Gravity-based Local Clock Synchronization in Wireless Sensor Networks*

Markus Wälchli, Reto Zurbuchen, Thomas Staub, and Torsten Braun

Institute of Computer Science and Applied Mathematics
University of Bern
Neubrückstrasse 12
3012 Bern - Switzerland
{waelchli,zurbuche,staub,braun}@iam.unibe.ch

Abstract. Contention-based MAC protocols follow periodic listen/sleep cycles. These protocols face the problem of virtual clustering if different unsynchronized listen/sleep schedules occur in the network, which has been shown to happen in wireless sensor networks. To interconnect these virtual clusters, border nodes maintaining all respective listen/sleep schedules are required. However, this is a waste of energy, if locally a common schedule can be determined. We propose to achieve local synchronization with a mechanism that is similar to gravitation. Clusters represent the mass, whereas synchronization messages sent by each cluster represent the gravitation force of the according cluster. Due to the mutual attraction caused by the clusters, all clusters merge finally. The exchange of synchronization messages itself is not altered by LACAS. Accordingly, LACAS introduces no overhead. Only a not yet used property of synchronization mechanisms is exploited.

Keywords Wireless sensor networks, synchronization, virtual clustering, Networking 2009

1 Introduction

Energy-efficient, contention-based MAC protocols maintain low duty cycles. This means the sensor nodes follow periodic listen/sleep cycles. In the listen cycle they are able to communicate with neighbor nodes and can forward pending data. In the sleep cycle they shut down their radio to preserve energy. In order to synchronize their listen/sleep cycles with neighboring nodes, SYNC messages are periodically exchanged. Nodes maintaining the same listen/sleep cycle are organized into clusters by this synchronization mechanism. This synchronization of common listen/sleep cycles is called virtual clustering. In theory it is possible that different, possibly disjoint, listen/sleep cycles occur.

* The work presented in this paper was partly supported by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

In [1] it has been shown that the existence of multiple virtual clusters is also common in reality. Already in a multi-hop network consisting of 50 nodes, four virtual clusters have been encountered. Accordingly, border nodes interconnecting the clusters in [1] had to wake-up up to three times more than normal cluster nodes. This implies decreased sleep cycles and higher energy consumption for those border nodes. Therefore, it is desirable to agree on a common schedule to discharge the border nodes. On the other hand, maintaining a global schedule within the whole network is unnecessary and implies overhead. Common schedules are only locally required, because nodes can only communicate with their neighbors.

To achieve local synchronization we propose a Local Adaptive Clock Assimilation Scheme (LACAS). Its basic idea is similar to the principle of gravitation. In the context of virtual clusters this means that larger clusters attract smaller clusters more than vice versa, until the clusters finally merge. This applies to all clusters which are present in a network or will evolve later. Therefore, different present clusters always converge towards one single cluster. Of course, this requires that the clusters are connected to each other, which presumes the existence of border nodes.

The rest of the paper is organized as follows: In Section 2 relevant related work is presented. Section 3 introduces the local clock synchronization scheme. Simulation results are provided in Section 4. The paper ends with conclusions in Section 5.

2 Related Work

S-MAC [2], T-MAC [3] and DW-MAC [4] are energy-efficient contention-based protocols for wireless sensor networks. All three are based on low duty-cycles and require SYNC messages to synchronize the listen/sleep schedules of the nodes. T-MAC furthermore supports the adjustment of its wake-up period according to pending data traffic. In both protocols each node maintains its own listen/sleep schedule. These schedules are synchronized whenever possible in order to reduce control traffic overhead. Nodes maintaining the same listen/sleep schedule build a virtual cluster. New sensor nodes initially listen to the wireless medium for a specific amount of time to overhear and adapt an existing schedule. If no SYNC message has been received during this period, a node choses its own schedule. Having determined its schedule, any subsequently overheard unknown schedule is adapted too. Thus, virtual clusters are interconnected. The interconnecting nodes are called border nodes and follow multiple schedules, i.e., the schedule of each cluster they are a member of. Accordingly they consume much more energy than normal cluster nodes. Apart from virtual clustering, SYNC messages are also used to adjust clock drifts between network nodes.

