An Asymmetric Network Architecture for Sensor Networks

Jonas Neander, Ewa Hansen, Mikael Nolin and Mats Björkman

Mälardalen Real-Time Research Centre

Mälardalen University, Sweden

{jonas.neander, ewa.hansen, mikael.nolin, mats.bjorkman}@mdh.se

MRTC report ISSN 1404-3041 ISRN MDH-MRTC-181/2005-1-SE,

Mälardalen Real-Time Research Centre, Mälardalen University, August, 2005

Abstract

In this paper, we present an asymmetric sensor network architecture. We use an existing infrastructure as base station support for the sensor network. The network architecture communication topology is asymmetric, in that the base station can reach its sensor nodes in one hop – but there is no guarantee that the sensor nodes reach the base station directly. To handle sensor nodes with different demands, and to save energy at the sensor nodes, we schedule the sensor nodes with time-division multiplexed access (TDMA).

We provide an initial comparison between our architecture and LEACH, a well-known TDMA cluster-based sensor network architecture. The comparison shows that our architecture outperforms LEACH and its variants LEACH-C and LEACH-F in large networks and that our architecture performs almost as well as the LEACH protocols in small networks.

1. Introduction

In this paper, we present an asymmetric sensor network architecture. We also present a comparison between our architecture and LEACH [4], a similar, and well-known, symmetric sensor network architecture.

With the growing interest in sensor networks, efficient communication infrastructures for such networks are becoming increasingly important. Among the interesting application areas for sensor networks are environmental surveillance and surveillance of equipment and/or persons in, e.g., a factory or a hospital. Common for application areas considered in this paper are that sensor nodes are typically left unattended after deployment, the communication is wireless, and the power supply is limited.

Deploying unattended sensor nodes with limited power supplies implies that one important feature of sensor networks is robust functionality in face of network nodes dropping out of the network after some time of activity. Another implication is that, if the network is to survive a longer period of time, new nodes will have to be added to the existing network. Thus the network topology must be dynamic, even if the sensor nodes themselves may not be mobile.

Some sensor nodes may not be able to communicate directly with the base station. The traffic from these sensor nodes must be forwarded by other sensor nodes, hence routing schemes are necessary. However, routing of traffic through other sensor nodes will increase the power consumption of the forwarding sensor nodes. Therefore, routing decisions must be carefully evaluated in order to maximize network lifetime. We use a semi-centralized approach where resource-adequate infrastructure nodes can act as base stations and, hence, be used to off-load sensors and thus prolong network lifetime.

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 and thereby prolong the lifetime of the sensor networks. The infrastructure network can act as a, possibly fault tolerant, base station backbone for sensor nodes collecting data or monitoring patients.

Industrial and hospital infrastructure networks are relatively static and they do not have limited energy as sensor nodes do. In this paper we assume that the base stations are stationary. The infrastructure network could be wired, wireless or a combination of both, see Figure 1.

A base station in the our architecture has large radio coverage and has the potential to accept all the sensor nodes that are receiving the signal from the base station. One possible solution in order to reduce the amount of traffic in the network is to build clusters of sensor nodes as proposed in e.g. [3, 4, 8]. Some sensor nodes become cluster heads and collect all traffic from/to their cluster. A cluster head aggregates the collected data and then sends it to its base station. In our architecture, asymmetric communication is possible. That is, the base station reaches all the sensor nodes directly, while some sensor nodes cannot reach the base station directly but need other nodes to forward its data.

The most power-consuming activity of a sensor node is typically radio communication [9]. Hence, communication must be kept to an absolute minimum. All activities involving communication are power-consuming and the most important way to save power is to turn off the radio as long time as possible. This applies to transmission and reception, but also

