Ripple-2: A Non-Collaborative, Asynchronous, and Open Architecture for Highly-Scalable and Low Duty-Cycle WSNs

Agnelo R. Silva
Electrical Eng. - Electrophysics
University of Southern California
Los Angeles, CA
agnelosi@usc.edu

Mingyan Liu
EECS Department
University of Michigan
Ann Arbor, MI
mingyan@eecs.umich.edu

Mahta Moghaddam
Electrical Eng. - Electrophysics
University of Southern California
Los Angeles, CA
mahta@usc.edu

Abstract

[v.010] The design of Ripple-2, a wireless in-situ soil moisture sensing system is presented in this paper. The main objective of such system is to collect high fidelity and fine grained data both spatially and temporally compared to radar remote sensing, which is the more traditional way of capturing soil moisture, and to use the former to validate and calibrate the latter. To do so, the in-site sensor network must cover a sufficiently large area, on the order of at least a few square kilometers. At the same time, cost constraints (both in deployment and in maintenance) puts a limit on the total number of sensor nodes, resulting in a very sparse (on average) network. The main challenge in designing the system lies in achieving reliability and energy efficiency in such a *sparse* network. For instance, in our pilot deployment, a 200mx400m area is covered by 22 nodes (average inter-node distance ¿50m). Traditional WSN technology typically calls for many more nodes to be deployed in such an area. Ripple-2 is introduced as a non-traditional WSN architecture where (1) the network is physically and logically segmented into isolated clusters, (2) a regular node (or end device, ED) only communicates with the cluster head (CH) of its segment, and (3) the ED-CH communication is distinct from the CH-sink (or CH-Data Server) and both links can use virtually any kind of point-to-point wireless technology. We use both simulated and empirical results to demonstrate the effectiveness of Ripple-2; it proves to be ideal for low duty-cycle data collection applications due to its exceptional small network overhead (typically smaller than 1%) and its robustness to the size of the network.

Categories and Subject Descriptors

C.3 [**Special-purpose and application-based systems**]: Real-time and embedded systems

General Terms

Design, Experimentation, Performance

ACM MiSeNet'12, August 26, 2012, Istanbul, Turkey.

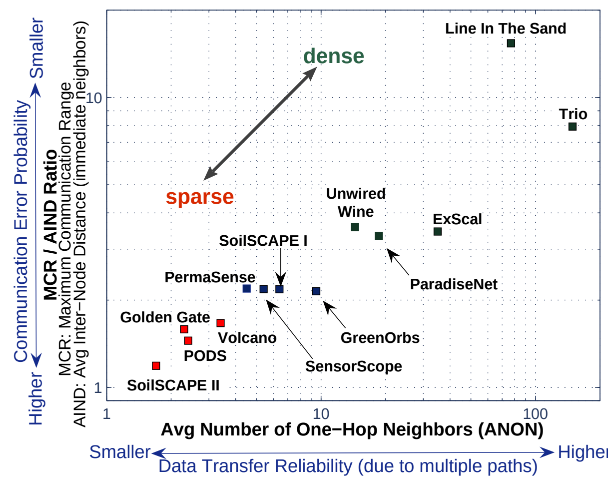

Figure 1. AIND-ANON metrics (sparsity degree of a network) applied to 13 large WSNs.

1 Introduction

The design and implementation of Ripple-2, a novel Wireless Sensor Network (WSN) architecture for in-situ soil moisture sensing systems, is presented in this paper. The main goal of the system is to collect high fidelity and fine grained data in order to validate and calibrate a radar remote sensing system. Due to the large areas involved (few square kilometers or more), the cost constraints puts a limit on the total number of nodes, resulting in a very sparse network. The first version of our system, Ripple-1, was based on traditional ZigBee technology [1, 2] and we faced scalability problems associated with large and sparse deployments. In such scenarios, the network performance is strongly impacted, as shown in Fig.1. Large physical areas are also associated with higher costs due to the frequent need of exchanging batteries, in particular if the network overhead increases (e.g., $> 5\%$) and the energy consumption among nodes is not balanced among the nodes. To address such problems, we envisioned the design of a WSN specifically tailored to the mentioned soil application but with superior scalability and energy-efficiency characteristics.

