Distributed Networking in Autonomic Solar Powered Wireless Sensor Networks

Shusen Yang*, Xinyu Yang†, Julie A. McCann*, Tong Zhang†, Guozheng Liu†, and Zheng Liu†

*Department of Computing, Imperial College, London, UK,
†Department of Computer Science and Technology, Xi'an Jiaotong University, China,

***Abstract*—Recent advances in solar harvesting technologies pave the way for sustainable environmental-monitoring applications in the emerging solar powered wireless sensor networks (SP-WSNs). The complexities associated with the low-resourced, high-dynamic, and vulnerable sensor nodes operating in potentially unattended or hostile environments require a high degree of self-management and automation. Guided by autonomic communication principles, this paper presents AutoSP-WSN, a novel distributed framework to achieve sustainable data collection while also optimizing end-to-end network performance for SP-WSNs. Initially, we present the energy-aware support component that provides reliable energy monitoring and prediction. This drives the power management component, which is adaptive to time-varying solar power, avoiding battery exhaustion as well as maximizing the per-node utility. Finally, to demonstrate the key design issues of the network protocol component, we propose two self-adaptive network protocols, a routing protocol SP-BCP and a rate control scheme PEA-DLEX. We show that the individual components seamlessly integrated as a whole, and the AutoSP-WSN framework exhibits the properties of context-awareness, distributed operation, self-configuration, self-optimization, self-protection and self-healing. Through extensive experiments on a real SP-WSN platform, and hardware-driven simulations, we show that the proposed schemes achieve substantial improvements over previous work, in terms of reliability, sustainable operation, and network utility.**

I. Introduction

Environmental monitoring is one of the most important applications in Wireless Sensor Networks (WSNs) [1]. Battery-powered sensor nodes have to be manually replaced when they are depleted. However, in many outdoor deployments, it is difficult or even impossible to physically access the sensor nodes. As a result, the bounded lifetime of WSNs has become a restriction impacting their use in such applications.

Recently, the development of photovoltaic harvesting techniques (e.g. [2]), are demonstrating that solar power is indeed a viable first step towards autonomous WSNs [3]. However, due to the limited sizes of micro solar panels, harvested solar energy remains scarce [4]. To make best use of this resource one needs to know how the node uses power while understanding the dynamics of energy generation. However, hardware power usage is complex and the nature of solar power is highly dynamic. Further complications come about because of the heterogeneous spatial harvesting capabilities across different nodes in a sensing space due to shading or cloud coverage as shown in Figure 1. These have significant impacts on the design of reliable multi-hop solar-powered WSNs (SP-WSNs). For instance, a routing protocol should adaptively selects and dynamically adjusts end-to-end path to avoid the time-varying routing hot-spots (i.e. nodes with low harvesting power). This work therefore aims to provide sustainable data collection capability, while also maximizing end-to-end network performance (e.g. goodput) for SP-WSNs.

Fortunately, the emerging autonomic computing and communication paradigms [5], [6], can help to address the above issues in the design of SP-WSNs: Firstly, sensor node's internal power system should be realistically modelled (e.g. solar power and battery recharging/discharing process) to provide reliable *energy awareness*. Secondly, the capacity of self-management for each individual node is required to ensure sustainable operation while optimizing its long-term solar power usage (this is its *power management* function). Thirdly, through adapting to the time-varying and heterogeneous distributed solar harvesting opportunities, distributed and adaptive *network protocols* should be derived for SP-WSNs to achieve autonomous multi-hop data collection.

The current studies on energy harvesting WSNs, per-node power management schemes [4], [7], [8] and network-wide protocols [9]–[13], aim to address the above issues separately. Furthermore, most of them are theoretical work, and therefore are likely to perform poorly or even fail in real-world SP-WSNs. In this paper, we incorporate autonomic communication principles into the design of SP-WSNs, and present a systematical study covering both individual components (i.e. energy awareness, power management, and network protocols) *and* the SP-WSN as a whole. The contributions of this paper are summarized as follows:

1. Guided by autonomic communication principles, we develop and implement AutoSP-WSN, to our knowledge, the first distributed framework for practical SP-WSNs. The overarching goal of AutoSP-WSN is to optimize the usage of solar power as well as the end-to-end network performance, and more importantly, to achieve Energy Neutral Operation (ENO) [7], i.e. to guarantee that no node will run out of energy at any point in real-world deployments.

2. A set of self-managing functions and adaptive algorithms are developed for AutoSP-WSN, including Energy Awareness (EA) support, solar prediction, power management, and end-to-end network protocols.

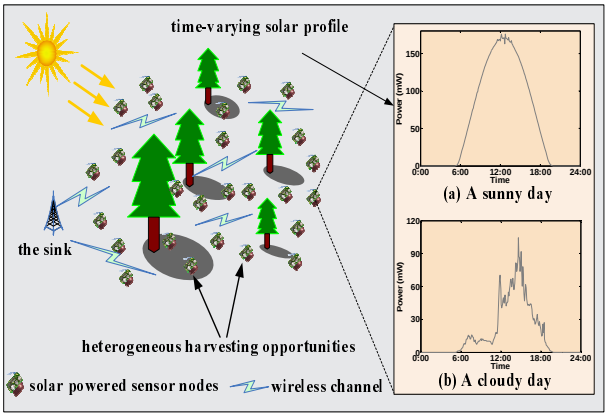


Fig. 1. Illustration of a typical multi-hop SP-WSN.

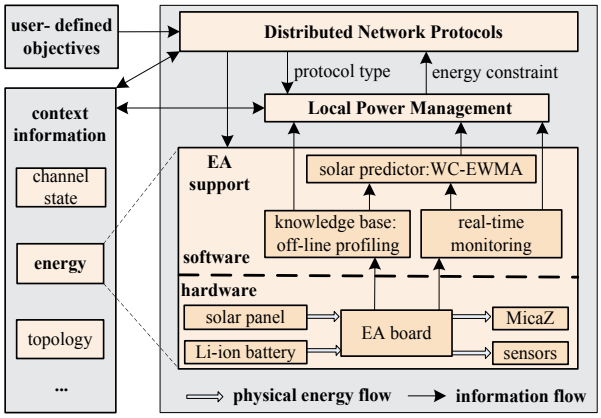


Fig. 2. AutoSP-WSN architecture.

3. Through extensive evaluations on both a real SP-WSN platform and the Tossim simulator [14], we show that both the proposed individual schemes and the whole AutoSP-WSN achieve substantial improvements over previous approaches.

The rest of paper is organized as follows. The next section presents the overall AutoSP-WSN architecture. Section III presents EA support component. The energy prediction algorithm, WC-EWMA, is described in Section IV. Section V presents the details of the LPM component. Two solar-aware network protocols are proposed in Section VI. Evaluations are presented in Section VII. Section VIII describe the related work, and finally, Section IX concludes this paper.

II. System Overview

A. AutoSP-WSN Architecture

Figure 2 illustrates our AutoSP-WSN framework. The system consists of three functional components:

The EA support component provides realistic hardware-driven power models and reliable off-line parameter estimations (e.g. energy translation efficiency). Besides, EA component also provides precise online measurements for power parameters (e.g. real-time solar power). Furthermore, a lightweight algorithm, WC-EWMA, are developed to forecast the future solar profile. These self-monitoring and forecasting functionalities offer solid EA supports for AutoSP-WSNs.

The LPM component computes the energy consumption constraint for each individual node to achieve ENO and optimal long-term solar power usage, based on the power information provided by the EA support component. Since the complex hardware details are encapsulated by LPM, various network protocols can simply use this energy consumption constraint for their decision makings. Besides, LPM can self-configure its parameters to provide short-term or long-term energy constraints for different types of network protocols.

The network protocol component consists of several communication protocols such as Medium Access Control (MAC), routing, and sensing rate control, to enable different nodes in a multi-hop SP-WSN cooperatively carry out data collection tasks in a distributed manner. Besides energy constraint provided by LPM, network protocols should be aware and adaptive to other environment context such as channel condition and topologies. In this paper, we present a routing protocol SP-BCP, and a rate control protocol PEA-DLEX to demonstrate how autonomic communication principles aid the designing of network protocols in SP-WSNs.