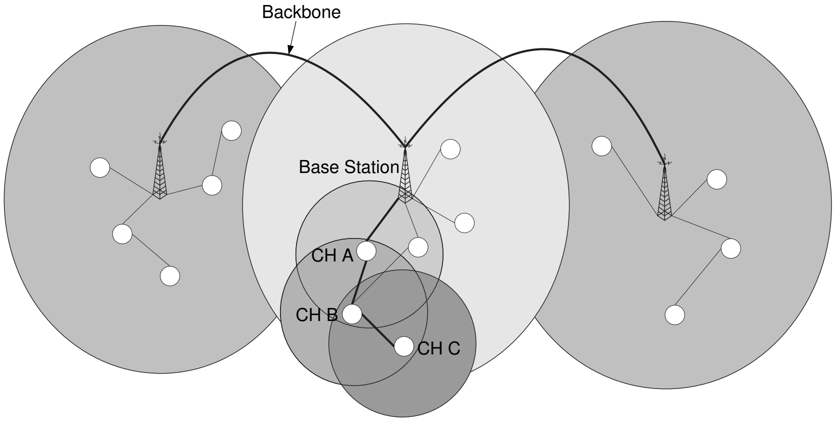

Figure 1. Overview of the architecture.

to listening for data. Hence, we use time division multiplexed access (TDMA) schemes for sensor node communication in our architecture which allows the radio to be turned off for long periods of time.

In this paper, we also provide an initial comparison between our architecture and the LEACH sensor network architecture and its variants LEACH-C and LEACH-F [4]. The main focus of the comparisons is to study the energy consumption when transfering data from the sensor nodes to the base station. We do these comparisons in order to verify that forwarding data is more energy efficient in large networks, than sending it directly to the base station.

We show that our architecture delivers more messages to the base station than LEACH and its variants given the same amount of energy. We also show that our architecture has more sensor nodes alive at any given time, after the first demised sensor node in LEACH. The sensor nodes that are alive can be found throughout the entire network thus providing coverage of the whole monitored area. Our results show that our architecture, in its preliminary non-optimized form, outperforms LEACH and its variants when the network size increases.

The rest of this paper is outlined as follows: in Section 2 we describe, related work. In Section 3, our architecture is presented. Section 4 describes the comparisons between our architecture and the LEACH protocols, and Section 5 presents the results from the comparisons. Finally, we conclude and outline future work.

2. Related Work

LEACH (Low-Energy Adaptive Clustering Hierarchy) [4] 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 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) [4] has been developed out of LEACH and the basis for LEACH-C is to use a central control 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. According to [4], the nodes may get their current location by using a global positioning system (GPS) receiver that is activated at the beginning of each round. After that, the base station runs the centralized cluster formation algorithm to determine the clusters for that round. To determine clusters and select cluster heads, LEACH-C uses simulated annealing [7] to search for near-optimal clusters. Before running the algorithm that determines and selects the clusters, 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 TDMA slot used for data transmission. Then, the node goes to sleep until it is time to transmit data to its cluster head.

A further development is LEACH-F (LEACH with Fixed clusters) [4]. 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. Furthermore, LEACH-F does not handle node mobility.

TEEN (Threshold-sensitive Energy Efficient sensor Network protocol) [5] and APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network protocol) [6] are both designed for time-critical applications. Both TEEN and APTEEN uses asymmetric communication between the base station and the sensor nodes. Further, they build clusters with cluster heads that perform data aggregation and then send the aggregated data to the base station or to a cluster head.

In TEEN, the cluster head broadcasts a hard and a soft threshold to its members. The hard threshold aims at reducing the number of transmissions by allowing the nodes to transmit only when the sensed attribute is in the range of interest. The soft threshold further reduces the number of transmissions by eliminating all the transmissions which might have occurred otherwise when there is little or no change in the sensed attribute. The soft threshold can be varied, depending on how critical the sensed attribute and the target application are.

APTEEN is a hybrid protocol that changes the periodicity or threshold values used in the TEEN protocol according to the user needs and the type of the application. In APTEEN, the cluster head broadcasts physical parameter attributes important for the user. APTEEN sends periodic data to give the user a complete picture of the network. APTEEN also responds immediately to drastic changes for time-critical situations.

Both TEEN and APTEEN are modified to reduce the amount of messages in the network, hence, increasing the lifetime of the network. However, a comparison between TEEN and APTEEN with LEACH and its variants, as in [5] and [6], is not directly suitable. LEACH sends data periodically to the base station while TEEN and APTEEN only send data after a certain threshold. This will result in longer delay times and prolonged network life time. LEACH and LEACH-C delivers more data than TEEN and APTEEN to the base station. Hence, LEACH and LEACH-C consume less energy per message than TEEN and APTEEN. Since TEEN and APTEEN are protocols for longevity only and do not consider the data throughput to the base station, it is beyond the scope of this paper to compare them with our architecture. It is more suitable to compare our architecture with LEACH and its variants because they also send data periodically to the base station.

