Reducing Energy Consumption in a Sheep Tracking Network Using a Cluster-based Approach

Ragnar Stølsmark
Institute of Computer Science and Electrical Engineering
University of Stavanger
Stavanger, Norway
rstolsmark@live.no

Erlend Tøssebro
Institute of Computer Science and Electrical Engineering
University of Stavanger
Stavanger, Norway
erlend.tossebro@uis.no

***Abstract*—Sheep tracking ease the work of the farmer when retrieving the sheep. The current commercial sheep tracking solution is popular but not energy-efficient. It only uses GSM and GPS and has no interaction between the sheep. Sheep often walk in clusters. If a farmer knows the location of all clusters and also which sheep are in each cluster, he does not need to know the exact location of each sheep. By using a new cluster-based localization and data retrieval approach this paper show, through energy measurements and simulations, that it could be possible to reduce the average energy consumption by more than 50 % in flocks were the sheep walks in large clusters. The reduced energy consumption could be used to either increase the update frequency or to reduce the battery capacity. The cluster–based approach can also eliminate the need for GSM and GPS modules on part of the sheep nodes, thereby making the system more affordable for farmers. The reduced energy consumption and cost makes the solution described in this paper better than the currently available commercial one.**

Keywords-Wireless Sensor Networks; Animal Tracking; Energy Consumption.

I. INTRODUCTION

Sheep farmers in Norway send their sheep to graze on the fertile mountain grass during the summer. This is important since the grass on the farm then is allowed to grow freely and can be harvested and used as food for the sheep during the winter. One of the big problems with this custom is to locate the sheep when the summer is over. The common method for sheep localization has been that the farmer searches for the sheep manually. This search will typically start in the area where the farmer thinks the sheep will be located and expand outwards to the less likely areas. The search continues until all sheep are found, or the farmer grows tired of looking for them. This process can take a week or more of walking in often challenging terrain.

In the last few years there have come a solution to this inefficient retrieval method. Telespor [1] is a commercially available system based on the electronic shepherd research project [2]. It tracks the sheep using GPS and sends the position of the sheep back to the farmer using the GSM network. However, it is far from perfect. The biggest problem is cost. Each unit cost approximately € 200, which is too expensive for most farmers, at least if they want to equip their entire flock with these devices. Another limiting factor is the battery capacity. The batteries should not have to be replaced during the season which last around 100 days. Since there is a size and cost limitation on the batteries, it is important to reduce the energy consumption of each update to allow for more position updates and thereby increase the utility of the system.

This paper looks at the possibility of mitigating both the cost and energy consumption problem by taking advantage of the fact that sheep tend to move in clusters. In the current Telespor system every sheep find their own position using GPS. The farmer however, only needs to know which sheep are in each cluster and the position of one of the sheep in every cluster. Knowing this, it is possible to construct an algorithm where only the sheep with the highest battery level in a cluster use the GPS to find its location. The rest of the cluster only needs to report which cluster they belong to. This will reduce the energy consumption since GPS localization requires fairly high amounts of energy.

Using a cluster-based approach it is also achievable to have some of the sheep only carry radio transceivers, not GPS receivers and GSM transceivers,. It will only be possible to locate these cheaper equipped sheep when they are in a cluster with at least one sheep with a GPS receiver. It can be argued that this information is of limited value when retrieving the sheep. The farmer has to retrieve every cluster anyway and will therefore find these extra sheep. It can however be useful during the season for a farmer to know if a sheep is doing well. If for instance a sheep that has been following a cluster for some time suddenly disappears, it might be a signal that it is injured or dead.

Two systems have been developed and tested during this research project. The difference between the two systems lies in how the sheep transport data back to the server. In the system called Distributed GSM, every sheep reports their position. In the Centralized GSM approach, only the sheep responsible for finding the GPS position of the cluster, report its position along with the information of which sheep are in the cluster.

