

Efficient Cluster Formation for Sensor Networks

Ewa Hansen, Jonas Neander, Mikael Nolin and Mats Björkman
Mälardalen Real-Time Research Centre
Mälardalen University, Sweden
ewa.hansen@mdh.se

Abstract

The interest in wireless sensor networks is growing and the development of energy efficient infrastructures for such networks is becoming increasingly important. In this paper we investigate the usefulness of enforcing a minimum separation distance between cluster heads in a cluster based sensor network, i.e. prolonging network lifetime by lowering the energy consumption.

The simulations were performed in order to determine how much we can lower the energy consumption in the sensor network by separating the cluster heads. We have also investigated how the number of clusters affect the energy consumption for a given minimum separation distance.

We show that our sensor network performs up to 150% better when introducing a minimum separation distance between cluster heads, comparing the number of messages received at the base station. The simulations also show that the minimum separation distance that result in the lowest energy consumption in our network varies with the number of clusters.

1. Introduction

The need for energy-efficient infrastructures for sensor networks is becoming increasingly important. Wireless sensor networks are networks consisting of many sensor nodes that communicate over a wireless media. A sensor node is equipped with a sensor module, a processor, a radio module and a battery. Since the battery limits the lifetime of the sensor nodes it also limits the lifetime of the sensor network, thus energy efficiency is a major issue for sensor networks.

An important goal in many sensor networks is to monitor an area as long time as possible. Hence, it

is important to distribute energy consumption evenly across the network. When the energy consumption is evenly distributed, the major part of the sensor nodes will stay alive approximately equally long time. This enables continued information gathering throughout the whole network area during the lifetime of the network.

The most power-consuming activity of a sensor node is typically radio communication [10]. Hence, radio communication must be kept to an absolute minimum. This means that the amount of network traffic should be minimized. In order to reduce the amount of traffic in the network, we build clusters of sensor nodes as proposed in e.g. [1, 3, 9]. Some sensor nodes become cluster heads and collect all traffic from their respective cluster. The cluster head aggregates the collected data and then sends it to its base station. When using clustering, the workload on the cluster head is thus larger than for non-cluster heads. The cluster heads should therefore be changed several times during the lifetime of the sensor network in order to distribute the extra workload and energy consumption evenly.

Our hypothesis is that the geographical distribution of the cluster heads severely influences the overall energy consumption of the network, thus prolonging its lifetime. Simulations presented in this paper indicate that introducing a minimum separation distance between cluster heads improves network lifetime.

For our simulations we have used the AROS architecture, Asymmetric communication and ROuting in Sensor networks [7]. AROS is an extension of LEACH-C [2] which is a well known cluster-based sensor network architecture. The AROS architecture is based on cluster groups using base stations with “unlimited” energy and “enough” bandwidth in the back-

bone network. Often, the base stations can be situated in existing infrastructures. For instance, there are infrastructure networks built in hospitals and industrial factories that could be used to host base stations. The infrastructure network can act as a, possibly fault tolerant, base station backbone for sensor nodes collecting data or monitoring patients.

In order to be able to turn off the radio of the sensor nodes as long as possible to save energy, we use Time Division Multiple Access (TDMA) to schedule the communication of the sensor nodes. Furthermore, we use clusters to ease the scheduling of the sensor nodes. When using clusters we can aggregate or fuse data to lower the communication needs in the sensor network.

AROS is based on clusters where the cluster heads gathers data from their cluster nodes and then transmit it to the base station. AROS has an asymmetric topology where the base station is able to transmit information to all its sensor nodes directly. All cluster heads may however not be able to transmit data directly to the base station. Hence, traffic from these cluster heads must be routed through other cluster heads in order to reach the base station. However, routing of traffic through other cluster heads will increase the power consumption of the forwarding cluster heads. Therefore, routing decisions must be carefully evaluated in order to maximize network lifetime. In AROS we use a centralized approach where the resource-adequate base stations perform all the calculations necessary to evaluate routes and schedules, thus relieving sensor nodes from the energy-consuming task of executing convex distributed decision algorithms.

In our simulations we have experimented with a minimum separation distance between cluster heads. We have also investigated how the number of clusters used, together with this minimum separation distance, affects the energy consumption in the network. The minimum separation distance is the smallest distance that is allowed between cluster heads. The distance can be larger than the minimum separation distance but should not be smaller. The simulations were performed in order to investigate the effects on the energy consumption when using a minimum separation distance between cluster heads.

