

Provided by the author(s) and University College Dublin Library in accordance with publisher policies. Please cite the published version when available.

Title	Bailigh: Low Power Cross-Layer Data Gathering Protocol for Wireless Sensor Networks
Author(s)	Bober, Wojciech; Bleakley, Chris J.
Publication date	2009-10-24
Conference details	International Conference on Ultra Modern Telecommunications (ICUMT), St. Petersburg, Russia, 12 - 14 October, 2009
Publisher	IEEE
Item record/more information	http://hdl.handle.net/10197/7082
Publisher's statement	© © 2009 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Publisher's version (DOI)	http://dx.doi.org/10.1109/ICUMT.2009.5345543

Downloaded 2018-02-21T19:37:07Z

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa) 

Some rights reserved. For more information, please see the item record link above.



Bailigh: Low Power Cross-Layer Data Gathering Protocol for Wireless Sensor Networks

Wojciech Bober

*Complex & Adaptive Systems Laboratory
School of Computer Science and Informatics
University College Dublin
wojciech.bober@ucd.ie*

Chris Bleakley

*Complex & Adaptive Systems Laboratory
School of Computer Science and Informatics
University College Dublin
chris.bleakley@ucd.ie*

Abstract

Data gathering systems are an important class of Wireless Sensor Networks (WSNs). As the goals of such systems are long lifetime, high reliability, and unattended operation, efficient use of limited energy is crucial. This paper presents Bailigh, a low power cross-layer protocol designed for low rate periodic data collection. Bailigh schedules the network to wake up at regular intervals for brief periods of data collection. To achieve this Bailigh integrates synchronous low power listening and network level scheduling. The proposed synchronous low power listening technique mitigates clock drift with low overhead and reduces unnecessary radio usage. Integrated scheduling enables staggered communication, which effectively reduces collisions and increases delivery rate. We use simulations to compare the proposed approach with a non cross-layer approach. Results show an average duty cycle of 0.1% and delivery rate of 95%. With duty cycle 8.7 times lower than LPL, Bailigh offers network lifetime of 5.8 years.

1. Introduction

Data gathering has received significant attention as a Wireless Sensor Networks (WSNs) application. The promise of cheap sensors deployed at large scale, enabling precise observations, is attractive in areas such as microclimatic research [1], habitat monitoring [2], and precision agriculture [3]. WSN nodes are autonomous devices equipped with a microcontroller, sensors, and a short range radio. In data gathering applications WSNs deliver sensor measurements to a central sink node for further processing. Although various methods of energy harvesting have been proposed, so

far a battery is the most common method of powering nodes. Often nodes are deployed in remote and harsh locations, where replacing battery is difficult if not impossible. Therefore, reducing energy consumption in order to extend network operation time is of significant importance. Of all node components the radio is the main power consumer [4]. Moreover, the energy of transmitting useful data is only a fraction of the overall radio power consumption [5]. Precious energy is wasted on idle listening, collisions, overhearing, and in control overhead [6].

In this paper, we present Bailigh a low power cross-layer data gathering protocol. The proposed protocol is designed for low rate periodic data collection, i.e., the time between consecutive data collects, ranges from minutes to tens of minutes. Bailigh schedules network activity using the data collection period as an explicit protocol parameter. Between data collects network activity is stopped. To achieve this, Bailigh integrates two components: Synchronous Low Power Listening (S-LPL) and network level scheduling. S-LPL is a low power, one-hop communication technique, which does not require maintaining time synchronization. It depends on knowledge of the communication period. This information is determined by the network schedule. Bailigh uses a staggered schedule, where nodes' communication time depends on the data collection period and on a nodes distance from the sink. Each parent node assigns individual communication slots to its children. Bailigh constructs a distributed TDMA schedule based on S-LPL, which reduces collisions and increases delivery rate. In a network of 50 nodes with collection period of five minutes the average duty cycle is 0.11%. Assuming as a power source two AA batteries with capacity of 2700 mAh each, this network is able to operate for 5.8 years. Compared to 1.26

year lifetime of the same network under LPL, Bailigh improves deployment lifetime by 4.6 times.

2. Related work

From the research on energy efficient communication in Wireless Sensor Networks two important approaches have emerged: scheduling and low power listening. Scheduling is a synchronous approach to communication, which reduces idle listening by ensuring that sender and receiver wake up at the same time. In S-MAC [6] and T-MAC [7] nodes form clusters which use the same schedule. The network can be partitioned in multiple clusters and nodes within a cluster are synchronized by one node (cluster head) by means of periodically broadcasted synchronization packet. Scheduling eliminates idle listening but requires maintaining time synchronization or introducing a guard time. This is necessary because nodes' clock exhibit certain drift. This increases control overhead and limits the energy efficiency.