TDMA-based MAC protocols such as [5], [6] require the exchange of periodic SYNC messages. However, unlike contention-based protocols, the operation of TDMA-based protocols is based on the concept of clusters. In general, they require a cluster leader which allocates slots to its cluster members. Thus, the problem of virtual clustering is not present as the nodes are per se organized into clusters. On the other hand, TDMA-based protocols require very precise synchronization and scale rather poorly.

Apart from protocols that require synchronization, asynchronous contention-based MAC protocols [7], [8], [9], [10] have been proposed. [7], [8] and [9] are based on preamble-sampling. These protocols send long preambles to reach neighboring nodes that are currently asleep. RI-MAC [10] avoids the transmission of preambles. In RI-MAC receiver nodes announce their availability by beacon messages. Based on the reception of such a beacon, a sender node transmits its pending data to the receiver. The approach achieves low duty cycles. Asynchronous protocols do not face virtual clustering, but require the exchange of preambles or beacons. Moreover, broadcast operations are poorly supported.

The problem of virtual clustering, i.e., of coexisting schedules, has been addressed in [1]. To solve the problem, an additional schedule age is introduced. The authors motivate that different schedules must have entered the network at different time points and thus have different ages. The schedule age is announced in the SYNC message. Over time all nodes converge toward the oldest schedule in the network. To prevent network partitions all other schedules need to be temporary stored too. The maintenance and distribution of the schedule age requires additional information. This paper will show that no schedule age is needed if local schedule consistency is sufficient.

3 Local Adaptive Clock Assimilation Scheme (LACAS)

Nodes implementing an energy-efficient contention-based MAC protocol such as S-MAC follow periodic listen/sleep schedules. Nodes with the same schedule are virtually organized into clusters. To support communication between different virtual clusters, border nodes interconnecting the according clusters are required. This synchronization mechanism is shown in Fig. 1.

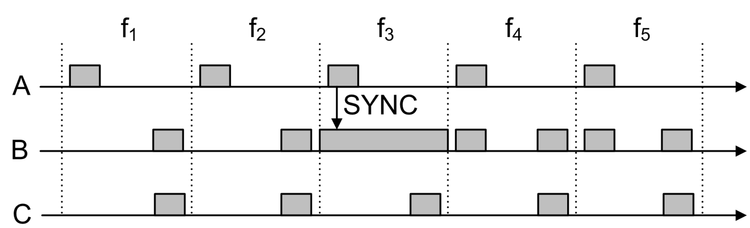


Fig. 1. Periodic sleeping and virtual clustering in S-MAC and T-MAC.

Node A is member of a virtual cluster, whereas nodes B and C are members of another disjoint virtual cluster. This is indicated by the listen periods in Fig. 1. All nodes could be in transmission range of each other. However, in the example above, it is only required that node A can hear node B and node B can hear both nodes A and C. From time to time each node remains awake for an entire frame length f_i in order to scan for present schedules. In Fig. 1 this is node B in frame f_3. B learns the cluster of node A in frame f_3, because it overhears the SYNC message sent by node A. Only this SYNC transmission is shown in Fig. 1. Node B becomes a border node as it interconnects two clusters. This means that node B synchronizes to both known schedules after frame f_3.

Experiments have shown [1] that in S-MAC already in a multi-hop network consisting of 50 nodes four different virtual clusters evolved. Moreover, it has been shown that border nodes had to listen to up to three different schedules. In all four experiments more than 44% of all network nodes followed at least two schedules. In two of the four experiments 34%, respectively 47%, of all network nodes had to listen to three virtual clusters. Obviously, the border nodes thus have a higher average energy consumption than normal cluster nodes.

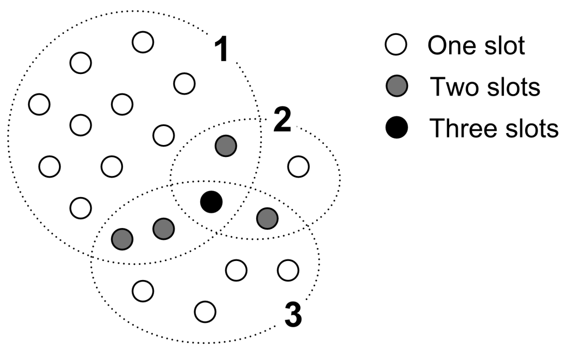


Fig. 2. Drawback of virtual clustering.

