WSN for Cattle Monitoring – Related Work and Design Proposal

Herman Lundkvist*, Attila Para†, Vipul Mahawar‡ and Omar Elshal§
*Student Id: 0973534
Email: h.e.lundkvist@student.tue.nl
†Student Id: 0975194
Email: a.para@student.tue.nl
‡Student Id: 0980015
Email: v.mahawar@student.tue.nl
§Student Id: 0980295
Email: o.a.m.elshal@student.tue.nl

I. Introduction

A. Background

The livestock industry constitutes a considerable part of the worlds economy. In fact, it generates around € 8.6 billion every year in the Netherlands alone [1]. At the same time, there is a clear trend of automation in this sector that aims to increase the efficiency and decrease the amount of human labour.

However, one of the major costs for the farmers in this field is due to the diseases contracted by their animals. By developing a system that could detect such diseases, and other abnormal behavior, one could potentially reduce this cost by a great amount.

B. Application Description

The main purpose of this project is to design a system that can help farmers monitor the health of cattle while they are grazing. This will be done using wireless sensor network (WSN) technology, because of its ability to deliver real-time monitoring at a very low cost. However, on account of range limitations of WSN transceivers, the system will be designed for a relatively small field of 10 ha which is in the range of an average dairy farm in the Netherlands [2].

The network of the system will consist of three types of nodes: a base station, which acts as a data sink; sensor nodes, one for each head of cattle, recording health characteristics; and a number of relay nodes on fixed positions in the field, forwarding data from the sensor nodes to the base station. The reason for using relay nodes, is to be able to cover the majority of the field.

The system will be used both to detect different types of diseases, for example: fever, lameness and mastitis, and to detect if a cow is in estrus. This can be accomplished by the of use accelerometer-, microphone-, and temperature sensors in the sensor nodes [3]. Every 5 minutes, the sensor nodes will send the acquired data in a processed form to the base station for storage. The base station can analyse the sample values to detect patterns that correspond to different diseases or the onset of estrus.

C. Related Work

Wireless livestock monitoring has became a widely researched area in recent years due to the increased availability and lower costs of wireless sensor network technologies. WSN have been utilized in many livestock monitoring applications around the world such as animal localization, behavior analysis, health monitoring or pregnancy detection in cows. Several studies are concentrated to raw data collection [4], [10] for later data processing and research. The main drawback of these studies is the high energy consumption of the sensor nodes, which limits the lifetime of the nodes to only a few days at most. Also the storage and processing of collected data would be a problem in longer term.

Most of the livestock monitoring WSNs are designed for cattles [4]–[15] but there are also examples of studies for sheep health and behavior monitoring [19] and chicken monitoring for avian influenza surveillance in poultry farms with ultra low power wireless sensor nodes [17], [18]. The wireless health monitoring of cows has been one of the most important researched topic among authors. Kumar et al. [7] and Wang et al. [10] both developed cattle health monitoring WSNs based on the IEEE 802.15.4 protocol. Hwang et al. [9] showed that continuous monitoring and comparison of cattle activity can be used for disease prediction and prevention. In addition to disease prediction, pregnancy detection is also possible using continuous monitoring [13].

Wireless sensor networks can be used for livestock localization indoor [20] or outdoor [12], [14]. Panckhurst et al. [12] developed a GPS based wireless positioning system, while Huircan et al. [14] demonstrated that the link quality indication (LQI) feature of the ZigBee protocol can also be a candidate for outdoor livestock localization. Livestock positioning can have different purposes around the world, like the prevention of pastureland degradation and desertification in Mongolia [16] or cattle rustling prevention in Africa [8].

The ZigBee protocol and a carrier frequency of 2.4 GHz are widely used [7], [8], [14], [19], however some of the authors [6], [12], [17], [18] prefered sub-gigahertz carrier frequencies due to the higher communication distance. Sousa Silva et al. [6] have demonstrated that a Floating Base Sensor Network (FBSN) is also a feasible solution for data collection in large mobile wireless sensor networks. Kwong et al. [15] have analysed the problem of signal penetration through animal's body and they proposed a two antennae scheme for an optimized radio coverage of the cattle's neck collar. In this project we will not consider the interference caused by the body of the cows.

II. Application Specific Challenges

A. Mobility

Mobility presents a major challenge for sensor nodes mounted on cattle since they are subject to frequent changes in location. The animal monitoring system must be able to support animal mobility. The network topology and routing paths should therefore be dynamic, able to respond to frequent animal movement while optimising packet delivery.