Under this design, a regular sensor node acts *selfishly* in that no message relaying is performed in order to minimize its energy consumption. This is a very different design concept from that commonly used in ad hoc networks as well as many wireless sensor networks, in which *cooperation* and *re-*

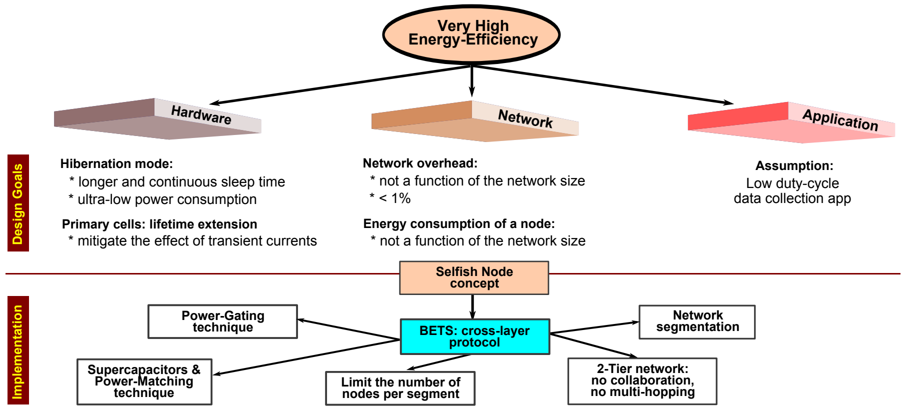

Figure 2. Ripple-2 architecture: selfish node, network segmentation, BETS protocol, and power-gating/matching.

laying are used and/or required. To avoid message relaying, the network is segmented and each segment has a star-like topology with a maximum number of nodes. At the center of the star, there is a cluster head (CH) node. This scheme calls for an efficient time synchronization solution and also for a provision to make the CH sleeps as much time as possible. To this end, we developed the Best-Effort Time Slot Allocation (BETS) protocol. In addition, hardware techniques called Power-gating and Power-matching are employed to achieve maximum energy-efficiency, as shown in Fig.2.

Because there is no communication between segments, the solution in one segment can be replicated many times without impacting the network. Therefore, Ripple-2 is a very scalable architecture and based on simulated and empirical results, it also has a very small network overhead (typically smaller than 1%) for a network segment with 30 regular nodes and assuming 20-min measurement cycles. Finally, Ripple-2 is open to support different wireless technologies, not only the ones associated to WSNs. Besides its expected use in soil moisture measurement applications, Ripple-2 can be potentially applied to other low data-rate environmental monitoring applications. The tradeoffs of the architecture are higher data latency and lack of mobility support.

The rest of this paper is organized as follows: In Section 2, the main concepts behind Ripple-2 are discussed. In Section 3, the power managements techniques used in Ripple-2 are presented. In Section 4, the core component of Ripple-2, the BETS protocol, is discussed. The paper is concluded with the simulated and empirical results of the overall system in Section 5.

2 Motivation and Main Concepts

The motivation for the Ripple-2 design, the importance of the AIND-ANON metrics (detailed below), and the concepts of selfish-node, network segmentation, power-gating, and power-matching are presented in this section.

2.1 Motivation

While analyzing the real requirements behind in-situ soil moisture measurement applications, it becomes clear that certain features of WSNs, as non-planned (ad-hoc) deployments, collaboration among nodes, multiple data-paths, mobility support, etc. are not strictly necessary for our solution. Therefore, we decided to remove such features in order to achieve an extremely light WSN solution in terms of energy-efficiency which is also invariant to the network size, as shown in Fig.2. The pilot site of our project involves a 200m x 400m area covered by 22 nodes [1]. The second deployment (Summer 2012) involves a bigger area (3km x 3km) that must be covered with 150 nodes, each one with 3 soil moisture sensors. Therefore, the solution must scale without increasing the density of nodes. If traditional WSN technology is used in both cases, the poor network performance will eventually lead to the deployment of a significant number of extra nodes. We faced such problems in our previous implementation based on ZigBee technology (Ripple-1 [1]).