The problem is illustrated in Fig. 2. The gray and black nodes operate as gateway nodes between the clusters and have to listen to multiple schedules. Accordingly, these nodes sleep less and their batteries deplete sooner. If this happens, network connectivity might be broken and the network might be disconnected, even though sufficiently many working network nodes could still exist. Therefore, it is desirable to avoid virtual clustering. In [1] it has been shown that surprisingly many nodes follow multiple schedules. The temporary failure of communication links and effects of strongly varying radio ranges (communication gray zones [11]), have been identified as main reasons.

While the authors of [1] use a global mechanism to solve the problem, we propose a local adaptive clock assimilation scheme (LACAS) that achieves local synchronization. A global solution requires system-wide synchronization towards one global schedule. This implies overhead in terms of signaling and requires the storage of fallback mechanisms, i.e., of temporary valid local schedules. LACAS avoids these drawbacks. Maintaining a global schedule is unnecessary, because of the locality of communication links between network nodes. LACAS avoids the drawback of virtual clustering and leads to a uniform distribution of the energy consumption that is required for synchronization.

LACAS implements a mechanism similar to gravitation. Translated into the virtual clustering problem, this means that larger clusters attract smaller ones more than vice versa, until the clusters finally merge. In LACAS, the cluster nodes represent the mass and the number of sent SYNC messages represent the gravitation force. Because all sensor nodes implement the same contention-based transmission scheme, large clusters broadcast in average more SYNC messages than small ones and accordingly cause more attraction.

LACAS only exploits the information exchanged by synchronization messages. Therefore, no additional control traffic is generated. Moreover, the loss of SYNC messages does not affect the principle of LACAS, but only temporarily decreases the gravitation force of a cluster.

Only border nodes are part of multiple clusters. Accordingly, only border nodes attract clusters. Whenever a border node evolves, it spans its listen period over all schedules it knows. Furthermore, every node adapts its own listen/sleep schedule to a given percentage α (e.g., α = 5%) towards the schedule of the cluster from which it has received a SYNC message. Accordingly, the parameter α controls the attraction caused by a SYNC message. In a first step of a merging process, the schedule of a border node is expanded and then it starts to contract again.

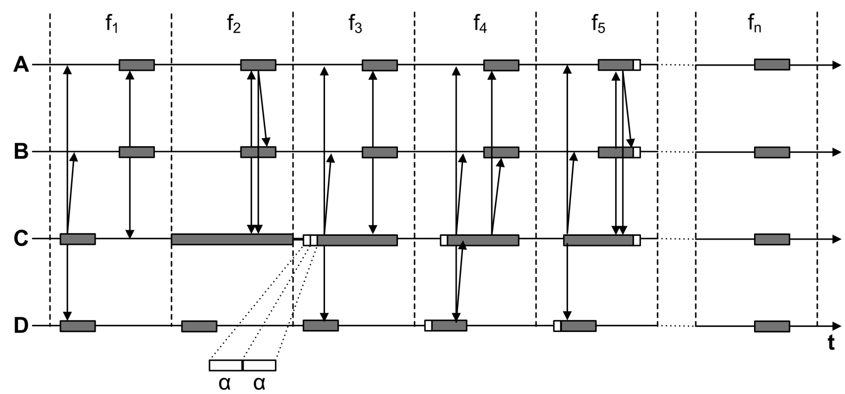


Fig. 3. Gravitation principle of LACAS: Neighbor clusters are detected in specific frames. Then, the clusters fuse (slowly).

A merging process is shown in Fig. 3. The dark bars indicate listen periods, while the white bars indicate schedule adaptations. Node C stays awake for a whole frame f_i, i.e., a whole listen/sleep period, in f_2. Having detected another schedule, it spans its listen period over both known schedules (see f_3). This listen period contracts then to the normal schedule length merging both connected clusters. The parameter α controls the gravitation force. High values for α lead to high attractions and fast convergence. On the other hand, this can temporary break connections between clusters. However, only the convergence time of the clusters is affected. The gravitation mechanism itself is not compromised. In the worst case, nodes are successively transferred from the smaller cluster to the larger. The problem is discussed in detail below. The contraction of the schedule of node C continues after period f_5 and ends in period f_n.

