

Energy Efficient Clustering using a Wake-up Receiver

Johannes Blanckenstein*, Jirka Klaue*, Holger Karl†

*EADS Innovation Works
Dept. IW-SI-CO
81663 Munich, Germany
johannes.blanckenstein@eads.net

†Universität Paderborn
Institut für Informatik
Warburger Str. 100
33098 Paderborn
holger.karl@upb.de

Abstract—We present a new energy-saving method for creating node clusters with similar sensor readings. The idea is to wake up a node/main transceiver only when data correlated to the node's own data is transmitted. This is achieved by encoding relevant parts of local data in a wake-up receiver's data pattern.

Through this new protocol the building of clusters becomes highly energy-efficient, even if re-clustering is required, the energy consumed for clustering is still low. For environments where re-clustering is often necessary a clustering scheme can be used to further reduce the energy consumption. This further reduction can be achieved efficiently through the transmission of the sensor readings from only one node of the cluster to the data sink. One reading is therefore sufficient to approximate the readings of the whole cluster because all nodes within this cluster have similar sensor readings.

I. INTRODUCTION

Applications of wireless sensors are expected to be very beneficial in future aircrafts, especially in structural health monitoring (SHM) and flight test installations (FTI). Apart from the general saving of cabling (weight) and installation effort, permanent monitoring of the structural health with wireless sensors is one of the premises of maintenance on demand, where the aircraft will only be disassembled if necessary. Downtimes would be decreased and maintenance would be simplified. Of course, there are stringent requirements on such wireless sensors for SHM applications. The first and most important requirement is autonomous operation for more than 10 years. Installed in hard-to-reach places and possibly embedded in the structure, replacing batteries is not an option. Energy harvesting for wireless sensors is a promising approach which could solve that problem. A lot of research has been performed within this area in the last years [1, 2] and working prototypes of energy harvesting and power management modules are now available [3, 4].

The cabling effort for the installation/deinstallation of hundreds and sometimes thousands of sensors for flight test campaigns is huge and wireless sensors would be a good alternative. Though there is no need for long-term autonomous operation as in the SHM domain, the wireless sensors should still work for some time on batteries. Energy-efficient operation is thus another key requirement for both application domains.

Since the radio chip is the most significant consumer, it is commonly tried to switch it off as often as possible. One approach is to use the radio in a duty-cycled manner [5]. However, polling does not scale very well with the number of sensors. Another approach uses special wake-up receivers (WUR), which are always on duty. Such a WUR is a compromise between energy consumption, sensitivity, and wake-up time [6].

To efficiently use the energy-saving WUR while fulfilling the requirements of the applications, a protocol has to be developed. Almost all applications in the SHM and FTI domains require periodic measurement and transmission of physical parameters like temperature or strain. The number of sensors is high but constant and known a priori. The measurement and transmission rates are also defined and constant. Thus, TDMA as the basic protocol scheme is the natural choice. TDMA also makes the certification easier since there is no inherent non-determinism in the protocol. The determinism of the TDMA protocol provides an additional feature to higher-layer protocols. When using a fixed TDMA schedule, the node who is sending the packet is implicitly known because for each time slot only a certain node is permitted to send packets; so when the slot is known the sender is also known.

SHM can generate a lot of sensor readings and the transmission of these will consume a lot of energy. Much research has been undertaken in the last years to decrease energy consumption [7, 8]. One approach is to predict the sensor readings in the sink and therefore decrease the number of readings that have to be sent to it. For example in [8] the protocol PAQ is presented. PAQ divides the nodes into clusters in which each node generates similar sensor readings. The cluster leaders then use a time series forecasting model and send the model coefficients together with the cluster membership lists to the sink. By utilising the membership lists the sink can predict the readings for each cluster. Additionally, because of the data similarity within a cluster, the sensor readings for all cluster members are implicitly predicted as well. But as re-clustering is often necessary, the clustering process itself will consume a lot of energy.

In this paper a proof of concept of a new clustering approach is presented which uses "wake-up messages" to disseminate

similarity information about the sensor readings. During clustering it is sufficient to use only the main transceiver to send the “wake-up messages” but for receiving the messages the main transceiver can be turned off. That is the case because the wake-up receiver can still receive the “wake-up messages” and thus has the necessary information about the data similarity. We show the feasibility of this concept by simulation.

The paper is structured as follows. As preliminaries, in Section II the functionality of a wake-up receiver and the mapping of sensor readings to wake-up patterns are described. The wake-up receiver dependent clustering scheme is described in detail in Section III-A, followed by the presentation of a more traditional clustering approach in Section III-C. To prove the concept of the new clustering approach, simulation results are presented in Section IV, and Section V concludes the paper.

II. PRELIMINARY

A. Wake-up Receiver

A wake-up receiver (WUR) is a low-power receiver that is used to wake-up a sleeping node after receiving a wake-up message. There are several concepts for wake-up receivers available, all differing in complexity and performance. For example, wake-up can be achieved by simply detecting the energy of a wake-up message. An alternative approach, on which this paper focuses, can be achieved through correlating the received wake-up message with a known one. Such a wake-up receiver was developed within the EU project CHOSeN [6]. It uses OOK as modulation. This WUR is capable of correlating 64 bit wake-up messages—further referred to as patterns—when the correlation result exceeds a specific threshold the node is triggered to wake-up. In comparison with simple energy detection, the received pattern is now compared to predefined patterns. This makes it possible to distinguish between several patterns; therefore a kind of wake-up id can be transmitted which is normally used for addressing the node/group of nodes which shall be woken up.