Low Power Listening (LPL) presented in B-MAC [8] and WiseMAC [9] is an asynchronous approach, which uses periodic channel polling to minimize idle listening. Each node periodically checks channel for activity. Once activity is detected the node keeps the radio on to receive incoming packet. This approach does not require a sender and a receiver to be synchronized, instead a long preamble is used to wake up the destined node. LPL suffers from the overhearing problem, because long preamble cause nodes which are not intended for reception to wakeup. Improvements in form of synchronized polling [10] and using a trail of packets as a preamble [11] were proposed.

Synchronous Low Power Listening, proposed in this paper is a one hop communication technique, that combines LPL with scheduling: it schedules receiver and sender communication time and enables LPL for a very brief period of time.

Protocols which focus specifically on continuous data gathering include FPS [12] and Dozer [13]. In FPS [12] a parent node is responsible for scheduling its children, thus the pairwise communication is contention free. However, collisions are possible due to transmissions between other nodes in the tree, FPS handles this using CSMA. FPS requires global and fine grained time synchronization for its operation, due to common schedule and short slots. Dozer [13] uses local schedules and pairwise time synchronization. Each parent schedules its children and schedules between different levels are not synchronized. Beacons are used to maintain time synchronization. Alignment

of schedules is solved by introducing a variable random period between consecutive synchronization beacons.

Unlike, Bailigh does not maintain synchronization between data collects. This allows for a significant reduction in network activity. Bailigh schedules the whole network to send the data during very brief collection periods. Bailigh uses a staggered schedule as originally proposed in DMAC [14]. However, in DMAC nodes do not have individual transmission slots as in Bailigh. Moreover the authors do not discuss how the tree depth necessary to construct the schedule is shared among nodes in the network, which is addressed in Bailigh. The proposed protocol uses a cross-layer approach previously proposed in [15] and [16], but focuses on low data rate applications.

3. Bailigh Overview

Bailigh achieves low power operation because the network is active only for brief periods of time. The rate of data collection is determined by a global parameter - the data collection period. For a typical monitoring application the data collection period ranges from minutes to tens of minutes. Note that the data collection period does not determine how often sensor samples are taken. It is possible that several samples are delivered in one data collect. For example, sensors may be sampled every minute, but collection takes place every ten minutes. Bailigh integrates two techniques which allow intermittent operation: S-LPL and network scheduling.

S-LPL is a synchronous, low power, one-hop, communication technique. It depends on both sender and receiver knowing the time of next communication. During communication, the sender synchronizes its clock to the receiver. In S-LPL, the receiver is responsible for mitigating clock drift. The receiver extends its listen period taking into consideration maximal drift which could have developed since the last communication. During this extended listen period Low Power Listening is used to reduce the energy consumption. This is important since the extended listening period is significant when communication periods are long. S-LPL does not require the maintenance of time synchronization (e.g., sending periodic synchronization beacons) and makes intermittent operation possible.

Waking up nodes at the same time to send data may lead to packet storms. Packet storms cause high contention and negatively affect packet delivery rate. In Bailigh, these issue is addressed by introducing network level scheduling. Bailigh creates a staggered schedule. In this schedule, the moment of communication between nodes depends on their level in tree. At