To better highlight the problem that motivated this work, it is important to analyze large WSN deployments, as reported in [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13] and shown in Fig.1. To this end, a metrics called AIND (Average Inter-Node Distance among immediate neighbors) -ANON (Average Number of One-hop Neighbors) is introduced. Given the Maximum Communication Range (MCR) of a node, the ratio MCR/AIND reflects the probability of communication errors due to high inter-node distances. More specifically, the smaller the MCR/AIND value is (close to 1 or even smaller), the higher the probability of errors. The other parameter, ANON, is related to the probability of delivery success due to the existence of multiple data paths. Therefore, the worst scenario for WSNs is the one that involves smaller values for both MCR/AIND and ANON, that is, a sparse network, as shown in Fig.1. The AIND-NON metrics is evaluated for 13 large-scale WSN deployments (number of nodes, coverage, or both). Observe that our previously mentioned pilot deployment (labeled as SoilSCAPE II) has a high probability of problems, as we empirically confirmed. In Section 5, we will see that the performance of a Ripple-2 solution for this scenario is achieved with success even considering an average communication error rate of around 11%.

2.2 *Selfish Node* Concept

The fundamental concept underlying Ripple-2 is a non-collaborative approach based on the notion of selfish nodes, where a node wakes-up, takes measurements, and sends/receives data related solely to *itself*. If we disregard communication channel errors or collisions, such scenario is the ideal one because the network overhead becomes negligible. In contrast, typical WSN protocols have a relatively high overhead (¿3%) even for small networks. Moreover, it is well known that, depending on the topology, some of the nodes in a WSN can potentially have a smaller lifetime due to their role in the multi-hopping scheme. By adopting the selfish concept, all regular nodes have strictly the same energy consumption assuming the same application duty-cycle.

Clearly, the simplest way to implement the selfish node concept is by using time slots (TDMA-like approach). Because all communication inside a segment ends up in the CH node, this node is the one that can provide the proper time-synchronization to avoid channel collision. A pure selfish node implementation does not require any form of channel overhearing mechanism because an ED believes that the network is always *ready* for itself. However, because communication errors can still occur, a basic handshaking mechanism exists during the duration of the time slot allocated to that node. It is possible to implement such concept by using off-the-shelf WSN nodes without removing existing data link protocols (or developing low-level protocols). However, in order to avoid hardware modifications in the WSN module, it is usually easier to use an additional processor/microcontroller (MCU) to run the Ripple-2 software and turn on/off the WSN module by simply considering such device as a point-to-point radio transceiver. In our implementation of an ED (end device) node, we followed this *overlay* approach to allow the solution to be flexible enough to accommodate different wireless technologies according to the scenario. For instance, we already know that inside a dense forest, a 433MHz WSN node will have better performance than our usual choice (IEEE 802.15.4, 2.4GHz module). The only requirement for such off-the-shelf WSN nodes/transceivers is the realization of a point-to-point (ED-CH) communication. The contention and reliability aspects are ultimately provided by a Ripple-2 component called BETS protocol no matter if the underlying WSN node/transceiver provides or not such functionalities.

2.3 Network Segmentation

The implementation of the selfish-node concept relies on the availability of a CH node ready to collect data from the EDs. However, such solution assumes that all EDs of the same segment are in the communication range of the CH node of that segment. This fact clearly imposes a strong limitation on the network planning and it explains why mobility is not supported under Ripple-2. Similarly, if too many nodes belong to the same network segment, there is a possibility that the sum of the individual time slots be higher than the length of the measurements cycle. Therefore, in order to avoid both problems, the network must be divided into logical segments, each one with a maximum number of nodes. The concept of logical segment is stronger than the physical one in the sense that even if a node is in the communication range of 2 or more CHs of different segments, such ED node must be member of a single segment. Such provision is usually achieved by the use of different frequency channels.