3. Asymmetric sensor network architecture

Our architecture is based on clusters with a Base Station (BS) with "unlimited" energy and "enough" bandwidth in the backbone channels, see Figure 1. The BSs are connected to each other by wire, wirelessly or both. To be able to turn off the radio of the sensor nodes as long as possible, we propose to use TDMA to schedule the communication of the sensor nodes. Furthermore, we propose to build clusters where the BSs are the masters in the network. Using clusters will ease the scheduling of the sensor nodes. Further, when using clusters we can aggregate data to minimize the communication in the network. The BS can reach all its sensor nodes directly and a similar TDMA scheme as used in LEACH could be used in our topology.

All clusters have a Cluster Head (CH) that can aggregate and fuse data received from sensor nodes in its cluster. CHs are the only sensor nodes that send and forward data to the BS. All CHs may not be able to communicate directly with the BS. Some CHs need other CHs in order to forward the traffic to the BS. For example, CH B in Figure 1 is located on the fringe area, and its radio power does not reach the BS. CH B needs to use CH A to forward its traffic. CH B in its turn has to help CH C with forwarding of traffic. Thus, we propose an asymmetric topology where the BS reaches all its sensor nodes while the sensor nodes might not reach the BS directly.

The BS will make route decisions and manage topology changes for its sensor nodes. The BS will construct a TDMA schedule for its sensor nodes and provide the information to each sensor node about their assigned time slot. The BS will look at other BS schedules and ensure that its sensor nodes do not interfere with adjacent sensor nodes. The sensor nodes only need to focus on their own tasks and thereby save energy that otherwise would be used to, e.g., do extra computations or exchange messages with other sensor nodes, in order to maintain the network topology. The BS will change existing routes to save highly exposed sensor nodes from draining their batteries. When a BS receives a message from a new sensor node, it assigns that node to the most suitable BS. When a BS is assigned a new sensor node, the BS will compute the best route and inform any other concerned sensor nodes about the changes. The BS will also check if the network would benefit from rearranging old routes to new ones. No, or little, knowledge of the network is needed at the sensor nodes. The BS can make optimizations that a pure sensor node network would not consider cost-effective. Issues to be considered by the BS include:

- Mobility: Mobile sensor nodes will make the scheduling decisions more complex.
- Energy: When is it worth to reroute the sensor nodes in order to save energy?
- Optimization: What are the network optimization goals and when do we execute the optimizations?
- New sensor nodes/dead sensor nodes: When to do rerouting and optimizations when a new node enters the cluster or demises?
- New sensor nodes added to the network: Which BS does the sensor node try to send its join request to? Does a sensor node need help from other sensor nodes with forwarding of its whereabouts to the BS?
- Timing issues: After what time can a new sensor node be guaranteed to be inserted into a cluster?
- What happens if a BS disappears or a new BS enters the network?

Depending on the TDMA scheme used, the maximum allowed clock skew will be known. From this, and from knowledge about the drift of the local clocks, the maximum time interval between clock synchronizations can be calculated. This in turn implies a maximum sleep time for the sensor nodes, i.e. how often they must listen to the radio in order to keep their clocks in synchronization with the TDMA schedule.

Some sensor nodes in the network could be scheduled for optimized energy saving, while others could be scheduled for Quality of Service (QoS). In our architecture, we can handle sensor nodes with different demands without e.g., involving the whole sensor network for reorganization. The BS will handle all the extra workload, and only the sensor nodes concerned will have to be rescheduled or reclustered. Depending on the application running on the sensor node, i.e. the requested QoS, the BS will schedule the sensor nodes differently. A sensor node with low QoS demands could/would be scheduled to sleep during several TDMA cycles. Sensor nodes with higher demands could/would be scheduled every TDMA cycle (or more often if necessary). Having sensor nodes with low QoS sleep during several TDMA cycles will increase the delay for topology changes and messages from the sensor nodes to the BS. Different QoS demands in the network imply high complexity. Sensor nodes within a cluster must be grouped in a smart way to e.g., guarantee response time.

