Christian Renner, Stefan Unterschütz, Volker Turau, and Kay Römer. 2014. Perpetual Data Collection with Energy-Harvesting Sensor Networks. ACM Trans. Sensor Netw. 11, 1, Article 12 (September 2014), 45 pages. DOI: http://dx.doi.org/10.1145/2566675

Perpetual Data Collection with Energy-Harvesting Sensor Networks

CHRISTIAN RENNER, University of Lübeck
STEFAN UNTERSCHÜTZ, Hamburg University of Technology
VOLKER TURAU, Hamburg University of Technology
KAY RÖMER, Graz University of Technology

A sustainable, uniform, and utility-maximizing operation of energy-harvesting sensor networks requires methods for aligning consumption with harvest. This paper presents a light-weight algorithm for online load adaptation of energy-harvesting sensor nodes using supercapacitors as energy buffers. The algorithm capitalizes on the elementary relationship between state of charge and voltage that is characteristic for supercapacitors. It is particularly designed to handle the non-linear system model, and it is lightweight enough to run on low-power sensor node hardware. We define two energy policies, evaluate their performance using real-world solar-harvesting traces, and analyze the influence of the supercapacitor's capacity and imprecisions in harvest forecasts. To show the practical merit of our algorithm, we devise a load adaptation scheme for multi-hop data-collection sensor networks and run a four-week field test. The results show that (i) choosing a duty cycle a priori is infeasible, (ii) our algorithm increases the achievable work load of a node when using forecasts, (iii) uniform and steady operation is achieved, and (iv) depletion can be prevented in most cases.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols; C.2.4 [**Computer-Communication Networks**]: Distributed Systems

General Terms: Algorithms, Design, Performance, Measurement, Performance, Reliability, Experimentation

Additional Key Words and Phrases: Energy management, duty-cycle adaptation, energy harvesting, network protocols, data collection

1. INTRODUCTION

Sensor networks have been around for more than a decade. From the early stages of this emerging technology, one critical aspect has accompanied its presence and triggered continuous research activity: the conservation of energy to prolong lifetime.

As with mobile devices and low-power electronic convenience products—such as home weather stations—battery capacity is limited. Despite the low-power hardware, this bottleneck is further tightened by the tiny dimensions of sensor nodes and their batteries, which are mandatory for achieving goals such as non-intrusive operation, space limitations, and low costs. With typical batteries, lifetime of a sensor node is restricted to a few days, unless energy-saving strategies are applied. The key term in this context is energy efficiency, which essentially translates to switching hardware components off when they are not needed or, when this is not possible, duty-cycling their operation. The latter usually implies a performance penalty, e.g., duty-cycling the radio decreases network performance in terms of latency and throughput. However, energy efficiency can merely prolong lifetime to a few months in most cases. To ensure the success of a sensor network application, reliable

Work supported by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) under contract TU 221/4-1.
Author's addresses: C. Renner, Institute of Computer Engineering, University of Lübeck; S. Unterschütz and V. Turau, Institute of Telematics, Hamburg University of Technology; K. Römer, Institute for Technical Informatics, Graz University of Technology.

DOI: http://dx.doi.org/10.1145/2566675

lifetime predictions prior to deployment are mandatory. This is a non-trivial task, because consumption within a network is influenced by inaccessible factors such as wireless connectivity—which constitutes a research area of its own. Improper or imprecise methods for battery state-of-charge assessment, battery self-discharge, and consumption variation of cheap sensor node hardware aggravate this problem. For low-power applications, lifetime assessment of sensor networks is hence error-afflicted and vague. Since battery capacities are not expected to rise in orders of magnitude within the near future, provided that larger-sized batteries are not an option, and since replacement of batteries is usually infeasible, research has embarked on a different strategy.

To meet lifetime requirements, or even achieve sustained operation of wireless sensor nodes, the potential of powering these devices with energy harvested from the environment has been explored. Here, various harvesting solutions can be employed, e.g., sunlight, radio frequency, wind, vibration, and temperature differences. Particularly sunlight is highly promising, since it is available in many deployment scenarios. Moreover, it produces a sufficient amount of energy to supply wireless sensor nodes, which draw currents between several microampere in their lowest power state and some milliampere in fully operational mode. To maintain the miniature size of the sensor nodes, the dimensions of the harvesting sources are yet restriced. As a result, the amount of produced energy is not large enough to dispense with energy-saving techniques such as radio duty-cycling. Morever, the amount of harvested energy is usually neither constant nor continuous. In the instance of solar harvesting, energy can only be harvested during the day, while its amount depends heavily on weather conditions, the time of the year, and the actual placement and orientation of the sensor node's solar cell. Moreover, hardware aging, dirt, and environmental changes may affect and decrease energy production over time. This issue has two major implications. Firstly, energy must be buffered in order to prevent nodes from temporal energy shortage and restricted periods of operation. Secondly, nodes must align their consumption to the available harvest online, because the latter varies from node to node, changes throughout node lifetime, and is difficult to determine prior to deployment. A prospective option for buffering energy are supercapacitors. They are environmentally-friendly, relatively small, can reliably supply a node for a few days, have a virtually unlimited number of charge-discharge cycles and long shelf life. Moreover, the stored energy can be estimated from their terminal voltage with low computational effort. This is indispensable for depletion-safe online load adaptation.

One driver for maximizing the utility of harvestable energy are economic reasons—e.g., to achieve the same work load with smaller and cheaper hardware. Another driver is to increase a node's utility (in the network) by reducing its use of energy-saving methods, hence relieving the imposed performance penalty and improving, e.g., routing performance in terms of latency and throughput. Here, greedy consumption is not an option: Many applications demand uniform and reliable node operation to guarantee constant sampling rates, achieve low latency, ensure connectivity in large-scale multi-hop networks, and prevent depletion. Several solutions to this problem have been presented. They usually require a linear system model or do not work well for low-capacity energy buffers. Thus, they are generally incompatible with many low-cost, energy-harvesting sensor nodes that solely use supercapacitors as energy buffer. Moreover, existing load adaptation algorithms tend to be rather intricate and overdeveloped when load adaptation boils down to finding the maximum supported duty cycle for each sensor node in a data-collection network. Firstly, this problem differs from traditional energy-aware scheduling, since a constant load—or uniform operation, respectively—suffices for steady data collection in absence of hard delay limits. Secondly, assigning individual loads to different times of a day may decrease predictability of sensor node behavior and lead to unsteady operation. For example, when nodes run a duty-cycled MAC protocol, the energy consumption caused by waiting for the receiver to wake up depends on the duty cycle of the receiver. If nodes change their duty cycles frequently, they affect the energy consumption of their neighbors, requiring these to subsequently adapt their duty cycles or provoking unexpected consumption, possibly leading to depletion. This might be overcome by exchanging information of duty-cycle schedules among neighboring nodes at the cost of dramatically increased communication overhead, energy expenditure, and protocol complexity. Besides, the degree of complexity

requires a thorough setup by the system maintainer or researcher, who is likely not to be a sensor node, computer science, or electrical engineering expert. Simple approaches are hence desirable for mass-market sensor network deployments.

1.1. Goal and Contributions

The goal of this paper is to enable perpetual and maintenance-free data collection for energy-harvesting sensor networks. Unlike most existing methods, we (i) focus algorithmic simplicity and manageability to support setup by non-experts, (ii) use local knowledge only to avoid communication and energy expenditure, and (iii) consider a non-linear energy-flow model.

Our approach is two-tiered. First, we devise a method for determing the maximum average current consumption of a sensor node, such that its energy reserve never falls below a critical threshold. Second, we show how the knowledge of the maximum average consumption can be utilized to set up and maintain the duty cycle of a low-power data collection protocol without additional communication. We evaluate the proposed concept through simulation and a real-world experiment.

In particular, we make the following contributions:

— Based on existing research, the requirements of an energy-harvesting sensor node are formulated for typical sensor node monitoring applications. A prototype using a solar cell as harvester and a supercapacitor as energy buffer is built, and an energy-flow model is derived. The practical usability and precision of this model has been evaluated and pointed out in our previous work [Renner et al. 2012a; Renner and Turau 2012b].
— We introduce the method of predictive load adaptation and present an algorithm to predict a node's future state of charge. It is based on the energy-flow model and particularly considers the non-linear charging and discharging characteristics of supercapacitors. We show that harvest forecasts improve harvest utility—i.e., the ratio of consumed vs. harvestable energy—while achieving uniform operation—i.e., steady node activity and availability for packet forwarding.
— To achieve depletion-safe operation and to maximize the amount of harvested energy, we devise the concept of energy policies. The latter define a set of energy reserve conditions that must be satisfied within the prediction horizon of the load adaptation algorithm. We analyze the impact of two energy policies w.r.t. supercapacitor capacities and harvest forecast errors.
— The developed concepts are finally implemented for real sensor node hardware and evaluated in a real-world case study. To show their practical relevance and feasibility, a simple yet effective scheme for practical load adaptation for low-power duty-cycling routing protocols—such as ORiNoCo [Unterschütz et al. 2012] and ORW [Landsiedel et al. 2012]—in multi-hop sensor networks is devised and analyzed. This scheme achieves automatic, energy-aware load balancing, while it is based on local reckoning and does hence not infer overhead in terms of data and status exchange in the network.

Hence, this article significantly contributes to the current state of research on energy-harvesting, supercapacitor-powered, and load-adapting sensor networks.

The remainder of this paper is organized as follows. In Sect. 2 we briefly present and discuss existing research. Sect. 3 presents the energy harvester prototype, the corresponding design considerations, and the energy-flow model. The concept of predictive load adaptation with energy policies is motivated and explained in detail in Sect. 4. Sect. 5 contains the evaluation. Our scheme for practical load adaptation is motivated and discussed in Sect. 6. Sect. 7 describes and evaluates the real-world case study. The paper is concluded in Sect. 8.

2. BACKGROUND AND RELATED WORK

This section presents related research in context of this paper. First, we review existing energy harvesters for sensor nodes as basis for designing the prototype in Sect. 3. Second, we investigate harvest forecasting techniques for use in our predictive load adaptation algorithm in Sects. 4 and 6. Third, we discuss load adaptation algorithms and point out their deficiencies.

Table I. Overview of energy-harvesting sensor node platforms in chronological order of publication

Platform	Energy Source	Energy Buffer	Specifics
Prometheus [Jiang et al. 2005]	solar (136 mW)	EDLC (22 F) and Li-polymer	direct charging, two-tiered energy storage
Heliomote [Raghunathan et al. 2005]	solar ($\sim$ 220 mW)	NiMH	static MPP direct charging
Everlast [Simjee and Chou 2006]	solar (450 mW)	EDLC (100 F)	open-circuit solar voltage MPPT
AmbiMax [Park and Chou 2006]	solar (400 mW), wind (500 mW)	Li-polymer and EDLCs (10, 22 F)	open-circuit solar voltage MPPT, rotor-speed-based MPPT for wind generator
Enviromote [Kyriatzis et al. 2007]	solar (2 400 mW)	NiMH	static max. power range, max. 500 mW usable charging power
Batteryless [Brunelli et al. 2009]	solar (84 mW)	EDLC (50 F)	open-circuit solar voltage MPPT with additional solar cell

2.1. Energy-Harvesting Sensor Nodes

Powering sensor nodes from ambient, regenerative energy sources opens the door to unattended, uninterrupted, and virtually unlimited operation. The process of converting ambient into electrical power is frequently referred to as energy harvesting or energy scavenging; both terms are interchangeable. A power supply based on regenerative energy sources is called harvester. The output of the harvester, e.g., power or current, is called harvest.

Since sensor nodes are low-power devices, energy harvesters rarely have to exceed the dimensions of the nodes. Tiny system size is thus maintained. Energy harvesting mitigates the problems of energy buffer self-discharge and capacity variation, since buffers are refilled timely. Several energy-harvesting power supplies for sensor nodes have been designed and presented in the last decade. They are introduced in this section, and Table I summarizes their main characteristics.