2.4 2-Tier network

Once the data from the sensor nodes is collected by the CH in a segment, the next step is to transfer the data from the

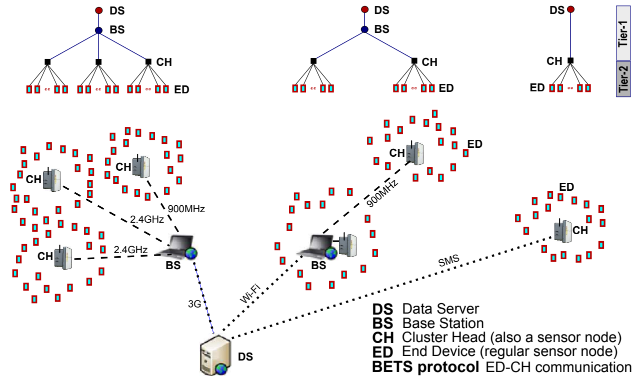

Figure 3. Examples of different topologies in the Ripple-2 architecture already tested in our work.

CH to the Data Server (DS) which controls the application. Two possibilities are envisioned under Ripple-2, both implemented in our experiments, as shown in Fig.3. One option is the direct CH-DS connection using a variety of communication links: Wi-Fi, 2G/3G/4G, SMS, long-distance radio modems, satellite, etc. The second option is to concentrate data from multiple CHs in a single point, the Base Station (BS) node, which relays such data to DS. No matter what scheme is used, CH-DS or CH-BS-DS, such data transfer is not necessarily synchronized with the ED-CH transfer. Although such characteristic of the Ripple-2 architecture allows the network to increase in size without disturbing each segment, it can also increase the data latency. Also, observe that the central component of the architecture, the BETS protocol, deals specifically with the ED-CH communication.

3 Power Management Techniques

3.1 Power-Gating (PG) Technique

The power-gating (PG) technique basically refers to the use of an electronic switch to turn on/off modules such as, the radio transceiver, the sensors, a voltage regulator, etc. However, the main tradeoff of PG is the significant delay imposed by such solution. Low-cost electronic power switches typically require a significant amount of time (hundreds of ms) to turn-on a load. Typically, WSN protocols turn-on/off the radio transceivers multiple times per second. Therefore, it is not a surprise that such technique is not usually used in WSN nodes.

Because Ripple-2 assumes the use of a low application duty-cycle (e.g., $< 1\%$) and no collaboration among ED nodes, it is possible to allow a node to have a very long sleep time (power hibernation), such as in terms of minutes or even hours. In this case, the switching delay of PG is significantly smaller compared to the total sleep time of a node and PG becomes a feasible technique. In some cases, the energy savings with PG are significant, as shown in Fig. 4. In this figure, the expected lifetime of a typical WSN node without any special power management technique is compared with the case where PG (and also power-matching, explained next) is employed for different application duty-cycles. Although the sleep current of individual modules of a WSN node can be very small (tens of μA), the sum of these sleep currents can be still be significant according to the application duty-

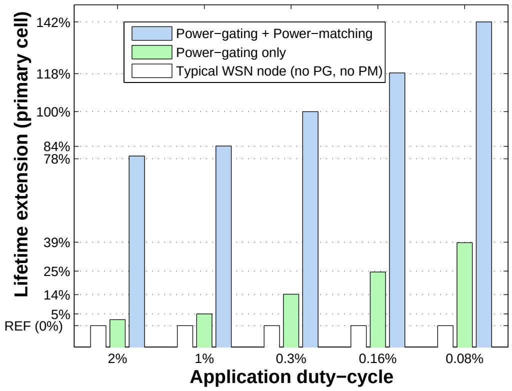

Figure 4. Expected lifetime extension of a primary battery considering the use of the power-gating and power-matching techniques.