4. Simulations

In order to verify our assumptions that forwarding will reduce the amount of energy for large network sizes, we have set up a fixed, single BS, network in NS 2 [1], created with the centralized cluster formation algorithm that LEACH-C uses, see Section 2. The BS does not make any optimizations such as i.e., recalculation of the best cluster formation or, the optimal sleep time. Below, we show that our architecture with asymmetric communication and forwarding of packets outperforms the LEACH protocols with respect to the amount of energy consumed by the sensor node per data packet sent to the BS. Here we assume that the sensor nodes are clock synchronized, and the sensor nodes know their position.

We have set up the system using the MIT uAMPS LEACH ns Extensions (uAMPS) [2]. uAMPS was developed on the Network Simulator platform (NS 2) [1]. Test simulations were performed to verify the LEACH and LEACH-C protocols. We have implemented the LEACH-F protocol in NS 2 and the results were verified based on the simulation results in [4].

	1:st simulation	2:nd simulation
Network size	$100X100$ m	$400X400$ m
BS location, x,y	$50, 175$	$200, 475$
Nodes	100	100
Radio prop. speed	3x10^8 m/s	3x10^8 m/s
Processing delay	50 μs	50 μs
Radio speed	1 Mbps	1 Mbps
Data size	500 bytes	500 bytes

Table 1. Characteristics of the network

First, the simulations were configured as in [4] i.e., a network size of 100x100 meters with 100 nodes randomly distributed and the base station located at position $x = 50$, $y = 175$. That is, the BS was placed 75 meters outside the area where the sensor nodes were deployed. The BS reschedules the CHs every 20:th second. Each node sends a message to its CH during a given slot. The optimal number of clusters for this kind of network is somewhere between 1 and 6 according to [4]. In order to be able to study the behavior of forwarding, we have chosen to use 4 clusters. We placed 2 clusters close to the BS, to forward data from the 2 clusters placed at the back of the network. According to [4], the most energy efficient cluster formation is between 3-5 clusters in a 100x100 meter network. The sensor node starts with 2 Joules of energy and the simulation continues until all the sensor nodes in the network have consumed all of their energy. All sensor nodes have an equal amount of energy when the simulation starts. In order to make comparisons possible, we have used the same channel propagation model, radio energy model and beam forming energy model as in LEACH [4]. The energy consumption of the radio transmitter is according to [4] $\epsilon_{friss-amp} = 10pJ/bit/m^2$ for distances under 87 meters and $\epsilon_{two-ray-amp} = 0.0013pJ/bit/m^4$ for distances over 87 meters. The radio electronics cost/energy was set to $E_{elec} = 50nJ/bit$. The data size was 500 bytes/message plus a header of 25 bytes, $b = (500bytes + 25bytes) * 8 = 4200bits$. The equation for calculating the amount of energy used for sending a message d meters is:

$$E_{Tx} = \begin{cases} b * E_{elec} + b * \epsilon_{friss-amp} * d^2 & : d < 87m \\ b * E_{elec} + b * \epsilon_{two-ray-amp} * d^4 & : d \geq 87m \end{cases} \tag{1}$$

and the amount of energy used when receiving a message is:

$$E_{Rx} = b * E_{elec} \tag{2}$$

Further, all the parameters, such as radio speed, processing delay and radio propagation speed were the same as in [4], see Table 1. However, the energy model can benefit from improvements but is outside the scope of this paper.

In the second simulation, the network size was increased to 400x400 meters. The amount of sensor nodes randomly distributed in the network remained the same as in the first simulation, i.e. 100 nodes. Also in this case, we placed the base station 75 meters outside the monitored area, at location $x = 200$, $y = 475$. According to the equation in [4], the optimal number of clusters for this network size is somewhere between 1 and 24 clusters, considering the energy consumption. Simulations with LEACH show that the most energy efficient cluster formation is between 4 and 5 clusters, see Figure 2. In order to study the behavior of forwarding, we have chosen to use an even number of clusters. We put half of the clusters in the front and the other half in the back of the network, from the BS' point of view. The clusters in the back of the network use the clusters in the front to forward their data to the BS. When using even number of clusters, the