This paper is structured as follows: Section II contains related work. System design and equipment is covered in

Section III. Section IV contains the results found by analyzing the different systems, measuring the energy consumption and performing simulations. Section V concludes the paper and looks at future work.

II. RELATED WORK

The Telespor [1] system described in [2] is used as a base line reference throughout this paper and is therefore a closely related work. Telespor is described in Section III as a benchmark version was made to work on the same equipment used in the rest of this paper. This benchmark version was built on the principles described in the electronic shepherd paper.

Stølsmark and Tøssebro looked at the possibility of localization via trilateration in a sheep tracking network [6]. Some of the sheep would be equipped with GPS and the rest of the flock would find their position by using GPS sheep as beacons for RSSI-based trilateration. They found that the position estimates were not accurate enough to be useful. This was mainly due to the RSSI being highly dependent on non-distance related factors, such as weather and topography.

Huircán et al [7] tried to locate cattle in a field using RSSI. They were able to achieve a high level of accuracy but only by having a dense beacon placement, with around 80 m between each beacon. This makes such a localization scheme unsuitable for a large area where sheep typically graze.

In the Zebranet project [8, 9] Juang et al. tried to track zebras in an area without GSM coverage. The zebras would store logs of their own positions and exchange these logs when they met other zebras. The logs could then be downloaded by the data collecting scientists driving around the savannah with radio transceivers.

The WildSensing project [10] tries to monitor badgers in the wild in an energy-efficient manner. Dyo et al. equipped the badgers with RFID tags and set up RFID readers at strategic locations where the badgers would likely be. Since the badgers sleep during the day, the readers would only be active at night. Markham et al. even tried monitoring the badgers in their underground burrows using magnetic fields [11].

Polastre et al. monitored seabirds on the remote Great Duck Island [12] using a tiered architecture to save energy and money. This meant that the wireless sensor network nodes would send their data to a base station on the island which would provide WAN connectivity to send data back to the scientists.

Surveys on different localization methods can be found as part of the extensive wireless sensor network survey performed Yick et al. [13] and in the survey by Akyildiz et al. [14].

Much research effort has been put into localization, but different applications require different solutions. A solution that works for badgers is not necessarily suitable for sheep. In the case of sheep, it is possible to design a system that benefits from the fact that lambs follow their mothers. The application-specific requirements will make it difficult to design one tracking system that can handle every situation. It could be possible to identify a few different scenarios that would fit most applications and design solutions to them. An example of such a scenario could be tracking a large animal in an area without GSM coverage. Identifying the suitable scenarios and creating solutions to them is still an open problem.

III. SYSTEM DESIGN AND EQUIPMENT

A sheep tracking system has to be able to provide the farmer with the position of his sheep, at regular time intervals, throughout an entire season, without any manual local intervention. Any method requiring manual synchronization or battery replacement is therefore not an option. It should also be possible to change the frequency of position updates during the season since it is more important to have frequent updates when collecting the sheep.

This paper looks closer at three possible sheep tracking solutions. They all use GSM for data transport between the sheep and the farmer. They also use GPS as part of the localization process. The difference between them lies in the amount of interaction between the sheep in a flock.

A. Telespor

Telespor is the system farmers are currently using to locate their sheep. It is a simple but working solution with no interaction between the different sheep in a flock. At regular intervals the sheep calculates their location using GPS and sends this location to a server via the GSM network. The benefit of this approach is that since it does not communicate between the sheep, it is not necessary to equip the sheep with an additional transceiver and antenna. One of the problems the farmers have reported with the Telespor system is the poor performance in areas with little GSM coverage [3]. Sometimes, it could take weeks between sheep position updates. The results presented in [4] show that adding sheep to sheep communication equipment is a good solution to the coverage problem. Therefore it is reasonable to argue that sheep to sheep transceiver equipment should be added anyway, and thereby making it possible to implement the two other systems studied in this paper.