B. Time Horizon and Protocol Types

Due to the dynamic nature of solar power, modelling time is critical to AutoSP-WSN. Continuous time is divided into discrete durations as shown in Figure 3. We define a *slot* (of several minutes, dependent on weather and sensor nodes' surroundings) as the atomic duration in which solar power can be considered to be static. Our WC-EWMA algorithm predicts solar power within a future *prediction interval* (explain later), based on the historical solar profile of previous prediction intervals, over the current day and multiple previous days.

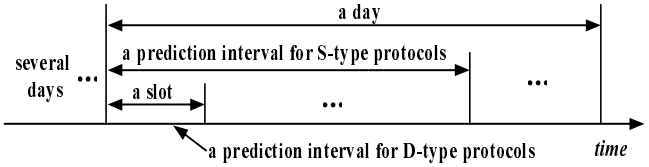


Fig. 3. Time horizon and duration terms.

A prediction interval can consist of either one slot or multiple slots, depending on the types of network protocols: 1. *Static type (S-type) protocols.* This kind of protocols such as rate control algorithms DLEX [10] and Quickfix [15], require relatively long-term static local resource constraints, since they have relatively heavy control overheads (e.g. the transmission of end-to-end control information). 2. *Dynamic type (D-type) protocols* such as routing schemes CTP [16] or BCP [17] often have the light control overheads (e.g. one-hop beacons), and therefore they can quickly adapt to network resource change. Hence, we define *a prediction interval consists of one slot for D-type protocols and multiple slots for S-type protocols.*

C. Characteristics of Autonomic Communication

Besides context awareness, AutoSP-WSN exhibits the following self-management behaviors:

1. *self-optimization*. A key objective of AutoSP-WSN is to optimize the usage of solar power. Specifically, LPM optimizes

the per node utility over time, while network protocols aim to achieve a near optimal end-to-end network performance (e.g. network goodput and fairness).

2. *self-protection and self-healing*. A foundation aim of AutoSP-WSN is to achieve ENO. Several schemes are also proposed to detect and recover errors of solar power prediction, resulting in a minimal risk of network failure.

3. *self-configuration*. LPM can automatically tune its parameter for different types of network protocols.

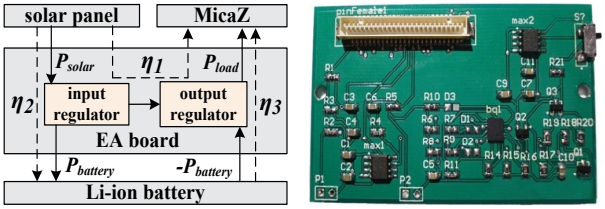
III. Energy-Awareness Support

This section briefly discusses the EA support component, including the system hardware, off-line knowledgebase establishment, and online measurement. Solar power prediction is also an important function of the EA support component, which will be discussed in detail in Section IV.

A. Brief Description of the Hardware

The objective of the hardware design for AutoSP-WSN is to achieve low cost, high efficiency, and more importantly, energy awareness support. It is worth noting that AutoSP-WSN does not rely on any specific hardware. However, in order to evaluate real-world performance and demonstrate our design methodology, we developed a simple solar powered sensor node shown in Figure 4 (a). The hardware system consists of: a solar panel (4.5V, 50mA, $9 \times 3.8\text{cm}^2$), a rechargeable Li-ion battery (2.7–4.2V, 800mAh, nearly 100% recharging efficiency), a MicaZ mote [18], and our own developed EA board shown in Figure 4 (b).

In Figure 4(a), P_{solar}, P_{load}, $P_{battery}$ and $-P_{battery}$ represent the powers of solar, load, battery recharging, and discharging, respectively; and $0 < \eta_1, \eta_2, \eta_3 < 100\%$ are the energy translation efficiencies from the solar panel to the load (direct power), from solar panel to the battery (recharging) and from battery to the load (discharging) respectively.



(a) hardware architecture (b) The EA board

Fig. 4. EA support hardware and its architecture.

The EA board efficiently powers both the load (MicaZ mote and sensors) and the battery, as well as providing the realtime physical power information. If $P_{solar} > P_{load}$, the solar panel will power both the battery and the load; otherwise, both the solar panel and the battery will power the load; automatically controlled by the input regulator BQ24070 [19].

B. Off-line Modelling and Profiling

The knowledgebase of each sensor node is established off-line by modeling and profiling its power system and corresponding parameters, which is then used by other components at runtime.

1) Energy Translation Model and Efficiency Parameters: We first model the battery recharging and discharging process as follows

$$P_{load}/\eta_1 + P_{battery}/\eta_2 = P_{solar} \qquad P_{battery} \geq 0 \qquad (1)$$

$$\eta_1 P_{solar} + \eta_3(-P_{battery}) = P_{load} \qquad P_{battery} < 0 \qquad (2)$$

In the recharging model (1), part of the solar power P_{load}/η_1 is provided directly for consumption by the MicaZ, while the remaining power $P_{battery}/\eta_2$ is stored in the battery. In the discharge model (2), load power is supplied by both solar ($\eta_1 P_{solar}$) and battery ($-\eta_3 P_{battery}$). Using these two models, the three efficiency parameters η_{1-3} can be easily estimated by using least square methods. For instance, η_{1-3} of a typical EA board are estimated as 79%, 85%, and 84% respectively, by using an oscilloscope.

2) Off-line P_{load} Profiling: Most networking protocol operations can be translated to the scheduling of the major energy consumers: microcontroller unit (MCU), wireless transceivers (e.g. the CC2420 radio), and sensors. Since the power consumption of a given operating state (e.g. MCU idle, CC2420 on, sensor on) is stable at runtime, it is feasible to profile such major solar powered sensor nodes by referring to data sheets or using oscilloscopes.

3) Off-line Battery Modeling: The residual energy level of a battery can be measured by estimating the quantitative energy-voltage relation. We have established piecewise linear models for typical Li-ion batteries. However, such models can only be used to estimate the initial battery levels off-line, because the variation of the voltage is too small to be reliably measured as the residual energy changes at runtime.

C. Online Monitoring

A sensor node can compute its realtime solar power P_{solar}, based on the measured voltage and current of the solar panel. The realtime battery energy can be computed according to Eq. (12) in Section V, based on the estimated efficiency parameters η_{1-3}, the initial off-line estimated battery level, the online monitored P_{load} and P_{solar}. Since all these parameters can be estimated or monitored precisely, AutoSP-WSN can obtain precise realtime battery level.

IV. Solar Power Prediction: The WC-EWMA Algorithm

Our AutoSP-WSN framework requires forecasting solar power of a future prediction interval, i.e. one future slot for D-type protocols and multiple future slots for S-type protocols, with high accuracy and low complexity (Section V will discuss how to utilize the predicted solar power). Yet current solar prediction algorithms [20]–[22] cannot achieve this. Therefore, we develop a novel weather-aware solar prediction scheme, WC-EWMA, to meet all the requirements of AutoSP-WSN.

WC-EWMA is based on both long-term seasonal and short-term daily solar profiles. *A day is divided into* M *non-overlapping prediction intervals, each of which consists of* L *slot(s) with a duration* T *(i.e.* $\mathrm{M} \cdot \mathrm{L} \cdot \mathrm{T} = 24$ hours*)*. We can use the triple (i, l, d) to refer to a slot i in the prediction interval l of the d^{th} day.

A. Reference Power

Let $P_{re}(i,l,d)$ be the reference solar power in slot (i, l, d) to reflect the seasonal stable solar pofile governed by the long-term geographical climate. The reference power vector is only updated once at the end of a day as follows:

$$P_{re}(i,l,d+1) = \begin{cases} P_{re}(i,l,d), & wv(d) \geq wv_T \\ \alpha_{re} P^{real}_{solar}(i,l,d) + (1-\alpha_{re}) P_{re}(i,l,d) & \text{otherwise} \end{cases}$$

where $p^{real}_{solar}(i,l,d)$ is the real solar power metered in slot (i, l, d); $\alpha_{re} \in [0,1]$ is the weighting factor; $wv(d)$ is weather condition level of the d^{th} day (explained in next subsection), i.e. the more sunny the d^{th} day is, the smaller $wv(d)$ is; wv_T is a predefined cloudiness degree threshold. Consequently, $P_{re}(i,l,d)$ is only updated when the d^{th} day is not quite cloudy, which aims to filter the influence of bad weather days (noise) on the seasonal stable reference power.