The operation in Fig. 3 is discussed in the following: Initially nodes A and B form cluster 1, while nodes C and D form another cluster 2 (see frame f_1 in Fig. 3). Because T-MAC is used, all nodes stay periodically awake for an entire frame f_i in order to detect other clusters. In Fig. 3 this is node C in frame f_2. Having learned both clusters, C spans its own listen period over both schedules. Moreover, as it has received two SYNC messages from nodes A and B from cluster 1, node C moves its listen period for 2α towards the listen period of cluster 1 (frames f_2 and f_3 in Fig. 3). In frame f_3, C is able to transmit its own SYNC message and receives another one from node B from cluster 1. Accordingly, the listen period of node C again moves for α towards the listen period of cluster 1. Due to SYNC from node C, node D is attracted too (f_4 in Fig. 3). In frame f_4 node C is able to transmit a SYNC message in both schedules. Accordingly, cluster 1 and node D are attracted towards C. C itself moves towards D as it has received a SYNC message from D (frame f_5 in Fig. 3). The merging process continues in Fig. 3 after frame f_5. After a while, both clusters will fuse. In the example, cluster 1 transmits in general more SYNC messages than cluster 2 (It has three members, whereas cluster 2 has only two). Accordingly, both clusters will merge closer to the original schedule of cluster 1.

The choice of the adaptation parameter α is crucial considering performance. A small value implies a long merging period. On the other hand, if a large value has been chosen for α, the fast convergence towards a large cluster might disrupt the connection between a border node and its smaller cluster. This disconnection is no basic problem, because the clusters are connected again when a border node remains awake for a whole frame, but it increases merging time. In the worst case, nodes pass successively from the smaller cluster to the larger. The convergence of LACAS is however not affected. Moreover, growing clusters have in general also a growing number of border nodes and therefore their gravitation force increases too. In the current deployment we have chosen a value for α of 5%. Thus, all sensor nodes are able to synchronize within a few minutes. Expecting a network lifetime of at least several months, the synchronization time seems tolerable. Finally, only the listen periods of the according MAC protocol are optimized. Any subsequent data exchange period is not affected by LACAS.

4 Evaluation

LACAS has been implemented on top of T-MAC in the network simulator OMNeT++ [12]. All network nodes wake up randomly within the first 30 simulation seconds, and begin immediately to synchronize. T-MAC enhanced with LACAS has been used as MAC protocol for a topology control algorithm [13]. Because

LACAS has been implemented using a cross-layer approach, we evaluate the performance of LACAS in a larger context. The topology control mechanism establishes a routing backbone after a synchronization and neighborhood learning period of 400 seconds. Non-backbone nodes temporarily shut-down their radios to save energy. This has some impact on the convergence time of LACAS as fewer SYNC messages are sent due to the temporary unavailability of the non-backbone nodes.

The parameters of T-MAC have been set according to [3]. All nodes follow a periodic listen/sleep frame of 610 ms, whereof they are awake for at least 13.5 ms, i.e., if no data transmission is pending. This minimum wake-up period consists of the synchronization period, which is 7 ms, and the traffic-adaptivity period TA, which is required by T-MAC and has a duration of 6.5 ms. Each node remains awake for a whole frame in every 35th frame, i.e., every 21.35 s. This is required in order to detect neighbor nodes which follow different listen/sleep cycles.

Three different network sizes of 50, 100 and 200 nodes have been simulated. Each node has in average 12 neighbors. The network topology was randomly generated taking network connectivity into account. Any experiment has been repeated 20 times. The spectrum of the schedule lengths present at a specific time point is indicated by the standard deviation.

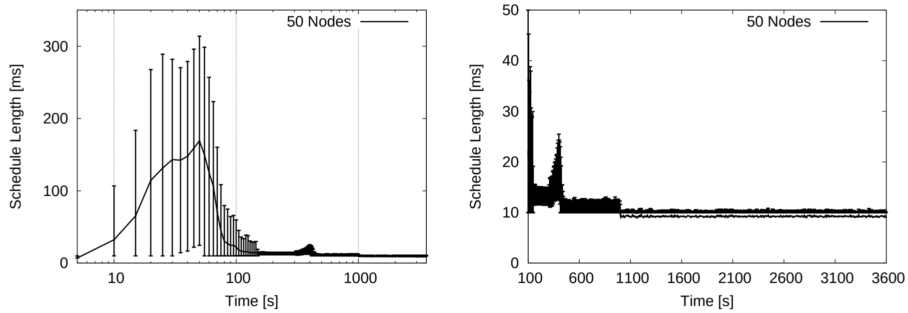
The properties of the sensor nodes are configured according to values from the Embedded Sensor Board (ESB) platform [14]. The nodes operate in the 868 MHz frequency band, the transmission range is about 37 m, whereas the interference range is approximately 52 m. The data rate is 115.2 kbps. The energy consumption in transmission mode is 5.2 mA. Idle listening and receiving both require about 4.7 mA, whereas the radio in sleep mode only needs 5 μA.