Telespor is used as a baseline reference when studying the performance of the different systems. It is especially interesting to look at how much energy consumption can be reduced by using a cluster-based positioning approach, to see if it is worth the effort and added complexity. Algorithm 1 describes the Telespor solution.

```
On init(){
        SleepUntil(first update);
}
On update(){
        pos = GPS.getPosition();
        GSM.SendToServer(pos, id);
        SleepUntil(next update);
}
```

Algorithm 1. The Telespor algorithm.

B. *Distributed GSM*

The Distributed GSM solution is identical to the Telespor system in the way that each sheep send their position back to the server. The positioning differs due to the cluster-based approach. The sheep are synchronized and wakes up from sleep at regular intervals. Upon wake up, every sheep will calculate a delay based on the amount of energy left in their battery. In experiments 20 seconds was found to be a sufficient maximum value for this delay. If a leader message is received during this time the receiving sheep will become a follower of the leader sheep. If a sheep does not receive any leader message before the delay expire, it will become a leader sheep itself and send out a leader message. A leader sheep and all of its followers are considered to be one cluster. The leader will locate its position using GPS and broadcast this position to the followers. The followers, as well as the leader, will report the same position back to the server via the GSM network. Distributed GSM require every sheep to individually report its own position, no in-network aggregation or collection is performed in this solution. The Distributed GSM algorithm can be seen in Algorithm 2.

```
On init(){
        Clock.Synchronize();
        SleepUntil(first update);
}
On update(){
        delay = CalculateDelay(battery.status);
        Wait(delay);
        if(HasReceived(leaderMessage)){
           //become follower
           leader = leaderMessage.leader;
        }else{
           //become leader
           leader = this;
           leaderMessage.leader = this;
           Broadcast(leaderMessage)
           pos = GPS.getPosition();
           Broadcast(sheepid, pos);
        }
        GSM.SendToServer(leader.id, id);
        SleepUntil(next update);
}
```

Algorithm 2. The Distributed GSM algorithm.

C. *Centralized GSM*

Centralized GSM is identical to Distributed GSM when it comes to leader choice and sheep localization. However, instead of each sheep individually reporting its position to a server via the GSM network, the follower sheep send a notification to the leader. The leader sends its position along with the list of followers to the server. This has the potential of saving energy since the energy consumption of each follower is reduced at the expense of increased energy consumption for the leader. The increased consumption at the leader is due to more communication and an extra listening period when waiting for the messages from the followers. It is also possible to use the Centralized GSM solution to reduce costs by not equipping every sheep with GSM transceivers. Centralized GSM is described in Algorithm 3.

```
On init(){
        Clock.Synchronize();
        SleepUntil(first update);
}
On update(){
        delay = CalculateDelay(battery.status);
        Wait(delay);
        if(HasReceived(leaderMessage)){
            //become follower
            SendToLeader(id);
            SleepUntil(next update);
        }else{
           //become leader
           numFollowers = 0;
           followers = Ø;
           Wait(followerdelay);
           For(each receivedFollower){
              followers.Add(receivedFollower) ;
              numFollowers++;
           }
           pos = GPS.getPosition();
           GSM.SendToServer(pos, id,
           numFollowers, followers);
           SleepUntil(next update);
        }
}
```

Algorithm 3. The Centralized GSM algorithm.

IV. RESULTS

To find the best algorithm, three different investigation methods were used: Analysis, experiments and simulations.

A. *Analysis of the different solutions*

To be able to better understand the difference in energy consumption between the different solutions an analysis of them was performed. The analysis tries to calculate the expected number of updates each sheep can perform before running out of battery.

TABLE 1. LIBELIUM WASPMOTE ENERGY FIGURES.