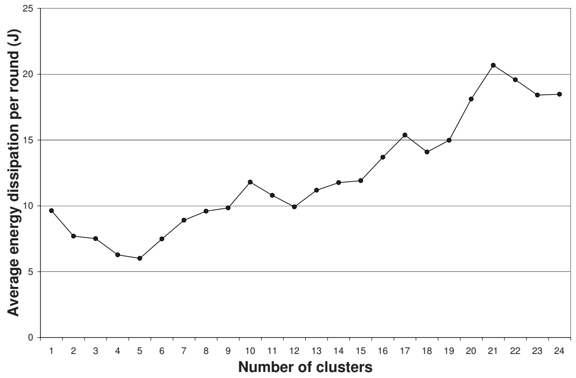

Figure 2. Simulation results showing the average energy consumption per TDMA-cycle in LEACH.

lowest amount of energy is consumed when using 4 clusters, as can be seen in Figure 2. All the parameters, except the BS' location and the network size, are the same as in the first simulation setup, see Table 1.

We used LEACH-Cs centralized cluster formation algorithm to create the clusters in our architecture. The clusters were then manually changed to better suit 4 clusters with forwarding. It is not always the case that the clusters generated by the centralized cluster formation algorithm create cluster formations where forwarding of data can be studied. In some cases it creates one cluster far away from the BS and three clusters beside each other nearby the BS. This was the case when trying to create a suitable cluster formation for our architecture using 4 clusters. However, earlier simulations in LEACH-C with 5 clusters showed a cluster formation suitable for 4 clusters when 3 of the clusters where merged into 2. This cluster formation is also used for LEACH-F in order to simulate the same cluster scenario.

The sensor nodes are scheduled to send their data to a cluster head during a given slot. The cluster heads furthest away from the BS i.e., Cluster C and Cluster D, see Figure 3, were modified to send their aggregated data to the cluster heads in Cluster A and Cluster B respectively, instead of sending it directly to the BS. The cluster heads in Cluster A and Cluster B forwards the aggregated data directly to the BS after receiving it.

The length of the TDMA cycle for a cluster depends on how many nodes there are in the cluster. The length of the TDMA cycle is updated every 20:th second, at the same time as the network is rescheduled. Cluster A and Cluster C might have different TDMA cycle length, due to different number of nodes in the cluster. To simplify the forwarding schedule, we used the longest TDMA cycle of Cluster A and Cluster C plus some overhead as cycle lengths for Cluster A and Cluster C. The same was done for Cluster B and Cluster D.

Data fusion might save energy even more, this is something to consider in future work. The TDMA cycle for the

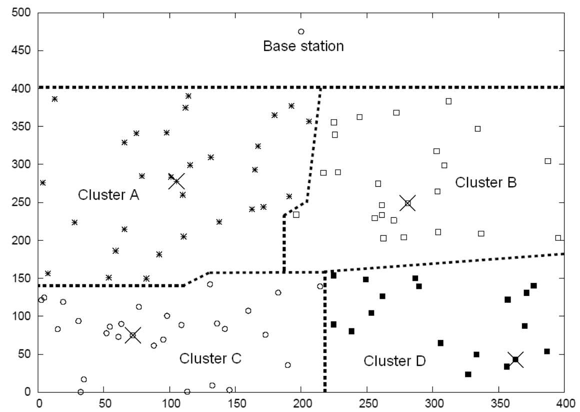

Figure 3. Cluster formation of the simulated network using 4 clusters and a network size of 400x400 meters.

clusters should be optimized, and clusters with different length should be scheduled into a merged TDMA scheme.

5. Results

The results from our experiments with a 100x100 meter scenario, show that our architecture perform almost as well as LEACH-C and LEACH-F, depicted in Figure 4. In spite of the fact that the CHs in our architecture send the data a shorter way towards the BS, the extra receive and send when forwarding data sometimes use more energy than to send it directly to the BS. Our architecture sends almost as much data to the BS as LEACH-C and LEACH-F. The data from the clusters furthest away has a longer delay time before the BS receives the data. This is due to the prolonged TDMA-cycle of the smaller cluster, see Section 4, and due to the extra hop the data needs to travel. Our architecture will perform even better when optimizing the cluster formations and the data routing.