4.1 Convergence Time of Schedule Length with LACAS

In this section the convergence time of LACAS is investigated. The independent initial wake up of the network nodes in the first 30 s of the simulation leads to multiple coexisting schedules in the beginning. The evolution of the schedule length of each network node has been monitored over the first hour of operation. The schedule length has been captured every 5 s. The evolution of the mean schedule length in a network consisting of 50 nodes is shown in Fig. 4.

Fig. 4(a) shows the evolution of the schedule length over the whole first operation hour. The schedule length converges to a length of approximately 10 ms within the first 200 seconds. Of course, in this convergence period the distribution of the schedule length is high in the network. There are nodes that follow common schedules and thus have already a short schedule length. On the other hand, there are numerous nodes interconnecting different schedules, which results in a temporary large schedule length.

Fig. 4(b) shows the evolution of the mean schedule length after the first 100 seconds. The peak at second 400 is due to the backbone scenario as described above. At second 300 the parameter α is adapted from 0.05 to 0.5 to achieve faster convergence. This leads to the temporary peak. After the adaptation a little better performance can be achieved, though. Sleeping non-backbone nodes

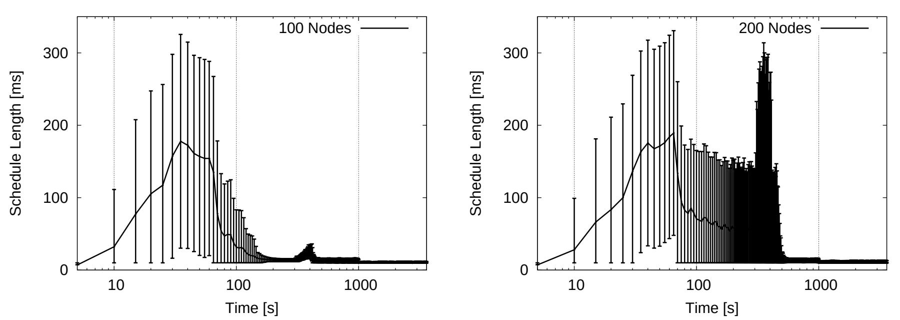


(a) Average schedule length evolution over the first hour (log-scaled).

(b) Average schedule length evolution ignoring the first 100 seconds.

Fig. 4. Convergence of schedule length in a network consisting of 50 nodes.

lead to a smaller amount of SYNC messages, which further reinforces the effect. The peak is in the order of a duplication of the schedule length. The adaptation of α could be implemented in LACAS without cross-layer optimization too. The mean schedule length converges to 13 ms without adaptation and to 10 ms with adaptation. The average schedule length remains stable after 200 seconds without adaptation. With adaptation stability is achieved after 450 seconds. The point in time when stability is achieved is protocol-dependent, though.



(a) Network consisting of 100 nodes.

(b) Network consisting of 200 nodes.

Fig. 5. Average schedule length evolution over the first hour (log-scaled).

Fig. 5 shows the impact of the network size. The performance of LACAS in a network consisting of 100 nodes is depicted in Fig. 5(a). The performance is very similar to the performance in the network consisting of 50 nodes. However, the convergence needs more time in the network consisting of 200 nodes (Fig. 5(b)). Compared to an intended network lifetime of several months or more, this delay is still insignificant, though. The convergence time of LACAS increases with network size. This is due to the hop-by-hop impact of the gravitation principle. Due to gravitation, clusters show an impact similar to the movement of a ripple through water over multiple hops.

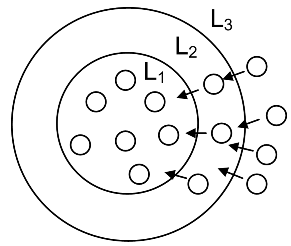


Fig. 6. Ripple effect of gravitation over multiple hops.