B. Size of Field

A key challenge in a wireless network of this kind is the coverage of the wireless nodes. In cattle monitoring WSNs the network architecture should be able to cover enough area of the fields and must be scalable accordingly when needed for larger fields. The average size of a dairy farm field in the Netherlands is around 100,000 m^2 and the maximum range of Zigbee WSN protocol is approximately 500 m, so theoretically, only one or two relay nodes could cover a sufficient area of the field. In reality, transceivers with such range capabilities are not used in WSN applications, because of cost and power constraints.

C. Wireless Communication

As the animals move freely on the field, wireless technology is considered the only feasible method to establish and maintain communication between a base station and network nodes attached to the cattle. The wireless communication imposes several challenges on the design, like signal attenuation in the medium, interference from other radio signals, crosstalk from the other nodes, etc. As the radio signal generated from the node is weak owing to preserve battery on the node, a significant amount of wireless signal is attenuated by the animal tissue. For maximum signal coverage from relay node the antenna on the animal is placed on a neck collar with two antenna on both sides of the neck to have spatial diversity. Also the radio element is switched off as soon as the data transmission is finished so as to maximize battery life.

D. Energy Consumption

The battery usage is one of the major constraint in cattle monitoring networks because the radio collars used for cattle monitoring are expected to run for up to several years without battery replacement. As there is limited battery power available per node, low powered, lightweight radio antennas should be considered for the design. The network protocol should be designed so as to use limited battery power and only communicate with the base station, to provide adequate amount of data required for monitoring health of cattle. The processors used by the nodes should be chosen so as to consume a minimum amount of power, while still being able to perform the necessary operations on the data as required by the application.

E. Cost of the System

The design of a WSN should not be too expensive. Furthermore the wireless nodes must be low-cost with high lifespan and low maintenance in order to reduce the cost of managing the system. Another reason for the sensor nodes to be low-cost is the potentially high number of nodes needed for monitoring an entire herd.

III. Application Specific Parameters

A. Herd size

According to Eurostat, the statistical office of the European Union, the average number of livestock units (LSU) per ha is 3.35 in the Netherlands. [28] LSU is a reference unit to compare the feeding needs of different livestock species, in the case of dairy cows, this value is 1.0. The result of this is that on average on a field of 10 ha, 33.5, or approximately 30 cows will held.

B. Resting Times

Cows lay down for a considerable amount of time every day. In fact, according to milkproduction.com, a site owned by DeLaval, cows lay down for approximately 14 hours every day, with each lying down period typically lasting between half and hour and three hours. [29]

The consequence of this is that the network needs to be able to cope with cows staying out of coverage for at least three hours.

C. Data size

One of the most important requirements for the system is to be able to handle the amount of data generated by the sensors. Estimating this amount is thus also very important. For storing the temperature of the cow after processing the raw data from the sensors, one byte is enough to give a reasonable resolution. This is because a dairy cows normal rectal temperature lies between 38.3°C and 38.9°C [21] and one could limit the byte to represent values between 30°C and 50°C without losing any information. Using the same argumentation, one byte could also be used to store the average heart rate, because the normal heart rate for a dairy cow lies between 40 and 84 beats per minute [21].

In addition, to store the data gathered from the accelerometers, one could devise a similar strategy to the one used in a study where the behaviour of sows were classified [22]. In this study, the activities of the animals were organized into four different sets, using a method specifically developed for low powered embedded devices. This resulted in a classification with an accuracy of close to 90 %. Using data in this format, the researchers were able to detect the onset of farrowing. In this study only four sets of behaviours were used, if it is possible to develop similar sets for behavioural studies on cows, one byte would be more than enough to describe the behaviour.

If five samples for each category are generated, the total data amount needed to be sent each five minute interval would be 15 bytes.

IV. REQUIREMENTS

To ensure that the designed system can meet the challenges described in section II, taking into account the parameters presented in section III, it needs to meet the following requirements:

1) The network must be implemented in such a way that if a sensor node moves out of coverage, the sensor node will be reassigned to the network once it moves into coverage again.
2) The network must continue to function whether nodes are added or removed.
3) When out of coverage, each sensor node must be able to store sensor values for a period of up to four hours.
4) The end-to-end latency between the sensor nodes and the base station must be less than 2 minutes for at least 90 % of data packets.
5) For every sensor node at least 70 % of the recorded sensor data must be delivered to the base station during a 5 hour period.
6) The network must function and be scalable for up to 30 sensor nodes.
7) The time until the first sensor node of the network fails must be greater than one year.