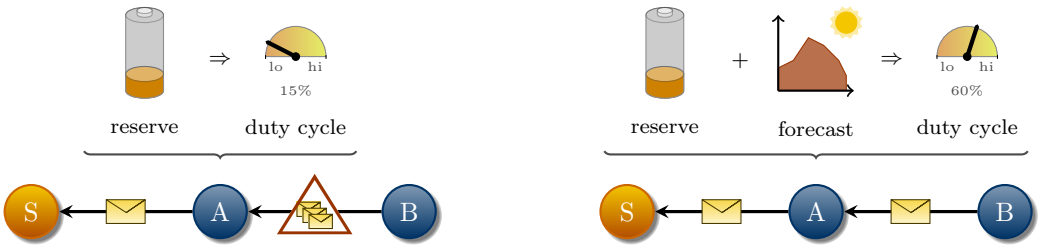
Prometheus [Jiang et al. 2005] uses a two-tiered energy storage system composed of a supercapacitor and a Li-polymer battery. The supercapacitor acts as primary energy buffer of the system to reduce the number of recharge cycles of the Li-polymer battery, which acts as a back-up energy buffer. The solar cell directly charges the supercapacitor. Near-perpetual operation is achieved by Heliomote [Raghunathan et al. 2005], a solar-powered sensor node. To prevent power loss caused by maximum-power-point tracking (MPPT) of the solar cell, the authors used a direct charging circuit. Two AA type NiMH batteries serve as energy buffer. Pointing out that batteries are the lifetime-limiting factor of energy-harvesting systems, the authors of the Everlast platform base their system on a supercapacitor as sole energy buffer [Simjee and Chou 2006]. A solar panel harvests energy from the environment with a software MPPT based on the open-circuit voltage of the solar cell.

A multi-source harvesting platform, called AmbiMax, is presented in [Park and Chou 2006]: It scavenges energy with a solar cell and a wind generator. AmbiMax makes use of supercapacitors to improve harvesting efficiency, but relies on a Li-polymer battery as energy buffer.

A solar cell serves as energy source and NiMH batteries buffer energy on the Enviromote platform [Kyriatzis et al. 2007]. To reduce system overhead, no MPPT is employed. Brunelli et al. presented a batteryless, i.e., supercapacitor-only, energy harvester for sensor nodes [Brunelli et al. 2009]. Their platform is solar-powered with a MPPT that employs the open-circuit voltage method, where an additional light sensor is used for MPPT.

2.2. Harvest Forecasting

A sensor node can increase its duty cycle or, in general, its activity—e.g., in terms of sensing rates or responsiveness—while keeping the risk of energy depletion at a low level, if the node is aware of its expected future energy harvest:



(a) reserve-only (b) harvest-aware

Fig. 1. Duty-cycle adaptation based on (a) energy reserve only and (b) an additional harvest forecast: The network traffic jam in (a) is prevented by considering the forecast in (b), because node A is enabled to maintain a higher radio duty cycle, since the low energy reserve does not pose a depletion-threat due to expected energy intake

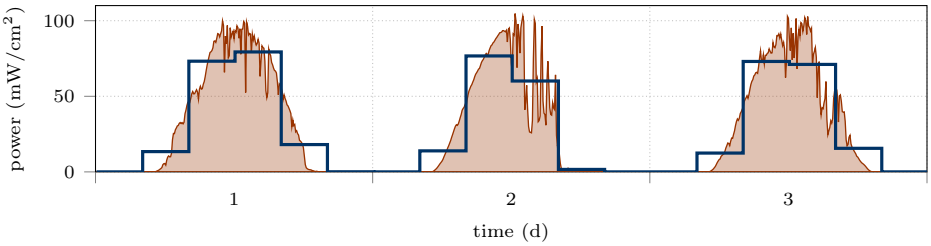


Fig. 2. Excerpt of a solar power density trace (red)—recorded at Elizabeth City State University (ECSU), NC, USA, [US Department of Energy 2011]—and its slot representation (blue) with six slots per day

— Throttling the radio duty cycle is unnecessary, if the energy buffer is close to depletion, but energy can be harvested in the near future. Figure 1 illustrates this point.
— Short periods of high harvest can be used for fast and high-volume data exchange or collection, because there is no need for radio duty-cycling.
— More stable routing paths can be established, if perspective energy harvest is integrated into path establishment.
— Delay-tolerant tasks can be deferred and sampling rates reduced, if energy reserves and expected harvest are low.

The actual energy intake varies among equally equipped sensor nodes due to hardware variation and aging, node placement, environmental impacts, seasonal changes, and dirt. In many outdoor scenarios, it is impossible to foresee the precise energy condition of each single node in a deployment a priori. Generating forecasts based on local historical data is a remedy to this problem. It is generally feasible, if energy harvest exhibits a pattern, as is the case, e.g., for outdoor solar harvesting. Figure 2 shows an excerpt of a real solar power density trace with diurnal harvest pattern.

Forecast techniques that exploit the quasi-cyclic harvest pattern have been introduced to wireless sensor nodes [Kansal et al. 2007; Recas Piorno et al. 2009; Ali et al. 2010; Bergonzini et al. 2010; Sharma et al. 2010]. Due to the restricted memory resources, the general idea behind these approaches is to divide a cycle into a relatively small number of time slots with equal length, e.g., 48 slots with a length of 30 min for solar harvesting. For each time slot, the mean value of harvested energy is calculated. To compensate for influences, such as seasonal effects and harvester aging, the newly calculated value of a slot and its historic data are smoothed by applying an exponentially weighted moving average (EWMA) filter. This filtering technique is frequently used in sensor network protocols to reduce the memory footprint: no historic data has to be stored. The smoothed values already serve as a harvest forecast; thus, no additional effort for creating a forecast is re-

quired [Kansal et al. 2007]. Figure 2 shows an example slot distribution. Here, the value stored for one time slot is used as harvest forecast for the same time slot on the next day.

The weather-conditioned moving average (WCMA) forecast algorithm [Recas Piorno et al. 2009; Bergonzini et al. 2010] was proposed to improve over simple EWMA filtering. The forecast for the directly following slot is refined by taking the harvest trend of the current slot and the last few days into consideration. WCMA has only been shown to elevate short-term forecasts of a single slot [Ali et al. 2010]. It has no proven advantage over EWMA-based forecasts of a complete pattern cycle (long-term forecast). We have carried out a detailled study about existing forecast techniques in [Renner and Turau 2012a].

Other approaches on sunlight and solar forecasting exist. Machine learning for predicting solar power was discussed in [Sharma et al. 2011]. SunCast predicts sunlight conditions through finding the most similar historical sunlight trace [Lu and Whitehouse 2012]. Currently, sensor nodes cannot profit from these promising results, since resource demands overshoot those offered by sensor nodes.

2.3. Online Load Adaptation

Load adaptation is an active research area in energy-harvesting sensor networks. This section motivates their necessity and introduces existing approaches.

Energy-Reserve-Only Approaches. Nano-RK [Eswaran et al. 2005] is a real-time operating system for sensor nodes. It supports preemptive multi-tasking and enforces task deadlines. Nano-RK can record the energy consumption of each task (with a software tracking system) to enable modifying execution periodicity and to meet lifetime goals. Energy-intake forecasts are not supported.

Vigorito et al. present a duty-cycle adaptation algorithm based on adaptive control theory for energy-harvesting sensor nodes [Vigorito et al. 2007]. They employ an optimal linear quadratic tracking solution, called LQ-Tracker, with the goal of achieving energy-neutral operation while minimizing the variance of a node's duty cycle. The optimal duty cycle is smoothed with an EWMA filter to enforce more uniform and predictable operation of each node in the network. The algorithm aims at keeping the battery's state of charge at near-constant level, so that this method is not compatible with supercapacitors. Harvest forecasts cannot be combined with the LQ-Tracker, either.

In [Zhu et al. 2009], the authors present an algorithm to adjust the consumption—or activity level, respectively—of an energy-harvesting sensor node that uses a supercapacitor as only energy buffer. The algorithm determines the maximum (average) current consumption, such that a minimum lifetime (time before depletion) is achieved. The output of the algorithm is a conservative estimate, since future harvest is not considered. Supercapacitor leakage is explicitly modeled.

Harvest-Aware Approaches. Assessing energy reserves exclusively is no optimal solution as explained in Sect. 2.2. Several solutions incorporating harvest forecasts have thus been devised.

Hsu et al. present a harvest-aware power management system [Hsu et al. 2006; Kansal et al. 2007] that combines energy-neutral operation with optimal system performance. The concept of energy neutrality defines that a node must not spend more energy than it harvests from the environment. The authors suggest using energy-intake forecasts based on historical data and EWMA-filtering. The authors formulate the optimal duty-cycle distribution (among time slots) as a linear program (LP) and present a low-complexity solution approximation that assigns higher duty cycles to slots with high energy intake. Uniform operation is hence not achieved.

Moser et al. investigate real-time scheduling in [Moser et al. 2007]. The authors devise a lazy scheduling algorithm that handles energy constraints and task deadlines. Results are obtained through simulation of a linear energy-flow system. In particular, the energy source is modeled to produce power that is only affected by environmental conditions. Although the authors discuss the potential use of supercapacitors in their system model, they do not consider non-linear effects on the energy production of the harvesting source.

A hierarchical control algorithm for achieving low-complexity duty-cycle adaptation is presented in [Moser et al. 2008; 2010]. Prediction of solar intake is split into one daily and one hourly estimator. The intention is to feed the results to two different controllers, of which the first is responsible for

achieving long-term operation while the second manages short-term adaptations. Both controllers solve a linear program (LP), where computation and resource complexity are reduced by splitting the problem into two sub-problems. However, the LP introduces notable computation complexity, while uniform node operation is not preserved.

In [Fan et al. 2008], Fan et al. address the problem of allocating individual sensing rates to solar-powered sensor nodes in a network. Their solution also considers the data flow in the network and the corresponding energy consumption but assumes a linear energy model. A centralized algorithm based on a linear program (LP) is presented. In addition, the authors propose a simplified, distributed algorithm without the need of solving an LP but with the constraint that the routing paths in the network must be known. Here, maximum rates are determined by an iterative process of sending control packets on each path from a leaf node to the sink.

Zhang et al. suggest a set of Harvesting Aware Speed Selection (HASS) algorithms for energy management in time-critical sensor networks [Zhang et al. 2010; 2011]. The algorithms aim at maximizing the minimum energy reserve of a single node in the network to ensure operability in emergency situations, when load is high and latency must be low. A linear program is formulated, for which a centralized (optimal) and a distributed (approximate) solution algorithm are derived.

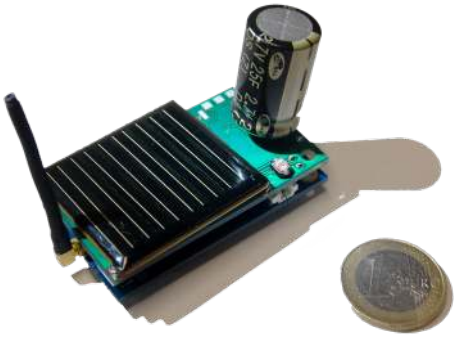
Conclusion. Energy-aware load adaptation is an active research field. To maximize the utility of a sensor node in terms of using as much harvestable energy as possible, linear programs have been formulated, and simplified algorithms, meeting the resource constraints of sensor node hardware, have been proposed. While guaranteeing a high utility in case of perfectly predictable harvest, these solutions are questionable in real-world deployments, in which energy intake is partly predictable only and with large errors. Moreover, steady and reliable operation—i.e., guaranteeing gapless sensed data and continuous network operations—is favorable in traditional data-collection sensor networks, the predominant use case as of today: Sacrificing utility, e.g., in terms of network responsiveness or delay, prevents loss of data caused by energy depletion. Squeezing the last drop of utility may lead to accidental and unexpected energy shortage. Steady network operation also improves the predictability of network behavior, since duty cycles and routing paths require less frequent changes.