The effect is illustrated in Fig. 6. Clusters of nodes such as the nodes in L_1 attract nodes at the boundaries. The nodes in L_2 have again an impact on their border nodes, i.e, on the nodes in L_3. The effect causes complex mutual influences. Moreover, due to the increased time needed for dissemination, clusters located far away from each other have a longer lasting impact on each other than nearby clusters. The probability of presence of such clusters grows with network size. Thus, the convergence time increases with network size too.

Considering the network size of 200 nodes in Fig. 5(b), LACAS is not able to converge to a short schedule length before cross-layer adaptation occurs. The cross-layer impact leads again to a temporary duplication of the average schedule length, which becomes in this case much longer. However, the schedule length converges quickly to 10 ms after the adaptation. This fast convergence is again due to the cross-layer approach. Without optimization the convergence would look similar to Fig. 4(a) or 5(a). It would only require some more time.

The average schedule length has converged in all evaluated network topologies and sizes to a length of approximately 10 ms. This is not surprising, because the convergence (gravitation) is basically a local process. Mainly local communications have an impact on local convergence and any local communication is independent of the network size. On the other hand, due to the ripple effect it is not possible to achieve the schedule length of 7 ms of T-MAC. Accordingly, there is a trade-off between avoiding the border nodes and the achievable schedule length. On node-level, every border node with disjoint schedules consumes more energy than any node running LACAS. Concerning the overall energy consumption, LACAS preserves energy if the following inequation applies ($\forall i : n_i \in \mathbb{N}$):

$$n_1 \cdot 7ms + n_2 \cdot 14ms, +... + n_k \cdot k \cdot 7ms > n_t \cdot 10ms; \quad \sum_{i=1}^{k} n_i = n_t \qquad (1)$$

where n_i is the number of nodes maintaining a given number of schedules and n_t is the total number of nodes in the network. Inequation 1 assumes that the different schedules are disjoint. Else, overlaying schedules would need to be included.

4.2 Analysis of Power Consumption

Unlike the convergence of LACAS, a realistic virtual clustering is difficult to simulate. Virtual clustering mainly occurs due to physical impacts such as communication gray zones [11] and temporary unavailable communication links, which are hardware- and environment-dependent and accordingly difficult to simulate properly. Approximating those impacts in simulations might falsify the simulations rather than improve them. Finally, these effects have little impact on the convergence of LACAS due to the robustness of the gravitation principle.

In order to assess the power consumption needed by LACAS we adopt the results obtained in real world experiments in [1]. The costs of T-MAC and LACAS are computed according to these values and inequation 1. The long-sleep impact of non-backbone nodes has not been considered in this evaluation, because it is based on a cross-layer optimization. Accordingly, all nodes follow a normal listen/sleep schedule. The sensor network in [1] consisted of 50 nodes running S-MAC. The percentage of network nodes maintaining a certain number of schedules are listed in Table 1.

Table 1. Percentage of nodes maintaining a certain number of schedules (from [1]).

	Number of Schedules			
	1	**2**	**3**	**4**
Exp. 1	56%	44%	-	-
Exp. 2	32%	68%	-	-
Exp. 3	-	66%	34%	-
Exp. 4	9%	44%	47%	-

The results would be the same if T-MAC had been used due to the identical synchronization mechanism. As mentioned above, ESB nodes need 4.7 mA in idle listening state. We take this value to estimate the power consumption of LACAS. Furthermore, we assume that the different schedules, which evolved in the experiments in [1], are disjoint (see also inequation 1). SYNC messages that would have to be sent in the synchronization periods are not considered. Table 2 shows the power consumption of T-MAC and LACAS to maintain all schedules of all network nodes in one listen/sleep cycle. The results apply as soon as the networks are stable, i.e., after the convergence to the common schedule length of 10 ms in the case of LACAS, respectively after all virtual clusters have evolved in the case of T-MAC. Therefore, the values in Table 1 can be used. LACAS maintains only one schedule. Accordingly, the expected power consumption of LACAS would be the same in all four experiments.

Table 2. Power consumption (in mAs) per listen/sleep cycle for T-MAC and LACAS.

	T-MAC	LACAS
Exp. 1	2.37	2.35
Exp. 2	2.76	
Exp. 3	3.85	
Exp. 4	3.92	

The estimations in Table 2 show that in a network consisting of 50 sensor nodes, depending on the experiment, more or less power can be saved with LACAS, i.e., between 0.02 and 1.57 mAs in one listen/sleep cycle of 610 ms. Even though LACAS has a slightly longer minimal schedule length than T-MAC, LACAS is estimated to perform at least as well as T-MAC in all four experiments performed in [1]. Considering a network lifetime of months or more, the possible energy savings are promising. The estimations shown in Table 2 concern the average energy consumption over all nodes. Border nodes with disjoint schedules consume more energy and would therefore even sooner run out of energy.