B. Computing The Weather Volatility Value $wv(d)$

From Figure 5, it is obvious that solar power varies smoothly on sunny days (e.g. the 8^{th} day) and fluctuates on cloudy days (e.g. the 6^{th}) day. Based on the above observations, we use two metrics; *fluctuation frequency* $wv_0(d)$ and *fluctuation intensity* $wv_1(d)$ to profile the solar curve on the d^{th} day. At the end of the d^{th} day, $wv_0(d)$ is calculated as

$$wv_0(d) = \sum_{i=3}^{\text{M·L}} (h(i,d) \oplus h(i-1,d)) \tag{3}$$

where $\oplus$ is the XOR operator, and for $i \in [2, \text{M} \cdot \text{L}]$,

$$h(i,d) = \begin{cases} 1 & \text{if } P^{real}_{solar}(i,d) > P^{real}_{solar}(i-1,d) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

It is clear that $wv_0(d)$ is the total number of peaks and troughs of the d^{th} day's solar curve. $wv_1(d)$ is calculated as

$$wv_1(d) = \sum_{i=3}^{\text{M·L}} ((h(i,d) \oplus h(i-1,d)) \wedge g(i,d)) \tag{5}$$

where $g(i,d)$, $i \in [2, \text{M} \cdot \text{L}]$, is

$$g(i,d) = \begin{cases} 1 & |P^{real}_{solar}(i,d) - P^{real}_{solar}(i-1,d)| \geq FI_T \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where FI_T is a predefined threshold. We can see that $wv_1(d)$ represents the total number of the d^{th} day's peaks and troughs, where variation of two successive time slots are no smaller than FI_T. Combining both the strong and weak weather volatilities, $wv_1(d)$ and $wv_0(d) - wv_1(d)$, we get

$$wv(d) = \alpha_{wv} \cdot wv_1(d) + (1-\alpha_{wv}) \cdot (wv_0(d) - wv_1(d))$$

where $0 \leq \alpha_{wv} \leq 1$ is the weighting factor. Figure 5 shows that $wv(d)$ can reflect the daily weather conditions accurately.

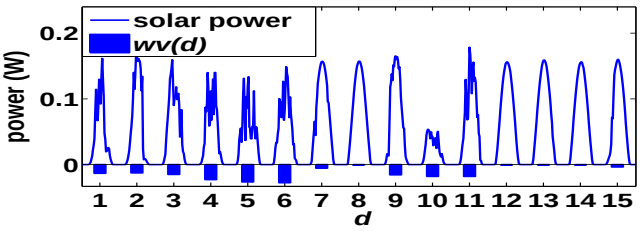


Fig. 5. Solar profile of 15 days and weather volatility $wv(d)$ for each day d with $\alpha_{wv} = 0.8$.

C. Adaptive Solar Power Prediction

At the beginning of slot (1, l, d), WC-EWMA simply predicts the solar power $p^{pred}_{solar}(i,l,d)$ for all slot(s) (i, l, d), $1 \leq i \leq \text{L}$ (i.e. a future prediction interval) as follows

$$P^{pred}_{solor}(i,l,d) = \frac{\sum_{i=1}^{\text{L}} P^{real}_{solar}(i,l-1,d)}{\sum_{i=1}^{\text{L}} P_{re}(i,l-1,d)} P_{re}(i,l,d) \tag{7}$$

V. Local Power Management

At the beginning of every prediction interval l, LPM computes the maximum feasible load rate (energy budget) for all slots in l, i.e. from slot (1, l, d) to slot (L, l, d), based on the energy information provided by EA support component. For S-type protocols, LPM sets L>1 and enforces maximum feasible load rate to be equal at every slot during l. For D-type protocols, LPM sets L=1 so energy budget are provided for each future slot. Let P_{idle} and P_{active} (Watts) represent the load power when a sensor node is idle (MCU idle, CC2420 off, and sensor off) and active (MCU active, CC2420 on, and sensor on). Since both P_{idle} and P_{active} are nearly constant at runtime [7], [18], we use the maximum feasible duty cycle $D^{max}_x(l)$ of a sensor node x to represent its maximum feasible load rate in a prediction interval l. Section VI will show how $D^{max}_x(l)$ can be used by network protocols.

A. LPM for S-type protocols

Let PS^i_x and B^i_x respectively be the predicted solar power of sensor node x at slot i, and x's battery level at the beginning of slot i, $1 \leq i \leq \text{L}$. $D^{max}_x(l)$ can be computed by solving the following piecewise linear programming problem:

maximize $$D^{max}_x(l) \tag{8}$$

subject to

$$B^{\text{L}+1}_x \geq \varphi \tag{9}$$

$$\text{B}_{\max} \geq B^i_x \geq 0 \tag{10}$$

$$\text{D}_{\min} \leq D^{max}_x(l) \leq 1 \tag{11}$$

$$\begin{aligned} B^{i+1}_x =& B^i_x + (1 - D^{max}_x(l))\text{T}\eta_2 |PS^i_x - \text{P}_{\text{idle}}/\eta_1|_+ \\ & -(1 - D^{max}_x(l))\text{T}|\text{P}_{\text{idle}} - \eta_1 PS^i_x|_+/\eta_3 \\ & + D^{max}_x(l)\text{T}\eta_2 |PS^i_x - \text{P}_{\text{active}}/\eta_1|_+ \\ & - D^{max}_x(l)\text{T}|\text{P}_{\text{active}} - \eta_1 PS^i_x|_+/\eta_3 - \text{E}_{\text{leak}} \end{aligned} \tag{12}$$

where $\mathrm{E_{leak}}$ represents the battery leakage in a slot ($\mathrm{E_{leak}} \approx 0$ for the Li-ion battery); $\mathrm{D_{min}} > 0$ is the user-defined minimum duty cycle; and the operator $|\cdot|_+$ represent $\max(0,\ \cdot)$ (i.e. for any real number a, $|a|_+ = a$, if $a > 0$; $|a|_+ = 0$, otherwise).

Input: Energy Parameters, $\mathrm{D_{min}}$, and protocol type.

1: $D_x^{max}(l) \leftarrow 1$;
2: $i, j \leftarrow 1; E \leftarrow B_x^1; K_1, K_2, K_3 \leftarrow 0$;
3: $\mathrm{P}_1 \leftarrow \eta_3(\mathrm{P_{active}} - \mathrm{P_{idle}})\mathrm{T}; \mathrm{P}_3 \leftarrow \frac{\eta_2}{\eta_1}(\mathrm{P_{active}} - \mathrm{P_{idle}})\mathrm{T}$;
4: **while** $i \leq \mathrm{L}$ **do**
5: **if** $\eta_1 PS_x^i < \mathrm{P_{idle}}$ **then** /* case 1 */
6: $P \leftarrow (\eta_1 PS_x^i - \mathrm{P_{idle}})\mathrm{T}/\eta_3 - \mathrm{E_{leak}}$;
7: $B_x^{i+1} \leftarrow B_x^i + P - D_x^{max}(l)\mathrm{P}_1$;
8: $K_1 \leftarrow K_1 + \mathrm{P}_1$; $E \leftarrow E + P$;
9: **else if** $\mathrm{P_{idle}} \leq \eta_1 PS_x^i < \mathrm{P_{active}}$ **then** /* case 2 */
10: $P \leftarrow (PS_x^i - \mathrm{P_{idle}}/\eta_1)\mathrm{T}\eta_2 - \mathrm{E_{leak}}$;
11: $P_2 \leftarrow (PS_x^i - \mathrm{P_{idle}}/\eta_1)\mathrm{T}\eta_2 - (\eta_1 PS_x^i - \mathrm{P_{active}})\mathrm{T}/\eta_3$;
12: $B_x^{i+1} \leftarrow B_x^i + P - D_x^{max}(l)P_2$;
13: $K_2 \leftarrow K_2 + P_2$; $E \leftarrow E + P$;
14: **else** /* case 3 */
15: $P \leftarrow (PS_x^i - \mathrm{P_{idle}}/\eta_1)\mathrm{T}\eta_2 - \mathrm{E_{leak}}$;
16: $B_x^{i+1} \leftarrow B_x^i + P - D_x^{max}(l)\mathrm{P}_3$;
17: $K_3 \leftarrow K_3 + \mathrm{P}_3$; $E \leftarrow E + P$;
18: **end if**
19: **if** $B_x^{i+1} > \mathrm{B_{max}}$ **then** /* record overcharging slot as j */
20: $B_x^{i+1} \leftarrow \mathrm{B_{max}}$; $j \leftarrow i + 1$;
21: $E \leftarrow \mathrm{B_{max}}$; $K_1 \leftarrow 0; K_2 \leftarrow 0; K_3 \leftarrow 0$;
22: $i \leftarrow i + 1$;
23: **else if** $B_x^{i+1} < 0$ **then** /* reduce $D_x^{max}(l)$ */
30: $D_x^{max}(l) \leftarrow E/(K_1 + K_2 + K_3)$;
24: $E \leftarrow B_x^j$; $K_1 \leftarrow 0; K_2 \leftarrow 0; K_3 \leftarrow 0$;
25: $i \leftarrow j$;
26: **else if** $B_x^{i+1} < \varphi \wedge i = \mathrm{L}$ **then** /* reduce $D_x^{max}(l)$ */
35: $D_x^{max}(l) \leftarrow (E - \varphi)/(K_1 + K_2 + K_3)$;
27: $E \leftarrow B_x^j$; $K_1 \leftarrow 0; K_2 \leftarrow 0; K_3 \leftarrow 0$;
28: $i \leftarrow j$;
29: **else** /* $D_x^{max}(l)$ is maximum feasible before i */
30: $i \leftarrow i + 1$;
31: **end if**
32: **end while**
33: **return** $\max(\mathrm{D_{min}}, D_x^{max}(l))$;