Battery capacity (Bat)	6600 mAh
GPS consumption (GPS)	27.5 mA
GSM consumption (GSM)	100 mA
Processor consumption (Proc)	9 mA
XBee broadcast consumption, full power (XBc)	160 mA
XBee receive consumption (XRcv)	73 mA

The analysis use the energy consumption figures for the Libelium Waspmote [5]. This is the hardware that was used to measure the energy consumption of the different algorithms. The relevant hardware figures are listed in Table 1.

The analysis is based on a scenario where the sheep is always part of the same cluster and every sheep starts with a fully charged battery. Defining N as the number of sheep in a cluster, each sheep will be leader in 1/N of the execution cycles. U is defined as the average number of position updates a sheep can perform before running out of battery power. The following formulas were used for the different algorithms:

1) *Telespor*

$$U = \frac{\mathrm{Bat}}{\mathrm{GPS} + \mathrm{Proc} + \mathrm{GSM}} \tag{1}$$

2) *Distributed GSM*

$$U = \frac{\mathrm{Bat}}{\frac{1}{N} * (\mathrm{GPS} + 2 * \mathrm{XBc}) + \frac{N-1}{N} * (2 * \mathrm{XRcv}) + \mathrm{GSM} + \mathrm{Proc}} \tag{2}$$

3) *Centralized GSM*

$$U = \frac{\mathrm{Bat}}{(\frac{1}{N} * (\mathrm{GPS} + 2 * \mathrm{XBc} + (N-1) * \mathrm{Xrcv} + \mathrm{GSM}) + \frac{N-1}{N} * (2 * \mathrm{XRcv} + \mathrm{XBc}) + \mathrm{Proc})} \tag{3}$$

When performed with a varying number of sheep in the cluster, the analysis was able to provide some insight into the properties of the different algorithms. Telespor is the preferred algorithm when there is only one sheep in a cluster, making it good for small flocks. For clusters with more than one sheep Distributed and Centralized GSM is better than Telespor. Distributed and Centralized GSM has almost the same energy consumption, however the gap between them increase with cluster size in favor of Centralized GSM. The analysis results can be viewed in Fig. 1.

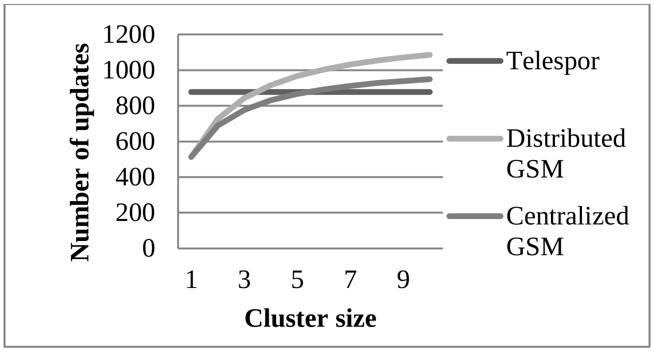

Figure 1. Analysis of average number of updates per sheep with different cluster sizes.

One algorithm was analyzed, in addition to the three mentioned earlier. This was a Centralized GSM algorithm were the nodes would broadcast every message they received to extend it into a multi-hop network, since this could increase flock size. It was decided that this algorithm would be dropped after the analysis showed that it performed worse than the other algorithms, especially in clusters with more than 10 sheep. Another point is that if the cluster covers a very large area, it gets harder for the farmer to find the sheep. With a multi-hop network it becomes more difficult to define the maximum area of a cluster. Limiting the number of hops could be a possible solution.