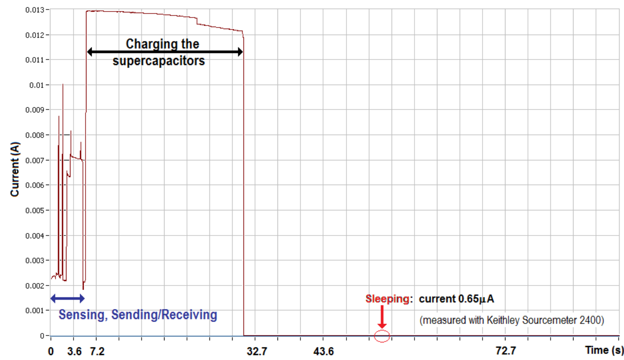

Figure 5. When using the PG and PM techniques, the current drawn from the battery is carefully controlled. In this case (ED node), the current is always below 13mA.

cycle. If such duty-cycle is not small, the savings with PG are negligible. However, as the duty-cycles decreases, the energy spent in inactive mode becomes significant.

3.2 Power-Matching (PM) Technique

Although energy harvesters, such as a solar panel, seem to be a proper choice for WSN nodes, in particular for outdoors, we envisioned a simpler system based on primary cells (non-rechargeable batteries). There are some advantages of this approach, such as the possibility of a more-concealed and mechanically robust solution for the node, no uncertainty related to the power source, and support for extreme weather conditions and temperatures. On the other hand, it has been reported that the use of such cells typically require constant exchange of batteries. In fact, a primary cell only provides its full nominal energy capacity if transient or pulse currents (temporarily high-currents) do not occur. Otherwise, the capacity of the battery is strongly reduced. Empirically, we determined that such reduction is between 60-90% for a typical WSN node due to the high current of the radio. In other words, if the lifetime of a battery for a certain WSN node is calculated as 20 months based on the *average* consumed current, it is possible that such battery only lasts 2 months. In fact, this was the result of one of our initial experiments.

The power-matching (PM) technique is applied to avoid the negative effect of transient current in primary cells. Instead of connecting a battery directly to the load, supercapacitor(s) is(are) slowly charged by a low-current (Fig.5 and quickly discharged driving a high-power load, such as a radio transceiver. As expected, PM can potentially introduce a transmission delay and it is only efficient if not used very frequently because such scheme also has a significant loss. However, such technique can be properly exploited under Ripple-2 because the radio is never used for relaying messages and the application duty-cycle is assumed small. As shown in Fig.4, when PM is used with PG, the realistic lifetime expectation can double for a node that follows a 30-min schedule and is active for only 6s in each cycle (0.3% duty-cycle). For such simulation, we assumed conservative values for the PM losses and our preliminary empirical results show even a better performance (i.e., +20% lifetime extension). Due to these results, we changed our initial approach of using solar panels to a solution which is solely based on primary batteries for EDs. In our current implementation, the expected minimum lifetime for an ED node following a 20-min schedule is 18 months.

4 BETS Protocol

Best-Effort Time Slot Allocation (BETS) protocol is a novel cross-layer protocol implemented as an application-level overlay. BETS operates at the MAC and upper-level networking layers and it also controls the activation of the modules of the node (PG). If an existing MAC protocol is available at the off-the-shelf radio transceiver used for the node, its MAC functionalities can be disabled or ignored if not causing significant overhead. BETS has two basic goals: the implementation of the selfish-node concept at the ED side and energy-efficiency at ED and CH sides. The former goal is implemented by a fair allocation of time slots for each node providing a contention-free and reliable communication channel for the ED-CH link. As shown in Fig.6, BETS assumes that a periodic sensing application is in place and it supports homogeneous or heterogeneous schedules for the nodes. The CH node captures the scheduling data sent by the main application to the nodes and defines a proper allocation of the time slots. Note that the term *schedule* refers to the same entity for both application and network discussions.

The implementation of BETS at the ED side is relatively simple because the ED does not have to control the access to the medium. In its time slot, which is defined with the same fixed value for all EDs, it must try to send its data and receive a new schedule. If it fails, it can repeat the process according to the remaining time in its time slot length. Once the time-slot reaches its end, the message that was not transmitted is lost and no storage/queue provision is supported. This characteristic partially explains the term *Best* in the BETS acronym. A second reason for this term is the sequential order of the time-slot given to a node: there is no guarantee that such order will be maintained. This is specially true if the nodes have different schedules but can also occur in case of communication errors, as explained next.

