direct, a limit on the selections of $C_{max} = 10$. For FREQ we also see a linear increase over time and thus although providing a high coverage probability the configurations change rather rapidly. This is clearly not a desired behaviour for monitoring groups.

The situation changes if we look at the results for the static selection probability with $p = 0.25$ in Figure 4.2(b). For this scheme a down-sampling in the selections is enabled. For DRS-direct and DRS-nolimit the coverage probability is again close or equal to one from the very beginning. For pure DRS it takes roughly $60\,\mathrm{s}$ to also get there. In spite of the delay it is clearly an improvement compared to the static case with $p = 1.0$. With a selection probability smaller than one DRS is well able to properly form CoF configurations in high density deployments and now performs clearly



(a) p=1.0



(b) p=0.25

Figure 4.3: Low entity density, static selection probability

better than FREQ. Compared to DRS-direct it takes some time for each entity to correctly estimate the number of its friends and thus we also see a large number of exchanges at the beginning. But compared to the previous case there is no linear increase over time and the selections are rather stable.

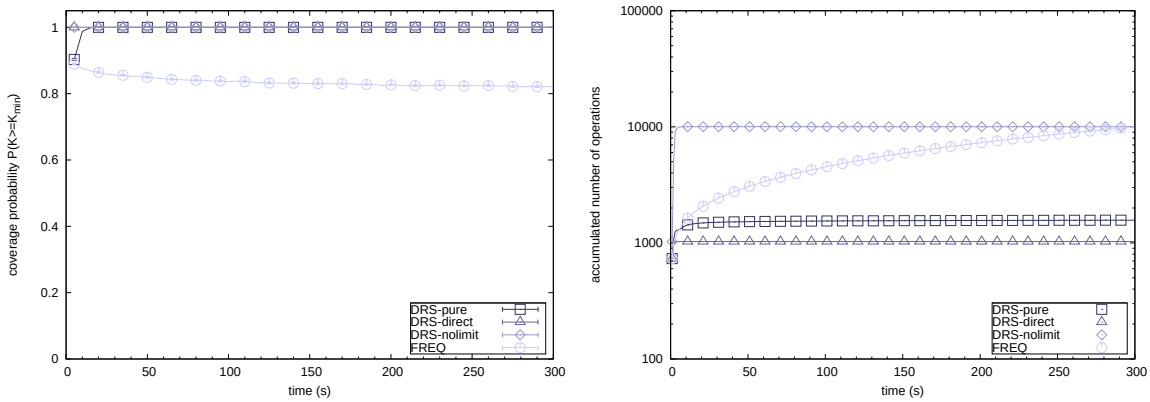
Low Entity Density

In contrast to high entity densities we can only expect few possible selections for each entity when looking at low entity densities. In this case there is no pressing need for a down-sampling of the join and exchange operations. Figure 4.3 shows again the results for the setups considered in the previous section and now the performance of DRS is comparable to DRS-direct and DRS-nolimit independent of the value for the static selection probability. DRS is also slightly better than FREQ but reaches only a coverage probability of at most 0.98 whereas DRS-direct and DRS-nolimit again are equal to one. The small gap to a coverage probability of one for DRS can be explained with the imminent under-estimation of K values which makes it necessary to select an entity more than K_{min} times to get an estimate equal to K_{min}. With entities selected by more entities than actually needed it becomes