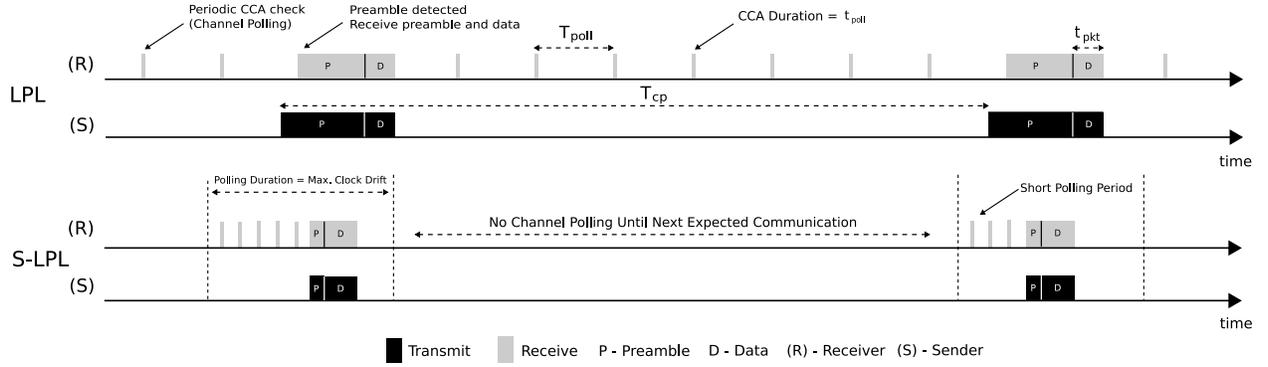


Figure 1. Comparison of LPL and S-LPL.

a given time only two levels of the tree communicate. To avoid collisions between nodes at the same level, a parent node assigns individual time slots to its children. The network schedule depends on S-LPL, because the length of each slot is determined by the receiver's extended listening period.

For tasks which cannot be handled with scheduled traffic (e.g., creating and updating network topology), Bailigh uses maintenance mode. In this mode, nodes use unscheduled and asynchronous communication. The maintenance mode is used on network bootstrap and afterwards is executed periodically to update the network topology.

4. Synchronous LPL

The efficiency of radio communications can be assessed by considering the *Duty Cycle (DC)* (1), the ratio between the radio active time and total operation time. The radio is active during packet transmission, reception, channel polling and channel clear assessment.

$$DC_{active} = DC_{rx} + DC_{tx} + DC_{poll} + DC_{cca} \quad (1)$$

If not active the radio is in a sleep mode, therefore:

$$DC_{sleep} = 1 - DC_{active}$$

Energy consumption per second can be calculated by multiplying duty cycle of each state by corresponding power consumption (2).

$$\begin{aligned} E &= E_{tx} + E_{rx} + E_{poll} + E_{cca} + E_{sleep} \quad (2) \\ &= P_{tx} \cdot DC_{tx} + P_{rx} \cdot DC_{rx} + P_{poll} \cdot DC_{poll} \\ &= P_{cca} \cdot DC_{cca} + P_{sleep} \cdot DC_{sleep} \end{aligned}$$

The amount of time spend by the radio in each state depends on the protocol communication scheme. To compare the proposed approach with LPL we analyze

combined duty cycle of receiver and sender in one-hop scenario. In LPL (Figure 1) the receiver periodically turns off the radio and checks the channel for activity by performing Clear Channel Assessment (CCA). This process is called channel polling. The duty cycle of channel polling (3) depends on time required to perform channel check t_{poll} and polling period T_{poll} i.e. interval between consecutive checks.

$$DC_{poll} = \frac{T_{cp} \cdot t_{poll}}{T_{poll}} \frac{1}{T_{cp}} \quad (3)$$

Once the receiver detects channel activity it stays on listening to receive a packet. The duty cycle of reception (4) includes time required to receive the packet t_{pkt} and preamble. Because the sender and the receiver are not synchronized, on average, half of the preamble will be received. We exclude time required to poll the channel since it was included in polling duty cycle calculations.

$$DC_{rx} = \frac{0.5T_{poll} + t_{pkt} - t_{poll}}{T_{cp}} \quad (4)$$

In order to wake up a receiver, the sender must first send a preamble which is as long as channel polling period T_{poll} plus the time required to poll the channel. Only after that a packet of length t_{pkt} can be send. Before initiating the sender checks the channel for activity in order to avoid collision with ongoing transmission. The duty cycle of the sender are therefore given by equations (5) and (6).

$$DC_{tx} = \frac{T_{poll} + t_{pkt} + t_{poll}}{T_{cp}} \quad (5)$$

$$DC_{cca} = \frac{t_{cca}}{T_{cp}} \quad (6)$$

Note that time required to perform CCA t_{cca} and poll the channel t_{poll} are different. This is because the former relates only to probing the channel whereas the later includes time required to turn on the transceiver,

stabilize crystal oscillator, and finally probe the channel.

To minimize energy consumption, an optimal polling period can be calculated by minimizing DC with respect to the channel polling period (7).

$$\frac{\delta DC_{LPL}}{\delta T_{poll}} = -\frac{t_{poll}}{T_{poll}^2} + \frac{1}{T_{cp}} + \frac{1}{2T_{cp}} = 0 \quad (7)$$

$$T'_{poll} = \sqrt{\frac{2}{3} t_{poll} T_{cp}} \quad (8)$$

The optimal polling period T'_{poll} (8) depends mainly on communication period T_{cp} . If the communication period is unknown the channel polling period can be set to an arbitrary value. In general, decreasing channel polling period decreases latency, but increases energy consumption.