At the CH side, the BETS implementation is not so simple as in the ED case. According to the second goal of the protocol, it is necessary that CH sleeps as much as possible. When the nodes have different schedules, potential energy inefficiency can occur at the CH node. Even the use

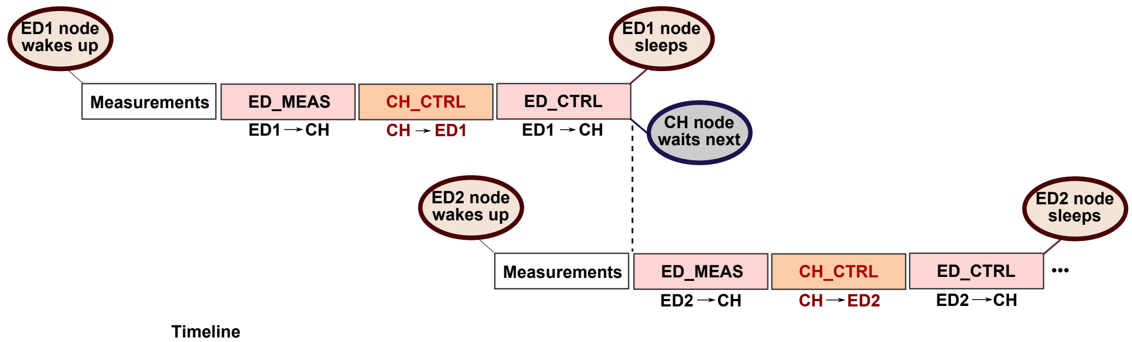

Figure 6. BETS functionality (ED side): an implementation of the *selfish node* concept.

of a single schedule for all sensor nodes does not guarantee the maximum energy performance at the CH side. For instance, some ED nodes can follow the sequence 0-5-10-15.. for a 5-min schedule while other follow the 3-8-13-18.. sequence. In this case, the CH node cannot have a longer sleep although it is clear that it would be possible to achieve this goal. Therefore, BETS provides a mechanism to synchronize all schedules in order to always allow the best possible energy-efficiency also at the CH size and the mentioned problems do not occur. Moreover, in our implementation PG and PM techniques are used in the CH side and its energy profile is very close to the sum of the energy profiles of the ED nodes in that segment. In other words, BETS provides a very deterministic support for the energy management of the system. Besides the measurements, all nodes also send to CH the information about battery level, number of communication errors, and number of initializations (or power shortages). With this data, it is possible to accurately control the lifetime of the network.

In one of our current BETS implementations mentioned at the beginning of this paper, the average ED-CH distance is around 210m (the AIND is 50.5m). Even in such relatively critical case, the total data loss is found to be smaller than 1.8%. In this case, the time slot length still allows a second transmission if the first one fails (not shown in Fig.6. Naturally, the larger the time slot is, the higher is the reliability of the solution and smaller is the energy-efficiency. In order to provide such flexibility for different deployment scenarios, the time slot length can be dynamically defined for each network segment.

Major Time Slot (MTS): once CH is initialized (boot), the time line is divided into fixed periods called MTSs. In our current implementation the duration of an MTS is 5min. Accordingly, the application schedules are given in multiple of this value. During some MTSs, called *inactive MTSs*, the CH node is sleeping considering that all nodes at that segment are also sleeping. The *active MTS* is divided into three parts. The first part is the ED Time-Slot (ETS) and it is used for the communication with ED nodes. Therefore, the length of ETS depends on how many ED nodes are active at that active MTS. In our current implementation, up to 30 ED nodes per segment are supported and if all these nodes are active at a given MTS, the ETS length achieves its maximum value.