When the network was increased to 400x400 meters, LEACH did not perform well. The nodes furthest away from the BS demised early and data from that area could not be received at the BS. The early drop out of the nodes is due to the radio transmission, draining the node when they are trying to send data to the BS. Our architecture, on the other hand, handles this by sending its data shorter distances. The total amount of energy consumed, E_{tot}, when sending a message to the BS depends on the number, n, of forwarding CHs between the sending CH and the BS. Equation 1 and 2 are used to calculate the total energy consumed E_{tot} as:

$$E_{tot} = \begin{cases} E_{Tx_n} & : n = 0 \\ E_{Tx_0} + \sum_{k=1}^{n} (E_{Rx_k} + E_{Tx_k}) & : n > 0 \end{cases} \tag{3}$$

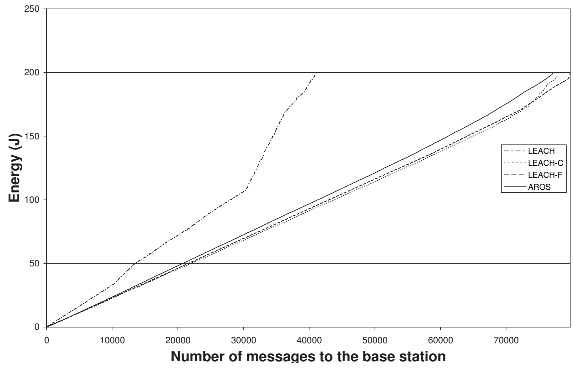

Figure 4. Total data received at the base station per given amount of energy in a 100x100 m large network with 4 clusters.

For example, consider a sending CH located 475 meters from the BS. The amount of energy consumed in LEACH, to send data to the BS is $E_{tot_{LEACH}} \approx 278mJ$, $n = 0$ (3). The amount of energy consumed when using our architecture with one forwarding CH is $E_{tot_{OUR}} \approx 53mJ$ (3). The CH that forwards the data in this example is located half-way between the BS and the sending CH, $d = 237,5m$. As one can see, LEACH consumes more than five times more energy than our architecture.

When comparing how much data the BS receives per Joule of energy in Table 2, we can see that our architecture performs 97% better than LEACH, 28% better than LEACH-C and 32% better than LEACH-F. This is also depicted in Figure 5.

Protocoll	Data/Energy (J)	OUR is
LEACH	$\frac{19160}{204.2} \approx 93.8$	97% better
LEACH-C	$\frac{29240}{202.2} \approx 144.6$	28% better
LEACH-F	$\frac{28581}{203.7} \approx 140.3$	32% better
OUR	$\frac{37979}{205.2} \approx 185.1$	

Table 2. Data received at base station per unit energy (J)

Figure 5 also shows that when LEACH-C and LEACH-F have used all of its energy and demises, our architecture still has 25% of its energy left and 54% of its energy left when LEACH demises. In Figure 6 we can see that our architecture has more than 73% of its nodes alive when LEACH-F has zero nodes alive in the network. When LEACH-Cs network demises our architecture has 68% of its nodes alive and if we compare to LEACH, our architecture has approximately 88% of its nodes alive. This results in a situation where the BS can receive at least 9000 more messages from the network

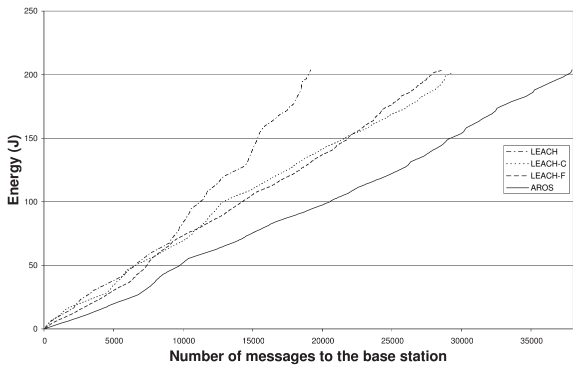

Figure 5. Total data received at the base station per given amount of energy in a 400x400 m large network with 4 clusters.

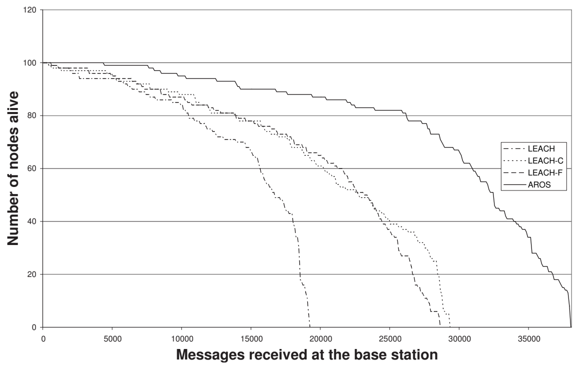

Figure 6. Number of nodes alive compared to the amount of messages received at the base station in a 400x400 m large network with 4 clusters.

before all energy is consumed.