Most algorithms are still computationally complex and overdeveloped for data-collection sensor networks. The majority of the presented algorithms have been evaluated through simulation only. It is unclear whether the simplified, linear system models used for system modeling are compatible with supercapacitor-powered sensor nodes. There is a lack of simple and practical algorithms for duty cycle adaptation in energy-harvesting sensor networks using supercapacitors as only energy buffer. This lack is particularly unsatisfactory, since supercapacitors keep drawing attention due to their advantages over alternative energy buffers.

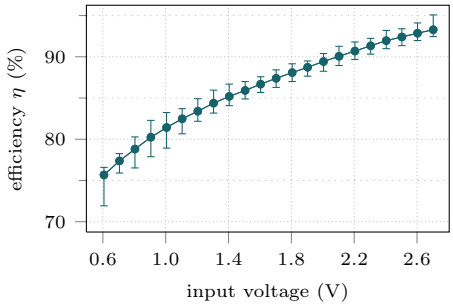
3. ENERGY-HARVESTING PROTOTYPE

The desire for perpetually operating sensor nodes has lead to power supplies based on renewable energy sources. The review of existing solutions in Sect. 2.1 produces the following insights:

— Solar energy is the most frequently used harvesting source. It meets the end of application-agnostic, universal, and ubiquitous outdoor monitoring. Self-sustained operation of a sensor node is supported by a relatively large amount and predictable pattern of energy intake.
— Supercapacitors offer a wide range of advantages. Particularly, they promise an outstanding ease of state-of-charge assessment and come at capacities large enough to supply sensor nodes for several days without recharge. They do not require a complex charging circuitry and offer a shelf life of ten years and more.
— Improving harvesting efficiency with maximum-power-point tracking increases circuit complexity and decreases system predictability, the core prerequisite for system modeling and algorithm evaluation.



(a) energy-harvesting sensor node



(b) regulator efficiency

Fig. 3. (a) Energy-harvesting power supply with 25 F supercapacitor and solar cell mounted on top of an Iris node with removed battery pack, and (b) efficiency of the regulator (mean and extreme values) for typical sensor node consumption profile (30 µA − 18 mA)

Hence, a solar-harvesting power supply using a supercapacitor as its energy buffer is a generic, representative prototype, that is well-suited for a wide range of sensor network applications. Moreover, it satisfies the requirements of energy-aware duty-cycle adaptation in sensor network applications as introduced in Sect. 2.3, because it particularly permits the assessment of the energy flow. This will become evident in the following. An evaluation based on such a prototype promises research results with a wide scope and practical merit. To perform a real-world analysis, we hence built a solar-harvester prototype with a supercapacitor as energy buffer.

3.1. Hardware

The hardware realization of the energy-harvesting power supply, referred to as harvester throughout the remainder of this paper, is introduced in the following. A fully functional energy-harvesting sensor node with the prototype harvester is displayed in Fig. 3a. An equivalent circuit is presented in Sect. 3.2.1. The prototype board is designed to comply with the dimensions of the Iris sensor node and its extension connector, but it is generally usable with any sensor node hardware.

A solar cell with a size of $39 \times 35\,\mathrm{mm}^2$ serves as harvesting source. It has a maximum working voltage of $4\,\mathrm{V}$ at a maximum working current of $35\,\mathrm{mA}$ [Conrad Electronics 2008]. Direct charging is preferred over maximum-power-point tracking (MPPT) to reduce circuit and system-model complexity, costs, size, and energy consumption of the charging circuitry. The current produced by the solar cell is almost unaffected by the cell's terminal voltage (according to the datasheet), so that equal intake can be expected under the same environmental circumstances. Forecasts of the intake are therefore enabled. Current backflow is prevented with a diode. The harvester provides a sensor for measuring the current produced by the solar cell.

The harvester is designed for supercapacitors with a maximum rated voltage of $V_{\max} = 2.7\,\mathrm{V}$. Supercapacitors with 25 to $100\,\mathrm{F}$ give a good trade-off between size, capacity, and price; e.g., a $50\,\mathrm{F}$ supercapacitor can operate an Iris node at a 1% radio duty cycle for more than two days without harvesting energy. The supercapacitor is connected to one of the node's ADC ports, so that its voltage can be measured by the sensor node. To protect the supercapacitor from overcharging, the harvester automatically disconnects the supercapacitor from the solar cell using a switch S_h, if its voltage exceeds $V_{\max}$. In this case, it is still possible to measure the current produced by the solar cell with the current sensor.

Since the voltage range of the supercapacitor is below the minimum required voltage of the Iris sensor node, a Texas Instruments TPS 61220 switching regulator [Texas Instruments 2009] is used to produce a constant output voltage of $V_\mathrm{n} = 2.7\,\mathrm{V}$. It has a measured minimum input voltage of $V_\mathrm{cut} = 0.5\,\mathrm{V}$ (the cut-off voltage); i.e., the regulator stops operation when $V_\mathrm{c} < V_\mathrm{cut}$. We call this instance depletion. After depletion, the regulator is kept disabled until V_c reaches the start-up voltage

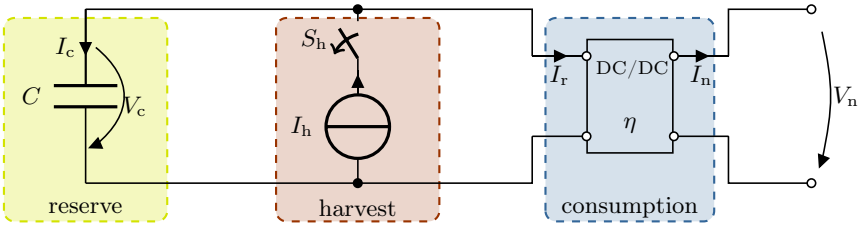


Fig. 4. Simplified, equivalent circuit serving as basis for system model equations

$V_{on} = 1.6\,V$ for the first time. The intention of this hysteresis is to avoid a node from frequently resetting after depletion when $V_c \approx V_{cut}$. Figure 3b portrays the overall conversion efficiency η for the prototype. Efficiency ranges from 72% to 95% (with an average of 86%) and depends on the input voltage, while the impact of the output current is negligible.

The prototype allows for estimating the actual capacity of the supercapacitor to cope with, e.g., manufacturing variation and aging. Details are omitted due to space constraints but are explained in [Renner and Turau 2010].

3.2. Energy-Flow Model

In the following, a simplified circuit of the harvester prototype is presented, from which an energy-flow model is derived.

3.2.1. Simplified, Equivalent Circuit. Figure 4 shows the simplified circuit that describes the fundamental energy flows of the hardware. While the presented circuit is derived from the prototype introduced in Sect. 3.1, it is generally applicable to any harvesting sensor node architecture with the following characteristics:

— The harvester produces a direct current (DC) that only depends on harvesting (i.e., environmental) conditions. There is, in particular, a negligible dependency of the produced current and the terminal voltage. This constraint is met by most direct charging circuits using solar cells.
— A supercapacitor serves as energy buffer.
— A DC/DC converter converts input to output power with a known efficiency.

The current $I_h \geq 0$ (the harvest) is produced by the solar cell, and the regulator current $I_r > 0$ is consumed by the regulator to supply the sensor node with $I_n > 0$ (the load or consumption) at the constant output voltage V_n. The current I_c flows into the supercapacitor with capacity C. If I_c is positive, harvest exceeds consumption and the supercapacitor is charged; otherwise, the supercapacitor acts as source and discharges to supply the sensor node. V_c is the voltage of the supercapacitor. Note that due to overcharging protection, V_c cannot exceed V_{max} and that the regulator will fail, if V_c underruns the cut-off voltage V_{cut}. The power-conversion efficiency η of the regulator is a function of I_n and V_c according to Sect. 3.1.

3.2.2. Analytic System Model. The current flowing into the buffer (supercapacitor) is the difference of the current harvested by the source (solar cell) and the current consumption of the regulator (to supply the sensor node):

$$I_c = I_h - I_r \,. \tag{1}$$

The regulator converts input power with efficiency η, so that

$$I_r \cdot V_c \cdot \eta(V_c, I_n) = I_n \cdot V_n \,. \tag{2}$$

Motivated by previous research results in [Renner et al. 2012a; Renner and Turau 2012b], we use an ideal capacitor model:

$$I_\mathrm{c} = C \cdot \dot{V}_\mathrm{c}\,. \tag{3}$$

The only parameter in this model is the capacity C. More accurate models of the charging and discharging behavior of supercapacitors are based on multi-layer cascade circuits of ideal capacities and resistors (cf. [Weddell et al. 2011]), and a piecewise linear model of leakage was presented in [Zhu et al. 2009]. While these models are completely suitable for simulating realistic voltage courses of supercapacitors, they require multi-parameter estimation not desirable for sensor node hardware. Moreover, it is not clear whether these parameters could be determined with the required precision on sensor node hardware.

Combing Eqs. (1) to (3) yields the final system model for normal operation:

$$C \cdot \dot{V}_\mathrm{c} \;=\; I_\mathrm{h} \;-\; \frac{I_\mathrm{n} \cdot V_\mathrm{n}}{V_\mathrm{c} \cdot \eta(V_\mathrm{c}, I_\mathrm{n})}\,. \tag{4}$$

This model is the foundation for predictive load adaptation.

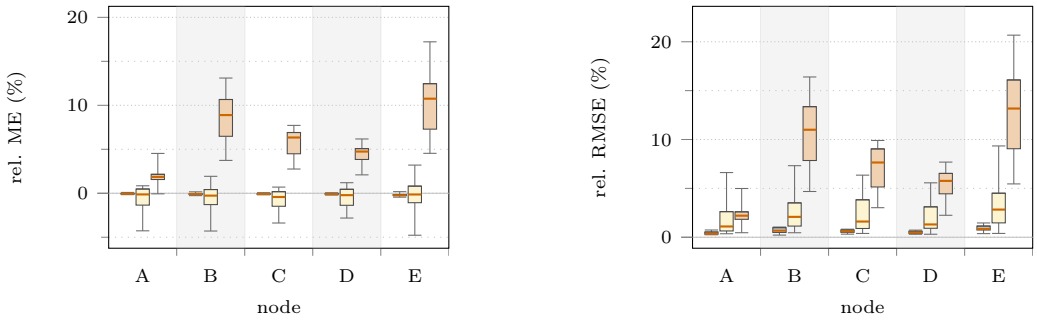
3.3. Validation of Model Appropriateness

Even though we already used the assumption of an ideal capacitor model to derive a manageable form for the energy-flow model, solving Eq. (4) analytically is infeasible due to the unknown functions I_n and I_h. A fine-grained numeric evaluation is possible yet computationally impracticable and futile. The precise temporal course of I_n and I_h would be required to achieve accurate results, whereas precision limitations—e.g., caused by consumption variation and imprecise harvest forecasts—would lead to error-prone results. To meet this end we will additionally assume a constant power-conversion efficiency η. As will be seen in Sect. 4.3, these assumptions will allow for an analytical approach for solving Eq. (4). In this section we present justifications for these assumptions. In particular, we evaluate the impact of two assumptions, namely (i) the ideal capacitor model and (ii) a constant power-conversion efficiency η. For a more detailed study, we point the reader to [Renner et al. 2012a; Renner and Turau 2012b].

We conducted an experiment with five sensor nodes equipped with the harvester with supercapacitors of 25 and 50 F nominal capacity. We used Samwha Green-Caps from the DS series with a 2.7 V-rating. All nodes ran four discharge cycles in the voltage range from 2.6 V to 1.0 V for five radio duty cycles ϕ ranging from 0.1% to 20%. The solar cell was disconnected. Supercapacitor voltage V_c was measured with a periodicity of 30 s. In total, we recorded 20 traces for each node.