Due to periodic channel polling LPL does not require synchronization between a sender and a receiver. This is suitable for applications which require low latency e.g., event detection networks, but is a significant and unnecessary overhead in data gathering applications. Therefore, we propose an application driven LPL which only enables the radio when communication is expected (Figure 1). We assume that the communication period is known. Nodes synchronize their clocks during packet transmission and switch off their radios until next expected packet transmission. During that time no synchronization information is exchanged. This, however, requires an efficient way of mitigating clock drift. Clock drift is present due to crystal oscillator frequency skew. The frequency skew is caused by temperature, aging, and manufacturing imprecision.

When regular synchronization is not present a guard time (9) can be introduced to mitigate for clock drift. The guard time takes into consideration crystal stability and time elapsed from last synchronization. Since a node does not know whether its clock is faster or slower in relation to other node's clock, the guard time has to be added before and after the expected moment of communication. This is done by extending listening period of receiver and extending packet preamble.

$$t_{guard} = 2 \cdot r_{skew} \cdot T_{cp} \quad (9)$$

$$t'_{guard} = 2 \cdot t_{guard} \quad (10)$$

We propose to use an asymmetric guard time (10) at receiver side and no guard time at senders side. This allows the receiver to use LPL with very short polling period. The duty cycle equations for reception and transmission remain the same as in case of LPL. The difference is in polling duty cycle (11). We assume that on average, only half of the guard time will be

required for a sender and a receiver to communicate, therefore the guard time is divided by two.

$$DC_{poll} = \frac{t'_{guard}}{2 \cdot T_{poll}} \frac{t_{poll}}{T_{cp}} = \frac{2 \cdot r_{skew} \cdot t_{poll}}{T_{poll}} \quad (11)$$

We can also calculate an optimal polling period T''_{poll} for the S-LPL using its active duty cycle:

$$\frac{\delta DC_{S-LPL}}{\delta T_{poll}} = \frac{1}{T_{cp}} + \frac{1}{2T_{cp}} - \frac{2 \cdot r_{skew} \cdot t_{poll}}{T_{poll}^2} = 0$$

$$T''_{poll} = \sqrt{\frac{4}{3} \cdot T_{cp} \cdot r_{skew} \cdot t_{poll}} \quad (12)$$

Since the optimal polling period T''_{poll} cannot be smaller than the physical time required by a radio to poll the channel, the following inequality must hold:

$$T''_{poll} > t_{poll}$$

$$\frac{4}{3} \cdot T_{cp} \cdot r_{skew} \cdot t_{poll} > t_{poll}^2$$

$$T_{cp} > \frac{3}{4} \cdot \frac{t_{poll}}{r_{skew}}$$

Assuming $t_{poll} = 3$ ms and $r_{skew} = 50$ ppm, the communication period must exceed $T_{cp} > 45$ s which satisfies our initial assumptions on handling low data rates.

5. Network Level Scheduling

S-LPL enables very low duty cycles because it uses information about expected transmission times and does not use radio unnecessarily. However, the low duty cycle of S-LPL comes with a price. The protocol cannot handle unanticipated traffic. In data gathering networks, this type of traffic is only present when nodes join the network or change routes. Therefore, we use a scheme which defines two modes of network operation - maintenance and scheduled. In maintenance mode, nodes use asynchronous LPL. This mode is used to initialize and update the network topology. In scheduled mode, nodes are synchronized and use S-LPL. Bailigh scheduled operation is intermittent with the time between consecutive data downloads determined by collection period. Bailigh schedule uses S-LPL, hence the data collection period corresponds to the communication period T_{cp} of S-LPL. All nodes in the network wake up instantly every T_{cp} seconds. The nodes then calculate a time offset used to handle communication at a given level. After all packets from a node and its children have been sent to its parent, the node sets a wakeup timer and goes into a sleep state for T_{cp} seconds.