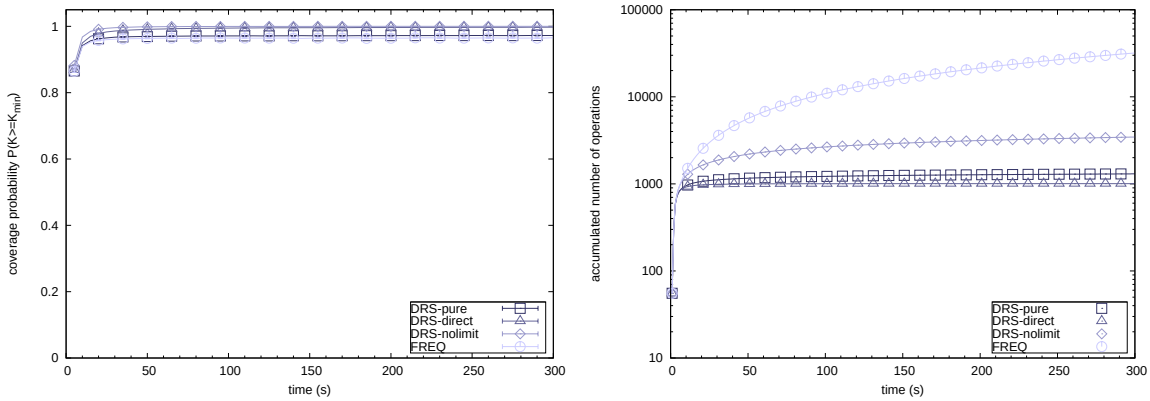
more likely that for some entities there is not enough capacity available to have a sufficient number of friends. In case of $p = 0.25$ as depicted in Figure 4.3(b) we can see that forming CoF configurations takes longer than for the case of $p = 1.0$ in Figure 4.3(a). Thus in contrast to high entity densities, a small selection probability is not preferable because it only delays the formation.
If we look at the accumulated number operations in Figure 4.3 we see that DRS is now very close to DRS-direct independent of the value of p. The fact that the number of operations stabilizes at a value close to 1000 indicates that the selection capacity of the entities is fully utilized even for this low density setup. For DRS-nolimit the accumulated number of operations is smaller than for high entity densities simply because there are fewer entities for selection. For FREQ the number of operations again rises linearly over time. This indicates that over time beacons from more then C_{max} distinct entities are received and old selections are subsequently exchanged.

Adaptive Selection Probability

From the above results it becomes clear that there are contradictory requirements for the choice of a fixed selection probability when considering different entity densities. The adaptive selection probability scheme given in Section 3.3 accounts to exactly this problem.
If we look at the coverage probability for low and high entity densities and the adaptive selection probability scheme in Figure 4.4 we can see that DRS performs comparably well independent of the entity density. For low densities we see again a small gap to a coverage probability achieved by DRS-direct and DRS-nolimit as for the static scheme. Compared to FREQ the performance is always better, albeit only slightly for low densities. The time needed to reach the final coverage probability for DRS is for low densities now comparable to the static case of $p = 1.0$ in Figure 4.3(a). At the same time we can reach a coverage probability of one for high entity densities as for the static case of $p = 0.25$ in Figure 4.2(b). The accumulated number of operations is close the DRS-direct for both low and high entity densities. Thus there are only few exchanges indicating a smooth formation of CoF configurations.
From the results we can conclude that is well possible to establish a minimum number of friends for each entity of a given group independent of the entity density. The choice of the selection probability. however, is critical with regard to the actual density. Using the adaptive selection probability scheme avoids this problem and shows good performance in terms of coverage probability for all low and high entity densities.



(a) High entity density



(b) Low entity density

Figure 4.4: Adaptive selection probability

Chapter 5

Related Work

In this report we have outlined a framework for a future herding system based on WSN technology. Several existing works also consider the goal of monitoring groups by using radio technologies. In the "virtual fence" system [11] GPS receivers are used to compare the coordinates of a cow with those of a virtual paddock defined by a set of linear fences. The cow wears a smart collar consisting of a GPS unit, a PDA with an 802.11 (WiFi) compact flash card and amplified speakers. When near to a fence the cow gets a sound stimulus inversely proportional to the distance to the boundary to keep the cow within the boundary. The WiFi connection is used to specify the parameters of the system, like the virtual boundary, from a base station and return acknowledgements and status information (of the collar). The goal of the system is to keep a group of cows within the boundary using the sound stimuli but not to monitor the togetherness of the group over time. The only form of cooperation among the collars in this system is the forwarding of messages using an ad-hoc messaging protocol.
Another work that considers a group of animals is the electronic shepherd system (ES) described in [12]. In this system sheep are tracked while they graze in the summer period. A two-tier system is used: sheep usually tend to cluster in flocks, and within each flock one individual is equipped with a gateway node possessing a GSM/GPRS modem and a GPS receiver. Furthermore, all individuals possess a short-range wireless communication system operating in the $433\,\mathrm{MHz}$ band (radio tag). Through this system the flock members report their identity and other sensed status data periodically (e.g. temperature) to the flock leader, who collects the data and transmits it through GPRS to a central server. Again, there is no cooperation among the radio tags to track individual sheep but in this system missing individuals can be identified (with some delay) and their last position of can also be estimated based on the position of the last forwarding flock leader.
There are also systems described that use RFID technologies or WiFi to locate persons (children) in theme parks (see for example [1], [2]). In these systems the geographical area is delineated implicitly by the transmission range of infrastructure devices like WiFi access points or RFID readers. The position of an individual is either given by the nearest reader or calculated using triangulation of WiFi access points. All the above systems have in common that the radio tags attached to each individual are just data sources (except for virtual fences) and only transmit information to the nearest base station or gateway node for further processing. There is no cooperation among the radio tags to track individual items. Furthermore, these system rely on the presence of specific infrastructure like GPS, GSM/GPRS, WiFi access points or RFID readers and are therefore bound to their designated geographical area. Our approach to herding is different in several aspects: (i) We want to enable the radio (WSN) tags to cooperatively monitor the togetherness of the group and identify missing items