To compare the recorded traces with our model, we calculated voltage courses through Eq. (4) with the following parameter sets: For each discharging trace, we derived the empirical capacity C by minimizing the root-mean-square error (RMSE). We used the constant efficiency $\eta = 86\%$, which is the average value in the corresponding voltage range (cf. Fig. 3b). We set $I_\mathrm{h} = 0\,\mathrm{mA}$, and we used a constant consumption I_n corresponding to the duty cycle ϕ. Here, we modeled the node being active (radio enabled) for the percentile ϕ of the time and sleeping otherwise, cf. [Renner et al. 2012a].

The relative root-mean-square error (RMSE) and mean error (ME) of three variants of discharging models versus the recorded traces are displayed in Fig. 5. The first model uses optimized values of capacity C for each individual trace to identify the smallest possible model error. The RMSE does not exceed 2% in this case, and the ME stays below 1%. The second model uses a single value of capacity C—one for each node—that was obtained by averaging the optimal capacities. In the worst case, the RMSE has a median of 2% and always remains below 5%. While the ME is unbiased—i.e., errors with negative and positive signs cancel out—its deviation is as small as 5%. The third model uses the nominal capacity (printed on the supercapacitors), giving an RMSE of up to 22% and medians of 3 to 13%. The distribution of errors (both RMSE and ME) varies notably among supercapacitors due to age, usage, and manufacturing deviation.



(a) relative ME

(b) relative RMSE

Fig. 5. Discharging model errors using capacity values optimized for each discharge curve (blue/left), a single estimated capacity value (yellow/center), and the nominal capacity (orange/right). The box plots show median, quartiles, and extreme values using all discharging curves at a granularity of 30 s

The evaluation shows that the proposed model—with the assumptions of a constant efficiency and ideal supercapacitor behavior—works in the field. Errors may be as large as 5%, but this number is small when considering that sensor nodes have (slightly) different consumption, yet only the mean consumption among all nodes was used for evaluation, and although a constant regulator efficiency was employed. Despite different charging currents and duration, voltage range, and capacity, all discharging traces exhibit an almost equal shape with high correlation between modeled behavior and recorded trace.

4. PREDICTIVE ONLINE LOAD ADAPTATION

Next, we propose and explain a new method for load adaptation in data-collecting sensor networks. First, we motivate the need for online load adapation in Sect. 4.1. It follows a description of the general approach and its components in Sect. 4.2. Finally, we give technical details in Sects. 4.3 and 4.4.

4.1. Motivation and Objective

While energy harvesting offers perpetual and maintenance-free operation of sensor nodes, it demands aligning a node's consumption with the harvest. As discussed in Sect. 2.3, this problem is non-trivial and has been addressed by many researchers. Choosing correct system parameters, such as the radio duty cycle, prior to deployment is dicey due to uncertain and frequently misjudged harvesting conditions.

In most multi-hop, delay-tolerant, sense-and-send sensor network applications, combining a low-power MAC protocol with tree routing guarantees easy implementation with off-the-shelf sensor network protocols. Here, opportunistic and greedy consumption (i.e., take what you get) is not an option. Many harvesters, e.g., solar cells, show an unsteady and location-dependent harvest pattern. Parts of the network may thus not be able to communicate—e.g., nodes on different sides of a building may only have non-overlapping periods of harvest. In addition, changing weather conditions will lead to unstable duty cycles. In contrast, uniform operation—i.e., predictable network behavior due to infrequent and small changes of duty cycles—proves beneficial, since a node's consumption is not only determined by its own duty cycle but is also affected by that of its parents in a routing tree (cf. Sect. 6.4).

Therefore, a node should aim at achieving an enduring, constant duty cycle with an energy consumption meeting two contradicting demands. On the one hand, energy consumption must be low enough to guarantee depletion safety; on the other hand, a node should wake up as frequently as possible—leading to a higher duty cycle and energy consumption—to prevent child nodes from high energy expenditure due to long waiting times. For this purpose, a node must combine its knowledge

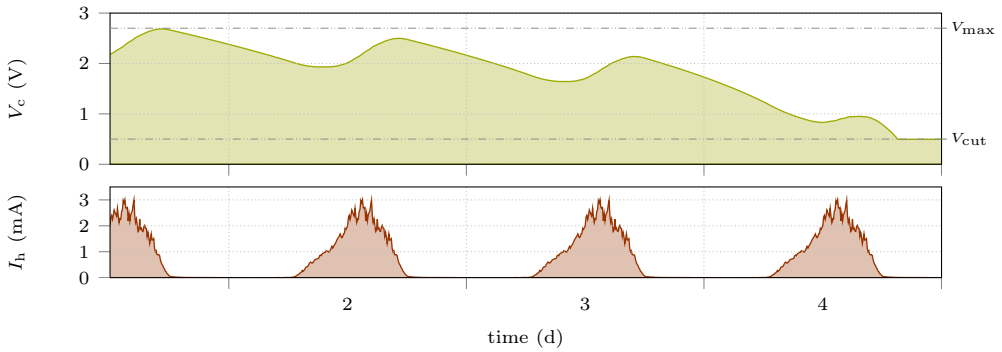


Fig. 6. Simulated development of supercapacitor voltage V_c (upper plot) in a fully predictable solar environment with constant consumption $I_n = 0.65\,\text{mA}$ equal to the mean value of harvester current I_h in a day. Capacity is $C = 50\,\text{F}$ and the regulator efficiency is $\eta = 100\%$

about the present energy state with a forecast of the future harvest. We aim at providing a simple yet effective method for this purpose. The method is simple due to dispensing with the formulation of a linear program—an approach fancied in most existing solutions, cf. Sect. 2.3. Simplicity reduces computational complexity, while it is less prone to producing overfitted results w.r.t. uncertain harvest and the resulting imprecise forecasts.

The proposed method is particularly designed to suit energy harvesters with a non-linear energy-flow model, an aspect frequently ignored in the literature. In this context, linearity means that the energy reserve after a time interval is a linear function of energy consumption and energy harvest. In particular, consumption and harvest are independent of each other in this case. This implies that different (power) consumption functions with the same integral value (in the same time interval) lead to the same change in remaining energy. Finding the maximum average power consumption can thus be performed with a small set of linear equations as in [Kansal et al. 2007].

Supercapacitor-powered, energy-harvesting sensor nodes generally exhibit a non-linear energy flow (cf. Sect. 3.2). In particular, harvest may depend on the energy reserve. In case of the harvester prototype, the solar current is only affected by the harvesting conditions while the produced power linearly depends on supercapacitor voltage V_c; such a dependency is expected for similar harvesters as discussed in Sect. 3.2.1. As a result, consumption does not only affect the state of charge at a later point, but it also influences the power production in the future. We illustrate this issue with an example consisting of two cases: In the first case, the sensor node's current consumption is chosen to perfectly match the (known) harvest, i.e., $I_n(t) = I_h(t)$. Assuming $\eta = 1$ for simplicity and an initial value of $V_c = V_n$ (note that $V_{max} = V_n$ in our hardware setup), Eq. (4) yields $C \cdot \dot{V}_c = 0$. Supercapacitor voltage V_c is hence constant and we achieve perpetual operation. In the second case, depicted in Fig. 6, current consumption is constant with the same integral value (energy consumption). At night, this leads to $C \cdot \dot{V}_c < 0$, so that V_c falls. As a result, the power $V_c \cdot I_h$ of the solar cell is lower than in the first case, so that depletion occurs on the fourth day ($V_c \to V_{cut}$). The example hence shows that finding a maximal and depletion-safe, uniform consumption is a non-trivial task. In the following, we derive a lightweight solution to this problem.

4.2. General Approach

In the following, we sketch our approach to online load adaptation. A detailed and technical explanation is subsequently given in this section.

The abstract goal of our load adaptation algorithm is to determine the maximum average current consumption I_n^* of a sensor node, such that its energy reserve never falls below a user-defined threshold. Here, the term load is an abstract placeholder for what the sensor node does, and we assume that this load is linked (e.g., through application parameters) to the consumption I_n. We

hence use both terms interchangeably. In [Renner et al. 2012a] we have shown that it suffices if the consumption I_n is periodic with a period of up to a few minutes and a mean current consumption $\bar{I}_\mathrm{n}$ equal to I_n^*.

The choice of the threshold and the procedure of turning the average current consumption into system parameters are application-specific. For the example of a multi-hop sensor network using a low-power MAC, the average current consumption can be translated into a radio duty cycle. We propose such a concept in Sect. 6 and evaluate it in Sect. 7. For the same example, the threshold of the energy-reserve could be chosen to ensure that a node may continue operation for a given amount of time that enables all its neighbors to identify a different routing path.

To achieve its goal, our algorithm uses the energy-flow model presented in Sect. 3.2 to produce I_n^* based on forecasts of the harvest (cf. Sect. 2.2) and the state of the energy buffer (in terms of the supercapacitor's voltage V_c) at the time of algorithm execution. In addition, the algorithm has to consider the (unknown) future course of V_c, e.g., to ensure that the energy reserve (i.e., supercapacitor voltage) never falls below the user-defined threshold. To install a more general concept here—there is more than one way to look at the course of V_c, cf. Sect. 4.4.1—we introduce *energy policies*. The latter is a set of rules that the future course of V_c must fulfill.

Unfortunately, it is not possible to obtain I_n^* directly from the functional equation defined by Eq. (3). It is however possible to derive the supercapacitor voltage V_c at the end of a time interval with a given harvest I_h and a given consumption I_n. Our algorithm uses this knowledge as follows:

(1) For the harvest forecast I_h (represented by a time series) of a time interval, the corresponding future time series of V_c is calculated for a given average consumption $I_\mathrm{n} = \bar{I}_\mathrm{n}$ (see last step). To achieve low-complexity computation and a small memory footprint, the concept of time slots from Sect. 2.2 is adopted for harvest forecasts and the timing of algorithm execution. Details are provided in Sect. 4.3.
(2) The time series of V_c is checked for compliance with the energy policy. If the policy is violated, the value of $\bar{I}_\mathrm{n}$ is too high and considered as an upper bound. Otherwise, $\bar{I}_\mathrm{n}$ is considered as a lower bound. Details about energy policies are provided in Sect. 4.4.1.
(3) A binary search on consumption $\bar{I}_\mathrm{n}$ is employed to identify the maximum load I_n^* complying with the energy policy. For each value considered during the search, the node's future energy course is simulated (predicted) using a harvest forecast (step 1), and it is tested for policy compliance (step 2). This method is called *predictive load adaptation with energy policies*. Finally, the maximum load can be used to derive a node's radio duty cycle. Such an approach is discussed in Sect. 6.

4.3. Simulating and Predicting Supercapacitor Voltage

This section introduces the method for simulating and predicting a node's future energy course in terms of its supercapacitor voltage V_c. A formal, mathematical solution is presented and a practical realization using slot-based harvest forecasts, as discussed in Sect. 2.2, is derived. The general validity of such an approach is backed by the evaluation Sect. 3.3 and [Renner et al. 2012a].

4.3.1. Simulating a Node's Energy Course. The energy-flow model in Eq. (4) from Sect. 3.2 is an ordinary differential equation of supercapacitor voltage V_c. Its solution gives access to the energy course in terms of $V_\mathrm{c}(t)$ for a known current consumption $I_\mathrm{n}(t)$ and harvest $I_\mathrm{h}(t)$. Solving Eq. (4) analytically is infeasible due to the unknown functions $I_\mathrm{n}(t)$ and $I_\mathrm{h}(t)$ and because the hardware characteristics imply additional constraints: The overcharging protection prevents V_c from exceeding V_max, and the switching regulator fails if $V_\mathrm{c} < V_\mathrm{cut}$ (cf. Sect. 3.1).