V. EVALUATION METHOD

The system was first implemented using the ContikiOs for TelosB/Sky nodes, and later evaluated using the Cooja simulaton software. The reason for choosing simulation over real life experiments, is in part due to time and resource contraints, and in part due to the fact that simulations allows for a more rapid work flow.

There is one downside to using simulations however. It is hard to accurately model the effects of the environment, for example the shadowing of vegetation and cows blocking the line of sight. Also, one of the limitations with the Cooja simulaton software is that nodes cannot be added after the simulation has started.

Another consequence of using simulations is that we cannot use actual sensors to generate data. Instead we use randomization to generate the data. Since processing the sensor's raw outputs would infer some extra complexity, this needs to be taken into account when evaluating the performance. The consequences of only doing simulations will be examined further in section VII-B.

Since we decided to make the simulation using Cooja on Contiki OS instead of using real hardware, we tried to make use of the software as much as possible and tested multiple scenarios on it. We also tried to make use of the different functionalities Cooja provides like mobility support and used BonnMotion to generate a more sophisticated mobility scenarios that can roughly match the cows movements which is described thoroughly in section V-A.

Three different things were measured during the simulations: the end-to-end latency between sensor nodes and base station, the PDR from sensor node to base station, and finally the energy consumption of the sensor nodes.

Ideally, each simulations would be run for at least one day to get a good picture of the network's performance. However, because of time and resource constraints, combined with our desire to test several parameters, we opted to run simulations with that were only 20 minutes in simulated time. To compensate for the short simulation time, we decided to stress test the network by increasing the generation and transmission of sensor packets fivefold.

A. Mobility Models

Since we are using simulation only to test our implementation, we wanted to generate a more sophisticated mobility scenario. So we used BonnMotion which is Java application that can create and analyse many different mobility scenarios. The only format than can be used by COOJA is wiseML format. But we found out that the library used for wiseML format is not implemented yet in the official page of COOJA and its different forums. There was an implementation for such format done by CONET (the Cooperating Objects NETwork of Excellence), but the cooja-wiseml extension was not yet publicly available and they have CONET version of Instant Contiki with all their implementations installed, but the download link expired. So we implemented a python script to convert the wiseml format mobility file to the dat format used by COOJA.

BonnMotion offers nearly 21 mobility scenarios, we used 2 mobility scenarios that matches our mobility requirements: Reference Point Group Mobility model (RPGM) (Fig. 1a) where sensor nodes are moving in groups uniformly and Nomadic Community Mobility model (Fig. 1b) where sensor nodes move in groups but some nodes can also move or wait alone. Finally, we ended up using only Nomadic Mobility model because we found it to simulates the cows behaviour more precisely than RPGM.

To generate the Nomadic Mobility models, we ran BonnMotion with the following parameters, where XX denotes the number of sensor nodes:

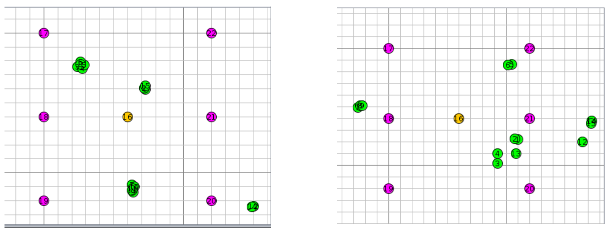
1)
```
bm -f Nomadic30t20 Nomadic -n XX\
     -d 1200 -i 3600 -x 320 -y 320\
     -h 5 -l 1 -p 10 -a 7 -r 7 -c 7
```

2)
```
bm -f Nomadic30t20 Nomadic -n XX\
     -d 1200 -i 3600 -x 320 -y 320\
     -h 3 -l 0.5 -p 7 -a 7 -r 7 -c 5
```

B. Test Scenarios

The network was tested with the following parameters:

- 5, 15, and 30 sensor nodes;



(a) RPGM (b) Nomadic

Fig. 1: Mobility models

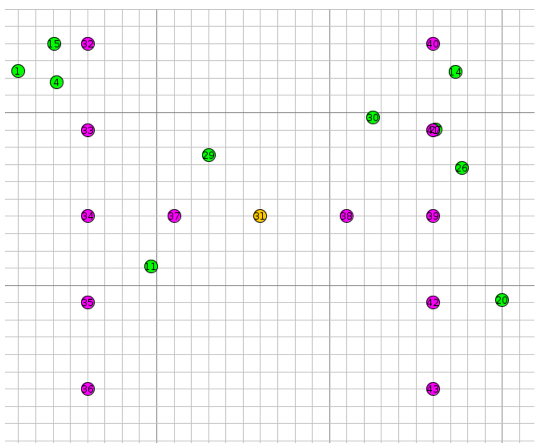


Fig. 2: Relay node configuration with 12 relay nodes

- 6 an 12 relay nodes, see Fig. 1 and Fig. 2 (relay nodes are magenta colored) for the layout of the relay nodes;
- 2 different configurations of the nomadic mobility model (see section V-A);
- a field size of 320 m by 320 m.