The length of this wake-up I is limited due to the small set of available 64 bit patterns with good correlation properties. Promising code-sets with low cross-correlation are Gold codes or Kasami codes [9], [10]. Gold codes have a code family size of 65 and Kasami codes have a family size of 520. It can be shown that the largest cross-correlation value R_{\max} of Kasami codes is lower than the largest of Gold codes but however the cross-correlation of Gold codes is lower on average than the one of Kasami codes; Table I summarizes some code set characteristics.

| Family | Family size | R_{\max} |
|------------------------|----------------------|-------------------|
| Gold codes | $2^n + 1$ | $1 + 2^{(n+1)/2}$ |
| Kasami codes—large set | $2^{3n/2} + 2^{n/2}$ | $1 + 2^{n/2+1}$ |

TABLE I
SOME CODE SETS OF LENGTH $(2^n - 1)$ FOR EVEN n [11]

B. Mapping Measurements to Wake-up Patterns

The idea of the new clustering protocol is to disseminate information about sensor readings by “misusing” wake-up patterns. To do this, the large set of sensor readings has to be mapped to a small set of available wake-up patterns.

The WUR patterns are arranged in a look-up table to map the measurements to a WUR pattern. The code set from which the WUR patterns are chosen will have a big influence on the performance of the WUR. So when using the CHOSeN wake-up receiver presented in Section II-A the look-up table contains a subset of the code set of 64 bit long patterns. The subset size n thereby defines the possible mapping co-domain or the size of the intervals, respectively. The interval width is δ . All measured values have to be mapped to such a WUR pattern. Thus the mapping corresponds to a quantizing problem and the width of one bin corresponds to δ . This means all measurements within one bin are similar to each other or in other words the difference of each other is smaller than δ .

With the relatively small available code set size there is a trade-off between possible values of δ and the mapping co-domain Δ , which can be supported without clipping. For example if there are 100 patterns available 100 bins are possible. If $\delta = 0.5$ is desired, the maximal mapping co-domain is $\Delta = 100\delta = 50$. Meaning for example when temperature is measured, values between -10°C and $+40^\circ\text{C}$ can be correctly mapped to a pattern. Therefore if the measuring co-domain of the readings differ from the mapping co-domain, the mapping may not fit. To support all measurements, even if they are outside the mapping co-domain, all measurements outside it have to be mapped to the nearest bin, despite the fact that they are not similar to the other measurements within this bin. This is the same as a hard clipping of the measured values at the border of the mapping co-domain. Alternatively, if a bigger value for δ is chosen, the mapping co-domain will increase and more values can be mapped without clipping. The calculation of the co-domain is done by $\Delta = n\delta$.

III. CONCEPT

The following section presents firstly the concept of the wake-up receiver clustering scheme and secondly a more traditional method for comparison with the WUR approach.

A. Clustering Scheme

The clustering scheme is divided into two phases. Phase I is the information gathering mode where all sensors broadcast their data and listen for similar data from other neighbours. More precisely, they broadcast a WUR pattern, which corresponds to their measured value. At the same time they listen for WUR patterns which are similar to their own patterns. This is done m times to strengthen the protocol against outliers of the sensor readings or correlation errors. So Phase I takes m frames for completion.

In Phase II—the clustering mode—the cluster leaders are selected. This takes three frames; the first frame is used to pre-select the cluster leaders by using the gathered information about data similarities from Phase I. To do this the cluster

leaders broadcast a special wake-up pattern which indicates cluster leadership. The next two frames are used to resolve any correlation errors which may have occurred. Due to the pre-selection the error resolving is still energy-efficient as it allows cluster members to listen only to the slots of their own cluster leaders. Therefore the main transceiver can be switched off for all other slots. With completion of Phase II, all nodes are divided into clusters and each cluster has a unique cluster leader.

B. Detailed Description

1) *Phase I—Information Gathering Mode:* During the information gathering mode all nodes measure their values at timestep k and store them in their queue V . This is repeated m times. Additionally, a WUR pattern is chosen which corresponds to the actual measured value v_k . To enable this, a mapping table exists to map a measurement to a WUR pattern. This pattern is then broadcast to disseminate the information of the current measurement v_k in a very energy efficient way within the network (Algorithm 1). With the main

Algorithm 1 Information Dissemination

```

1: for  $k = 1 \rightarrow m$  do
2:   wait for own slot
3:   read  $v_k$  from  $V$ 
4:   transmit  $pattern(v_k)$ 
5: end for

```

transceiver set to a deep sleep mode, only the WUR is listening for similar WUR patterns. When such a pattern is received, the microcontroller is woken up. The awoken node knows implicitly from the fixed TDMA schedule which node was sending this WUR pattern. It then includes the id of this node in its neighbour list (NL) and initializes the associated hit counter c . Respectively if the node was already listed in the neighbour list, the corresponding hit counter will be incremented by one. After m frames all nodes change from the information gathering mode to the clustering mode. This procedure can be seen in Algorithm 2.