For these reasons, solving the differential equation is simplified—or rather enabled—by assuming that I_h and I_n are piecewise constant functions. This simplification matches the actual consumption of a sensor node, which consists of discrete consumption states. It also complies with the knowledge about the harvest, for which only discrete samples exist. These samples can be converted into a piecewise constant function of I_h. In addition, we assume a constant regulator efficiency η; Fig. 3b shows that this assumption holds for small time intervals and, therefore, small changes of V_c. How-

ever, such a degree of detail is not even required: Using the average consumption of a node and the average harvest in time intervals of up to a few minutes usually suffices, as we have shown in [Renner et al. 2012a]. The results in [Renner and Turau 2012b] indicate that the error of using a constant η is low and can be neglected.

For $I_\mathrm{h} = \mathrm{const.}$, $\eta = \mathrm{const.}$, $I_\mathrm{n} = \mathrm{const.}$, Eq. (4) reduces to the ordinary, first-order differential equation (ODE)

$$\dot{y} = b - \frac{a}{y} \qquad \left(y = V_\mathrm{c}\,,\ a = \frac{V_\mathrm{n} \cdot I_\mathrm{n}}{\eta \cdot C} \geq 0\,,\ b = \frac{I_\mathrm{h}}{C} \geq 0 \right). \tag{5}$$

This equation has the implicit solution

$$0 = y - y_0 + \frac{a}{b} \cdot \log\left(\frac{a - b \cdot y}{a - b \cdot y_0} \right) - b \cdot \Delta t = f(y, y_0, \Delta t)\,, \tag{6}$$

in which the overcharging protection and regulator cut-off voltage are neglected for the moment. Here, $y = y(t)$ and $y_0 = y(t_0)$ with $t \geq t_0$ are points in time with $\Delta t = t - t_0$. This equation can be solved with, e.g., Newton's Method:

$$y_{n+1} = y_n - \frac{f(y_n, y_0, \Delta t)}{f'(y_n, y_0, \Delta t)} = g(y_n, y_0, \Delta t) \tag{7}$$

for a given number of iteration steps or until an absolute or relative error ϵ is achieved. Due to the overcharging protection, the final result of the iteration must be reduced to V_max, if it exceeds this value. Moreover, the case $V_\mathrm{c} < V_\mathrm{cut}$ has to be handled separately because it is equivalent to node depletion.

In two cases Eq. (6) cannot be used due to mathematical and numerical reasons:

(1) If $I_\mathrm{h} = 0$ and thus $b = 0$, the explicit solution $y = \sqrt{y_0^2 - 2a \cdot \Delta t}$ must be used.
(2) In steady state, i.e., $y_0 = \frac{a}{b}$, the solution is $\dot{y} = 0 \Rightarrow y = y_0$.

4.3.2. Online Prediction of the Energy Future. The approach used for simulation is turned into an algorithm that generates a prediction of the future supercapacitor voltage. For this purpose, a forecast of the harvest I_h is required. This end is met by the time slot method introduced in Sect. 2.2. Such a forecast provides the expected average harvest $\hat{\mu}_\mathbf{s}$ for a time slot $\mathbf{s}$ of length $\ell_\mathbf{s}$. For the prototype harvester, $\hat{\mu}_\mathbf{s} = \bar{I}_\mathrm{h}$ is the expected harvest in milliampere. The expected intake does (by definition of the forecasts) not change within a slot.

The consumption of a duty-cycled node is uniform in time intervals of several minutes or hours: Firstly, wake-up intervals are usually in the range of some hundred milliseconds. Secondly, asymmetric events, such as waiting for the receiver's wake-up for data transmission, are rare for low sampling and packet rates. Therefore, node consumption is modeled by its average load $\bar{I}_\mathrm{n}$. This simplification additionally reduces computation complexity, since there is no need to determine intermediate voltages (within a slot).

All parameters are available for determining the expected voltage at the end of a time slot $\mathbf{s}$ by solving Eq. (6) with Eq. (7). By considering a series $\mathcal{S}$ of slots, this method is able to predict a voltage course for the available forecast horizon. Here, the predicted voltage after one slot is used as initial voltage of the following slot, giving an iterative solution. In general, updates of this voltage course have to be calculated only if a new forecast is available, i.e., if a slot elapses.

4.4. Policy-Based Uniform Load Maximization

This section introduces the concept of energy policies and presents an algorithm to identify the maximum constant load satisfying a given policy. Here, the approach is to use policies as a heuristic for finding the maximum constant load.

4.4.1. Energy Policies. An energy policy defines a set of rules that the future course of energy reserve (i.e., supercapacitor voltage V_c) has to fulfill—e.g., the voltage should stay above a threshold.

However, the algorithm for predicting V_c in Sect. 4.3.2 produces a time series $\mathcal{V} = \langle\, v_0, \ldots, v_{|\mathcal{V}|-1} \,\rangle$ of supercapacitor voltages rather than a function description. Therefore, we define an energy policy $\mathcal{P}$ as a conjunction of predicates (rules) that is evaluated for a time series $\mathcal{V}$ of supercapacitor voltages. Since $\mathcal{V}$ has been calculated based on an average sensor node consumption $\bar{I}_\mathrm{n}$, evaluating a policy $\mathcal{P}$ answers the question whether $\bar{I}_\mathrm{n}$ complies with $\mathcal{P}$—e.g., answers the question whether $\bar{I}_\mathrm{n}$ will ensure that V_c stays above a threshold.

In the following, we introduce and explain two policies. The first aims at preventing depletion by keeping V_c above a user-defined, critical threshold. The second tries to additionally improve the final value of I_n^* by keeping the harvester in a better power point.

Depletion Safety. Achieving perpetual operation implies depletion-safe operation, hence the voltage V_c of the supercapacitor must not fall below the cut-off voltage V_cut. Due to uncertain energy intake in the future, it appears useful to define a critical voltage $V_\mathrm{crit} \geq V_\mathrm{cut}$ that must never be undercut, so that the policy guarantees backup energy. The corresponding depletion-safe policy is formally defined by

$$\mathcal{P}_\mathrm{DS}(\mathcal{V}) := \ \forall v \in \mathcal{V} \ : \ v \geq V_\mathrm{crit}\,. \tag{8}$$

Maximum Power Point. The harvester introduced in Sect. 3.1 operates at a better harvesting power point for higher values of V_c—for given values of I_h and I_n, the difference in Eq. (4) mainly depends on the value of V_c: the larger V_c, the lower the current subtracted from I_h. If the difference is positive, this means that the supercapacitor is charged quicker. If the difference is negative, this means that the supercapacitor is discharged slower. This observation is closely related to the example in Sect. 4.1.

For our hardware, the maximum power point is at $V_\mathrm{c} = V_\mathrm{max}$, so that it is desirable to operate the supercapacitor at this voltage. Requiring a high voltage V_c at all times yet implies a low load I_n: Keeping V_c close to V_max at times of low or no harvest (e.g., at night) demands near-zero consumption. To achieve uniform operation, the idea is to formulate a policy ensuring that V_c reaches V_max (at least) once a day:

$$\mathcal{P}_\mathrm{MPP}(\mathcal{V}) := \ \exists v \in \mathcal{V} \ : \ v = V_\mathrm{mpp} \ \wedge \ \mathcal{P}_\mathrm{DS}(\mathcal{V})\,, \tag{9}$$

where the inclusion of $\mathcal{P}_\mathrm{DS}$ is mandatory to prevent depletion. For the prototype harvester, $V_\mathrm{mpp} = V_\mathrm{max}$ is the natural choice.

4.4.2. Maximum Policy-Compliant Load. Finding the maximum load I_n^* complying with an energy policy $\mathcal{P}$ is achieved by predicting a node's voltage course. Here, I_n^* is constant w.r.t. the prediction horizon in order to achieve stable, uniform operation. The maximum average consumption of the sensor node is defined by the value of I_n^*; the actual consumption should be symmetric and uniform with an average of $I_\mathrm{n}^* = \bar{I}_\mathrm{n}$ (cf. Sect. 4.2).

The algorithm FINDMAXLOAD in Fig. 7 uses binary search to find the maximum load I_n^* complying with policy $\mathcal{P}$. Starting with the current supercapacitor voltage V_c, the algorithm calculates the series of intermediate voltages $\mathcal{V}$ via SIMVC, where the individual elements $v \in \mathcal{V}$ are the intermediate voltages at the beginning (or ends, respectively) of time slots. The lengths and expected harvest of the time slots are defined by the forecast in its slot representation $\mathcal{S}$. The series $\mathcal{V}$ is finally checked for compliance with $\mathcal{P}$ and the bounds of the search are adjusted. The maximum number of iterations is limited by an absolute tolerance tol, so that the binary search has $\log_2\left(I_\mathrm{n,max}/\mathrm{tol}\right)$ iteration steps. Here, $I_\mathrm{n,max}$ is the maximum allowed consumption.

The algorithm SIMVC implements the functionality explained in Sect. 4.3. Iterative approximation of V_c at the end of a time slot is repeated until the absolute error falls below ϵ or a maximum number of iteration steps N has been executed.

The optimality of the algorithm requires the optimality (applicability) of the binary search. This is the case, if the policy is violated for all loads larger than I_n^*, and if all loads smaller than I_n^* comply with the policy. The two presented policies fulfill this requirement. The sketch of the proof is as follows. First, we observe that a lower consumption I_n cannot lead to a smaller voltage V_c, cf. Eqs. (4) and (5). For $\mathcal{P}_\mathrm{DS}$ we can then show that if a load (consumption) complies with $\mathcal{P}_\mathrm{DS}$ (i.e.,

```
function FINDMAXLOAD(V_0, S, P)
    I_n^- ← 0
    I_n^+ ← I_n,max
    repeat
        I_n^* ← (I_n^- + I_n^+)/2
        v ← V_0
        V ← ⟨ v ⟩
        for all s ∈ S do
            v ← SIMVC(v, ℓ_s, μ̂_s, I_n^*)
            V ← V ∪ ⟨ v ⟩
        end for
        if P(V) then
            I_n^- ← I_n^*
        else
            I_n^+ ← I_n^*
        end if
    until I_n^+ − I_n^- ≤ tol
    return I_n^*
end function
```

```
function SIMVC(V_0, Δt, I_h, I_n)
    if V_0 < V_cut then
        I_n ← 0
    end if
    a ← (V_n · I_n)/(C · η)
    b ← I_h/C
    y ← V_0
    if I_h ≪ I_n then
        y ← √(y^2 − 2a · Δt)
    else if y ≠ a/b then
        j ← 0
        repeat
            j ← j + 1
            y_old ← y
            y ← g(y, V_0, Δt)
        until |y_old − y| < ϵ or j ≥ N
        y ← min(y, V_max)
    end if
    return y
end function
```

Fig. 7. (left) Algorithm for finding the maximum supported load I_n^* complying with policy $\mathcal{P}$ and (right) algorithm for simulating the supercapacitor voltage for known harvest and consumption

no intermediate voltage is smaller than V_crit), any smaller load also complies. For $\mathcal{P}_\mathrm{MPP}$ we must additionally show that if a load produces at least one intermediate voltage with value V_mpp, this is also true for any smaller load.

4.4.3. Estimating the Critical Voltage. Ensuring depletion-safe operation is a major driver behind the policy concept. Achieving this goal is sensitive to the choice of V_crit, since it defines the amount of energy reserved to absorb energy-intake deficits due to erroneous harvest predictions. For this reason, we derive an equation to choose V_crit properly.

The risk of depletion is particularly high, when $V_\mathrm{c} = V_\mathrm{crit}$ at the beginning of a slot—i.e., when running FINDMAXLOAD—while the expected harvest $\hat{\mu}_\mathbf{s}$ for the following slot $\mathbf{s}$ and the maximum policy-compliant load I_n^* are high. In this case, a prediction error resulting in a harvest deficit causes V_c to fall considerably below V_crit. Even if not resulting in depletion, FINDMAXLOAD would produce the lowest possible value of I_n^* (i.e., $I_\mathrm{n}^* \approx 0\,\mathrm{mA}$) at its next execution, rendering the node useless for the network and increasing its risk of depletion.