This resulted in twelve different test scenarios.

C. Data Extraction

To measure the latency and the PDR, data was extracted by using Cooja's built in printf functionality which outputs data to the serial UART port. At all places where a packet was transmitted or received, a printf with a string identifying: the node type; the node id; if the packet was sent, received, or delivered to the base station; and data related to the packet type. Additionally, printfs were used to show the sensor samples that were generated in the sensor node. The data was finally saved to a text file using Cooja's loglistner.

In order to evaluate the power consumption of the sensor nodes during the simulations, we used the Powertrace power profiling tool. This tool, is part of the ContikiOS, and has a 94% accuracy in estimating the power consumption compared to hardware-based power measurements [26]. The output of the tool is useful in examining the time spent in different states (e.g. transmission, reception, idle) of the transceiver and MCU. The data from the powertrace tool was saved in the same text file as mentioned above.

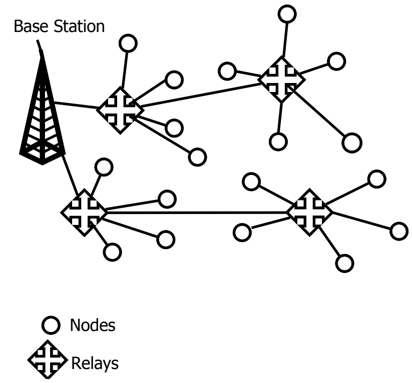


Fig. 3: An overview of the network topology.

The recorded data was analyzed with a Matlab script that calculated the average PDR, the average end-to-end latency, and the average energy consumption.

While this method is very powerful in that a lot of data can be gathered, using a lot of printfs has a noticeable effect on the cpu usage. This can in turn affect the energy consumption.

VI. Implementation

A. Network Topology

The network topology of the implementation has a tree like structure with the base station acting as the root of the tree, see Fig 3. The relay nodes act as the branches of the tree, being connected to the base station through each other, while the sensor nodes behave like the leaves of the tree.

The sensor nodes send data to any relay node that is within transmission range. Once received by a relay node, the data will be forwarded to the base station with the help of a routing protocol.

B. Physical Layer

In order to get an increased range and mitigate the problems with coverage, carrier frequencies between 902 MHz and 928 MHz of the IEEE 802.15.4 standard, were originally considered. However, as these frequencies are not supported by the TI CC2420 transceiver of the Telos B, a frequency of around 2.4 GHz was used instead.

For the modulation, the direct-sequence spread spectrum is used, because it is implemented in the transceiver mentioned above.

Different transmission powers are used for the different node types used in the implementation. The base station and relay nodes both use the maximum transmission power of the transceiver to ensure that the relay nodes can be placed as far apart as possible. The sensor nodes, on the other hand, use half of the maximum transmission power to save energy. Because the sensitivity of the Telos B nodes cannot be changed, this introduces an asymmetry between the nodes: relay nodes can reach the sensor nodes at maximum range while the converse is not possible. Even though this reduces the coverage compared to the case where maximum transmission power is also used in the sensor nodes, there is still one benefit: few relay nodes are needed to create a chain to a relay node placed far away from the base station (for example in a remote corner of the field). Thus the sensor nodes will still have a short distance to the nearest relay node.

According to an experiment performed in an outdoor environment, the average propagation path loss at 60 m for a link between two Telos B Nodes was estimated to be 92.3 dB. Using this value, along with 0 dBm, the maximum transmission power of the Telos B node, in the path propagation loss formula:

$$P_r[dBm] = P_t[dBm] - P_L[dB]$$

where P_r is the received power, P_t is the transmitted power, and P_L is the propagation path loss gives -92.3 dBm which is greater than -94 dBm, the nominal receiving sensitivity of a Telos B node. Taking this into account, the maximum transmission distance of the nodes are set to 60 m in the simulation.

C. MAC Layer

For the MAC layer, there are two modes to be used in the IEEE 802.15.4 standard: nonbeacon-enabled mode and beacon-enabled mode. In the former a unslotted CSMA/CA mechanism is used. In the latter one, a more elaborate scheme is used, wherein special nodes of the network, called coordinators, regularly send out beacons to synchronize with the nodes they are associated with. The period between two beacons is called a superframe, and within this the nodes use a slotted CSMA/CA mechanism.