The second part of an active MTS is the BS Time-Slot (BTS) and it is used for the CH-BS (or, for some topologies, CH-DS) communication. Instead of sending data to BS every time the CH receives data from EDs, it is possible for the CH node to postpone such transmission and even to realize some form of data aggregation/compression. This aspect is not under control of the BETS protocol, however, the communication with the BS/DS node must never impact the BETS performance of the ED nodes. Finally, the third part of an active MTS is the Sleeping Time Slot (STS) and it refers to the unused time of an active MTS when the CH is inactive and potentially sleeping.

Emergency Mode (EM): in normal operation, all nodes in a segment are synchronized by the CH node. Following their assigned time-slots, the wireless channel is potentially contention-free. However, when the node is a) deployed for the first time, or b) restarted, or c) does not receive the return from the CH node (CH_CTRL message at Fig.6), the network no more is contention-free. In this case, the network is not *convergent* and the problematic node(s) are said to be in emergency state. The message from a synchronized node can potentially collide with a message from a non-synchronized one and the former also enters in emergency state. While in non-convergent state, the network has significant energy penalties. Therefore, an important feature of BETS is to quickly converge a network when it faces problems. This is not a trivial task because an ED node never overhears the channel, even in emergency state. Such provision exists in order to benefit nodes that employ the PM technique. Through the use of random backoff times, the ED nodes avoid the cyclical contention effect. Moreover, because the CH node knows exactly how many nodes are expected for each active MTS and also how many EDs exist, it can extend the time of ETS in order to synchronize the nodes in emergency mode. In our real-world deployments, the network always converges in less than 2 measurement-cycles.

5 Performance Evaluation

We developed a specific network simulator for BETS in Matlab and we considered a single segment with 30 ED nodes. In relation to the communication errors, a uniform distribution is considered. For the power profile, we established the same values for the real implementation of the node: 5mW, 30mW, and 70mW for the active power of processor, sensors, and radio, respectively. Also, 0.01mW, 0.1mW, and 0.1mW for the inactive power of the same modules, respectively. In relation to the timing, the node periodically wakes up, performs some processing (1s), takes measurements (5s), sends/receives data to/from the CH node (3s), performs additional processing (1*s*) and sleeps again. Assuming a low duty-cycle application (i.e., $< 1\%$) and no communication errors, then the overhead of BETS is always $< 0.75\%$ due to the message structure shown in Fig.6. However, if the average communication errors increases, the overhead of BETS is drastically impacted due to the fact that when a node enters in emergency mode it can potentially cause channel contention.

For the empirical evaluation, data from multiple weeks are collected in relation to 2 scenarios/sites called SoilSCAPE I [1] and SoilSCAPE II [2] that are also shown in Fig.1. Although the AIND/ANON metrics can be used to evaluate the network challenges for a certain topology, the Ripple-2 architecture only depends on the communication in relation to the CH node. Accordingly, the first site has an average ED-CH distance of 77m and the second site an average of 240m. The average communication errors are around 0.6% and 11% for the sites SoilSCAPE I and II, respectively. The average effective packet losses are 1.4% and 1.7%.

The simulated and empirical results are shown in Fig.7 where the lifetime of a node is given as a function of the network overhead for different application duty-cycles. As the network overhead increases, the lifetime of the node decreases and this effect is particularly severe for low duty-cycle applications. If a network has an overhead smaller than 5%, the energy savings are significantly high. For an error-free network, such overhead for BETS is smaller than 0.75% making BETS one of the best solutions possible for this scenario. However, as the average communication error increases, the overhead due to BETS significantly increases. Using the empirical data for 20min-scheduling, we can observe that SoilSCAPE I has an expected lifetime only 3.4% below the ideal one for BETS. For the best of our knowledge, no current WSN solution provides such energy performance with the such level of reliability. On the other hand, for SoilSCAPE II, the average communication error of that site makes BETS to have an overhead of almost 2.2%. In other words, if a WSN solution has a smaller overhead while maintaining the same topology, number of nodes, and packet loss rate, so such solution is better than BETS. Again, there is no reference in the WSN literature about a network with such degree of sparsity operating with a small network overhead as demonstrated by the case studies in Fig.1.