Algorithm 2 Information Gathering

```

1: for  $k = 1 \rightarrow m$  do
2:   read  $v_k$  from  $V$ 
3:   listen with WUR for  $pattern(v_k)$ 
4:   if  $pattern(v_k)$  found then
5:     get id of current slot
6:     if  $id \in NL$  then
7:       get  $(id, c)$ 
8:       increment  $c$ 
9:     else
10:      create  $(id, 1)$ 
11:       $NL \leftarrow NL \cup (id, 1)$ 
12:    end if
13:  end if
14: end for

```

2) *Phase II—Clustering Mode:* In the clustering mode the cluster leaders will be selected. This takes three frames. At the beginning of the first frame the cluster list (CL) is created. To do this, a node stores its own id together with the maximum hit count $maxCount = m$ in the cluster list. It looks in the neighbour list for all entries with a hit counter c larger than the threshold $thold$. These entries will also be stored in the cluster list. With that, each node has a cluster list of all the nodes within its communication range which measure values within a certain bound δ , see Section II-B for mapping measurements to patterns.

For all slots of the first frame all nodes listen with their WUR for a special pattern $pattern(CH)$. This pattern does not represent a certain measurement but represents leadership notification. When such a pattern is found, the id of the node of this time slot is stored in the leader list (LL) (Algorithm 3). Additionally each node checks the cluster list to identify the

Algorithm 3 Reception of Leader Notification Pattern

```

1: listen with WUR for  $pattern(CH)$ 
2: if  $pattern(CH)$  found then
3:   get id of current slot
4:    $LL \leftarrow LL \cup id$ 
5: end if

```

node with the smallest id. If the selected node is identical to itself then it becomes a tentative cluster leader. All nodes which are tentative cluster leaders check, before their own slot occurs, if a node from their cluster list has already sent a $pattern(CH)$. When no other member from its cluster list has already sent such a pattern, the node becomes a validated cluster leader and will send its $pattern(CH)$ message to inform the other nodes about its status as a cluster leader. However if another node from its cluster list has already sent a leader notification, the tentative cluster leader recognizes this node as its cluster leader and becomes its cluster member. In this case it will not send a $pattern(CH)$. When communication errors occur, it may happen that more than one member of the cluster list has sent a leader notification. If so, the leader with the smallest id is chosen as the validated cluster leader.

At the end of the first frame all nodes that are not cluster leaders will check if they have received a leader notification from one of the nodes of their cluster list. If they haven't received such a leader notification, they will become validated cluster leaders themselves (Algorithm 4).

Algorithm 4 Tentative Cluster Head Selection

```

1:  $CL \leftarrow (ownID, maxCount)$ 
2:  $CL \cup \{(id, c) \in NL | c \geq thold\}$ 
3: tentative leader  $\leftarrow \min(id \in CL)$ 
4: if  $((tentative\ leader = ownID) \wedge (LL \cap CL = \emptyset))$  then
5:   validated leader = tentative leader
6:   send  $pattern(CH)$ 
7: end if

```

The second frame is used to check if the cluster membership

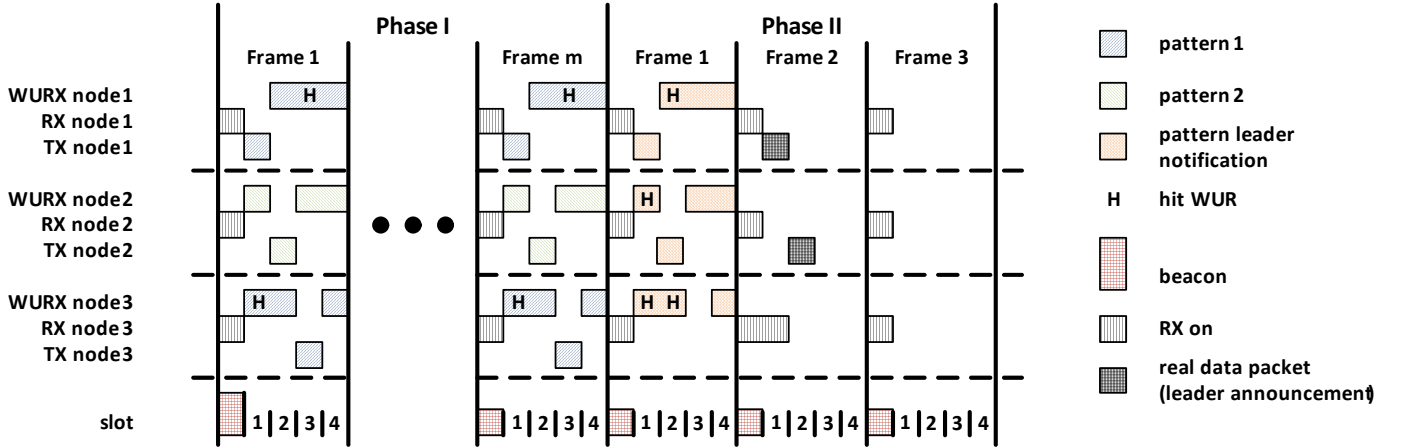


Fig. 1. Example for the clustering protocol for three nodes without communication errors.

is correctly recognized. To do this each cluster leader will send a real data packet instead of a wake-up receiver pattern. This packet contains the id of the cluster leader together with the list of cluster members—the cluster list—and is used as the reliable announcement of the leadership and the associated memberships. So all nodes which regard themselves as cluster members will switch on their main transceiver for the slot of their associated cluster leader to receive this announcement. They can then check if they are included in the cluster member list. Again, if they are not included in the member list they will become cluster leaders themselves and will send a leader announcement packet in the third frame of Phase II, see Algorithm 5. All other nodes will sleep for the third frame.