Because of the need for synchronization, the beacon-enabled mode might cause problems if a sensor node moves out of the coverage of a coordinator for a longer period of time, or if a sensor node needs to switch to a different coordinator. The sensor node might have to listen for a very long time to receive the next beacon, possibly even the whole time while it is out of range, thus consuming a lot of power.

On the other hand, the unslotted CSMA/CA mechanism of the nonbeacon-enabled mode, in addition to being less complex, does not require the nodes associated with a coordinator to regularly receive a beacon, and is thus suitable for the mobile nature of our application. The main drawback is the power consumption required for continuously listening at regular intervals. However, since the relay nodes and the base station are the only ones that need to receive data, only these will have to listen continuously, and not the sensor nodes. In view of these arguments, the design will use the nonbeacon-enabled mode. However, as contiki does not implement unslotted CSMA/CA, we will use one of the congestion-based MAC layers that contiki provides, in its place.

Contiki Operating System uses a little bit different network stack than the usual 5 layer TCP/IP model. The Medium Access Control (MAC) Layer usually resides between the Physical and the network layer. Contiki uses 2 more layer under the MAC layer and above the physical layer which are: Radio Duty-Cycle (RDC) and Framer layers as in the figure. The three different layers can be changed through the global variables NETSTACK_FRAMER, NETSTACK_RDC and NETSTACK_MAC, these variables are defined in compilation time.

Framer layer has some auxiliary functions for building the frame with data to be transmitted or parsing the received data. Since we only want the basic functions from this layer, we used the protocol framer-contikimac.

Radio Duty-Cycle (RDC) layer is the most crucial layer as it is the one responsible of the sleep period of the nodes, it also manages when the packets will be transmitted and at the same time makes sure that the nodes are awake when receiving packets. We used ContikiMAC since it uses a simple but efficient timing scheme to allow its wake-up mechanism to be highly power efficient [25]. Results mentioned in [24] show that using ContikiMAC reduced the average duty-cycle from 17% to 9%, which makes the energy consumption of the nodes drastically more efficient.

For the MAC layer, we used Carrier-Sense Medium Access (CSMA) protocol, as it contains addressing, sequence number and retransmissions which are crucially needed in our application other than the default nullmac protocol.

D. Network- and Application Layer

1) Packet Structure: For the network to function correctly, each node needs to be able to determine which type of packet it has received. This is achieved by including a one byte field in each packet whose value correspond to different packet types.

There are 5 types of packet used in the network.

- Sensor packets are broadcasted by the sensor nodes, and received by any relay node within range. These packets contain 15 bytes of data generated by the sensors.
- Init packets are broadcasted by the relay nodes and by the base station. The packets are used to configure the routing mechanism.
- Aggregated packets are broadcasted by the relay nodes and are used to aggregate several sensor packets into bigger packets.
- There are a also two types of acknowledges: sensor acknowledge, for acknwoledging sensor packets, and aggregated acknowledge for acknowledging aggregated packets.

2) Routing Mechanism: The network uses directed gossiping to route data from the sensor nodes to the base station. In our implementation, this method works by making sure that each relay node has a *hop count* that specifies the number of communication hops between it and the base station.

When a sensor node wants to transmit a sensor packet, it broadcasts the packet which will be received by any relay nodes within range. The relay nodes, after accumulating enough sensor packets, will broadcast an aggregated packet containing the aforementioned sensor packet and the hop count of the transmitting relay node. When a relay node receives an aggregated packet, it will only forward the packet if the relay node's hop count is lower than the hop count of the packet. Upon forwarding, the relay node also decrements the hop count of the aggregated packet by one.

To determine the hop count of each relay node, a configuration process is performed at regular intervals. The process is initiated by the base station, which broadcasts an init packet with a hop count and a sequence number, both set to zero. When receiving an init packet, the relay nodes compares the sequence number of the packet with the sequence number of the last received init packet. If the received packet's sequence number is higher, the relay node resets its hop count to a very large default value (initially, the hop count is also set to this default value). After this, the relay nodes compares the hop count of the received packet to its own hop count. If the former is lower, it means that the route that the init packet took has a lower hop count, the hop count of the relay node is therefore set to this value, and the init packet is forwarded with the hop count incremented by one. If the former is higher, then there is at least one shorter route and the packet is simply dropped.

3) Sensor Node: The sensor nodes has two process working in tandem. The first process periodically generates sensor data and puts it inside a queue. The second process tries to send the first item in the queue a few times, sleeping a short time in between, and then sleeps for a longer time before repeating itself. As soon as the sensor nodes receives a sensor acknowledge packet from a relay node, it will remove the first item in the queue, and try to transmit the next one.