5 Conclusions

In this paper a simple local clock synchronization scheme has been proposed. In general, nodes are not synchronized after deployment. Merging nodes with similar sleep/listen cycles into clusters leads to the problem of high energy consumption for the nodes connecting the clusters. Therefore, we have proposed a simple local synchronization mechanism (LACAS). LACAS provides system-wide local clock consistency and avoids the drawback of virtual clustering. It has been shown that the overhead of LACAS is marginal. Moreover, the synchronization procedure converges fast, i.e., within minutes for the simulated networks and remains stable thereafter.

In simulations the fast convergence of the algorithm has been shown. LACAS just exploits attraction information exchanged by SYNC messages. If messages are lost, attraction is decreased temporarily, but the functionality of LACAS is not affected. The according cluster just shows currently lower attraction. Due to the difficulty to realistically simulate virtual clustering, the energy consumption of LACAS has been estimated offline based on real-world results collected in related work. The estimations indicate that LACAS would save energy in a real-world implementation.

References

1. Li, Y., Ye, W., Heidemann, J.: Energy and latency control in low duty cycle MAC protocols. In: Proc. of the IEEE Wireless Communications and Networking Conference (WCNC '05), New Orleans, LA (March 2005) 676–682
2. Ye, W., Heidemann, J., Estrin, D.: Medium access control with coordinated adaptive sleeping for wireless sensor networks. IEEE/ACM Transactions on Networking **12**(3) (June 2004) 493–506
3. van Dam, T., Langendoen, K.: An adaptive energy-efficient MAC protocol for wireless sensor networks. In: Proc. of the 1st international conference on Embedded networked sensor systems (SenSys '03), Los Angeles, CA (2003) 171–180
4. Sun, Y., Du, S., Gurewitz, O., Johnson, D.B.: DW-MAC: a low latency, energy efficient demand-wakeup MAC protocol for wireless sensor networks. In: Proc. of the 9th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '08), Hong Kong, Hong Kong, China (2008) 53–62
5. van Hoesel, L., Havinga, P.: A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches. In: Proc. of the International Conference on Networked Sensing Systems (INSS '04), Tokyo, Japan (June 2004)
6. Rajendran, V., Obraczka, K., Garcia-Luna-Aceves, J.: Energy-efficient collision-free medium access control for wireless sensor networks. In: Proc. of the 1st international conference on Embedded networked sensor systems (SenSys '03), Los Angeles, CA (2003) 181 – 192
7. El-Hoiydi, A., Decotignie, J.D.: WiseMAC: An ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. In: Proc. of the 9th IEEE International Symposium on Computers and Communications (ISCC 2004). Volume 1., Alexandria, Egypt (June 2004) 244–251
8. Polastre, J., Hill, J., Culler, D.: Versatile low power media access for wireless sensor networks. In: Proc. of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys 2004), Baltimore, MD, USA (November 2004) 95–107
9. Buettner, M., Yee, G.V., Anderson, E., Han, R.: X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In: Proc. of the 4th international conference on Embedded networked sensor systems (SenSys '06), Boulder, Colorado, USA (2006) 307–320
10. Sun, Y., Gurewitz, O., Johnson, D.B.: RI-MAC: a receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In: Proc. of the 6th ACM conference on Embedded network sensor systems (SenSys '08), Raleigh, NC, USA (2008) 1–14
11. Zhao, J., Govindan, R.: Understanding packet delivery performance in dense wireless sensor networks. In: Proc. of the First ACM Conference on Embedded Networked Sensor Systems (SenSys '03), Los Angeles, CA (November 2003) 1–13
12. Varga, A.: Omnet++: a component-based, modular and open-architecture simulation environment (December 2008)
13. Wälchli, M., Zurbuchen, R., Staub, T., Braun, T.: Coordinated sleep MAC for energy-constrained wireless sensor networks. In: Proc. IEEE Global Communications Conference (IEEE GLOBECOM 2009). (2009) submitted.
14. Scatterweb: The self-organizing wireless communication platform (October 2007)