Algorithm 5 Validation of Member Status

- 1: listen with main transceiver at tentative leader slot
- 2: **if** announcement received **then**
- 3: **if** ownID \notin received CL **then**
- 4: send own leader announcement next frame
- 5: **end if**
- 6: **else**
- 7: send own leader announcement next frame
- 8: **end if**

Figure 1 shows one example of the clustering protocol for three nodes. Without loss of generality four TDMA slots are assumed. In this case node 1 and node 3 measure the same values while node 2 measures different ones. Node 1 and node 3 will build a cluster with node 1 as its cluster leader. Node 2 becomes cluster leader for its own cluster. With perfect communication no pattern or packet is lost and therefore in the third frame of Phase II no packets are sent.

C. Traditional Approach

As a comparison to the new clustering method a traditional approach is presented. The basic concept uses the clustering algorithm from the PAQ protocol, but is transferred to a TDMA scheme. This approach is similar to the new one

presented above, but because no wake-up receiver is used they will now send real data packets instead of wake-up receiver patterns. Similar to the new clustering approach, in Phase I the information gathering is performed and in Phase II the cluster leader selected. Therefore, the packets in Phase I will contain the real measurements and in Phase II they will contain the leader announcements together with the corresponding memberships. Of course, to receive these packets the main transceivers have to be switched on for all slots.

The approaches differ in Phase II, as with the traditional approach it is possible to receive all leader announcements and the corresponding memberships within one frame. This is possible because the main transceiver has a much higher data rate.

In Phase I of the traditional approach no patterns are sent indicating data similarity, the similarity has to be calculated for each received measurement. To do this, each received sensor reading will be compared to its own sensor reading and when the difference between these two is smaller than δ , this reading is considered similar. The corresponding node is then treated as a cluster member as in the new approach.

The main difference between the new and traditional approach is that the main transceiver has to be switched on more often in the traditional approach. The traditional approach is more reliable because the main transceiver is always used with its higher sensitivity and the whole clustering algorithm in the traditional approach can be performed in $m + 1$ frames instead of $m + 3$. Nevertheless the major advantage of the new approach is that it is highly energy efficient in comparison to the traditional approach, as shown in Section IV. Only this high energy efficiency will enable the usage of clustering protocols in specific environments.

IV. SIMULATION

A simple scenario is implemented in the OPNET network simulator as proof-of-concept of the clustering algorithm. In this scenario several nodes are randomly distributed in a 5×5 m

square and the sink is centrally positioned. The modeled wireless sensor node uses relatively new components. The relevant components on this node are the microcontroller, the main transceiver and the wake-up receiver. The following chips are assumed: for the microcontroller the EFM32G210 [12] from Energy Micro, for the main transceiver the AT86RF212 [13] from Atmel and for the WUR the Wake-up Receiver [14] from the EU project CHOSeN described in Section II-A.

Deduced from their capabilities, the settings from Table II are used for the simulation. Additionally, in the simulated

| | |
|-----------------------|-------------|
| data rate | 40 kbps |
| slot length | 12.8 ms |
| frame length T | 1 s |
| guard interval slot | 100 μ s |
| guard interval beacon | 50 μ s |
| slots per frame | 77 |
| beacon size | 96 bit |
| WUR pattern size | 64 bit |
| packet size | 512 bit |
| carrier frequency | 868 MHz |
| transmit power | 0 dBm |
| P_{tx} | 39 mW |
| P_{rx} | 27, 6 mW |
| P_{wur} | 0.0024 mW |
| P_{micro} | 17.28 mW |

TABLE II
SIMULATION SETTINGS

scenario each node is within communication range of each other and has knowledge about the fixed TDMA schedule. It knows implicitly who is sending at any given time.

A. Preliminary

1) *Wake-up Receiver:* In order to model the detection behavior of the Wake-up Receiver two error rates—the correlation errors—are introduced. A false wake-up rate P_{fwr} and a missed detection rate $\overline{P_{det}}$. For the missed detection rate holds $\overline{P_{det}} = 1 - P_{det}$, where P_{det} is the probability for a successful detection. The missed detection rate covers the situation when the wake-up receiver missed a pattern for which it was listening. On the other hand the false wake-up rate covers the situation when the wake-up receiver detects a pattern for which it was not listening but nevertheless thought that a “correct” pattern arrived. Later on it will be shown that these two error rates have a considerable influence on the performance of the WUR-protocol.