4) Relay Node: To reduce the number of packets needed to be transmitted, each relay node keeps a current aggregated packet, which it places received sensor packets inside, instead of fordwarding them immediately. When the current aggregated packet is full, or when a timeout timer has run out (the timer is reset upon receiving a sensor packet), the packet is locked for transmission, and is not updated until it has received an aggregated acknowledge.

When the current aggregated packet is locked, new sensor packets are placed in a temporary aggregated packet. The contents of the temporary packet are written to the current packet, when the latter is unlocked.

When a relay node receives an aggregated packet from another relay node, it will try to forward it according to the routing mechanism explained in section VI-D2.

Every time a sensor packet is successfully saved in a relay node, a sensor acknowledge will be sent to the sensor node that sent this packet; if the packet was dropped in a relay node, no acknowledge is sent. This system is designed to prevent packet loss: sensor nodes will keep sensor packets until they have been received by a relay node, and each relay node will keep aggregated packets until they have been received by another relay node.

5) Base Station: The base station simply sends init packets at regular intervals, to initiate routing configuration and listens for aggregated packets sent by the nearest relay nodes.

TABLE I: Latency and PDR Using 6 Relay Nodes

Number of sensor nodes	**Mobility Type**	**Latency**		**PDR (%)**
		Avg. (sec)	**<2 min (%)**	
5	1	44.85	90	31.58
	2	48	97.3	77.89
15	1	41.71	94.1	41.75
	2	24.73	97.3	77.54
30	1	14.9	99.6	81.75
	2	19.17	98.6	62.11

TABLE II: Latency and PDR Using 12 Relay Nodes

Number of sensor nodes	**Mobility Type**	**Latency**		**PDR (%)**
		Avg. (sec)	**<2 min (%)**	
5	1	87.02	70.8	50.53
	2	65.28	87.1	89.47
15	1	36.16	98.3	41.4
	2	32.42	95.5	85.26
30	1	16.34	98.6	90.88
	2	21.86	98.3	82.81

VII. Performance Evaluation

A. Results

According to the results of the simulation, the network seems to reassign nodes that move in and out of coverage correctly. Because we used mobility models, and the relay nodes did not cover the field completely, there were several instance where the sensor nodes moved out of coverage. In all of these cases, the sensor nodes were all able to deliver sensor packets, once they moved into coverage again. From our experiments, the first requirement appears to be satisfied.

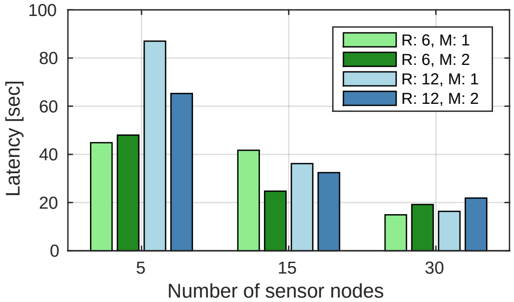
In our simulation, the network was shown to function correctly with different number of relay nodes and sensor nodes. But because of limitations with Cooja we could not explicitly test on-the-fly additions of relay nodes after a simulation had started.

Although we designed the buffers in the sensor nodes to be able to store more than four hours' worth of sensor data, we did not run any simulations with a length of more than four hours to test this. Hovewer, we were able to show that they were capable of storing at least one and a half hours' worth of sensor data.

The results from the simulations concerning the latency and the PDR are shown in tables I and II. From these tables, it is apparent that in all but two cases, the latency was less than two minutes for over 90 % of transmitted packets. In contrast, the PDR statistics are more diverse, with five out of twelwe simulations showing results below the requirement target.

The results are plotted in a bar diagram (Fig. 4) to illustrate how the results vary with respect to the different parameters. Looking at the average latency, the network seems to scale well with an increasing ammount of sensor nodes and it appears that the network is well optimized for 30 sensor nodes. At the same time the latency appears to be lower for 6 relay nodes compared to twelwe relay nodes. For the PDR the results are more mixed, but in general, the highest PDR is acheived with 30 sensor nodes and 12 relay nodes.