The simulations show that the minimum separation distance that result in the lowest energy consumption

in our network varies with the number of clusters. The simulations also show that it is up to 150% better to use a minimum separation distance between cluster heads than not using any minimum separation distance at all, measured with the number of messages received at the base station. By using a minimum separation distance between cluster heads we can make the network gather more messages from the network for a longer period of time.

The rest of this paper is outlined as follows: in Section 2, we describe some related work. In Section 3, we present the minimum separation distance algorithm and the simulation setup. In Section 4 we present the results from our simulations, and finally, in Section 5 we present our conclusions.

2. Related Work

LEACH (Low-Energy Adaptive Clustering Hierarchy) [3] is a TDMA cluster based approach where a node elects itself to be cluster head by some probability and broadcasts an advertisement message to all the other nodes in the network. A non cluster head node selects a cluster head to join based on the received signal strength. Being cluster head is more energy consuming than to be a non cluster head node, since the cluster head needs to receive data from all cluster members in its cluster and then send the data to the base station. All nodes in the network have the potential to be cluster head during some periods of time. The TDMA scheme starts every round with a set-up phase to organize the clusters. After the set-up phase, the system is in a steady-state phase for a certain amount of time. The steady-state phases consist of several cycles where all nodes have their transmission slots periodically. The nodes send their data to the cluster head that aggregates the data and send it to its base station at the end of each cycle. After a certain amount of time, the TDMA round ends and the network re-enters the set-up phase.

LEACH-C (LEACH-Centralized) [2] is a variant of LEACH that uses a centralized cluster formation algorithm to form clusters. The protocol uses the same steady-state protocol as LEACH. During the set-up phase, the base station receives information from each node about their current location and energy level. After that, the base station runs the centralized clus-

ter formation algorithm to determine cluster heads and clusters for that round. LEACH-C uses simulated annealing [4] to search for near-optimal clusters. LEACH-C chooses cluster heads randomly but the base station makes sure that only nodes with “enough” energy are participating in the cluster head selection. Once the clusters are created, the base station broadcasts the information to all the nodes in the network. Each of the nodes, except the cluster head, determines its local TDMA slot, used for data transmission, before it goes to sleep until it is time to transmit data to its cluster head, i.e., until the arrival of the next slot.

A further development is LEACH-F (LEACH with Fixed clusters) [2]. LEACH-F is based on clusters that are formed once - and then fixed. Then, the cluster head position rotates among the nodes within the cluster. The advantage with this is that, once the clusters are formed, there is no set-up overhead at the beginning of each round. To decide clusters, LEACH-F uses the same centralized cluster formation algorithm as LEACH-C. The fixed clusters in LEACH-F do not allow new nodes to be added to the system and do not adjust their behavior based on nodes dying.

BCDCP (Base-station Controlled Dynamic Clustering Protocol) [6] is a centralized routing protocol with a high-energy base station that makes all the high energy-consuming activities e.g. selecting cluster heads and routing paths, performing randomized rotation of cluster heads. The idea in BCDCP is to organize balanced clusters with uniformed placement of cluster heads where each cluster head serves an approximately equal number of member node.

During each setup phase the base station receives information on the current energy status from all the nodes in the network. BCDCP uses an iterative splitting algorithm to form clusters. The first step is to choose two nodes, among the eligible nodes, that have the maximum separation distance. Step two is to group the remaining nodes to one of the cluster heads, whichever is closest. Step three is to balance the clusters so that each cluster has approximately the same number of nodes. Step four is to start from step one and split the sub-clusters in to smaller parts. The iteration of the four steps continues until the desired number of cluster heads is attained.

3. Our Approach

In order to be able to see the effects on the energy consumption when using a minimum separation distance between cluster heads we have developed a simplified algorithm to find and select cluster heads.