or individuals by in-network processing of information. (ii) We do not consider a fixed geographical area. In our case a group can be mobile as a whole and its togetherness is investigated by the tags. (iii) We do not critically rely on the presence of infrastructure. Using a long-range communication system for indicating the loss of a group item is a reasonable add-on but not the core of our approach.
A specific approach that is also based on WSN technology and meets some of the properties above is SVATS (Sensor-network-based Vehicle Anti-Theft System) [13]. In this system sensor nodes are deployed in cars and form a network when parked in the same parking area. Each car is then monitored by several sensor nodes from neighboring cars based on periodic ”alive” messages. If a monitoring node misses a predefined number of such messages it starts a verification process. First it sends a challenge to the monitored node and waits for a response. If there is no response even after several retries the monitoring node confirms the theft detection and broadcasts an announcement to other nodes. Based on the reception of distinct announcements from different nodes each monitoring node makes a final decision on the detection of the theft and contacts a base station. Thus SVATS combines formation of monitoring groups for each sensor node and cooperation among the monitoring nodes to detect and verify the car theft. To maintain a group of monitoring nodes a node performs three phases: initial power-level estimation, neighbor discovery and neighbor maintenance. When activated it first listens to ”alive” messages from other nodes and orders the neighbors based on their transmit power level. It then sends a ”join” message with the list of discovered neighbors at a selected power level to reach a desired number of the discovered neighbors. Nodes receiving a ”join” message and find themselves in the transmitted list mark themselves as neighbors (monitoring nodes) and send a ”reply”. If enough ”reply” messages can be received the group of monitoring nodes is formed, otherwise the power-level is increased. To maintain the established monitoring groups the number of distinct nodes is periodically checked and the power level is adjusted, if needed. By design, SVATS is limited to static groups (of cars) and bound to a geographic area close to a base station. However, the communication with the base station is not needed for the detection of a theft but only for the immediate notification of the owner.
Another work with strong focus on the cooperation of radio tags is given in [14, 15]. Here a reverse problem to herding is considered: For the handling and storage of chemicals there are situations in which reactive materials should *not* be put together in proximity of each other. Containers holding the materials are equipped with sensor nodes that are aware of the type of material in the container and can also sense the distance to each other (ultrasonic). The devices share knowledge on materials and distances (facts) over wireless links in the $869\,\mathrm{MHz}$ band that is used to cooperatively identify hazardous situations based on a common set of rules.

Chapter 6

Conclusion and Future Work

Using wireless sensor technology for keeping arbitrary groups of persons or items together appears to be an interesting field of research. We have presented a specific framework that deals with core building blocks of a future herding system and bases on the local monitoring of entities provided by a rather small number of adjacent entities. Only if an entity is found in a critical state network-wide operations for verification and, if necessary, distribution of alarm notifications are required. Since one single entity is required to monitor only a fixed number of adjacent entities the approach in general scales well to large network densities. In a previous study [4] we have already investigated a specific scheme to the monitoring of an entity by a small group of adjacent entities. In this work we now focus on the problem of forming these small groups of monitoring entities, called circle of friends (CoF), and present an approach referred to as *Distributed Randomized Selection* (DRS). The truly distributed DRS approach enables entities to locally select adjacent entities for monitoring based on independent random decisions. At the same time DRS enforces that all entities of a group are monitored by a minimum number of entities. We have conducted a simulation study considering different parametrizations of the DRS approach and investigate the forming of CoF configurations for random entity deployments without mobility. The results show that with DRS it is well possible to cover all entities with a group of monitoring entities of minimum required size. This can be achieved for low as well as for high entity densities without additional memory and communication costs. Thus the DRS approach scales well with increasing entity densities. Critical factors for the performance of the approach in terms of entity coverage are the error in estimating CoF sizes and the down-sampling of entity selections. From the results we can conclude that an adaptive selection probability based on the local entity density is preferable for optimal performance independent from the entity density.