B. Measurements of Energy Consumption

The three different systems were implemented on the Libelium Waspmote [5] wireless sensor network platform. This is a platform were different modules can be attached when needed. The GPS and GSM modules were used during the energy consumption experiments. The GSM communication was performed by means of GPRS data packets with a maximum payload of 100 bytes. If a sheep cluster had more than 30 sheep, it would have to send additional packets and thereby slightly increase energy consumption. The communication between the nodes was carried out using an XBee 868 MHz transceiver attached to a 4.5 dBi antenna. This has five different power levels. On the highest power level it has an output of 315 mW and a mean range of 515 m [4]. The transceiver was set to transmit at the lowest power level during the energy measurements. This corresponds to an output of 1 mW, with an unmeasured range shorter than 515 m. The energy consumption analysis showed that the choice of power level was not a significant factor in the total energy consumption. There was only a 2% increase in battery life at the lowest power level compared to the highest. The battery used in the test had a voltage of 3.7 V and a capacity of 6600 mAh. All tests were performed with the nodes stationary, in close proximity and in an office environment.

TABLE 2. ENERGY CONSUMPTION MEASUREMENTS

Node type (number of nodes in test)	Average battery level percentage decrease per update	Standard deviation
Telespor (1)	0.5472	3.2
Distributed GSM leader (1)	0.1445	0.7
Centralized GSM leader (1)	0.4043	0.9
Distributed GSM leader (2)	0.2418	0.8
Distributed GSM follower (2)	0.0875	0.6
Centralized GSM leader (2)	0.7576	1.5
Centralized GSM follower (2)	0.0400	1.3
Distributed GSM leader (3)	0.3056	1.0
Distributed GSM follower (3)	0.0920	0.7

Six different setups were used for the experiments: Single node Telespor, Distributed GSM and Centralized GSM, two nodes running Distributed GSM and Centralized GSM and three nodes running the Distributed GSM algorithm. Every time one cycle of the algorithm had been completed the nodes would start the algorithm over again without sleeping, as the sleeping energy consumption should be equal among the three algorithms. All nodes reported their battery level back to the server in the same message used for reporting position. In the Centralized GSM case this information was sent via the current leader node. Each setup was tested with approximately 100 updates per node. The average decrease in battery level per update is displayed in Table 2 along with the standard deviation of the measurements. The most surprising result was the big difference between the energy consumption between the Centralized and Distributed GSM algorithms. It seems like the GSM use more energy than assumed in the analysis, however some of the difference might be caused by the battery level measurements not being 100 % accurate. Sometimes the battery level would actually increase between updates. This factor is probably the reason why Telespor has higher measured energy consumption than the Leader in the Distributed GSM algorithm. The Distributed GSM leader algorithm does everything the Telespor algorithm does, but also sends out two XBee messages and waits for synchronization with potential followers. This leads to the conclusion that some of the energy measurements must be wrong. Therefore we have chosen to conservatively set the Telespor energy consumption as 80% of the measured Distributed GSM consumption when comparing the algorithms in simulations.

C. Simulation Setup

The results from the energy consumption measurements was used as input to a Java simulator built for the purpose of evaluating the three different algorithms in a realistic sheep grazing scenario with between 50 and 250 sheep. The simulator placed the sheep randomly in a landscape measuring 5000 x 5000 meters. Every sheep was equipped with a transceiver that had a range which followed a gaussian distribution with a configurable average.

Figure 2. Libelium Waspmote [5].

Once placed, the sheep would discover the other sheep within their transceiver range and form clusters. The sheep would then start executing the leadership choice part of the two cluster-based algorithms. With a role as leader or follower the sheep would then deduct energy from the battery accordingly. This cycle of leader choice and energy deduction would continue until no sheep had any energy left in their battery. Each simulation scenario was repeated 1000 times and the figures presented in the results section are averages of these simulations. The simulations focus on number of updates instead of time until battery depletion. This is because all algorithms use the Waspmote hibernate mode between updates. In this mode the Waspmote consumes no energy from the main battery, since it only use the auxiliary clock battery to run the real-time clock. Therefore it is not necessary to consider the update interval when comparing the algorithms.

The Telespor solution was not simulated since it has no interaction between sheep. This means that it will have the same energy consumption no matter how the sheep are distributed. A simple calculation was done instead. If one Telespor update costs on average 0.244 % (80% of the measured consumption of the Distributed GSM leader) of the total energy in the battery, the battery would last approximately 410 updates.