3.1. Cluster head selection algorithm

In our cluster formation algorithm, we use the same simulated annealing as LEACH-C to minimize the energy consumption for cluster nodes when transmitting data to the cluster head. As LEACH-C, we randomly choose a node among the eligible nodes to become cluster head but we also make sure that the nodes are separated with a minimum separation distance (if possible) from the other cluster head nodes.

```

MSD = Minimum Separation Distance
dc = Number of desired cluster heads,
alive = Set of nodes alive,
energy(n) = Remaining energy for node n
avg =  $\frac{\sum energy(n)}{number\ of\ alive\ nodes}$ 
CH= {}
eligible = {n | energy(n) ≥ avg }
assert(|eligible| ≥ dc)

While (|CH| < dc)
  if ∃n: n∈eligible ∧ (∀ m∈CH, dist(m,n)) ≥ MSD
    add(n , CH)
    remove(n , eligible)
  else
    n ∈ eligible
    add(n, CH)
    remove(n, eligible)

```

Figure 1. Algorithm to select Cluster Heads (CH)

In the cluster head selection part, see Figure 1, cluster heads are randomly chosen from a list of eligible nodes. To decide which nodes that is eligible, the average energy of the remaining nodes in the network is calculated. In order to spread the load evenly, only nodes with energy above the average energy are eligible.

As long as it is possible, or as long as the desired number of cluster heads is not attained, we choose a node among the eligible nodes that is further away than the minimum separation distance from all other chosen cluster heads. If that is not possible, we chose

another node among the eligible nodes to become cluster head.¹ When all cluster heads have been chosen and separated, generally with at least the minimum separation distance, clusters are created the same way as in [2].

3.2. Simulation Setup

In the performed simulations we have varied the minimum separation distances between cluster heads, in order to see the effects on the energy consumption in the network. We have also investigated whether the number of clusters used, together with the minimum separation distance, has any effect on the energy consumption.²

All simulations presented in this paper were performed within one network setup. That is, we have used the same number of nodes and the same position of these nodes in all experiments presented in the paper.

The simulations were performed in the network simulator NS 2 [8], using a network size of 400x400 meters where 100 sensor nodes were randomly distributed in the network. We placed the base station 75 meters outside the monitored area, at location $x = 200$, $y = 475$. All sensor nodes start with a fixed amount of energy and the simulation continues until all the sensor nodes in the network have consumed all of their energy. Since AROS is an extension of LEACH, we have used the same simulation setup for our simulations as in LEACH [2], and all other parameters such as radio speed, processing delay and radio propagation speed were the same as in [2, 7].

In [2], Heinzelman has calculated how often the cluster heads should be changed, i.e. the round-time. The calculation was made for a 100x100 meters network. Due to the larger energy consumption of sending longer distances in a 400x400 meters network, we need to change cluster heads more often than every 20:th second, which is the round-time for the 100x100 meters network [2]. In our simulations we have chosen to change cluster heads every 10:th sec-

¹The algorithm is simplified for these simulations, i.e. the assert in Figure 1 will always be true.

²The minimum separation distance varied between 50 and 140 meters, and the number of clusters varied between 2 and 15 clusters.

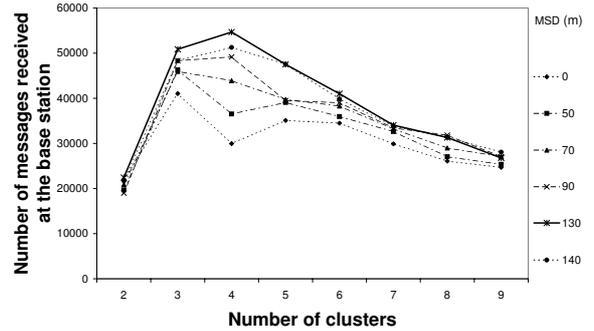


Figure 2. Messages received

ond. This is a tradeoff between rescheduling cost, efficiency and energy consumption balance. When the network reschedules new cluster heads are chosen and new clusters are formed at the same time.

4. Results

In Figure 2, we see how the minimum separation distance affects the energy consumption, i.e. the number of messages received at the base station during the lifetime of the network. We also see how the number of clusters used affects the energy consumption in the network. In the same figure we see that when using 2 clusters, the number of messages received at the base station is low in all our simulations. Further, we see that when using 4 clusters and a minimum separation distance of 130 meters between cluster heads, the base station receives the most messages. It is not always the case that 4 cluster yield the most messages to the base station. For some minimum separation distances 3 cluster heads yields the most messages. Below, we have therefore looked at the simulation results in more detail when using 3 and 4 clusters, respectively.

In Figure 2, we see that using a minimum separation distance between cluster heads is better than not to use any to control the placement of the cluster heads. By using a minimum separation distance between cluster heads we can make the network gather more messages from the network for a longer period of time. The figure also shows that a minimum separation distance of 130 meters delivers the most messages to the base station for almost all number of clusters.