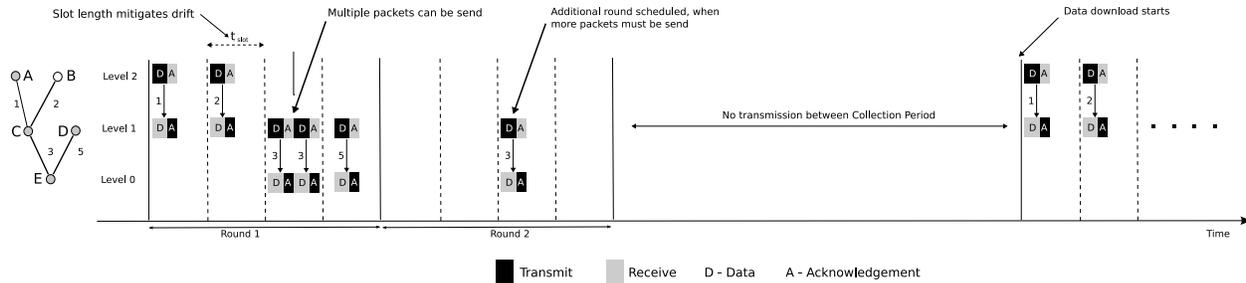


Figure 2. Staggered communication scheme.

5.1. Maintenance Mode

After booting, a node enters maintenance mode and enables LPL. A data gathering tree is created by disseminating beacons in the network. Only nodes connected to tree are allowed to disseminate beacons, and initially it is only the sink. Each beacon contains information about sender's hop count (distance from the sink), current tree depth, and time left to the next base period. For dissemination we use an approach based on Trickle [17]. Trickle is a gossiping dissemination protocol, originally developed for code dissemination. We use the Trickle gossiping mechanism to ensure that all nodes in the network learn depth of data gathering tree. This is required to calculate the timing offset in staggered schedule. In general, a node broadcast beacons at a variable rate. The rate is small when a node keeps receiving beacons with the same tree depth as the the one in its records. When a node receives a beacon with tree depth value greater then its own, the rate is set to a high value. Over a time the rate of dissemination will drop to the low value. This mechanisms ensures that any changes in the tree depth are announced in the network quickly, whereas in consistent state (i.e. all nodes sharing the same tree depth value) number of broadcasted beacons stays low.

Upon receiving a beacon, in addition to beacon details, a node stores information about packet timestamp and signal strength. An unconnected node after receiving first beacon waits for a random time before attempting to join the network. This allows to gather information about node's neighborhood. Details of each discovered neighbor are stored in a neighbor table. After this time the node selects a node with the highest signal strength and the lowest hop count as a parent and sends a join request to the selected node. The node which was selected as a parent might accept or reject the request, depending on the number of available children slots. The number of children slots is fixed and common for all nodes in the network. The node sends a join reply to inform the requesting

node about its decision. If the join request is accepted, the node aligns its clock to the selected parent and starts beacon dissemination. If not, the node selects the next node from the neighbor list and repeats the procedure. A node which cannot connect to a parent will stay in the maintenance mode.

Once the setup time is elapsed (nodes disseminate the remaining time in beacons), nodes enter the scheduled mode. The maintenance mode is enabled periodically to refresh the network topology or disseminate commands in the network.

5.2. Scheduled Mode

The operation of Bailigh scheduled mode is illustrated in Figure 2. All nodes in the network wake up at the same time. This moment starts a round. A round is the time required to send a packet from the furthestmost leaf node to the sink. Each node calculates its communication offset taking into account the maximal tree depth and its its distance from the sink. A round consists of frames. A frame is the period of time where adjacent levels communicate. In Figure 2 nodes A and B are in the same frame and send their data to parent C. A frame consists of slots. Each child communicating with the same parent has an individual slot. The length of slot is directly related to the guard time calculated by S-LPL. The slot is long enough to allow for clock drift which developed over the previous communication period and send fixed number of data packets. Each packet is acknowledged. In the example, in each slot two data packets can be send. The TDMA schedule created in the maintenance mode is not entirely collision free, because collisions might happen due to hidden nodes. Each packet transmission is preceded by a small contention window. Nodes use CSMA to detect channel activity and avoid collision. In that case a node will backoff for a packet duration and retransmit the packet. All transmission are stopped before end of the slot. Each parent keeps track of slots

Table 1. Transceiver characteristics [18], [19]

Parameter	CC1000	CC2420	Unit
P_{tx}	31.2	52.2	mW
P_{rx}	22.2	56.4	mW
P_{sleep}	3	3	μ W
P_{listen}	22.2	56.4	mW
P_{poll}	7.4	12.3	mW
t_{poll}	3	2.5	ms
t_{cca}	7	2	ms
Data Rate	19.2	250	kbps

assigned to its children. In case a slot is not assigned, the node will switch off its radio.