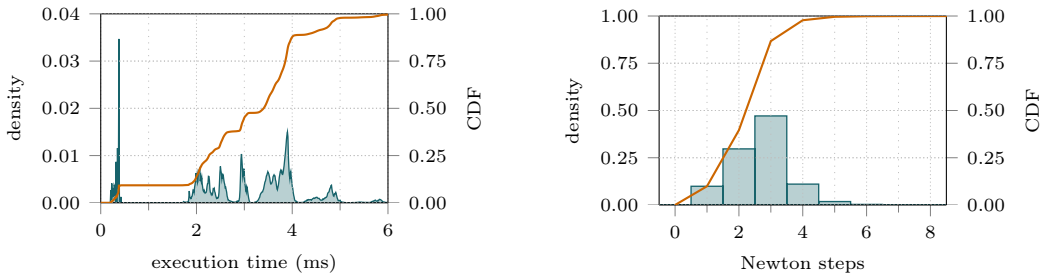
To avoid depletion, we must ensure that $V_\mathrm{c} > V_\mathrm{cut}$ for a time Δt of at least the length $\ell_\mathbf{s}$ of the following slot $\mathbf{s}$, when FINDMAXLOAD is run again to update I_n^*. The smallest required value of V_crit can be obtained from Eq. (1) with suitable bounds for I_h and I_r. A worst case error of harvest prediction leads to zero intake, i.e., $I_\mathrm{h} = 0\,\mathrm{mA}$. The current I_r in Eq. (1) cannot exceed $\hat{\mu}_\mathbf{s}$ because the voltage at the end of slot $\mathbf{s}$ would fall below V_crit, which is a contradiction to FINDMAXLOAD. Moreover, I_r is limited by means of Eq. (2) and $I_\mathrm{n} \leq I_\mathrm{n,max}$. However, these bounds of I_r are too weak and unrealistic, because they neglect that FINDMAXLOAD runs over a longer prediction horizon than a single time slot. The actual output I_n^* of the algorithm, and with that I_r, is usually much smaller and below the average harvest in the prediction horizon (we will discuss and show this in Sects. 5.1.3 and 5.2). Using the expected mean harvest $\mathrm{E}\{I_\mathrm{h}\}$ as an approximation for an upper bound on I_r and letting $I_\mathrm{h} = 0\,\mathrm{mA}$, we can hence employ Eq. (1) to obtain

$$C \cdot \dot{V}_\mathrm{c} \;=\; -I_\mathrm{r} \;\approx\; -\mathrm{E}\{I_\mathrm{h}\} \quad\Rightarrow\quad V_\mathrm{crit} \geq V_\mathrm{cut} + \frac{\mathrm{E}\{I_\mathrm{h}\}}{C} \cdot \Delta t\,. \tag{10}$$

For our prototype harvester in Sect. 3.1, $\mathrm{E}\{I_\mathrm{h}\} = 2\,\mathrm{mA}$, and a slot length of $\ell_\mathbf{s} = 1\,\mathrm{h}$, we find $V_\mathrm{crit} \geq 1.53\,\mathrm{V}$ for a $25\,\mathrm{F}$ capacitor and $V_\mathrm{crit} \geq 0.72\,\mathrm{V}$ for a $200\,\mathrm{F}$ capacitor, respectively. We will show in Sect. 5.2 that these limits give reasonable estimations for choosing V_crit.

Table II. Parameter setup for benchmarking the voltage simulation algorithm SIMVC

Parameter	Symbol	Unit	Set of Values
capacity	C	F	25, 50, 100, 200
initial voltage	V_0	V	0.5, 0.6, ..., 2.7
load	I_n	mA	0, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10
harvest	I_h	mA	0, 0.5, 1, 1.5, 2, 3, 4, 5, 10, 20, 30
slot length	Δt	min	10, 20, ..., 360



(a) (b)

Fig. 8. Evaluation of (a) the execution time of SIMVC and (b) the number of performed Newton steps

4.4.4. Implementation and Benchmarks. The algorithms in Fig. 7 were implemented for TinyOS. They are individual tasks in order to prevent blocking the microprocessor, which may cause side effects or failure in other parts of the software stack. Moreover, the Newton iteration in SIMVC is a third task. All calculation steps were performed in floating point arithmetic to avoid unforeseen calculation problems or errors. Calculations rely on the standard math library for square root and logarithm operations.

The implementation was benchmarked with an Iris sensor node. We recorded the execution time of SIMVC with microsecond resolution for the set of parameters shown in Table II. More than 320 000 results cover the practical ranges of all parameters with focus on frequency of occurrence, e.g., small harvest and load values are more likely than larger ones. The value of $\epsilon = 1\,\mathrm{mV}$ was chosen to achieve sufficient precision while keeping execution time low. Up to $N = 100$ Newton steps were allowed.

On average, 2.67 Newton iterations were required to calculate the resulting voltage after Δt, where the standard deviation was 0.95. The maximum number of Newton steps was 58, corresponding to 46.5 ms execution time. Mean execution time was 3.06 ms with a standard deviation of 1.26 ms. Figure 8 shows the distribution of execution times and the number of Newton steps of the experiment. Overall execution time of a single run of FINDMAXLOAD scales with the number of slots S and the number of binary-search steps. To achieve a granularity of $\mathrm{tol} = 20\,\mu\mathrm{A}$ for I_n^*, which is below the minimum consumption of an Iris sensor node [Renner et al. 2012a], ten binary search steps are sufficient. A back-of-the envelope calculation shows that for $S = 12$, a load adaptation at the end of a slot will require 367 ms on average. The energy expenditure caused by load adaptation can hence be neglected—e.g., the consumption of the radio running a 1% duty cycle would overshoot the consumption caused by load adaptation by a factor of 470 on an Iris node (with the values from [Renner et al. 2012a]).

5. EVALUATION OF PREDICTIVE LOAD ADAPTATION

This section presents the evaluation methodology and results of our predictive load adaptation algorithm. We show that the algorithm enables uniform node operation, and that the use of harvest

forecasts increases the achievable load by a factor of up to two. The maximum-power-point policy enables a higher node load for large capacities and reduces the risk of depletion in case of erroneous harvest forecasts in changing weather conditions.

5.1. Evaluation Methodology

5.1.1. Data Basis. The evaluation of the proposed energy policies is based on a 194-day harvest trace that was recorded by a sensor node powered by the harvester prototype from Sect. 3.1. It was placed outside on a window sill facing westwards on the fourth floor of the main University building. Readings are averaged over $5\ \mathrm{min}$ intervals to reduce experiment runtime.

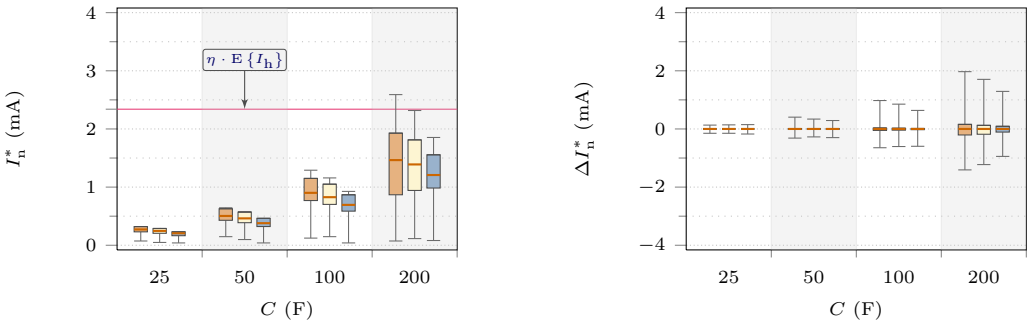
5.1.2. Methodology and Parameters. The voltage course is simulated with a time resolution of $5\ \mathrm{min}$. The initial supercapacitor voltage is $V_\mathrm{c} = V_\mathrm{on}$ in all experiment runs. In the event of temporary depletion (V_c falls below V_cut), the node resumes operation only after charging to $V_\mathrm{c} \geq V_\mathrm{on}$ (cf. Sect. 3.1). The node adapts its consumption $I_\mathrm{n} = I_\mathrm{n}^*$ through FINDMAXLOAD for a forecast horizon of 24 h. Consumption updates are performed at the beginning of a new time slot. The maximum consumption used in the binary search is $I_\mathrm{n,max} = 17.5\,\mathrm{mA}$ and $\eta = 86\%$ (cf. Sect. 3.3). This setup complies with the current consumption of an Iris node with enabled radio and the average efficiency of the harvesters' regulator (cf. Sect. 3.1). This setup additionally enables a straight-forward comparison with the results in Sect. 7. Supercapacitors were dimensioned from 25 F to 200 F. The algorithm parameters were $\epsilon = 0.1\,\mathrm{mV}$ and $\mathrm{tol} = 10\,\mu\mathrm{A}$. Critical voltages V_crit of $0.5\,\mathrm{V}$, $1.0\,\mathrm{V}$, and $1.5\,\mathrm{V}$ were tested according to Sect. 4.4.3.

To analyze the gain of using harvest forecasts, the performance of load adaptation in absence of a forecast was evaluated first. This is equivalent to the method proposed in [Zhu et al. 2009] (cf. Sect. 2.3). In this context, the only relevant (applicable) policy is $\mathcal{P}_\mathrm{DS}$, and the maximum load can be directly calculated for the prediction horizon (cf. Sect. 4.3.1). To compare the general behavior, advantages, and disadvantages of the policy concept and the individual policies, an evaluation with perfect forecasts was performed. For this purpose, perfect forecasts with and without time slots were generated and used, i.e., the actual I_h-data of the next 24 h and their slot representations were fed to FINDMAXLOAD. Finally, we assessed the performance in a realistic environment. Harvest forecasts were generated using the EWMA method introduced in Sect. 2.2. EWMA filter coefficients α of 0.2, 0.5, and 0.8 were used, and the number of slots ranged from $S = 6$ to 72.

5.1.3. Metrics. We assessed the performance of the policies with the following metrics. We analyzed the distribution of I_n^* obtained from the individual loads of each $5\ \mathrm{min}$-interval to cater for a fair comparison w.r.t. different slot lengths and slot distribution. If a node is (temporarily) depleted, a value of $0\,\mathrm{mA}$ is used. Load distributions are displayed with box plots showing the median along with minimum and maximum values plus upper and lower quartiles. The distribution is compared with the average usable harvest $\eta \cdot \mathrm{E}\{I_\mathrm{h}\}$, that defines the upper bound of average current consumption, cf. Fig. 6, since for our hardware $V_\mathrm{c} \leq V_\mathrm{n}$. This value is only reachable for $I_\mathrm{h} = I_\mathrm{n}$ and $V_\mathrm{c} = V_\mathrm{max}$, cf. Eq. (4). Furthermore, we analyzed the distribution of load adaptation deltas ΔI_n^* (the difference of two consecutive values of I_n^*) and show corresponding box plots. We assessed node downtime as a fraction of simulated experiment runtime, i.e., 194 days. In addition to these metrics, traces of V_c, $I_\mathrm{n} = I_\mathrm{n}^*$, and I_h are used to support our findings and provide details.