The energy consumed in the network is evenly distributed among the nodes in our architecture. Clusters far away from the BS in our architecture will survive until the end and continue to gather information. In contrast to LEACH-F where only the clusters closest to the BS are alive at the end and the clusters far away are demised, see Figure 7. At time 340, when Cluster D in LEACH-F is demised, LEACH -F has only 40% of its nodes left in the network. Our architecture on the other hand still has 61% of its nodes left in Cluster D and 56% of its nodes left in the network. This implies that our architecture still can collect data from the whole network area but LEACH-F can not because one cluster has demised. At time 400, when LEACH and LEACH-C demises, our architecture still collects data from the whole network with 28% of the nodes left in Cluster D, 30% of the nodes left in Cluster A, 29% of the nodes left in Cluster B and 54% of the nodes left in Cluster C. LEACH-F can only collect data from Cluster A, B and C with 20%, 29% respective 36% of its

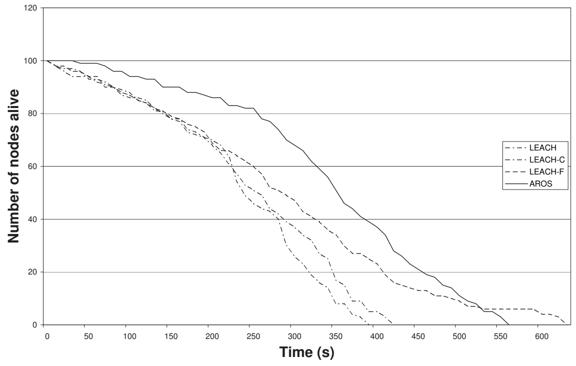

Figure 7. The amount of nodes alive over time in a 400x400 m large network with 4 clusters.

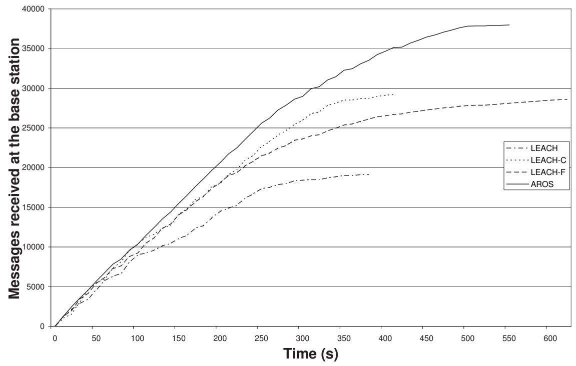

Figure 8. Number of messages received at the base station over time in a 400x400 m large network with 4 clusters.

nodes left alive. Until time 440, our architecture is able to collect data from the whole network with nodes alive in all 4 clusters. This is 30% longer time than with LEACH-F that only collects data from 3 clusters, Cluster A, B and C. At time 540 LEACH-F has one cluster left alive, Cluster A, with 6 nodes very close to the BS. Our architecture has 2 clusters left, Cluster A and C, with 2 respective 3 nodes left alive.

Reducing the energy consumption for sending data, each nodes' lifetime is prolonged and more data can be sent to the BS, as showed in Figure 8. This can also be seen in Figure 5, the total data received at the BS per given amount of energy. As a result for having more nodes alive our architecture can gather more data from a larger network area.