Fig. 6. Pseudocode of LPM algorithm for each sensor node x.

The constraint (10) ensures that the battery level should not exceed the battery capacity $\mathrm{B_{max}}$ nor be lower than 0. (11) states that $D_x^{max}(l)$ is x's duty cycle. Constraint (9) is the so-called final state constraint [4], [7]. The parameter φ ensures enough battery energy remaining for next prediction interval and the choice of φ influences long-term system performance. In AutoSP-WSN, LPM component sets φ to be a constant value for S-type protocols and dynamically adjusts φ for D-type protocols (next subsection). According to energy translation models (1) and (2), the constraint (12) states the following three battery updating cases from slot i to $i+1$:

- **case 1.** $\eta_1 PS_x^i < \mathrm{P_{idle}}$: discharging for both idle duration $\mathrm{T}(1 - D_x^{max}(l))$ and active duration $\mathrm{T}D_x^{max}(l)$. Recall that T is the duration of a slot.
- **case 2.** $\mathrm{P_{idle}} < \eta_1 PS_x^i < \mathrm{P_{active}}$: discharging when active and recharging when idle.
- **case 3.** $\eta_1 PS_x^i > \mathrm{P_{active}}$: recharging for both active and idle durations.

Due to the limited computational capacity of tiny sensor nodes, solving the problem (8) online would be prohibitive. Therefore, we propose a light-weight algorithm shown in Figure 6 to compute the optimal $D_x^{max}(l)$ at runtime.

The LPM algorithm looks a bit complicated due to the three detailed battery update cases. However, its logical flow is clear. The main idea is to monotonously and gradually reduce $D_x^{max}(l)$ from the upper bound $D_x^{max}(l) = 1$ (line 1), until constraints (9) and (10) are guaranteed. After initialization in lines 1-3, $D_x^{max}(l)$ keeps constant or is reduced in each iteration of the **while** loop (lines 4–32). Lines 5–18 update the battery levels from slot i to $i+1$ and other energy-relevant variables. Lines 19–22 deal with the battery capacity constraint and record the current slot as the overcharge slot j (note that B_x^j is always equal to B_x^1 or $\mathrm{B_{max}}$). Lines 23–28 process the two cases: $B_x^i < 0$ and $B_x^{\mathrm{L}+1} < \varphi$, for which the current $D_x^{max}(l)$ is reduced and the feasibility of the updated $D_x^{max}(l)$ will be rechecked from slot j. Finally, LPM algorithm returns, $D_x^{max}(l)$, the *optimal* solution to the problem (8), if it exists; otherwise, LPM algorithms returns $\mathrm{D_{min}}$. For brevity, we omit the proof of the optimality of the LPM algorithm.

In the worst case, LPM algorithm requires $\mathrm{O(L^2)}$ simple arithmetic calculations, when battery $D_x^{max}(l)$ is updated in every slot within prediction interval l. However, this rarely occurs in practice. In fact, after the initialization, $D_x^{max}(l)$ is typically updated once to guarantee the final state constraint (9). If there exists no battery overcharging and exhaustion, the number of simple arithmetic calculations reduces to O(L). Therefore, the computational overhead of the LPM algorithm is similar to a sorting operation for L elements, which is much more efficient than directly solving problem (8) online.

B. LPM for D-type protocols

Since L=1 for D-type protocols, we directly refer a slot $i \in$ [1, 2, ... , M] in the dth day without mentioning a prediction interval for readability. We rewrite the final state parameter φ as $\varphi_x^i(d)$ for sensor node x at slot i in the dth day. The minimum energy consumption of a sensor node in a slot is

$$\mathrm{E_{min}} = \mathrm{T}(\mathrm{D_{min}}\mathrm{P_{active}} + (1 - \mathrm{D_{min}})\mathrm{P_{idle}})$$

During the night, since $\eta_1 PS_x^i \mathrm{T} < \mathrm{E_{min}}$, a sensor node x may exhaust its battery even when it operates in the minimal required duty cycle $\mathrm{D_{min}}$. Therefore, to guarantee ENO, the node must preserve enough battery energy during the daytime to avoid battery exhaustion at night. To this end, LPM assigns $\varphi_x^i(d)$ as shown in Figure 7, where $i_0(d)$ and $i_1(d)$ are the first slots of daytime and night in the dth day respectively[1]. Let $B_x^i(d)$ be the battery level of sensor node x at the beginning

[1]The daytime and night are defined as follows: if $\eta_1 PS_x^i \leq \mathrm{E_{min}}/\mathrm{T}$, slot i is in daytime $[i_0(d),\ i_1(d)]$; otherwise, i is in night $[i_1(d),\ i_0(d+1)]$.

of a time slot i. It can be seen that $\varphi_x^i(d)$ is the lower bound of battery level $B_{i+1}^x(d)$ for every slot i. The calculation of $\varphi_i^x(d)$ is presented below.

With a duty cycle of $\mathrm{D_{min}}$, the battery energy updates during night $i \in [i_1(d), i_0(d+1)]$ as

$$B_x^{i+1}(d) = B_x^i(d) - \mathrm{E_{min}}/\eta_3 - \mathrm{E_{leak}}$$

The minimum battery energy B_{ENO} that a sensor node should store at slot $i_1(d)$ can be computed as follows

$$B_{ENO} = (\mathrm{E_{min}}/\eta_3 + \mathrm{E_{leak}})(i_0(d) + M(d) - i_1(d))$$

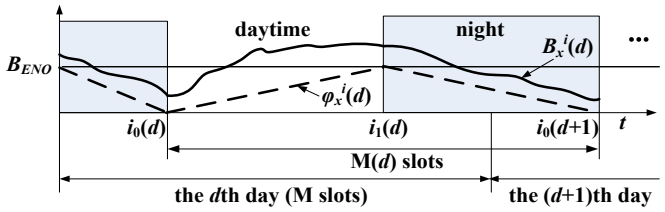


Fig. 7. Final state parameter $\varphi_x^i(d)$ and real battery level evolutions.

where $M(d)$ is the number of slots between $i_0(d)$ and $i_0(d+1)$, shown in Figure 7. This normally change seasonally but remain constant (i.e. $M(d)$=M) for multiple successive days (e.g. a week). Hence, we can set $M(d)$=M. Consequently, the following assignment of $\varphi_x^i(d), i \in [i_0(d), i_0(d+1)]$ is sufficient to ensure ENO for the whole day:

$$\varphi_x^i(d) = \begin{cases} \dfrac{(i - i_0(d))B_{ENO}}{i_1(d) - i_0(d)}, & i \in [i_0(d),\ i_1(d)) \\ \dfrac{(i_0(d) + \mathrm{M} - i)B_{ENO}}{i_0(d) + \mathrm{M} - i_1(d)}, & i \in [i_1(d),\ i_0(d+1)) \end{cases}$$

which can be calculated at $i_0(d)$[2] for each day d. As a result, node x can calculate its maximum feasible duty cycle $D_x^{max}(i)$ at the beginning of every slot i:

$$D_x^{max}(i) = \max_{\mathrm{D_{min}} \le D_x \le 1, \mathrm{B_{max}} \ge B_x^{i+1}(d) \ge \varphi_x^i(d)} D_x \qquad (13)$$

The LPM algorithm in Figure 6 can be used to efficiently solve the problem (13) in one loop, by setting L=1 and replacing φ and $D_x^{max}(l)$ as φ_x^i and $D_x^{max}(i)$ respectively.