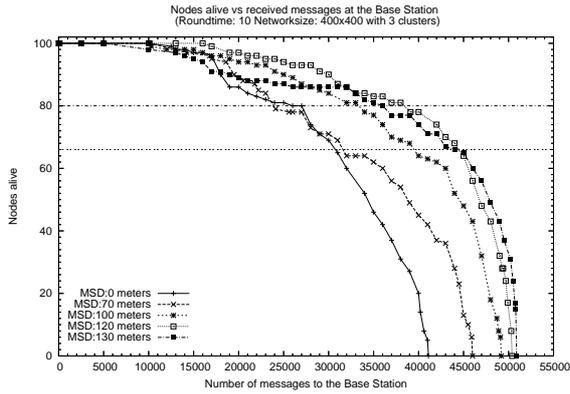


Figure 3. Using 3 clusters

4.1. Using 3 Clusters

In Figure 3, we present simulation results when using 3 clusters. In order to be able to see the curves more distinctively in the figure we have chosen to only show a subset of the curves.³ In Figure 3, we see that when not using a minimum separation distance between cluster heads, the base station receives approximately 41000 messages. However, when using 3 clusters and 130 meters as the minimum separation distance, the base station receives approximately 51000 messages, which is an enhancement of 24%, or 10000 messages. If we look at 80% tolerance limit⁴, illustrated with the upper horizontal line in Figure 3, we see that when not using a minimum separation distance the curve drops below the tolerance limit already at 26000 messages. When using 130 meters as the minimum separation distance the curve drops below the tolerance limit at 39000 messages, while when using 120 meters as the minimum separation distance the curve drops below the tolerance limit at 37000 messages.

Depending of the tolerance limit, different minimum separation distances yield the longest network lifetime, e.g., the crossover point between using 120 and 130 meters as the minimum separation distances is slightly above 65% sensor nodes alive, meaning that for tolerance limits above 65%, using a 120 meters

³All curves not represented in the figure are located in between the curves MSD: 0 meters and MSD: 130 meters.

⁴Most sensor networks have a lower limit on the number of nodes that must be alive in order for the network to still be functional, we call this limit the *tolerance limit*.

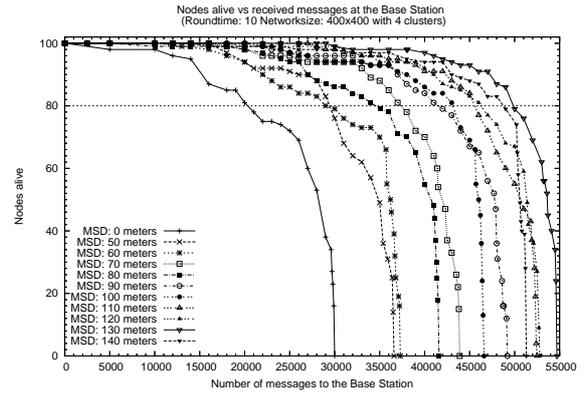


Figure 4. Using 4 clusters

minimum separation distance yields the longest network life (in terms of messages received at the base station). The 65% tolerance limit is illustrated with the lower horizontal line in Figure 3.

In general, the spread between minimum separation distances is small in the figure, and all curves in the figure have a rather gradual slope (see also discussion on slope below).

4.2. Using 4 Clusters

In Figure 4, we present simulation results when using 4 clusters. We show that when using 4 clusters and a minimum separation distance of 130 meters between cluster heads, the base station receives almost 55000 messages, compared to the simulation with 4 clusters and no minimum separation distance where the base station only receives approximately 30000 messages. The minimum separation distance of 130 meters between cluster heads thus gives an enhancement of 80%, or 25000 messages.

If we look at the 80% tolerance limit, we see that the 130 meters minimum separation distance curve crosses the limit at about 50000 messages, while the 0 meters minimum separation distance curve crosses the limit already at about 20000 messages. Using 130 meters as the minimum separation distance thus gives an enhancement of 150%, or 30000 messages, compared to when not using any minimum separation distance.