Each node has a queue which stores incoming packets. Due to the limited memory of nodes, these queues are limited in length. Therefore a parent might not be able to accept all packets sent by its children. In this situation more than one round is necessary to transfer the data to the sink. For example, node C in Figure 2 must send three packets: two generated by its children and one its own. Since only two packets can be send in a slot and additional round is required. This is achieved by including a number of data packets waiting in queue in each transmitted packet. Parent will schedule additional rounds until all children empty their queues. Additional round will be also scheduled, when a child node did not send all packet before slot end. Nodes which do not participate in communication (e.g. node D in Figure 2) enter sleep state.

Any type of scheduling requires time synchronization. In Bailigh a child node synchronizes its clock with its parent’s clock. This is done by adjusting clock using a timestamp from the acknowledgment packet. The timestamps are inserted by S-LPL before sending packet. The maximal difference in time between two consecutive levels is equal to the time drift since the last synchronization. The maximal time difference between the sink and the furthest leaf node is equal to $k \cdot t_{guard}$, where k is the tree depth. A node which loses synchronization (i.e. did not received any acknowledgment) will switch to the maintenance mode and wait for beacons to regain synchronization.

6. Analytical Results

The analytical analysis compares one-hop performance of S-LPL and LPL. The calculations are based on equations given in Section 4. Figure 3 compares the duty cycle of S-LPL and LPL as function of communication period. For each communication period the active duty cycle was calculated using an optimal polling period. On average, S-LPL has a duty cycle two orders in magnitude lower than LPL. It is also insensitive to

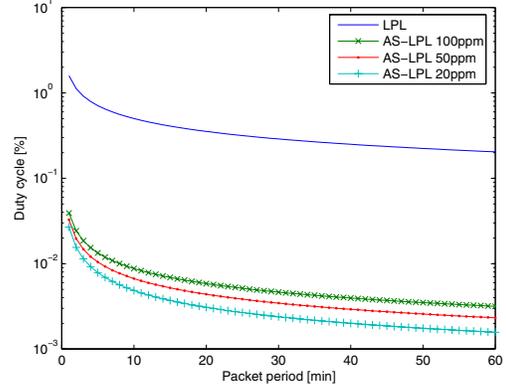


Figure 3. Duty cycle as a function of communication period.

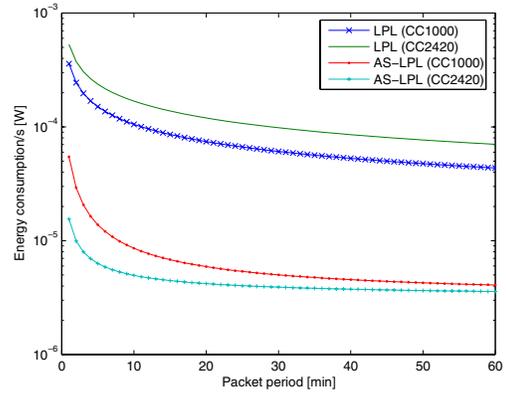


Figure 4. Energy consumption per second as a function of communication period.

different values of crystal stability. A change of 5x in crystal stability doubles the duty cycle. Figure 4 compares the energy consumption of S-LPL and LPL. For each communication period energy consumption was calculated using transceiver parameters given in Table 1. As in case of duty cycle S-LPL outperforms LPL by two orders in magnitude.

Table 2. Simulation parameters

Symbol	Description	Value
T_{sim}	Simulation time	1h
T_{setup}	Maintenance phase time	60 s
r_{skew}	Frequency skew	100 ppm
T_{poll}	Fixed polling period	100 ms (LPL) 10 ms (S-LPL)

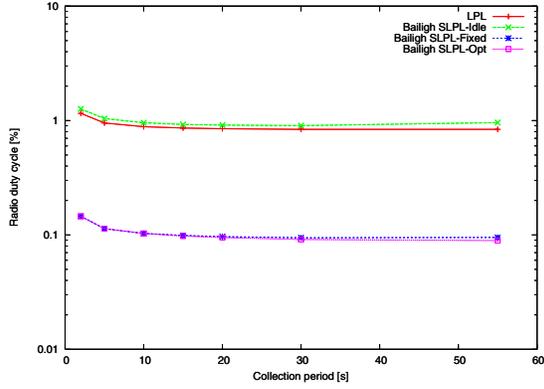


Figure 5. Average duty cycle as function of collection period.