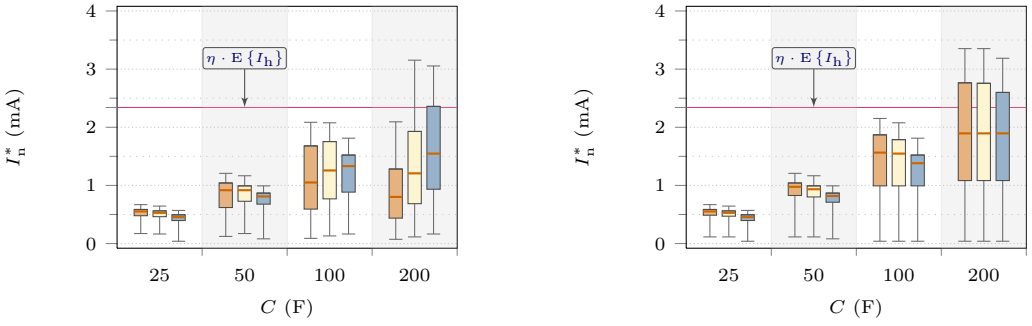
5.2. Evaluation Results

5.2.1. No Forecast. Figure 9 depicts the performance metrics for load adaptations performed every $5\ \mathrm{min}$. In Fig. 9a, it shows that the achievable load I_n^* increases with supercapacitor size. However, the gain is less than the capacity increase, e.g., the median I_n^* for $C = 100\,\mathrm{F}$ and $V_\mathrm{crit} = 1.0\,\mathrm{V}$ is approximately $0.83\,\mathrm{mA}$ while it is $1.39\,\mathrm{mA}$ for $C = 200\,\mathrm{F}$. Using $\mathcal{P}_\mathrm{DS}$ without forecast restricts the load to an upper bound, which is defined by the values of C, V_crit, and V_max—cf. Sect. 4.3.1 for the case $I_\mathrm{h} = 0$. This bound is visible in the figure; e.g., for a 25 F supercapacitor and $V_\mathrm{crit} = 1.0\,\mathrm{V}$, I_n^* cannot exceed the shown $0.29\,\mathrm{mA}$. On sunny days, large portions of the harvest are neither used by the node nor can they be stored in the buffer, once V_c reaches V_max. This behavior



(a) maximum load (b) adaptation deltas

Fig. 9. No forecast: Distribution of (a) I_n^* and (b) ΔI_n^* for $V_{crit} = 0.5\,V$ (red/left), $1.0\,V$ (yellow/center), $1.5\,V$ (blue/right). I_n^* was updated every 5 min



(a) $\mathcal{P}_{DS}$ (b) $\mathcal{P}_{MPP}$

Fig. 10. Perfect forecast: Distribution of I_n^* for $V_{crit} = 0.5\,V$ (red/left), $1.0\,V$ (yellow/center), $1.5\,V$ (blue/right)

is particularly pronounced in the case of small supercapacitors. While it could be circumvented—e.g., by allowing a node to spend more energy when V_c is close to V_{max}—the achievable benefit is questionable and has drawbacks, e.g., the demand for uniform load profiles (cf. Sect. 4.1) and the simple policy concept are undermined. Consumption is relatively uniform, which is indicated by the distribution of adaptation deltas in Fig. 9b. In most cases, the load is adapted by a few tenth of milliampere only.

5.2.2. Perfect Forecasts. The theoretical potential and conceptual weaknesses of the policies are revealed by providing a perfect forecast. Here, the node determines I_n^* and updates its consumption I_n every $5\,min$, i.e., with the granularity of the I_h-trace and voltage simulation. Hence, no downtimes are experienced.

Figure 10 shows the distribution of I_n^* for both policies. The plots reveal that all medians are considerably below the average harvest. For a $25\,F$ supercapacitor, only one fourth of the harvested current is effectively used by the node on average. The influence of V_{crit} for $\mathcal{P}_{DS}$ depends on capacity C, see Fig. 10a: In case of small C, a larger value of V_{crit} leads to a smaller load, since the usable range of V_c is reduced while small capacitors can be fully charged even on days with low harvest. Combining $V_{crit} = 1.5\,V$ with a $200\,F$ supercapacitor increases the average load, because the supercapacitor is operated at higher voltages and thus kept in a better (harvesting) power point. This effect is amplified by the fact that a larger V_{crit} increases the likelihood of V_c reaching V_{max}.

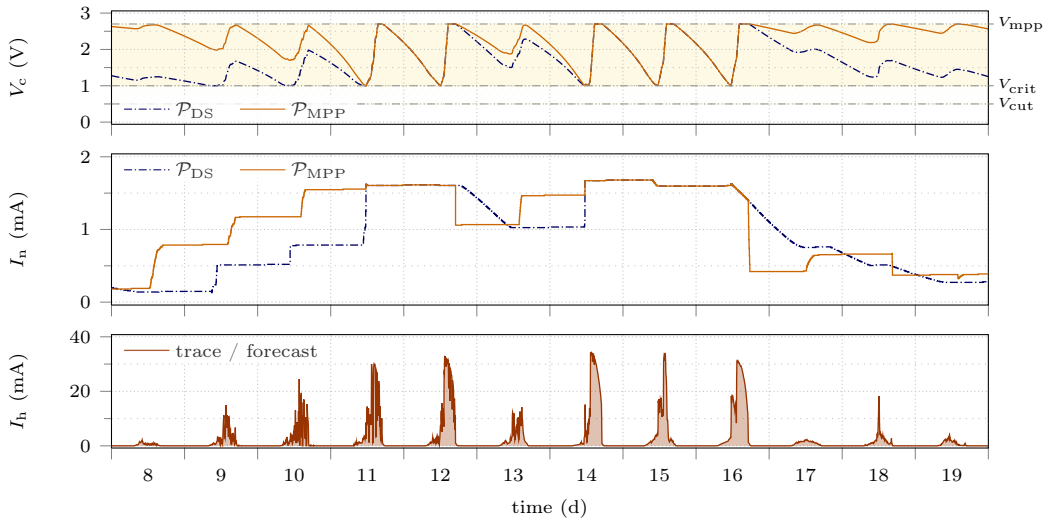


Fig. 11. Perfect forecast: Course of V_c and I_n for $V_{crit} = 1.0\,V$ and $C = 100\,F$. X-axis ticks indicate midnight

Yet, a too large value of V_{crit} will produce the contrary result: Despite a good power point, only a small fraction of energy can be drawn from the supercapacitor.

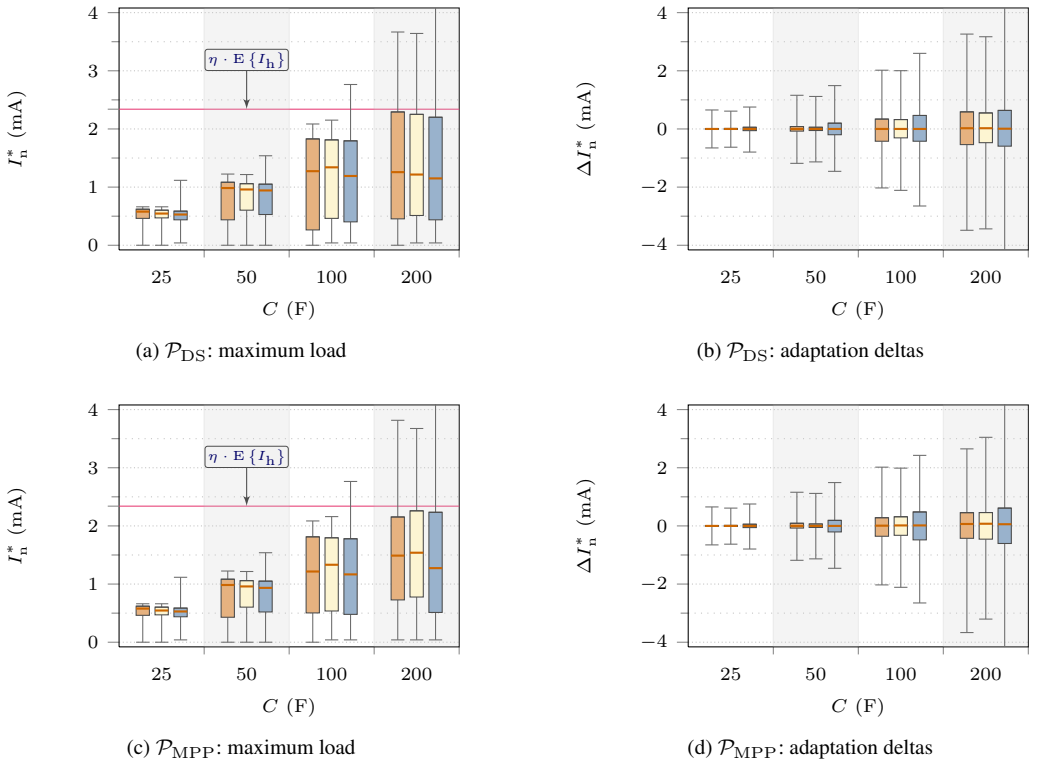
The maximum-power-point policy $\mathcal{P}_{MPP}$ brings improvements for $C \geq 100\,F$, see Fig. 10b. This supports the previous findings and shows that small supercapacitors are fully charged once per day in the setup, even without applying $\mathcal{P}_{MPP}$.

Figure 11 shows an energy course. On days with high harvest, $\mathcal{P}_{MPP}$ and $\mathcal{P}_{DS}$ show similar voltages traces. Differences are notable during days with low harvest. Particularly in the first days, $\mathcal{P}_{MPP}$ achieves higher loads, because V_c is held at a higher level. The price of this benefit is paid, when a good day is followed by a poor one (e.g., day 13): The load is suddenly decreased, whereas $\mathcal{P}_{DS}$ leads to a smoother descent.

Comparing the results to those in Sect. 5.2.1 admits several conclusions. Firstly, the average load is increased when using a harvest forecast for smaller capacities ($C \leq 100\,F$ in this evaluation), because forecasting relaxes the load limitation dictated by the capacity. In case of $C = 25\,F$ and $C = 50\,F$, the average I_n^* is more than doubled. For $C = 200\,F$, the choice of V_{crit} is critical when using $\mathcal{P}_{DS}$.

5.2.3. EWMA Forecasts. Load adaptations experience an expected performance decrease when uncertain forecasts are used. The general observation of the experiments is that the average load decreases, yet stays above the level of doing without forecasts. There is one exception: Using forecasts with $C = 200\,F$ and $\mathcal{P}_{DS}$ leads to lower values of I_n^*. This is caused by overestimating the harvesting conditions on a poor day after a series of good days. In this situation, V_c quickly falls below V_{crit} and leaves the harvester in a lower power point. Using $\mathcal{P}_{MPP}$ only pays off for $C > 100\,F$.

These results can be traced in Fig. 12, which primarily serves to study the influence of the number of slots. Figures 12a and 12c reveal that fewer slots may lead to a higher average load and produce fewer outliers of particularly large values of I_n^*. This effect is best visible for $C = 200\,F$ and $\mathcal{P}_{MPP}$. The reason for this behavior is that using more slots creates a more fine-grained prediction of V_c; combined with imprecise forecasts, the chance of policy violation is increased. Moreover, with fewer slots, the chance of missing temporary situations with $V_c < V_{crit}$ is increased. The characteristics of load adaptations shown in Figs. 12b and 12d exhibit a similar shape among policies and w.r.t. previous results.



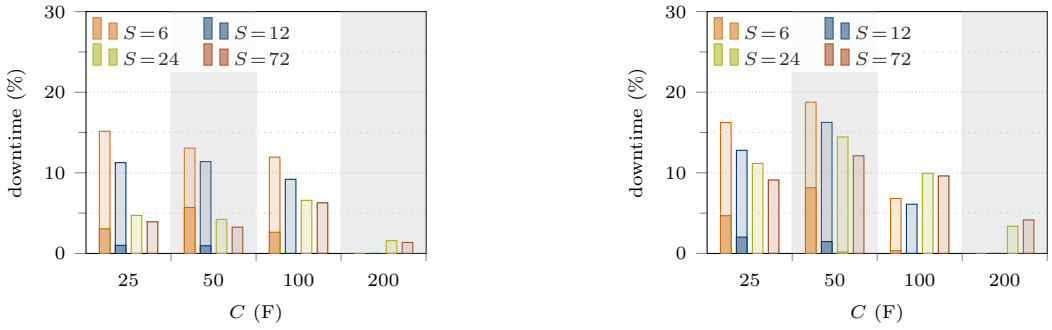
(a) $\mathcal{P}_{\mathrm{DS}}$: maximum load

(b) $\mathcal{P}_{\mathrm{DS}}$: adaptation deltas

(c) $\mathcal{P}_{\mathrm{MPP}}$: maximum load

(d) $\mathcal{P}_{\mathrm{MPP}}$: adaptation deltas

Fig. 12. EWMA forecast: Distribution of I_n^* and ΔI_n^* using 12 (red/left), 24 (yellow/middle), and 48 (blue/right) slots with an EWMA smoothing factor of $\alpha = 0.8$. The critical voltage is $V_{\mathrm{crit}} = 1.0\,\mathrm{V}$.