VI. Two Adaptive Network Protocols

The per-node energy constraint (i.e. maximum feasible duty cycle) provided by LPM offers an foundational support to develop new network-wide protocol and to extend existing battery based WSN protocols to SP-WSNs. To demonstrate how autonomic communication principles aid the designing of network-wide schemes in the highly dynamic and distributed SP-WSNs, this section presents two simple network protocols: a D-type routing protocol, called SP-BCP; and a S-type rate control protocol, called PEA-DLEX.

[2] The value of $i_1(d)$ can be estimated online based on the reference power vector in WC-EWMA algorithm.

A. A D-type adaptive routing protocol: SP-BCP

Recently, a D-type routing protocol, BCP [17], is developed for data collection in WSNs (please refer to [17] for the details of BCP). BCP is adaptive to sudden link fluctuations, queue hot-spots, and topology changes. However, without energy awareness, BCP can not provide any guarantee for efficient solar energy usage or long-term ENO in SP-WSNs. Therefore, we develop SP-BCP, an energy aware backpressure routing protocol for sustainable data collections in SP-WSNs.

Before presenting SP-BCP, we first introduce how to use duty cycle as energy constraints for D-type protocols. To ensure ENO, the active duration budget of a node x in slot i is $\mathrm{T}D_x^{max}(i)$, provided by LPM. Let the remaining active duration of node x be DR_x, initialized as $DR_x = \mathrm{T}D_x^{max}(i)$ at the beginning of each slot i. Let N_x be the set of x' all one-hop neighbors. To forward a packet over a link $(x,\ y),\ y \in N_x$, both the transmitter x and the receiver y will spend an active duration of $DP_{x,y}$[3], resulting in the reduction of remaining active durations $DR_x = DR_x - DP_{x,y}$ and $DR_y = DR_y - DP_{x,y}$. Therefore, to ensure ENO, a packet should not be transmitted over link (x, y), if $DR_x < DP_{x,y}$ or $DR_y < DP_{x,y}$, i.e. both the transmitter and the receiver must have enough remaining active durations for a packet transmission. Hence, D-type protocols can easily use DR_x as a local energy metric for every node x's decision making.

The operations of SP-BCP for a packet transmission are quite simple: **(1)** ***Energy aware weight calculation***: Each sensor node x computes the weights $w_{x,y}$ for all its neighbors $y \in N_x$. If remaining active durations of x and its neighbor y is larger than the estimated packet transmission duration (i.e. $DR_x \ge DP_{x,y}$ or $DR_y \ge DP_{x,y}$), it sets $w_{x,y} = (Q_x - Q_y)/DP_{x,y}$, where Q_x and Q_y represents the queue backlogs of nodes x and y respectively; otherwise $w_{x,y}$ is set as zero. (2) ***Routing***: Each node x selects link (x, y^*) with the maximum weight for optimal potential receiver $y^* \in N_x$. (3) ***Forwarding***: If $w_{x,y^*} > 0$, x forwards the packet to y^*. (4) ***Remaining activity duration update***: When a packet is transmitted, both the transmitter x and the receiver y^* update their remaining activity durations as $DR_x = DR_x - DP_{x,y}$ and $DR_{y^*} = DR_{y^*} - DP_{x,y^*}$.

The communication overhead of SP-BCP is quite light: every node x can dynamically obtain Q_y and DR_y by periodically broadcasting one-hop beacons or using overhearing. Evaluation results presented in Section VII demonstrate that SP-BCP achieves hard ENO guarantee, and therefore prevents any node running out of energy.

B. A S-type adaptive rate control protocol: PEA-DLEX

The key practical issue for such protocols is the accumulated prediction errors of multiple future slots in a prediction interval. For a sensor node, if the predicted solar power is larger than the real solar power, LPM would assign a larger load budget, resulting in a risk of battery exhausting; otherwise, solar harvesting opportunities can not be fully

[3] $DP_{x,y}$ can be easily obtain based on realtime link estimators (e.g. [17])

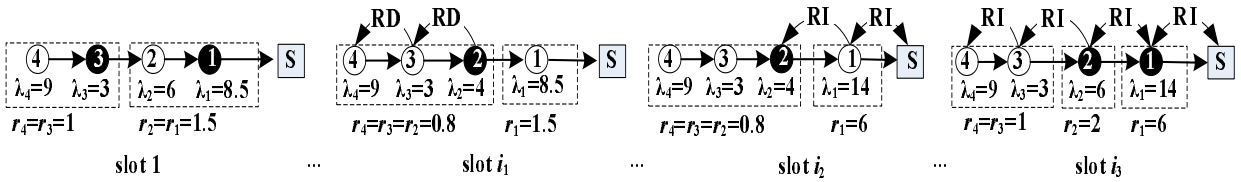


Fig. 8. An example of PEA-DLEX in a line topology SP-WSN in 4 slots $1 < i_1 < i_2 < i_3 \leq$ L during a prediction interval l. Saturated and unsaturated nodes are marked as black and white circles respectively.

utilized, leading to energy waste. For the whole network, the prediction errors may be heterogeneous across different individual nodes in a SP-WSN (e.g. future solar power is over-evaluated by some nodes but under-evaluated by others), which would further degrade the end-to-end network performance. To show how to use self-management principles to address above issue, we propose an adaptive Lexicographic Max-min (LM) rate control protocol PEA-DLEX, by modifying the static algorithm DLEX (please see details in [10]).

We consider a routing tree consisting of a set of sensor nodes N, rooted at a sink S in a prediction interval l. Assume all wireless links have the same capacity C (kbps). Let r_x be the sensing rate at which each node x measures the environmental data, and ST_x be the set of all nodes in the subtree rooted at x, excluding x. A rate assignment can be represented as a $|N|$-dimensional vector $R = (r_1, r_2, ..., r_{|N|})$ where the x^{th} entry represents the sensing rate of node x.

Definition 1. Feasible Rate Assignment. *A rate assignment R is feasible if under which every node can achieve ENO. Mathematically, for every node $x \in N$, $AF_x \leq \lambda_x$, where $AF_x = r_x + 2\sum_{y \in ST_x} r_y$ is x's actual forwarding data rate[4], and $\lambda_x = D_x^{max}(l)$C is the maximum allowed data rate.*

Under a feasible rate assignment, every node x must be in one of the two states: ***saturated***, if $AF_x = \lambda_x$; ***unsaturated***, if $AF_x < \lambda_x$.

Definition 2. LM Rate Assignment. *For two feasible rate assignment R and R' in non-descending order, if there exist a prefix (r_1,..., r_i) of R and a prefix (r'_1,..., r'_i) of R' such that $r_i > r'_i$, $r_j = r'_j$, $\forall 1 \leq j < i$, then R is lexicographically greater than R'. R is LM rate assignment if it is lexicographically greater than all other feasible rate assignments.*

For instance, consider slot 1 in Figure 8, the LM rate assignment is ($r_4 = 1$, $r_3 = 1$, $r_2 = 1.5$, $r_1 = 1.5$), which is lexicographically greater than any other feasible rate assignment such as ($r_4 = 1$, $r_3 = 1$, $r_2 = 1$, $r_2 = 2.5$). Recall that $D_x^{max}(l)$ provided by the LPM component of node x is for the whole prediction interval l. The DLEX algorithm [10] uses a static LM assignment (calculated in slot 1) for all slots $1 \leq i \leq L$. However, due to prediction error, nodes need to adaptively adjust their sensing rate to achieve ENO and better rate assignment (in terms of LM fairness). To this end, i.e. x tracks its state (i.e. saturated or unsaturated) and $\Delta B_x(i) = B_x^{real}(i) - B_x^{virtual}(i)$ in every slot $1 \leq i \leq$ L, where $B_x^{real}(i)$ and $B_x^{virtual}(i)$ represents the real measured battery level and the expected virtual battery level computed by LPM algorithm. If $|\Delta B_x^i|$ is larger than a predefined threshold DB_T, x updates $D_x^{max}(l)$ and λ_x by using the LPM algorithm. Let the $\Delta\lambda_x$ be the differential between the updated and previous λ_x.