When comparing the results from using 3 clusters and 4 clusters, we see that the number of messages received is larger for 4 clusters than for 3 clusters for the

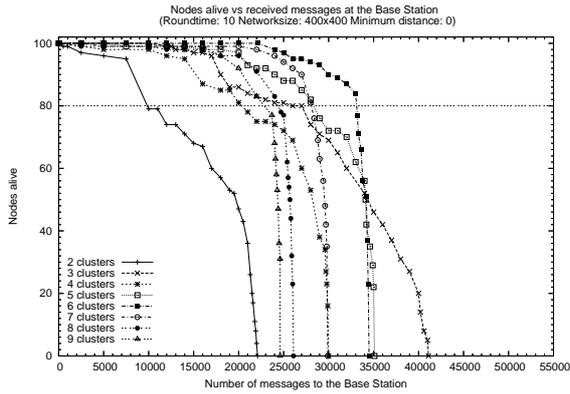


Figure 5. No Minimum Separation Distance

best minimum separation distances. We can also see in the figures that the spread between different minimum separation distances is much larger for 4 clusters than for 3 clusters, meaning that the choice of minimum separation distance becomes much more important. It can also be noted that most curves have a steeper slope when using 4 clusters than when using 3 clusters. This means that using 4 clusters can be more advantageous for high tolerance limits. In our figures, when using 130 meters as the minimum separation distance, the total number of messages received is 51000 and 55000 for 3 and 4 clusters, respectively, a relatively small difference, less than 10%. However, when comparing the same curves at the 80% tolerance limit, the number of messages received is 37000 and 50000, respectively. Here, the relative difference is around 30%. The conclusion from this example is that the slope of the curve matters, this will be further discussed below.

4.3. Minimum separation distance or not?

Figure 6 show results from simulations with a minimum separation distance of 130 meters and the number of clusters varied between 2 and 9. As mentioned above, when using 4 clusters and a minimum separation distance of 130 meters between cluster heads, the base station receives the most messages. When using the tolerance limit of 80%, the base station receives approximately 50000 messages.

The bad performance when using 2 clusters can clearly be seen in Figure 6, approximately 8000 messages are received when the 80% tolerance limit is

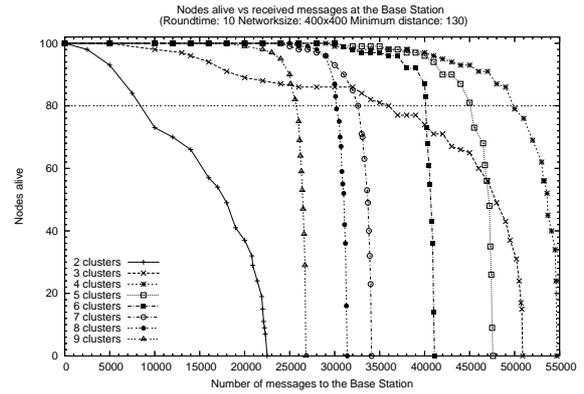


Figure 6. Minimum Separation Distance 130 meters

reached. The reason for this is that when using only 2 clusters, the communication distances between nodes become so long that the radio energy consumption (which is super-linear with communication distance) increases very much. It can also be seen in the figure that the slope when using 3 clusters is very gradual, as was observed earlier.

Figure 5 shows results from simulations without minimum separation distance, i.e., 0 meters as the minimum separation distance. We can see that when using the 80% tolerance limit and optimizing for maximum number of messages received at the base station, the best configuration of the sensor network is to use 6 clusters. The base station then receives approximately 33000 messages. When using 6 clusters and a minimum separation distance of 130 meters between cluster heads, depicted in figure 6, the base station receive approximately 40000 messages when using the 80% tolerance limit. Using 130 meters instead of not using a minimum separation distance thus yields an enhancement of 7000 messages.

Comparing Figure 5 and Figure 6 we see that regardless of how many clusters we choose to use in the network, using a minimum separation distance of 130 meters between cluster heads instead of not using any minimum separation distance will make the network stay alive longer and deliver more data to the base station.

4.4. Efficient utilization

Efficient utilization of the energy resources of the sensor nodes will increase the lifetime of the sensor network. In the ideal network, all sensor nodes would live exactly the same period of time.

In Figure 6, we see that the more efficient utilization of the sensor nodes' power makes the sensor network stay alive a longer period of time. In the figure we can also see that as soon as the sensor nodes in the network start to demise, the whole network demises shortly after for all number of clusters above 3, thus the utilization of the sensor nodes' energy has been efficient in these cases.

To be able to say that the utilization of the sensor nodes' energy has been efficient, we want the "knee" of the curve to be as sharp as possible, see Figure 6. The sharper the knee is, the better the energy consumption is distributed among the sensor nodes.