2) *Sensor Readings:* In order to test the clustering algorithm a simple data model is assumed. Each node generates a new sensor reading at the beginning of each frame. To represent several clusters of similar sensor readings all nodes are uniformly distributed into i groups. Of course this information is not publicly available within the network and only used to calculate the sensor readings. Within a group all readings have the same constant value v_i and between groups these values follow $v_{i+1} = v_i + D$, where $D > \delta$ is a constant offset between the values of adjacent groups.

B. Simulation Results

Several simulation runs are performed to test the protocols under different situations. First to highlight the function of the new WUR-clustering protocol, the cluster value $i = 3$ is fixed and the offset D is set to 1. Additionally, the correlation error rates are set to 0 to simulate ideal correlation within the wake-up receiver.

1) *Concept Presentation:* The total consumed energy for node 1 and node 31 is shown in Figure 2. They are selected as an example because the main power consumption of these nodes is separated by a distinct time period and therefore it is easier to distinguish between them. In this case, node 1 is a cluster leader and node 31 an associated cluster member. The consumed energy by the wake-up receiver and the microcontroller is modeled and included into the total energy consumption, but will not be visualized separately, because their influence is marginal compared to the energy consumed by the main transceiver. The first frame starts with the arrival of the first beacon at simulation time $t = 1$ s. At that time the first peak of consumed energy is recognizable for node 31. This peak occurs because the main transceiver is switched on for the reception of the beacon. Such a behavior can also be seen for each further beacon reception at $t = 2$ s, $t = 3$ s, $t = 4$ s, $t = 5$ s and $t = 6$ s. At time $t = 5$ s this small peak is not recognizable because the energy for listening at the directly following slot adds up to a big peak. Such a big peak can also be seen for each time when slot 31 is active and node 31 is sending a pattern. This happens for example in the first frame at $t \approx 1.39$ s. These peaks are higher than the peaks receiving a beacon because the time to send a pattern is longer than that for receiving a beacon. A cluster member will join its cluster at the end of frame 5 as no correlation errors will occur in this simulation setup. Therefore, no further packets have to be sent in frame 6.

A similar behaviour can be seen for node 1, the cluster leader. In the first three frames the consumed energy for node 1 is the same as for node 31 but consumed at slot 1 instead of slot 31. In frame 4 the *pattern(CH)* will be sent to publish a leader notification. This is followed by a real packet in frame 5 as the leader announcement. Again in frame 6 no packets will be sent, because in this simulation run no correlation errors occur.

For comparison, the total energy consumption for the standard-clustering protocol is shown in Figure 2(b). The energy consumption for node 1 and 31 are nearly equal. As can be seen, the main transceiver is switched on most of the time, so a near linear energy consumption is recognizable for the whole clustering protocol. As described in Section III-C in the traditional approach, the clustering process is done within four frames instead of six. But the traditional approach consumes approximately 110 mJ, significantly more than the corresponding 2.4 mJ for a cluster member in the wake-up receiver approach. The cluster leader is only using 3.0 mJ.

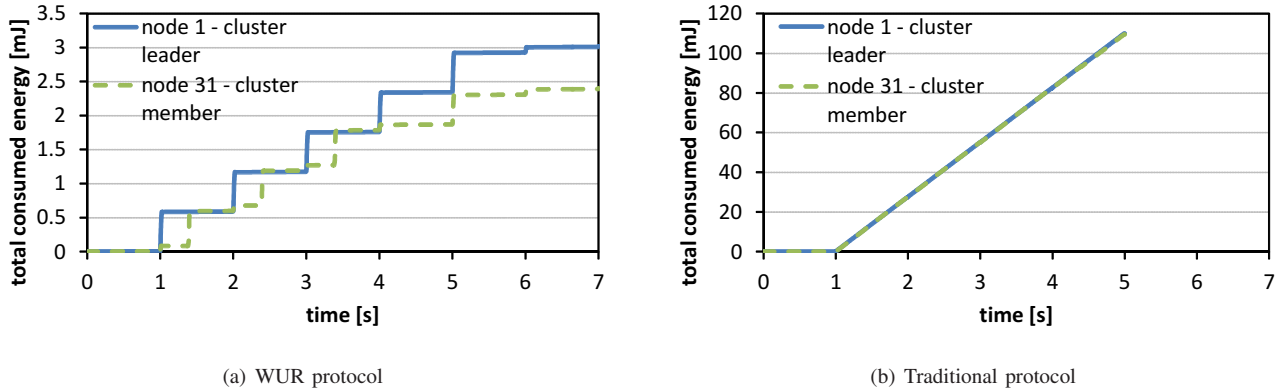


Fig. 2. Mind the different scaling — (a) and (b): total energy consumption by node 1 and 31 while establishing the clusters for all 77 nodes with $m = 3$ and $\text{thold} = 2$.

2) *Scalability*: A major difference between both protocols is that the energy consumption for the WUR protocol is independent of the number of nodes present. This situation changes for the standard protocol, which can be seen in Figure 3. The average energy consumption for the standard

detection will be examined. A missed detection is defined as the case where the correct wake-up id is transmitted but not correctly recognized. An error will have a different impact on the protocol depending on what kind of wake-up pattern is interpreted falsely, i.e. the $\text{pattern}(v_k)$ which disseminates the information about the sensor readings or the $\text{pattern}(\text{CH})$ which acts as the leader notification.