PEA-DLEX adjusts the rate assignment if one of the two events, Rate Decrease (RD) and Rate Increase (RI), is triggered by any node x's state change or $\Delta\lambda_x$ update as follows

A RD event *is triggered if $\Delta\lambda_x$ <0 or x's state changes from unsaturated to saturated.* In this case, x has to decrease AF_x to ensure ENO. To this end, x updates r_x and r_y, $y \in ST_x$[5], then multicasts the updated sensing rates carried by a RD packet to the nodes in ST_x.

A RI event *is triggered if x is saturated and $\Delta\lambda_x$ >0.* In this case, x sends a RI packet that contains the updated rates $r_y, y \in \{x\} \cup ST_x$ towards the sink. When x's parent, node z, receives the packet, if $r_z \leq r_x$ or z is unsaturated, it add its rate r_z into RI packet and forwards the updated RI packet to its parent; otherwise, z drops the RI packet. This process is repeated until the RI packet is received by the sink. The sink updates the RI packet and sends it back to x. When a node between the sink and x forwards this RI packet, it updates its rate as assigned by the sink. Upon receiving the RI, x updates its rate and multicasts the RI packet towards its upstream node(s). The RI packet sent by x is forwarded by unsaturated nodes but dropped by saturated nodes.

Take Figure 8 for instance, after the initialization in slot 1, node 2 triggers a RD event in slot i_1. Since $\Delta\lambda_2 = -2 < 0$, node 2 assigns the new rates $r_2 = r_3 = r_4 = 0.8$ and transmits the updated rates to nodes 3 and 4. In slot i_2, node 1 triggers a RI event($\Delta\lambda_1 = 5.5$), it sends RI packet to the sink which assigns the new rates (1,1,2,6), based on the previous information of nodes 2–4 in slot 1 and its current information in slot i_2. However, node 2 drops the RI packet sent by the sink since it is saturated. In slot i_3, node 2 triggers a RI event ($\Delta\lambda_1 = 2$). The sink computes the new rate (1,1,2,6) based on the information from nodes 3 and 4 in slot 1 and nodes 1 and 2 in slot i_3. The updated rates can be sent to every node. It can be seen that the reactions for RD and RI events aim to make a trade-off between communication overhead and LM optimality, guided by self-healing and self-

[4]Due to the half-duplex operation mode of the common commercial wireless radios, a node can not transmit and receive synchronously. Each node x has to receive and transmit the data traffic generated by all nodes in ST_x (i.e. $2\sum_{y \in ST_x} r_y$), as well as transmit its own data (i.e. r_x). Therefore, we have $AF_x = r_x + 2\sum_{y \in ST_x} r_y$.

[5]This calculation is based on previous information from nodes in ST_x recorded by x and Eq.(12) in [10].

TABLE I
EXPERIMENT PARAMETERS OF THE THREE PREDICTION ALGORITHMS.

EWMA	WCMA-PDR						WC-EWMA			
α	D	α	δ	γ	w	Ad	α_{wv}	α_{re}	FI_T	wv_T
0.5	4	0.7	0.7	0.55	4	1.8	0.8	0.5	5	3

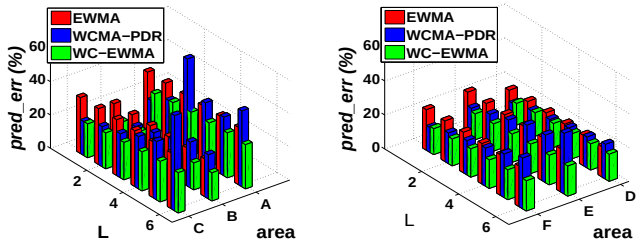


Fig. 9. Mean prediction errors for three prediction algorithms. Areas A-F respectively represent Aberdeen, Ashland, Dillon, Hermiston, Moab, and Madras.

optimization principles: adjusting nodes' rates locally needs no communication but may lead to poor fairness and battery exhaustion; while globally updating rate assignments in every slot can guarantee strict LM optimality and ENO theoretically, but will lead to heavy overheads in practice.

VII. EVALUATION

We implemented our AutoSP-WSN framework in TinyOS 2.1.1 [23] and evaluated them through our aforementioned SP-WSN platform and the Tossim simulator [14]. All real-world experiments used the on-line measured solar power, while all simulations were based on the public solar database [24]. For all experiments, the duration of a slot, T, was set as 30 minutes.

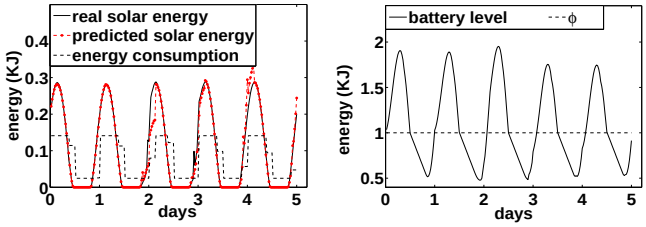
A. Evaluation of Solar Prediction Scheme WC-EWMA

We evaluated the performance of WC-EWMA algorithm by comparing it with the classic scheme EWMA [7], [20] and the state-of-the-art algorithm WCMA-PDR [22], based on real solar data in six geographical areas [24] with the duration of 100 days. Table I shows the simulation settings.

Figure 9 shows the impact of the prediction interval length L on the mean prediction error $pred\_err$ (defined in [21], [22]). It can be seen that WC-EWMA is more accurate than EWMA and WCMA-PDR (the average $pred\_err$ for WC-EWMA, EWMA, and WCMA-PDR are 18.7%, 26.7%, and 25.5% respectively), especially for large L scenarios. We can also see that WC-EWMA is relatively insensitive to L, but the prediction error of WCMA-PDR increases rapidly as L increases. Furthermore, WCMA-PDR requires to maintain solar profile and prediction errors of many days, and a large number of multiplication and division operations. However, WC-EWMA only requires to maintain two $\mathrm{M}\cdot\mathrm{L}$-dimensional vectors for reference and real power, and $\mathrm{O}(\mathrm{M}\cdot\mathrm{L})$ simple logical and arithmetic calculations for a whole day, resulting in much less computational and storage overheads.

TABLE II
EXPERIMENT PARAMETERS OF LPM (S-TYPE PROTOCOLS).

L	$\mathrm{B_{max}}$	B_x^1	φ	$\mathrm{P_{idle}}$	$\mathrm{P_{active}}$	$\mathrm{D_{min}}$
6	10656J	1000J	1000J	13.7mW	78.4mW	0

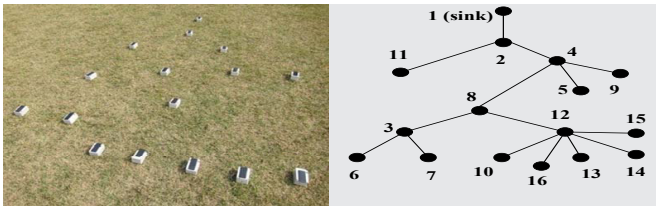


(a) solar and load energy evolution (b) battery level evolution

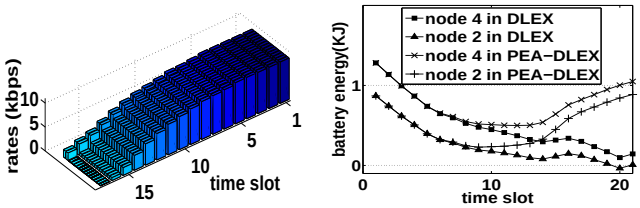
Fig. 10. Evaluations of LPM for S-type protocols.