We want the knee to drop as late as possible and when it finally drops the gradient should be as steep as possible. This indicates that the sensor nodes have been utilized efficiently, hence the network lives longer. This steep gradient also indicates that the whole network area is monitored almost until the whole network demises.

In Figure 5, we see a sharp knee and a steep gradient only when using 6 clusters. This indicates that most of the sensor nodes have been utilized efficiently when using 6 clusters. Looking for sharp knees and steep gradients in Figure 6, we can see that almost every choice of number of clusters have a steep gradient, except for 2 and 3 clusters, which have a more gradual slope.

When looking at Figure 6, we can see that when the number of clusters increases the sharper the knee becomes. Unfortunately this is a tradeoff between sharp knees and the total number of messages received. The figure show that despite of the fact that 8 and 9 clusters have the sharpest knees, using 4 clusters still delivers more messages to the base station at all times. When using 8 or 9 clusters the base station receives totally 31000 and 26000 messages respectively, while when using 4 clusters all nodes are still alive continuing to gather information, when the base station has received the same amount of messages. This means that even though 8 or 9 clusters have the sharpest knees, using

4 clusters is still a better choice, when comparing the number of messages received at the base station.

5. Conclusions

In this paper we have presented simulation results from our experiments with minimum separation distances between cluster heads. We have performed these simulations in order to be able to determine how much we can lower the energy consumption in the sensor network by separating the cluster heads, i.e. by distributing the cluster heads through the whole network.

We have presented a simplified energy-efficient cluster formation algorithm for the wireless multihop sensor network AROS.

We have shown that the minimum separation distance improves the energy efficiency, measured by the number of messages received at the base station. We have also shown that it is better, up to 150% in our simulations, to use a minimum separation distance between cluster heads than not to use any minimum separation distance. By using a minimum separation distance between cluster heads we make the network live longer, gathering data from the whole network area. We have also shown that the number of clusters used together with the minimum separation distance affects the energy consumption. Using 4 clusters and a minimum separation distance of 130 meters between cluster heads is the best configuration for our simulated network.

Our simulations have also shown that, depending on the number of dead nodes that can be tolerated, different minimum separation distances as well as different number of clusters affects the number of messages received before the given tolerance limit is reached. Looking at the slope of the curve can give a good feeling of how suitable a certain configuration is; the steeper slope the better.

Future work includes more thorough analysis in more scenarios with varying numbers of sensor nodes and network sizes, as well as evaluating alternative algorithms for cluster head selection. A comparison between the minimum separation distance algorithm and the BCDCP algorithm is also to be considered in the future.

References

- [1] M. Gerla, T. Kwon, and G. Pei. On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering. *Proceedings of IEEE WCNC 2000*, Sep 2000.
- [2] W. Heinzelman. *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, Massachusetts institute of technology, June 2000.
- [3] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. Maui, Hawaii, Jan 2000. In Proceedings of the 33rd International Conference on System Sciences (HICSS '00).
- [4] T. Maruta and H. Ishibuchi. Performance Evaluation of Genetic Algorithms for Flowshop Scheduling Problems. *Proceedings of the 1st IEEE Conference on Evolutionary computation*, 2:812–817, June 1994.
- [5] MIT uAMPS LEACH ns Extensions. World Wide Web, <http://www.ece.rochester.edu/research/wcng/code/index.htm>, 2005.
- [6] S. D. Muraganathan, D. C. F. Ma, R. I. Bhasin, and A. O. Fapojuwo. A centralized energy-efficient routing protocol for wireless sensor networks. *Communications Magazine, IEEE*, 43(3):8–13, March 2005.
- [7] J. Neander, E. Hansen, M. Nolin, and M. Björkman. Asymmetric Multihop Communication in Large Sensor Networks. *International Symposium on Wireless Pervasive Computing 2006, ISWPC*, Jan 2006.
- [8] NS 2, Network Simulator. World Wide Web, <http://www.isi.edu/nsnam/ns/ns-build.html>, 2004.
- [9] G. Pei and C. Chien. Low Power TDMA in Large Wireless Sensor Networks. *Military Communications Conference*, 1:347–351, 2001.
- [10] G. J. Pottie and W. J. Kaiser. Wireless Integrated Network Sensors. *Communications of the ACM*, 43(5):51–58, May 2000.