In the case of a missed detection error, a falsely interpreted $\text{pattern}(v_k)$ means that a similar sensor reading is recognized as dissimilar. This will have the effect that a neighbour with similar sensor readings may not be included in the cluster list in the case that enough similar sensor readings are treated as dissimilar (Algorithm 4). A missed leader notification will force the node that missed it to become a cluster leader itself. Ultimately, both cases lead to more cluster leaders but nevertheless the nodes within a cluster still have similar sensor readings. Therefore, the number of members per cluster will decrease with an increasing number of leaders.

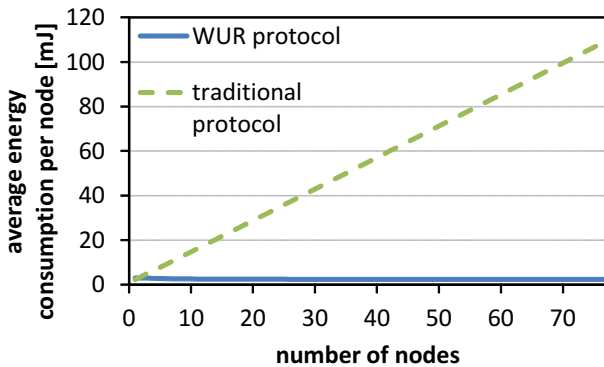


Fig. 3. Systemwide average of energy consumption per node over varying number of nodes.

protocol increases linearly with an increasing number of nodes, while the graph for the WUR protocol is independent from it and stays consistently around 2.4 mJ. This is the case because the WUR protocol has no frame where the main transceiver is switched on for more than one slot. In one frame either a pattern/packet will be sent or for one specific slot the receiver will be active. In the traditional protocol each node has to listen to all slots with their main transceiver which limits scalability. Conversely the WUR protocol is highly scalable because an additional node does not increase the overall power need.

3) *Impact of correlation errors*: In order to examine the impact of correlation errors on the clustering protocol, the false wake-up rate and the missed detection rate is alternately set to zero for the next simulation runs. To distinguish between the different impacts of the error rates, two different simulation sets are performed. Firstly, the impact of a missed

Secondly, the impact of a false wake-up error is examined. A false wake-up error is the case when a wake-up pattern is received and falsely recognized as one the WUR was listening for even though it was not. In the case of a false wake-up error a dissimilar sensor reading will be treated as similar. The impact of that is bigger because it could result in a cluster leader treating a node with dissimilar sensor readings as a valid cluster member. Therefore the number of members per cluster will increase with an increasing number of leaders. In this case the total number of nodes do not increase but a node will become a member in several clusters due to the misinterpreted sensor readings. This means that the data sink may receive a cluster list in which members with dissimilar sensor readings are listed. Therefore a data prediction model will produce incorrect values for these nodes. This effect can be mitigated when data pre-processing is done in the sink, because the sink can recognize that a node is member of several clusters and can act accordingly. In Phase II of the clustering protocol only one type of patterns will be sent—the $\text{pattern}(\text{CH})$ —and because of that a false wake-up error cannot occur in Phase II. As can be seen in Figure 4 three different curves

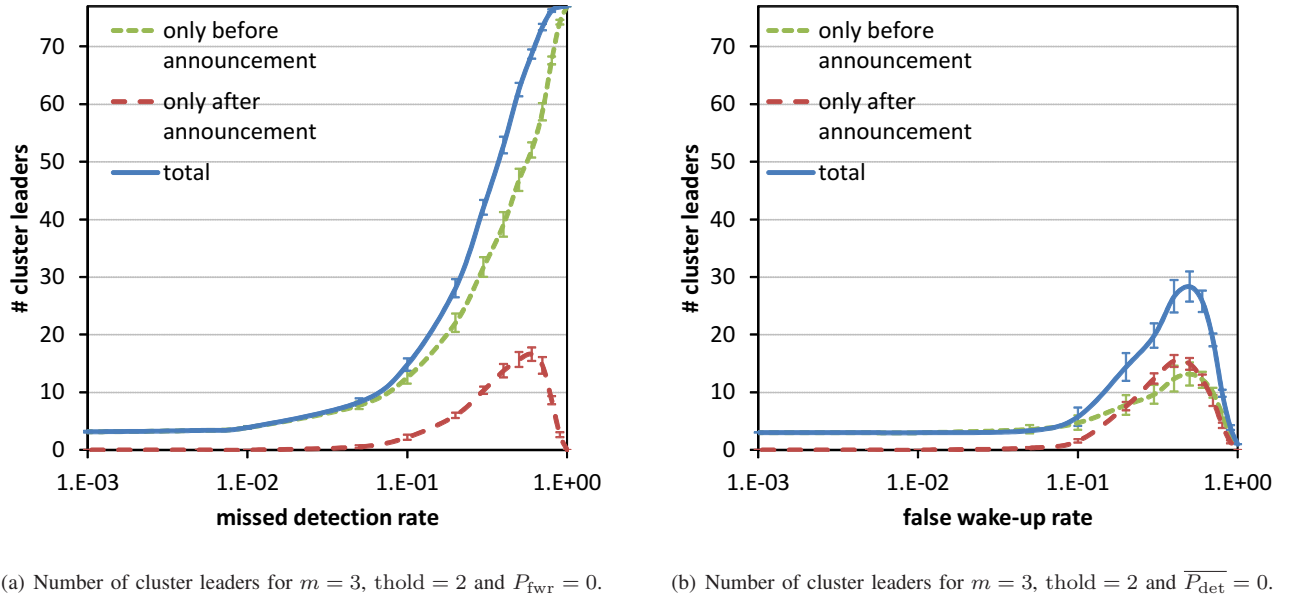
(a) Number of cluster leaders for $m = 3$, $\text{thold} = 2$ and $P_{\text{fwr}} = 0$.(b) Number of cluster leaders for $m = 3$, $\text{thold} = 2$ and $\overline{P}_{\text{det}} = 0$.