B. LPM (S-type protocols) and PEA-DLEX

The parameter settings of LPM experiments are shown in Table II. Since the Li-ion battery has a large capacity, we set the small initial battery level to show that our LPM algorithm is still suitable for small-size energy buffers. Figure 10 shows the results of S-type LPM algorithm for five days. With the maximum duty cycle $D_x^{max}(i)$ computed by LPM of node x at slot i, the corresponding energy consumption is computed as $E_x^{max}(i) = D_x^{max}(i)\mathrm{P_{active}}+(1-D_x^{max}(i))\mathrm{P_{idle}}$. It is obvious that both the node's battery level and energy consumption is adaptive to the harvested solar power dynamics. During the night, although $\mathrm{D_{min}}$ was set as zero, the battery level still linearly reduces, caused by idle CPU cycles ($\mathrm{P_{idle}}$) and battery leakage ($\mathrm{E_{leak}}$).



(a) network deployment and routing tree topology



(b) ranked rate assignment (c) battery level evolutions

Fig. 11. Real-world experiment results of DLEX and PEA-DLEX.

We first evaluated the short-term performance of PEA-DLEX scheme on a real 16-node SP-WSN during a prediction interval of L=20 slots (i.e. 10 hours). The network deployment and underlying routing tree are shown in Figure 11 (a). We set DB_T=20 J, and randomly set $B_x^1 = 1000 \pm 50\%$ J and

$\varphi_x = 1000 \pm 50\%$ J for different node x. All other parameters were set as the same in the S-type LPM experiment. Figure 11(b) shows the ranked sensing rate of every node in every time slot assigned by PEA-DLEX. Initially, the rate of every node is assigned to be equal (8.6 kbps), and there was only one saturated node (node 2) in the network. At the 6^{th} slot, node 4 changed from unsaturated to saturated, which triggered a RD event, and the network existed two level of max-min rates from then on. During the whole prediction interval, the total events triggered by nodes 2 and 4 are 14 and 12 respectively. Figure 11(c) shows the battery level evolutions of the two bottleneck nodes 2 and 4 for PEA- DLEX and DLEX respectively. For DLEX, they did not guarantee their final state constraints and the node 2 even exhausted battery energy in the 19^{th} slot. In contrast, PEA-DLEX prevented battery exhausting and maintained enough energy for the future poor solar harvesting situations.

To evaluate the long-term performance of PEA-DLEX of multiple prediction intervals, we also run a simulation for a 50-node SP-WSN with a randomly constructed routing tree during 12 days. The parameter setting was same as that in the real-world experiment. Figure 12 shows the ENO performance of PEA-DLEX and DLEX. In PEA-DLEX, no node run out of battery energy for the 12 days, while DLEX showed several bottleneck nodes failing, resulting in the disconnection of the whole network. Both testbed experiment and simulation results show that PEA-DLEX scheme manages to ensure ENO and achieves near LM optimality.

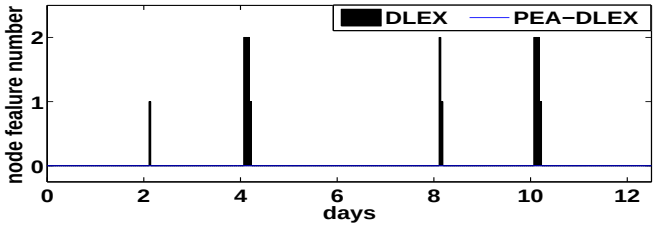
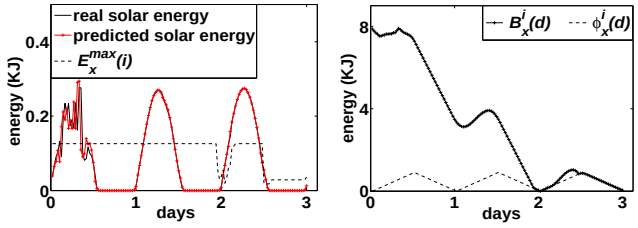


Fig. 12. Simulation results of DLEX and PEA-DLEX .

C. LPM (D-type Protocols) and SP-BCP

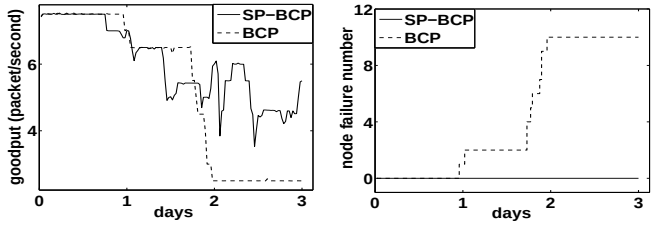
We evaluated the D-type LPM algorithm for three days. We set the $\mathrm{D}_{\min}$ =10%. Figure 13 shows that node's battery level is always above the $\varphi_x^i(d)$ curve, which means that the $\varphi_x^i(d)$ assignment scheme of LPM for D-type protocols can provide hard ENO guarantee.



(a) solar and load energy (b) battery level and $\varphi_x^i(d)$

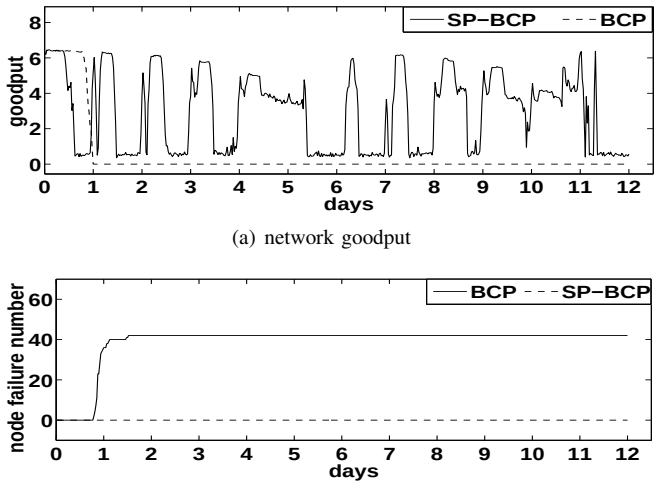
Fig. 13. Performance of LPM for D-type protocols.

To evaluate the performance of SP-BCP, we first compared the real-world performance of BCP and SP-BCP in our 16-node SP-WSN for three days. We set $\mathrm{D}_{\min}$=10%, the initial battery level as 1.2 KJ $\pm$ 30% for different nodes, and sensing (packet generation) rate as one packet per two seconds. As shown in Figure 14, for SP-BCP, no nodes run out of energy during the three days. Therefore, ENO were achieved. The sink continuously received data in every slot and achieved relatively high packet delivery ratio (about 76.8–98.9%) during the daytime. The main reasons of packet loss are limited data buffer size and inelastic sensing rates (a flow controller could significantly reduce the packet loss). For BCP, however, 10 nodes died during the second day, leading to network disconnection and significant degradation of network goodput.



(a) network goodput (b) node failure evolution

Fig. 14. Real-world experiment results of BCP and SP-BCP.



(a) network goodput

(b) node failure evolution

Fig. 15. Simulation results of BCP and SP-BCP.

Figure 15 shows the simulation results for a random deployed 50-node SP-WSNs for 12 days. Sensing rate of every node was set as one packet per seven seconds, and initial battery level were set as 200 J $\pm$ 30% for different nodes randomly. The simulation shows similar results to the testbed experiment, BCP failed quickly in the second day, but SP-BCP achieved sustainable data collection for the whole 12 days. In summary, both experiment and simulation results show that SP-BCP can improve the end-to-end performance of backpressure routing protocols in SP-WSNs.

VIII. Related Work

Autonomic WSNs. There are several efforts that apply the autonomic principles to the design of general WSNs [25]–[27], mobile sensor networks [28], and body sensor networks [29]. However, none of these focus on SP-WSNs, or more generally, WSNs with dynamic renewable energy sources. Furthermore, they are all evaluated based on simulations rather than real-world platforms.