D. Simulation Results

1) Effect of Transceiver Range: The transceiver range is adjustable by changing the transceiver power level. It is interesting to look at how this range affects the energy consumption. A shorter range will give smaller clusters and thereby increase the localization accuracy but it will also consume more energy. To test this, simulations with different transceiver ranges and a constant flock size of 250 sheep were performed. Table 3 shows the average number of updates per sheep in these simulations while Table 4 displays the average error.

The results indicate that it is possible to double the battery life by increasing the transceiver range from 100 to 500 m. An average error of approximately 300 m might be unacceptable, especially in areas with limited visibility such as forests.

TABLE 3. AVERAGE NUMBER OF UPDATES PER SHEEP WITH DIFFERENT TRANSCEIVER RANGES

Transceiver range/std. dev.	Distributed GSM	Centralized GSM
100 m/34 m	380	168
200 m/68 m	483	245
300 m/103 m	578	335
400 m/138 m	659	429
505 m/174 m	728	530

TABLE 4. AVERAGE ERROR [M] FOR EACH UPDATE WITH DIFFERENT TRANSCEIVER RANGES.

Transceiver range/std. dev.	Distributed GSM	Centralized GSM
100 m/34 m	17	20
200 m/68 m	77	90
300 m/103 m	149	165
400 m/138 m	223	234
505 m/174 m	299	318

Choosing a transceiver power level that gives a range of 300 m is a good trade-off between energy consumption and accuracy. When comparing the two algorithms it seems like the average error is quite similar.

The battery lasts longer using Distributed GSM than Centralized GSM, especially with a short transceiver range. There are simply not enough followers per leader to outweigh the added leader cost of the Centralized solution.

Clustering of Sheep: The two new algorithms presented in this paper use a cluster-based approach for localization and, in the Centralized version, also for data retrieval. Therefore, it is interesting to look at what effect the average cluster size has on the performance of the algorithms. To simulate sheep flocks with different tendencies to move in clusters, the sheep placement method was changed. First, a number of clusters were given a random position. Then, the sheep would be added to a random cluster and given the same position as that of the cluster. The simulations were performed with 250 sheep in the flock and the transceiver range set to 300 meters with a standard deviance of 103 meters. The average number of updates per sheep can be seen in Fig. 3. The centralized algorithm becomes better than the distributed when cluster size increases. The average energy consumption curves intersect at an average cluster size of approximately 8 sheep. This is not an unrealistic cluster size, especially if lambs are fitted with transceivers as well. The lambs always follow their sheep mother unless there has been an accident. Since each sheep typically have 2 lambs, there only needs to be 3 separate sheep in a cluster to get a total of 9 transceivers. If the lamb also carry transceivers the centralized algorithm is preferable. If not, the distributed algorithm will probably perform better, unless the flock is very large or in a small area. With a very small area there may not be much use for a sheep tracking system anyway.

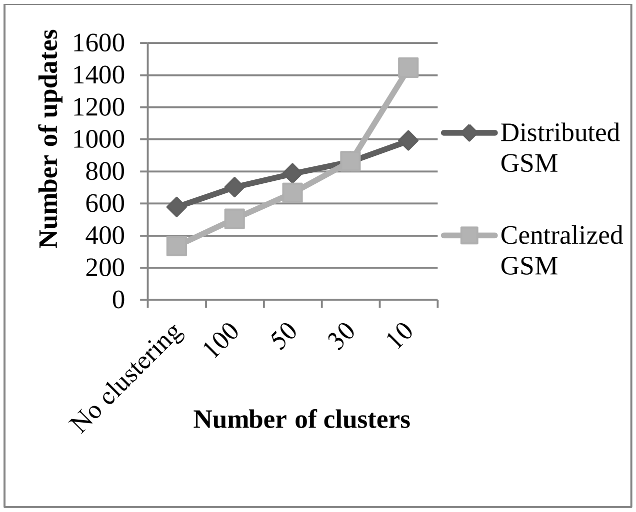

Figure 3. Average number of updates per sheep with different number of clusters.