It is clear that our specific framework is not the only way towards a future herding systems. It is well conceivable that other approaches use explicit constructions of the network communication graph G to check for connectivity and subsequently the togetherness of a group of entities. A natural measure for the connectivity of graph G is its *k-connectivity* which specifies the smallest number k of entities whose removal will disconnect the graph. Existing work in [16] indicates that there is a relationship between the vertex degree (number of edges) and the k-connectivity. One could now consider the number of monitoring entities for a given entity that is established by DRS as the vertex degree with respect to the communication graph G. This degree is smaller or equal to the real vertex degree and investigations on the relationship between the minimum CoF sizes and the k-connectivity of the communication graph are conceivable.

Within the scope of the DRS approach there are also several opportunities for further work. A clear

point for improvement is the local estimation of CoF sizes. We have used a very minimalistic approach for this study and have shown that it is reasonable to consider more sophisticated approaches that reliably minimize the error in the estimates. Until now, we also have not introduced a revoke rule as a connection between the forming of the monitoring groups and the classification results provided by these groups. Especially in mobile scenarios the maintenance of entity selections requires information provided by this classification to carefully hand over the monitoring of an entity that is moving within the group. But also for the initial selections more sophisticated selection rules that take link quality measures (e.g. the RSSI and LQI values of IEEE 802.15.4 packets, PRR estimates) into account are conceivable. Further investigations must show how sustainable the initial selections are and how reliable the communication among the monitoring entities can be performed for cooperation – if the entities are close, the gain from cooperation might be bigger.
Besides all this, the next logical steps should include the implementation of the DRS approach and its experimental evaluation, preferably in a large-scale testbed like the *TKN Wireless Sensor network Testbed* (TWIST) [17].

Bibliography

[1] Jonathan Collins. *Lost and Found in Legoland*. RFID Journal, April 2004.

[2] Alorie Gilbert. *Theme park takes visitors to RFID-land*. CNET News, September 2004.

[3] H. Karl and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. John Wiley & Sons, 2005.

[4] A. Willig, M. Kühm, and A. Wolisz. Cooperative distance classification using an IEEE 802.15.4-compliant transceiver. In *Wireless Communications and Networking Conference, 2009. WCNC 2009. IEEE*, pages 1–6, April 2009.

[5] 802.15.4-2006: IEEE Standard for Information technology- Telecommunications and information exchange between systems- Local and metropolitan area networks- Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), 2006.

[6] The Network Simulator NS-2. http://nsnam.isi.edu/nsnam.

[7] J. Zheng and Myung J. Lee. A comprehensive performance study of IEEE 802.15.4. In *Sensor Network Operations*, chapter 4, pages 218–237. IEEE Press, Wiley Interscience, 2006.

[8] MoteIV Corporation. Tmote Sky Datasheet. http://www.moteiv.com, Nov 2006.

[9] Kevin Fall and Kannan Varadhan. *The ns Manual (formerly ns Notes and Documentation)*. The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, 2.32 edition, Jun 2008.

[10] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 14–27, New York, NY, USA, 2003. ACM Press.

[11] Z. Butler, R. Corke, R. Peterson, and D. Rus. Dynamic virtual fences for controlling cows. In M.H. Ang and O. Khatib, editors, *Experimental Robotics IX, STAR 21*, volume Volume 21/2006, pages 513–522. Springer Berlin / Heidelberg, 2006.

[12] Bjoern Thorstensen, Tore Syversen, Trond-Are Bjoernvold, and Tron Walseth. Electronic shepherd – a low-cost, low-bandwidth, wireless network system. In *Proc. MobiSys 2004*, Boston, USA, June 2004.

[13] Hui Song, Sencun Zhu, and Guohong Cao. SVATS: A sensor-network-based vehicle anti-theft system. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages 2128–2136, April 2008.

[14] Martin Strohbach, Hans-Werner Gellersen, Gerd Kortuem, and Christian Kray. Cooperative artefacts: Assessing real world situations with embedded technology. In *UbiComp 2004: Ubiquitous Computing*, volume 3205/2004, pages 250 – 267. Springer Berlin / Heidelberg, 2004.

[15] Uwe Kubach, Christian Decker, and Ken Douglas. Collaborative control and coordination of hazardous chemicals. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 309–309, New York, NY, USA, 2004. ACM.

[16] Christian Bettstetter. On the minimum node degree and connectivity of a wireless multihop network. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 80–91, New York, NY, USA, 2002. ACM.

[17] TKN Wireless Indoor Sensor network Testbed. http://www.twist.tu-berlin.de, June 2009.