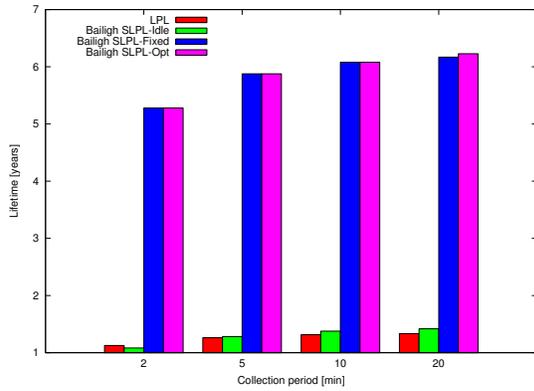


Figure 6. Lifetime as function of collection period.

7. Simulation Results

Multihop performance evaluation of Bailigh was done by simulation in the Castalia [20] framework for OMNeT++ [21]. In the evaluation Bailigh is compared with a non cross-layer scheme i.e. a tree routing with LPL. Nodes were uniformly deployed in a 50×50 field. Results are calculated from 10 independent simulation runs with parameters given in Table 2.

As in case of S-LPL the main metric of protocol efficiency is the duty cycle. Figure 5 shows the average duty cycle for a network of 50 nodes. Bailigh was tested in three configurations: optimal, fixed, and idle. The optimal configuration calculates polling period according to equation 12. In fixed configuration the polling period is fixed to 10ms. Idle configuration does not use S-LPL in guard time. Both fixed and optimal configuration, on average, achieve 8.7x lower duty cycle than LPL for all collection periods. Calculation of optimal polling period over fixed one is beneficial for very long collection periods. Idle configuration

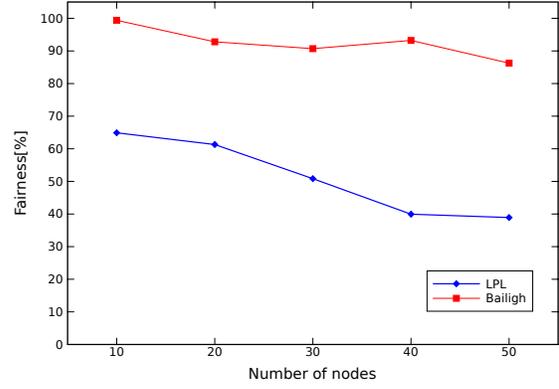


Figure 7. Average fairness as function of network size.

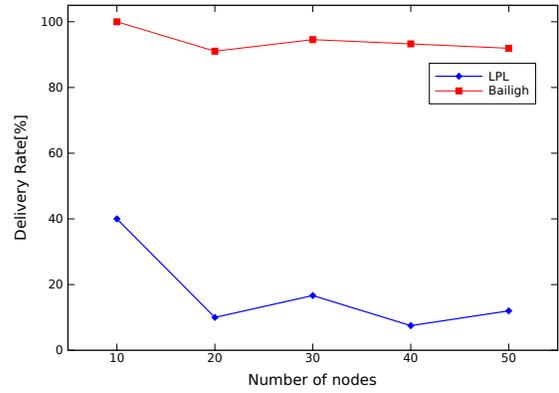


Figure 8. Delivery rate as function of network size.

performs worse than LPL, hence use of S-LPL during guard times is crucial for achieving very low duty cycle. In terms of lifetime (Figure 6) which is directly related to duty cycle Bailigh shows 4.6x improvement. Depending on collection period, network using Bailigh is able to operate for 5.8 years at 5 minutes collection period up to 6.2 years with 20 minutes collection period.

To assess impact of Bailigh scheduling we measured fairness and delivery rate. We define fairness as the percentage of packet transmissions which were successful on the first approach. Fairness allows us to measure the number of collisions detected and avoided by a node. Figure 7 shows the fairness of both protocols in function of network size. In all simulated cases, fairness of Bailigh is greater than 85%. The highest fairness of LPL is 64% for network of 10 nodes and drops to 35% for network of 50 nodes. Delivery rate (Figure 8) measures the percent of unicast packets successfully delivered i.e., acknowledged by receiver. A packet is lost when three consecutive transmission

retries were not successful. Bailigh has an average delivery rate of 95%, whereas the average delivery rate of LPL is 17%. High fairness and delivery rate are due to scheduling individual transmission slots. Thanks to this collisions are reduced, although some remain due to hidden nodes at different branches of the tree (hidden terminal problem). In LPL, the drop of fairness for higher network density is mainly due to the long preamble which is necessary to wake up a receiver. Since the channel polling of LPL was adjusted for maximal energy efficiency, the long preamble occupies the channel for considerable amounts of time preventing other nodes from accessing the channel. S-LPL uses very short preambles, therefore collisions caused by hidden nodes have less input.