2) GPS percentage: The sheep that are not leaders do not use GPS to find their position. To save money it could be possible to drop the GPS module from some of the sheep. These will then only have the possibility to be followers and can not find and report their position unless they are within range of another sheep with GPS. If this approach is to be implemented, it would be best to use the Centralized algorithm since it would then be possible to also abandon the GSM module on the nodes without GPS. The success of this method can be measured using the update failure ratio of the non-GPS nodes. The update failure ratio is the number of updates failed due to being out of range of a GPS-sheep, divided by the total number of updates performed by non-GPS nodes. Table 5 shows the update failure ratio for different flock sizes and number of clusters with a GPS percentage from 20-80 %. The transceiver range is kept at 300 m, with a standard deviance of 103 m and the centralized GSM algorithm is used. The GPS sheep are randomly chosen from the set of sheep. The results show that it is possible to drop the GPS and GSM modules on some of the sheep if the flock is big enough. 50 sheep is too few, but with more than 100 sheep and some clustering it is a good solution. Especially if lambs are included among those 100, since that increase the clustering effect.

TABLE 5. AVERAGE NON-GPS UPDATE FAILURE RATIO FOR DIFFERENT FLOCK SIZES AND NUMBER OF CLUSTERS.

Flock size and number of clusters	20 % GPS	50 % GPS	80 % GPS
50 sheep and no clustering	0.89	0.74	0.62
100 sheep and no clustering	0.79	0.55	0.39
250 sheep and no clustering	0.55	0.23	0.10
50 sheep and 30 clusters	0.65	0.33	0.18
100 sheep and 30 clusters	0.41	0.11	0.03
250 sheep and 30 clusters	0.12	0.01	0.00

TABLE 6. AVERAGE NUMBER OF UPDATES PER SHEEP WITH DIFFERENT FLOCK SIZES

Flock size	Distributed GSM	Centralized GSM
50 sheep	409	189
100 sheep	469	235
150 sheep	514	272
200 sheep	549	305
250 sheep	578	335

A good solution could be to equip the sheep mothers with GPS and the lambs with no GPS, since lambs would be able to report their position as long as they stay with their mother.

3) Flock size: The flock size is a factor in choice of algorithm, but not as important as the clustering effect. However, a bigger flock can lead to bigger clusters. The simulation results in Table 6 show the average number of updates with different flock sizes. The simulations were performed with no clustering effect and a 300 m transceiver range. The performance varies with flock size, but with no clustering effect the Distributed GSM outperforms the Centralized GSM for all common flock sizes. For flocks with less than 50 sheep and no clustering effect the most efficient algorithm is Telespor.

V. CONCLUSION AND FUTURE WORK

The energy measurements and simulations showed that it is possible to reduce the energy consumption of a sheep tracking network by more than 50% by using the Centralized GSM algorithm in a sheep flock with big clusters. In small flocks the Telespor solution is still the best since it has the lowest cost in clusters with only one sheep. In flocks with an average cluster size of approximately five sheep the Distributed GSM is the most energy efficient algorithm. The Centralized GSM is still preferable in these situations due to the possible cost reduction of not having to equip every sheep with GSM modules.

The best setup for a typical farmer with a flock consisting of more than 100 animals will be to equip all the sheep mothers with full functionality GSM and GPS nodes running the Centralized GSM algorithm. The lambs could then be equipped with a simpler version without GSM and GPS. This will reduce cost without increasing the update failure ratio since lambs follow their mothers.