Energy Harvesting WSNs. For solar prediction, the classic algorithm EWMA [7], [20] is lightweight but suffer from large prediction error, and the weather-aware scheme WCMA-PDR [22], is designed for one-slot solar power prediction. In contrast, our WC-EWMA can predict multiple-slot solar power with lower overheads and higher accuracy. Current power management schemes [4], [7], [8] focus on optimizing per-node utility rather than supporting end-to-end network protocols as our LPM. There also exist several network protocols, such as MAC [9], routing [11], [13], localization [30], rate control [10], and cross-layer schemes [15]. However, without hardware-driven EA and power management support, none of above is actually implemented in real SP-WSN platforms. In addition, most of them focus on specific components, while AutoSP-WSN not only improves the performance of individual components but also systematically integrates them together in a real SP-WSN. For a recent comprehensive survey of energy harvesting WSNs, we refer the reader to [31].

IX. Conclusion and Discussion

In this paper, we develop and implement AutoSP-WSN, the first autonomic framework for periodical data collection applications in SP-WSN. AutoSP-WSN achieves sustainable data collection, nearly optimal solar power usage, and high end-to-end performance. We show that adopting autonomic principles, especially context awareness and adaptiveness, benefit not only the design of AutoSP-WSN architecture as a whole, but also the individual components, including reliable energy awareness support and solar power prediction (WC-EWMA algorithm), local power management (LPM), as well as the network protocols (rate control protocol PEA-DLEX and the routing protocol SP-BCP). Extensive evaluations based on a real-world SP-WSN platform and the Tossim simulator demonstrate the effectiveness of the proposed algorithms.

AutoSP-WSN presents the fundamental tradeoffs between network performance and cost, which should be considered in developing new SP-WSN schemes. WC-EWMA can be directly used, and LPM can be easily modified for other SP-WSN platforms. Furthermore, with the real-time energy budget provided by LPM, various adaptive network protocols can be easily developed based on the methodologies of designing D-type and S-type protocols. In addition, beside battery-based SP-WSNs, AutoSP-WSN can also be used in supercapacitor-based SP-WSNs, where energy awareness can be better supported. However, due to the limited capacity and huge leakage of supercapacitors, it could be impossible to guarantee either QoS requirement or ENO during night.

Beside periodical data collection, event-based applications such as target tracking or information queries are also important WSN applications. Such applications normally require hard end-to-end QoS (e.g. delay) guarantees. It is straightforward to extend AutoSP-WSN to support such applications, by mapping local QoS metric $\mathrm{D}_{\min}$ to end-to-end QoS metrics. More broadly, for large-scale SP-WSNs with mobile sinks (or data mules), sensor nodes may store sensor data until a mobile sink pass by, rather than delivery real-time data through end-to-end paths. To support delay-tolerant applications in such networks (e.g. assisted-reporting garbage bins), new functionalities should be added to AutoSP-WSNs, such as classifying opportunistic data muling protocols and extending LPM by considering topology and buffer-size awareness for such protocols.

References

[1] J. Yick and B. Mukherjee, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, and M. Srivastava, "Design considerations for solar energy harvesting wireless embedded systems," in *Proc. ACM/IEEE IPSN*, 2005.

[3] R. Vullers, R. Schaijk, H. Visser, J. Penders, and C. Hoof, "Energy harvesting for autonomous wireless sensor networks," *IEEE Solid State Circuits Mag.*, vol. 2, no. 2, pp. 29–38, 2010.

[4] C. Moser, L. Thiele, D. Brunelli, and L. Benini., "Adaptive power management for environmentally powered systems," *IEEE Trans. Computers*, vol. 59, no. 4, pp. 478–491, 2010.

[5] M. Huebscher and J. A. McCann, "A survey of autonomic computing-degrees, models, and applications," *ACM Computing Surveys*, vol. 40, no. 3, pp. 1–28, 2010.

[6] S. Dobson, S. Denazis, A. Fernndez, D. Gaiti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli, "A survey of autonomic communications," *ACM Trans. Auton. Adapt. Syst.*, vol. 1, no. 2, pp. 233–259, 2006.

[7] A. Kansal, J. Hsu, S. Zahedi, and M. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embedded Comput. Sys.*, vol. 6, no. 4, 2007.

[8] C. S. Chen; P. Sinha; N.B. Shroff, "Finite-horizon energy allocation and routing scheme in rechargeable sensor networks," in *Proc. IEEE INFOCOM*, 2011, pp. 2273 – 2281.

[9] Z.A.Eu, H. Tan, and W. Seah, "Design and performance analysis of MAC schemes for Wireless Sensor Networks Powered by Ambient Energy Harvesting," *Ad Hoc Networks*, 2010.

[10] K. Fan, Z. Zheng, and P. Sinha, "Steady and fair rate allocation for rechargeable sensors in perpetual sensor networks," in *Proc. ACM SenSys*, 2008, pp. 239–252.

[11] Z. Eu, H. Tan, and W. Seah, "Opportunistic routing in wireless sensor networks powered by ambient energy harvesting," *Computer Networks*, vol. 54, no. 17, pp. 2943–2966, 2010.

[12] K. Zeng, K. Ren, W. Lou, and P. Moran, "Energy aware efficient geographic routing in lossy wireless sensor networks with environmental energy supply," *Wireless Networks*, vol. 15, no. 1, pp. 39–51, 2009.

[13] L. Lin, N. Shroff, and R. Srikant, "Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources," *IEEE/ACM Trans. Netw.*, vol. 15, no. 5, pp. 1021–1034, 2007.

[14] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: accurate and scalable simulation of entire tinyOS applications," in *Proc. ACM SenSys*, 2003, pp. 126–137.

[15] R. Liu, P. Sinha, and C. Koksal, "Joint energy management and resource allocation in rechargeable sensor networks," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[16] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proc. ACM SenSys*, 2009.

[17] S. Moeller, A. Sridharan, B. Krishnamachari, and O. Gnawali, "Routing without routes: the backpressure collection protocol," in *Proc. ACM/IEEE IPSN*, 2010.

[18] http://www.openautomation.net/uploadsproductos/micaz_datasheet.pdf.

[19] http://www.ti.com/lit/ug/sluu248/sluu248.pdf.

[20] D. R. Cox, "Prediction by exponentially weighted moving averages and related methods," *Journal of the Royal Statistical Society*, vol. 23, no. 2, pp. 414–422, 1961.
[21] J. R. Piorno, C. Bergonzini, D. Atienza, and T. S. Rosing, "Prediction and management in energy harvested wireless sensor nodes," in *Proc. Wireless Vitae*, 2009, pp. 6–10.
[22] C. Bergonzini, D. Brunelli, and L. Benini, "Comparison of energy intake prediction algorithms for systems powered by photovoltaic harvesters," *Microelectronics Journal*, vol. 41, no. 11, pp. 766–777, 2010.
[23] http://www.tinyos.net/.
[24] http://solardat.uoregon.edu/SelectArchival.html.
[25] D. Marsh, R. Tynan, D. O'Kane, and G. M. O'Hare, "Autonomic wireless sensor networks," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 7, pp. 741–748, 2004.
[26] M. Huebscher, J. McCann, and A. Hoskins, "Context as Autonomic Intelligence in a Ubiquitous Computing Environment," *International Journal of Internet Protocol Technology*, vol. 2, no. 1, pp. 30–39, 2007.
[27] T. Bourdenas, D. Wood, P. Zerfos, F. Bergamaschi, and M. Sloman, "Self-adaptive routing in multi-hop sensor networks," in *Proc. IEEE CNSM*, 2011, pp. 1–9.
[28] M. A. K.H. Low, W.K. Leow, "Autonomic mobile sensor network with self-coordinated task allocation and execution," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 36, no. 3, pp. 315–327, 2006.
[29] S. Thiemjarus and G. Yang, "An autonomic sensing framework for body sensor networks," in *Proc. ICST 2nd international conference on Body area networks*, 2007, pp. 61–70.
[30] G. Challen, J. Waterman, and M. Welsh, "IDEA: integrated distributed energy awareness for wireless sensor networks," in *Proc. ACM MobiSys*, 2010, pp. 35–48.
[31] S. Sudevalayam and P. Kulkarni, "Energy harvesting sensor nodes: Survey and implications," *IEEE Commun. Surveys Tuts.,*, vol. 13, no. 3, pp. 443–461, 2011.