Fig. 4. Number of cluster leaders before an announcement packet was send, the additional leaders after an announcement packet was send and the total number of cluster leaders; Each with 95 %-confidence interval. For (a) the missed detection rate and (b) the false wake-up rate.

are plotted; the number of leaders before the announcement packet was sent, the additional number of leaders after the announcement packet was sent and the total number of leaders. For all curves confidence intervals with a confidence level of 95 % are shown. The green curve represents all nodes which consider themselves as cluster leaders before the announcement packet is sent. Nodes which did not consider themselves as cluster leaders before the announcement packet and changed their status to cluster leader following the announcement are represented in red. The increased number of cluster leaders after the announcement packet is explained by nodes which did not find their id on the member list and consequently became leaders. The total number of leaders is the number of leaders at the end of the clustering process.

What can be seen in Figure 4(a) is that the number of leaders starts to change only around $\overline{P}_{\text{det}} \approx 0.01$. A lower missed detection rate did not have any influence on the clustering protocol. With an increasing rate of missed detections the number of leaders increases as well until all nodes become a cluster leader. As can be seen in 4(b) there is a point from which the number of additional leaders decrease until just one cluster exists. Meaning that all nodes consider their sensor readings as similar. As can be seen in Figure 4(b) the false wake-up rate did not have any influence on the clustering protocol for $\overline{P}_{\text{det}} \approx 0.05$. Only a very high error rate will have an impact on the functionality of the system and in normal circumstances will not occur.

To strengthen the protocol the number m of sensor readings which have to be above the threshold thold can be increased. This has the consequence of an overall increase of the duration of Phase I (Algorithm 1). However, the protocol is adjustable and can be manipulated to the requirements of the specific

application.

As can be seen in Figure 5 the clustering protocol becomes less sensitive to correlation errors with increasing m . E.g. with $m = 1$ a correlation error will instantly decrease the performance of the clustering protocol as can be seen in Figure 5(b). A good compromise between a longer clustering period and a strengthened protocol is $m = 3$ and $\text{thold} = 2$.

V. CONCLUSION

An energy-efficient clustering scheme was presented which divides nodes into clusters with similar sensor readings. Energy efficient wake-up receivers are used to disseminate the information about sensor readings throughout the network. Therefore the main transceiver can be set into a deep sleep mode and thus the power consumption is reduced. When using the new protocol the consumed energy for the clustering is independent from the number of nodes and therefore scales better than most of the other clustering protocols which uses data similarity. It was shown that correlation errors can have considerable influence on the clustering protocols, but when the correlation error rates are below $1 \text{ E} - 02$ the clustering protocol is able to compensate for them.

The next step will be to investigate the behaviour of the protocol in respect to varying sensor readings and to use real channel models in the simulation. The clustering protocol will be implemented on wireless sensor nodes to evaluate the protocol under real conditions. Also a data prediction model combined with the clustering protocol will be evaluated and a method to monitor the accuracy of the cluster prediction will be introduced.

Theoretically, pre-processing of the cluster member lists in the sink before the data prediction starts can increase the