If we compare our architecture and LEACH-F at time 340 again, when the first cluster demises in LEACH-F, we can se that our architecture gathers 80% more data until the whole network demises. When looking at the time after our architecture has demised, LEACH-F only gathers 468 messages during the last 75 seconds, and that data is only from one cluster closest to the BS, as mentioned earlier. At time 500 LEACH-F has almost no energy left and the few nodes left in the last cluster sends very few messages, see Figure 9. This means that LEACH-F prolongs the network lifetime

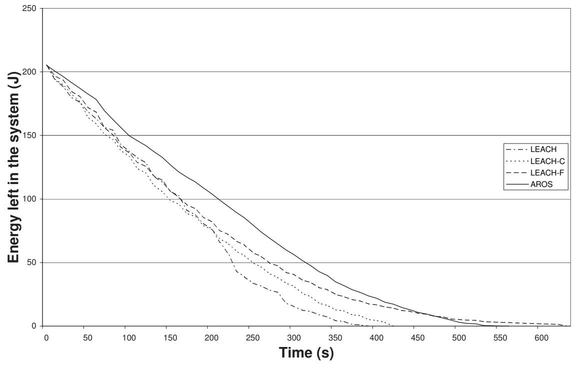

Figure 9. Energy left in the system over time in a 400x400 m large network with 4 clusters.

collecting data from a very small area. Even though LEACH-F lives slightly longer, our architecture collects data from sensors widely spread over a larger network area during its whole life time.

6. Conclusions

We have presented an asymmetric communication architecture for sensor networks. In our architecture, a base station acts as a master for the sensor nodes and can reach all its sensor nodes in one hop. However, all sensor nodes might not reach the base station in one hop. In order to minimize the communication between the sensor nodes, the base station will do route decisions and manage topology changes. The base station will also make a TDMA schedule for its sensor nodes and inform each sensor node about their assigned time slot. In this paper, the base station does not make any optimizations such as e.g., recalculation of the best cluster formation, sleep time. Our architecture is similar to LEACH, a cluster based protocol, and the clusters have cluster heads that can aggregate and fuse data received from the sensor nodes in its cluster.

In simulation studies, we have compared our architecture to LEACH and its two variants, LEACH-C and LEACH-F. 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. The simulations have shown that our architecture outperforms the LEACH protocols in large networks and that our architecture performs almost as well as the LEACH protocols in small networks.

In these simulations we have not used any advanced features of the base station (such as e.g., reclustering and rescheduling). Instead we have studied static network configurations. Even without advanced features, we have shown that our architecture is up to 32% better than LEACH-F, 28% better than LEACH-C and 97% better than LEACH in collecting data to a base station with the same total amount of energy. Because the energy consumed in the our architecture network is evenly distributed among the nodes, our architecture can collect data from sensors widely spread over a larger network area. Clusters far away from the BS will live longer and continue to gather information until the end. Our architecture has

25% of its energy left when the other LEACH protocols have used all of their energy and demised. We have shown, after sending the same amount of data to the BS, that our architecture has more than 73% of its nodes alive when LEACH-F have zero nodes alive in the network.

The simulations presented in this paper were performed in order to show that our architecture performs better than LEACH in large networks. For even larger networks, when using multihop, our architecture will perform even better. Optimizations and more complex TDMA scheduling will be investigated in future work.

Our next step is to design a TDMA scheduler for our architecture multihop networks and a base station implementation in NS in order to make dynamic simulations. The TDMA scheduler will optimize the network for energy saving, cluster formations and routing. Further, we will evaluate what types of scenarios our architecture is suitable for.

References

[1] NS 2, Network Simulator. World Wide Web, `http://www.isi.edu/nsnam/ns/ns-build.html`, 2004.

[2] MIT uAMPS LEACH ns Extensions. World Wide Web, `http://www.ece.rochester.edu/~wheinzel/code/index.htm`, 2005.

[3] M. Gerla, T. Kwon, and G. Pei. On Demand Routing in Large Ad Hoc Wireless Networks with Passive Clustering. *Procedings of IEEE WCNC 2000*, Sep 2000.

[4] W. Heinzelman. *Application-Specific Protocol Architectures for Wireless Networks*. PhD thesis, Massachusetts institute of technology, June 2000.

[5] A. Manjeshwar and D. P. Agrawal. TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. Parallel and Distributed Processing Symposium., Proceedings 15th International, April 2001.

[6] A. Manjeshwar and D. P. Agrawal. APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks. Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002, April 2002.

[7] 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.

[8] G. Pei and C. Chien. Low Power TDMA in Large Wireless Sensor Networks. *Military Communications Conference*, 1:347–351, 2001.

[9] G. J. Pottie and W. J. Kaiser. Wireless Integrated Network Sensors. *Communications of the ACM*, 43(5):51–58, May 2000.