An impressive but potentially hidden advantage of Ripple-2 is the balanced energy consumption among the regular nodes (EDs). In outdoors, this fact means a significant reduction on the costs to support the WSN. As a future work, we intend to customize a WSN node with the BETS functionality without the significant latency due to the overlay technique. Once reducing such latency, the energy savings in active mode are significantly enhanced. Moreover, with larger time slots, it is possible for the node to re-transmit its information more times if necessary. Alternatively, we can extend the current limit of 30 ED nodes per segment.

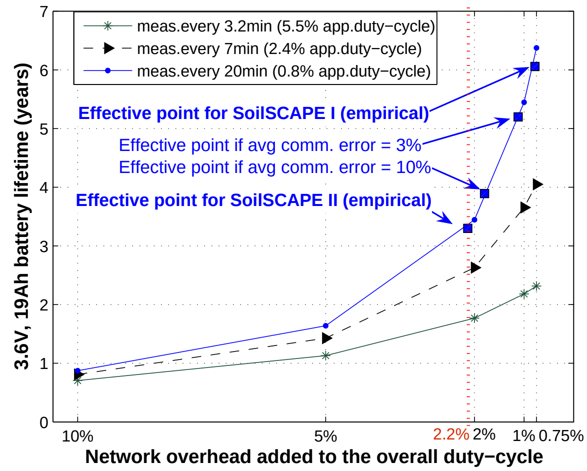

Figure 7. Expected lifetime of a WSN node considering different application duty-cycles and the network overheads. The network overhead due to BETS depends on the probability of communication errors of the network.

6 References

[1] M. Moghaddam, D. Entekhabi, Y. Goykhman, K. Li, M. Liu, A. Mahajan, A. Nayyar, D. Shuman, and D. Teneketzis, "A wireless soil moisture smart sensor web using physics-based optimal control: Concept and initial demonstrations," *IEEE Journal od Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 3, pp. 522–535, December 2010.

[2] M. M. et al., "Ground network design and dynamic operation for validation of spaceborne soil moisture measurements: Initial developments and results," in *In Proc. ESTF-2010*, 2010.

[3] E. S. Biagioni and K. W. Bridges, "The application of remote sensor technology to assist the recovery of rare and endangered species," *International Journal of High Performance Computing Applications*, vol. Vol 16, no. 3, 2002.

[4] R. Beckwith and et al., "Unwired wine: Sensor networks in vineyards," in *In Proc. of IEEE Sensors*, pp. 561–564, 2004.

[5] A. Arora and et al., "A Line in the Sand: A wireless sensor network for target detection, classification, and tracking," *Computer Networks*, vol. 46, no. 5, pp. 605–634, 2004.

[6] G. Werner-Allen and et al., "Fidelity and yield in a volcano monitoring sensor network," in *In Proc. of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 381–396, 2006.

[7] P. Dutta and et al., "Trio enabling sustainable and scalable outdoor wireless sensor network deployments," in *In Proc. IEEE IPSN' 06*, pp. 407–415, 2006.

[8] A. Arora and et al, "ExScal: Elements of an extreme scale wireless sensor network," in *In Proc. IEEE RTCSA*, pp. 102–108, 2005.

[9] S. K. et al., "Health monitoring of civil infrastructures using wireless sensor networks," in *In Proc. IEEE IPSN*, pp. 254–263, 2007.

[10] G. B. et al., "Sensorscope: Out-of-the-box environmental monitoring," in *In Proc. IEEE IPSN*, pp. 332–343, 2008.

[11] J. Beutel and et al., "PermaDAQ: A scientific instrument for precision sensing and data recovery in environmental extremes," in *In Proc IEEE IPSN*, pp. 265–276, 2009.

[12] A. Nasipuri and et al., "Wireless sensor network for substation monitoring: Design and deployment," in *In Proc. SenSys, demo session*, pp. 365–366, 2008.

[13] Y. Liu and et al., "Does Wireless Sensor Network Scale? A Measurement Study on GreenOrbs," in *In Proc. IEEE INFOCOM'11*, (Shanghai, China), 2011.