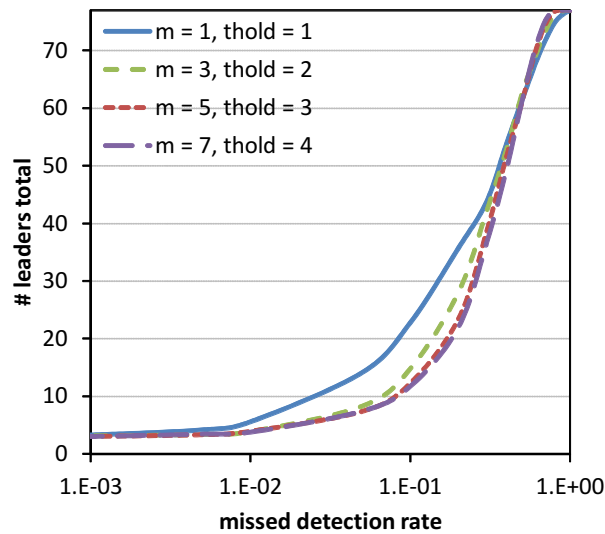
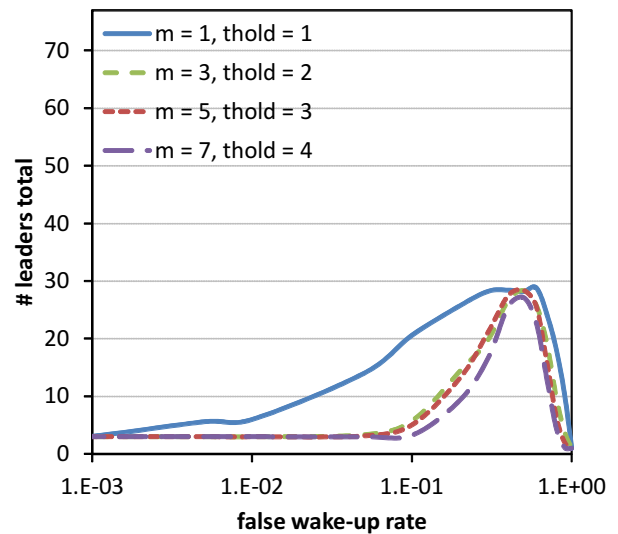
(a) Total number of cluster leaders for $P_{fwr} = 0$.(b) Total number of cluster leaders for $\overline{P_{det}} = 0$.

Fig. 5. Total number of cluster leaders for (a) the missed detection rate and for (b) the false wake-up rate.

performance of the prediction model and therefore would open a new field of research for this clustering protocol.

ACKNOWLEDGEMENT

The author would like to acknowledge funding by the German Federal Ministry of Education and Research project "AETERNITAS".

REFERENCES

- [1] L. Mateu, F. Moll, and U. Polit, "Review of Energy Harvesting Techniques and Applications for Microelectronics," in *VLSI Circuits and Systems II*, 2005, pp. 359–373.
- [2] U. Mitchell, Bradley J. (Snohomish, WA, U. Laib, Trevor M. (Woodinville, WA, and U. Callahan, Kevin S. (Shoreline, WA, "Energy harvesting devices," pp. 1–6, 2008. [Online]. Available: <http://www.freepatentsonline.com/y2008/0092937.html>
- [3] D. Samson, M. Kluge, T. Becker, and U. Schmid, "Wireless sensor node powered by aircraft specific thermoelectric energy harvesting," *Sensors and Actuators A: Physical*, vol. 172, no. 1, pp. 240–244, Dec. 2011. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0924424710005182>
- [4] M. Kluge, D. Samson, T. Becker, A. Gavrikov, B. Bennemann, and J. Nurnus, "Efficient Power Management for Energy Aware, Self-Sufficient Wireless Sensors in Aeronautic Applications," in *The 10th International Workshop on Micro and Nanotechnology for Power Generation and Energy Conversion Applications - PowerMEMS 2010*, no. 2, Leuven, 2010, pp. 2–3.
- [5] J. Sabater, M. López, and J. M. Gómez, "Energy Saving Algorithms for Wireless Sensor Communications: The Aeronautical Case Study," *Sensor Letters*, vol. 6, no. 3, pp. 386–394, 2008.
- [6] C. Hambeck, "Ultra-low Power Wake-up Receiver for Wireless Sensor Networks," Ph.D. dissertation, Technische Universität Wien, 2011.
- [7] D. Tulone and S. Madden, "An energy-efficient querying framework in sensor networks for detecting node similarities," in *Proceedings of the 9th ACM international symposium on Modeling analysis and simulation of wireless and mobile systems - MSWiM '06*, ser. MSWiM '06. New York, New York, USA: ACM Press, 2006, p. 191. [Online]. Available: <http://doi.acm.org/10.1145/1164717.1164768>
- [8] —, "PAQ: Time Series Forecasting for Approximate Query Answering in Sensor Networks," *Lecture Notes in Computer Science*, vol. 3868, pp. 21–37, 2006. [Online]. Available: <http://www.springerlink.com/content/f835265545h28304/>
- [9] U. Goni and A. Turkmani, "Software estimation of CDMA user capacity," *Spread Spectrum Techniques for Radio Communication Systems, IEE Colloquium on*, pp. 11/1–11/6, Jun. 1992. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=168656
- [10] A. Turkmani and U. Goni, "Performance evaluation of maximal-length, Gold and Kasami codes as spreading sequences in CDMA systems," in *Proceedings of 2nd IEEE International Conference on Universal Personal Communications*, vol. 2. IEEE, 1993, pp. 970–974. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=528523
- [11] J. Liu, "Generalized Kasami Sequences: The Large Set," *IEEE Transactions on Information Theory*, vol. 53, no. 7, pp. 2587–2598, Jul. 2007. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4252350>
- [12] Energymicro, "EFM32G210 data sheet." [Online]. Available: <http://www.energymicro.com/products/efm32g210f128>
- [13] Atmel, "AT86RF231 data sheet." [Online]. Available: http://atmel.com/dyn/products/product_card.asp?part_id=4338
- [14] C. Hambeck, S. Mahlknecht, and T. Herndl, "A 2.4 uW Wake-up Receiver for wireless sensor nodes with -71 dBm sensitivity." IEEE, May 2011. [Online]. Available: http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5937620