The next step in this project is to simulate and check the efficiency of the Centralized algorithm using real-world sheep location data. Telespor has allowed us access to a data set with over 400 000 sheep positions. Using this data will give a clearer indication of how much performance gain can be expected. It will be interesting to look at how big the clustering effect is, as some sheep prefer walking alone while others are more social. With real-world data it is possible to simulate movement and thereby look at the possibility of enhancing the algorithm if some sheep stay together in a group over time.

The final step in this project should be a real-world deployment with as many sheep as possible. This is the only way to really prove the performance of the Centralized GSM algorithm.

REFERENCES

[1] Telespor. Company Website. Available: http://www.telespor.no. Last accessed: 24th Apr 2012.

[2] B. Thorstensen, T. Syversen, T-A. Bjørnvold, and T. Walseth, "Electronic shepherd – a low-cost, low bandwith, wireless network system," Proc. of the 2nd international conference on Mobile systems applications and services, ACM, 2004, pp. 245-255.

[3] A. S. Haugset and G. Nossum, "Erfaringer med bruk av elektronisk overvåkningsutstyr på sau i 2010," Notat 2010:17 Trøndelag Forsking og Utvikling AS, 2010.

[4] R. Stølsmark and E. Tøssebro, ”Can a multi-hop solution improve GSM coverage for tracking networks?,” Proc. of the european conference on the use of modern information and communication technologies (ECUMICT 2012), Nevelland v.z.w Landegem, Mar 2012, pp. 95-106.

[5] Libelium Waspmote: Product information. Available: http:// http://www.libelium.com/products/waspmote. Last accessed 24th Apr 2012.

[6] R. Stølsmark and E. Tøssebro, “Uncertainty in trilateration: Is RSSI-based range estimation accurate enough for animal tracking?,” Proc. 1st international conference on sensor networks (SENSORNETS) 2012, ScitePress, Feb 2012, pp. 237-241.

[7] Huircán, J. I., Muñoz, C., Young, H., Von Dossow, L.,Bustos, J., Vivallo, G. and Toneatti, M., “ZigBee-based wireless sensor network localization for cattle monitoring in grazing fields,” in Computers and Electronics in Agriculture vol. 74(1), Elsevier, 2010, pp. 258-264.

[8] Juang P., Oki H., Wang Y., Martonosi M., Peh L.S. and Rubenstein D., “Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with ZebraNet,” Proc. of the 10th international conference on Architectural support for programming languages and operating systems (ASPLOS-X), ACM, 2002, pp. 96-107.

[9] Zhang P., Sadler C. M., Lyon S. A. and Martonosi M., ”Hardware design experiences in ZebraNet,” Proc. of the 2nd international conference on Embedded networked sensor systems (SENSYS ‘04) , 2004, ACM, pp. 227-238.

[10] Dyo V., Ellwood S. A., Macdonald D. W., Markham A., Mascolo C., Pásztor B., Scellato S., Trigoni N., Wohlers R. and Yousef K., “Evolution and sustainability of a wildlife monitoring sensor network,” Proc. of the 8th ACM Conference on Embedded Networked Sensor Systems (SENSYS ’10), 2010, ACM, pp. 127-140.

[11] Markham A., Trigoni N., Ellwood S. A. and Macdonald D. W., “Magneto-inductive tracking of underground animals” Proc. of the 8th ACM Conference on Embedded Networked Sensor Systems (SENSYS ’10), 2010, ACM, pp. 365-366.

[12] Polastre J., Szewczyk R., Mainwaring A., Culler D. and Anderson J., “Analysis of wireless sensor networks for habitat monitoring” in Wireless Sensor Networks, 2004, Springer, pp 399-423.

[13] Yick J., Mukherjee B. and Ghosal D., “Wireless sensor network survey” in Computer Networks vol. 52(12), 2008, Elsevier, pp. 2292-2330.

[14] Akyildiz I. F., Su W., Sankarasubramaniam Y., Cayirci E., “Wireless sensor networks: a survey,” in Computer Networks vol 38(4), 2002, Elsevier, pp. 393-422.