From the powertrace tool, we can estimate the average current consumption of the sensor node. In our simulations we



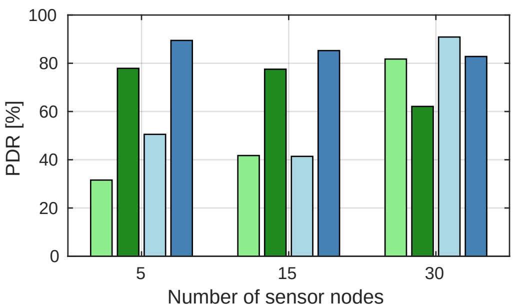


Fig. 4: Simulation results: average latency and packet delivery ratio (PDR). R and M denote the number of relay nodes and the mobility type, respectively

obtained an average current consumption of $I_{SN} = 0.93mA$. If we presume the use of two Energizer E91 Alkaline AA batteries [27], which has a capacity of $C_{battery} = 2500mAh$, we can, then, estimate the expected lifetime of a sensor node using Eq. 1:

$$T = \frac{C_{battery}}{I_{SN}} \tag{1}$$

With this calculation we obtain an expected average lifetime of approx. 112 days.

B. Discussion

Because the simulations were not run for five hours, and the fivefold increase in sensor packet generation does not equate to five hours' worth of sensor data, we cannot completely determine if the PDR requirement was fulfilled or not. In fact, as the results show that once sensor nodes were inside coverage, they were successfully able to deliver their packets to the base station, it is evident that the PDR is highly dependent on the mobility of the cows.

In addition, the average expected lifetime should be treated carefully and no final conclusion should be drawn from it. First of all this is only a simulation based estimation, in which we increased the sampling frequency of data generation. Also, the use of this tool requires more CPU usages which leads to higher consumption. On the other hand, though, we haven't included the current consumption of real sensors into the estimation.

Another consideration is that there are several parameters that could be tweaked to potentially increase the performance of the network. Firstly changing the timeout timers for sending aggregated packets could be possibly decrease the average end-to-end latency. Secondly, changing the timers of the sleeping times when a sensor node is out of coverage could also reduce the energy consumption.

As the simulation does not take into account fading effects, the energy consumption of sensors, and sensor data processing, real life experiments are needed to fully evaluate the performance of the network. Real life experiments can also show the effects of interference, though this is probably less important, as there is most likely not many users of competing bands in a typical cattle field.

Taking into account the unreliability of the PDR results, it is not wise to draw definite conclusions on how many relay nodes should be used. It is also possible that there are other relay node layouts that give a better performance.

VIII. Conclusion

Our simulations shows that the proposed design results in a correctly functioning network with an acceptable latency. However, further tests along with real life experiments are needed to fully evaluate the number of relay nodes needed, and the PDR- and energy usage requirements.

References

[1] https://www.government.nl/topics/agriculture-and-livestock/contents/livestock-farming, taken 12:30 22 sep 2015.

[2] D. B. Grigg, *The Agricultural Systems of the World: An Evolutionary Approach*, Cambridge University Press, november 1974, p. 188.

[3] A. Helwatkar, D. Riordan and J. Walsh, *Sensor Technology For Animal Health Monitoring*, Proceedings of the 8th International Conference on Sensing Technology, sep, 2014.

[4] Y. Guo, P. Corke, G. Poulton, T. Wark, G. Bishop-Hurley, and D. Swain, *Animal Behaviour Understanding using Wireless Sensor Networks,* in Proceedings 2006 31st IEEE Conference on Local Computer Networks, 2006, pp. 607614.

[5] K. Smith, A. Martinez, R. Craddolph, H. Erickson, D. Andresen, and S. Warren, *An Integrated Cattle Health Monitoring System,* in 28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2006. EMBS 06, 2006, pp. 46594662.

[6] A. C. de Sousa Silva, A. I. C. Arce, S. Souto, and E. J. X. Costa, *A wireless floating base sensor network for physiological responses of livestock,* Computers and Electronics in Agriculture, vol. 49, no. 2, pp. 246254, Nov. 2005.

[7] A. Kumar and G. P. Hancke, *A Zigbee-Based Animal Health Monitoring System,* IEEE Sensors Journal, vol. 15, no. 1, pp. 610617, Jan. 2015.

[8] P. K. Mashoko Nkwari, S. Rimer, and B. S. Paul, *Cattle monitoring system using wireless sensor network in order to prevent cattle rustling,* in IST-Africa Conference Proceedings, 2014, 2014, pp. 110.

[9] J. Hwang and H. Yoe, *Design and implementation of wireless sensor network based livestock activity monitoring system,* in Future Generation Information Technology, Springer, 2011, pp. 161168.

[10] H. Wang, B. Davies, and A. O. Fapojuwo, *Inter-wireless body area network scheduling algorithm for livestock health monitoring,* in 2015 IEEE Wireless Communications and Networking Conference (WCNC), 2015, pp. 21322137.