8. Conclusions

Data gathering is one of the driving applications for Wireless Sensor Networks. Developing efficient communication protocols is an important task, as the available energy is a very limited resource and radio communication exploits this resource most. In this work we presented Bailigh, a low power cross-layer data gathering protocol designed for energy efficiency in low data collection rate applications. Bailigh integrates novel a channel access method, Synchronous Low Power Listening, which enables very low duty cycles and efficiently mitigates clock drift. Operating on top of S-LPL we propose an application driven staggered scheduling, which limits collisions and increases delivery rates. To investigate the benefits of our approach we performed a number of simulations and compared Bailigh to a non cross-layer data gathering approach. Results show that Bailigh has an average duty cycle of 0.1% and delivery rate of 95%. Using Bailigh a network of 50 off-the-shelf nodes is able to operate for 5.8 years.

References

- [1] G. Tolle, J. Polastre, R. Szewczyk *et al.*, "A macrocope in the redwoods," in *Proc. of ACM SenSys*, 2005, pp. 51–63.
- [2] R. Szewczyk, A. Mainwaring, J. Polastre *et al.*, "An analysis of a large scale habitat monitoring application," in *Proc. of ACM SenSys*, 2004, pp. 214–226.
- [3] R. Musaloiu-E, A. Terzis, K. Szlavecz *et al.*, "Life under your feet: A wireless soil ecology sensor network," in *Proc. 3rd Workshop on Embedded Networked Sensors (EmNets 2006)*, May, 2006.
- [4] R. Jurdak, A. Ruzzelli, and G. O'Hare, "Adaptive radio modes in sensor networks: How deep to sleep?" in *Proc. of IEEE SECON*, June 2008, pp. 386–394.
- [5] P. Dutta, D. Culler, and S. Shenker, "Procrastination might lead to a longer and more useful life," in *Proceedings of HotNets-VI*, Atlanta, GA, November, 2007.
- [6] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, 2004.
- [7] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proc. of ACM SenSys*, 2003, pp. 171–180.
- [8] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. of ACM SenSys*, 2004, pp. 95–107.
- [9] A. El-Hoiydi and J.-D. Decotignie, "WiseMAC: an ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks," in *Proc. of ISCC*, 2004, pp. 244–251.
- [10] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle mac with scheduled channel polling," in *Proc. of ACM SenSys*, New York, NY, USA, 2006, pp. 321–334.
- [11] M. Buettner, G. V. Yee, E. Anderson *et al.*, "X-MAC: a short preamble mac protocol for duty-cycled wireless sensor networks," in *Proc. of ACM SenSys*, 2006, pp. 307–320.
- [12] B. Hohlt, L. Doherty, and E. Brewer, "Flexible power scheduling for sensor networks," in *Proc. of IEEE IPSN*, 2004, pp. 205–214.
- [13] N. Burri, P. von Rickenbach, and R. Wattenhofer, "Dozer: ultra-low power data gathering in sensor networks," in *Proc. of IEEE IPSN*, 2007, pp. 450–459.
- [14] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks," *Parallel and Distributed Processing Symposium, International*, vol. 13, p. 224a, 2004.
- [15] M. Sichitiu, "Cross-layer scheduling for power efficiency in wireless sensor networks," *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1740–1750 vol.3, March 2004.
- [16] A. G. Ruzzelli, G. M. P. O'Hare, and R. Jurdak, "Merlin: Cross-layer integration of mac and routing for low duty-cycle sensor networks," *Ad Hoc Netw.*, vol. 6, no. 8, pp. 1238–1257, 2008.
- [17] P. Levis, E. Brewer, D. Culler *et al.*, "The emergence of a networking primitive in wireless sensor networks," *Commun. ACM*, vol. 51, no. 7, pp. 99–106, 2008.
- [18] CC1000 Data Sheet. Texas Instruments.
- [19] CC2420 Data Sheet. Texas Instruments.
- [20] Castalia. National ICT Australia Limited.
- [21] A. Varga. Omnet++.