[11] K. H. Kwong, T.-T. Wu, H. G. Goh, K. Sasloglou, B. Stephen, I. Glover, C. Shen, W. Du, C. Michie, and I. Andonovic, *Practical considerations for wireless sensor networks in cattle monitoring applications,* Computers and Electronics in Agriculture, vol. 81, pp. 3344, Feb. 2012.

[12] B. Panckhurst, P. Brown, K. Payne, and T. C. A. Molteno, *Solar-powered sensor for continuous monitoring of livestock position,* in 2015 IEEE Sensors Applications Symposium (SAS), 2015, pp. 16.

[13] A. H. H. Nograles and F. S. Caluyo, *Wireless system for pregnancy detection in cows by monitoring temperature changes in body,* in 2013 IEEE 9th International Colloquium on Signal Processing and its Applications (CSPA), 2013, pp. 1116.

[14] J. I. Huircn, C. Muoz, H. Young, L. Von Dossow, J. Bustos, G. Vivallo, and M. Toneatti, *ZigBee-based wireless sensor network localization for cattle monitoring in grazing fields,* Computers and Electronics in Agriculture, vol. 74, no. 2, pp. 258264, Nov. 2010.

[15] K. H. Kwong, T. T. Wu, H. G. Goh, B. Stephen, M. Gilroy, C. Michie, and I. Andonovic, *Wireless sensor networks in agriculture: cattle monitoring for farming industries,* in Progress In Electromagnetics Research Symposium, 2009, vol. 5, pp. 3135.

[16] S.-G. Choi, E. Norinpel, and J. Yoo, *Cluster-based routing protocol in wireless sensor networks for tracking livestock movements in Mongolian nomadic herding,* in 2013 8th International Forum on Strategic Technology (IFOST), 2013, vol. 2, pp. 163167.

[17] H. Okada, H. Nogami, T. Itoh, and T. Masuda, *Development of Low Power Technologies for Health Monitoring System Using Wireless Sensor Nodes,* in 2012 Second Workshop on Design, Control and Software Implementation for Distributed MEMS (dMEMS), 2012, pp. 9095.

[18] H. Okada, H. Nogami, T. Kobayashi, T. Masuda, and T. Itoh, *Development of ultra low power wireless sensor node with piezoelectric accelerometer for health monitoring,* in 2013 Transducers Eurosensors XXVII: The 17th International Conference on Solid-State Sensors, Actuators and Microsystems (TRANSDUCERS EUROSENSORS XXVII), 2013, pp. 2629.

[19] E. S. Nadimi, R. N. Jrgensen, V. Blanes-Vidal, and S. Christensen, *Monitoring and classifying animal behavior using ZigBee-based mobile ad hoc wireless sensor networks and artificial neural networks,* Computers and Electronics in Agriculture, vol. 82, pp. 4454, Mar. 2012.

[20] M. Asikainen, K. Haataja, and P. Toivanen, *Wireless indoor tracking of livestock for behavioral analysis,* in Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International, 2013, pp. 18331838.

[21] https://en.wikivet.net/Bovine_Physiology_-_WikiNormals, taken 12:30 22 sep 2015

[22] G. F. Marchioro, C. Cornou, A. R. Kristensen and J. Madsen, *Sows activity classification device using acceleration data A resource constrained approach*, Computers and Electronics in Agriculture, issn: 01681699, jun, 2011, pages 110–117,

[23] IEEE Computer Society, LAN/MAN Standards Committee, Institute of Electrical and Electronics Engineers and IEEE-SA Standards Board, *IEEE standard for local and metropolitan area networks. Part 15.4*, isbn 978-0-7381-6684-1, Institute of Electrical and Electronics Engineers, 2011,

[24] MAC Protcols in Contiki - http://anrg.usc.edu/contiki/index.php/MAC_protocols_in_ContikiOS

[25] A. Dunkels, *The ContikiMAC Radio Duty Cycling Protocol.*, SICS Technical Report, December 2011.

[26] A. Dunkels, J. Eriksson, N. Finne and N. Tsiftes, *Powertrace: Network-level Power Profiling for Low-power Wireless Networks.*, Swedish Institute of Computer Science, Technical Report, March 2011.

[27] Energizer E91 Product Datasheet - http://data.energizer.com/PDFs/E91.pdf

[28] Livestock Density Index - http://ec.europa.eu/eurostat/tgm/table.do?tab=table&init=1&language=en&pcode=tsdpc450&plugin=1

[29] Cow comfort: 9) Resting - http://www.milkproduction.com/Library/Scientific-articles/Housing/Cow-comfort-9/