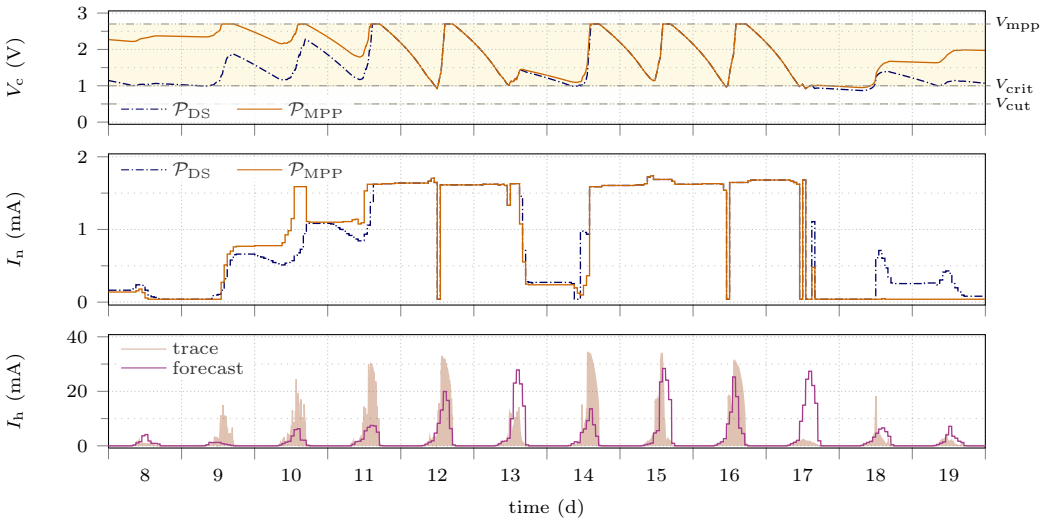


(a) $\alpha = 0.2$

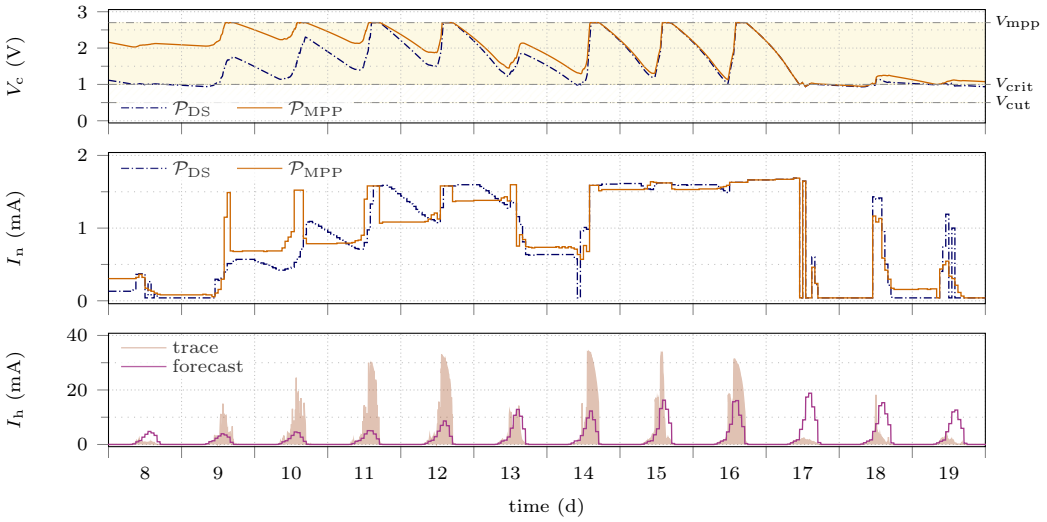
(b) $\alpha = 0.8$

Fig. 13. EWMA forecast with slots and $\mathcal{P}_{\mathrm{MPP}}$: Influence of the filter coefficient α on node downtime for $V_{\mathrm{crit}} = 0.5\,\mathrm{V}$ (light) and $V_{\mathrm{crit}} = 1.0\,\mathrm{V}$ (dark)

Downtimes vary with the chosen setup. They are close to 0% for $V_{\mathrm{crit}} \geq 1\,\mathrm{V}$, if at least 12 slots are used. An in-depth analysis of the results reveals that additional time slots reduce the risk of depletion only due to the increased number of load adaptations, i.e., policy control points. Forecast quality is not improved—details about this aspect are provided in [Renner and Turau 2012a].



(a) EWMA filter coefficient $\alpha = 0.2$



(b) EWMA filter coefficient $\alpha = 0.8$

Fig. 14. EWMA forecast with 24 slots: Course of V_c and I_n for $V_{crit} = 1.0\,V$ and $C = 100\,F$. Ticks on the x-axis indicate midnight

Influence of EWMA Smoothing Factor. The influence of the EWMA smoothing factor α on node downtime is studied in Fig. 13. It shows that downtimes are intolerable in case of $V_{crit} = 0.5\,V$. This is an expected result, because there is no backup energy in case of too optimistic forecasts. A value of $V_{crit} = 1\,V$ prevents node downtimes in most setups; only for small capacities and $S = 6$, a risk of depletion between 4 and 8% remains. A small value of α, see Fig. 13a, generally yields lower downtimes. From the perspective of downtime only, a small value of α appears favorable. In the following, it is shown that this observation is to be taken with care.

While α has low influence on the median value of I_n^* and, generally, on its distribution as well—no figure is thus presented—the detailed traces in Fig. 14 show remarkable behavior for both chang-

ing and stable weather conditions. Between days 9 and 12, when harvest is improving over the days, $\alpha = 0.2$ allows a quicker adaptation and therefore better precision of the harvest forecasts. This in turn extends the used range of V_c, resulting in larger values of I_n^* and I_n, respectively. However, operating V_c close to V_crit bears the threat of short break-ins of I_n, e.g., see the middle of day 12 in Fig. 14a. Upon the poor harvest on day 13, the forecast produced with $\alpha = 0.2$ has a higher error than the one obtained using $\alpha = 0.8$ (cf. Fig. 14b), so that the latter enables maintaining a higher load from the middle of that day on, because the too optimistic forecast using $\alpha = 0.2$ forced V_c close to V_crit. Both traces show a quick recovery after the single poor day, i.e., I_n is high between noon of days 14 and 17. The following series of days with low harvest, however, produces several insights. A quickly adapting forecast lowers the achievable load for $\mathcal{P}_\mathrm{MPP}$, because bringing V_c up towards V_max is enforced. In contrast, $\mathcal{P}_\mathrm{DS}$ enables a moderate load around 0.3 mA. A slowly adapting forecast ($\alpha = 0.8$) produces less accurate forecasts in this case, yet leads to a higher load regardless of the policy and without causing depletion.

Both traces share in common that $\mathcal{P}_\mathrm{MPP}$ produces load peaks when the forecast is consecutively improved of several days (e.g., from day 9 to 13). This can be explained as follows: Since the forecast is too low, V_c reaches V_mpp quicker than predicted. In this case, the maximum-power-point criterion of $\mathcal{P}_\mathrm{MPP}$ is satisfied, so that the policy effectively reduces to $\mathcal{P}_\mathrm{DS}$. This can be observed at noon of day 11, where both policies permit the same value of I_n^* at an equal V_c. When I_h undershoots I_n^* shortly before sunset and V_c thus falls below V_mpp, this effect is annulled. To comply with $\mathcal{P}_\mathrm{MPP}$, a reduced value of I_n^* is produced by FINDMAXLOAD until a point is reached, at which $\mathcal{P}_\mathrm{DS}$ and $\mathcal{P}_\mathrm{MPP}$ output the same value of I_n^*. This is the case, when the forecast is lower than the actual harvest in the prediction horizon and if the maximum-power condition of $\mathcal{P}_\mathrm{MPP}$ is dominated by depletion-safety regardless of the value of V_c. Such a situation is generally observable between noon of day 14 and noon of day 17.

5.2.4. Additional Results and Observations. The results presented in this section motivate two further investigations that are carried out in the following.

Using the current supercapacitor voltage V_0 for policy-compliance testing leads to abrupt load changes when V_0 falls below V_crit. The effect of ignoring V_0 was therefore studied. Improvements of the average I_n^* are below 2% in most cases, but load adaptations are less abrupt when V_c approaches V_crit. This comes at the cost of an increased downtime; e.g., downtime rises from 1% to 11% for $\mathcal{P}_\mathrm{DS}$ with $C = 50\,\mathrm{F}$, $V_\mathrm{crit} = 1\,\mathrm{V}$, and $\alpha = 0.8$. For $V_\mathrm{crit} = 1.5\,\mathrm{V}$, downtimes range from 0% to 2%. Overall, increasing performance (i.e., load) amplifies the risk of depletion. These figures, moreover, give a feeling for the performance in a real-world application. Here, preventing depletion may not be possible, because nodes cannot interrupt normal operation completely due to sensing tasks and inevitable data forwarding responsibilities.

The policy $\mathcal{P}_\mathrm{MPP}$ requires to fully charge the supercapacitor once a day. While this has been shown to generally improve I_n^*, it has also been identified to be a source of wasted harvest. If harvest overshoots its forecast, the supercapacitor is usually fully charged at times of high harvest, i.e., $I_\mathrm{h} \gg I_\mathrm{n}^* = I_\mathrm{n}$. Furthermore, the combination of a large capacity ($C = 200\,\mathrm{F}$) and consecutive days with low harvest lead to a low (almost zero) value of I_n^*, because charging to V_mpp is impossible within a day. To overcome these problems, the effect of choosing $V_\mathrm{mpp} = 2.5\,\mathrm{V}$ (instead of $V_\mathrm{mpp} = V_\mathrm{max}$) was investigated exemplary. With perfect harvest forecasts, the average I_n^* decreases by up to 10%, where the effect is lower for smaller C. When EWMA-filtered harvest forecasts are used, there is essentially no effect for $C \leq 50\,\mathrm{F}$. In some setups with $C = 100\,\mathrm{F}$ and all setups with $C = 200\,\mathrm{F}$, the average I_n^* is elevated by up to 3%. Downtime is not affected. For large capacities, a value of V_mpp slightly (e.g., $0.1\,\mathrm{V}$ to $0.2\,\mathrm{V}$) below V_max has the potential to increase the average achievable load in a practical scenario while not enlarging the risk of depletion.

5.3. Summary

The evaluation shows the applicability of the policy concept and confirms that operating a supercapacitor close to its maximum voltage (using $\mathcal{P}_\mathrm{MPP}$) enables a higher load and hence yields improved

mean utility with low variation. The gain vanishes for small capacities, as the solar cell fully charges these supercapacitors even on days with low harvest. This also causes a relatively low average load compared to the mean harvest, since a surplus $I_\mathrm{h} > I_\mathrm{r}$ cannot be buffered for later use. The problem vanishes with the use of smaller solar panels and larger supercapacitors. In fact, the solar panel of the prototype is over-dimensioned when combined with a 25 or $50\,\mathrm{F}$ supercapacitor; its average harvest of a day exceeds the storable energy of these small supercapacitors by a factor of 4 and 2, respectively.

Load variation of factor 2 and above—within and across days—stresses that choosing a fixed consumption or duty cycle a priori is difficult, if not impossible. With a static consumption setup, energy resources are wasted, or unacceptable and repeated node downtimes are risked. In contrast, the policy concept achieves rapid adaptation to the remaining energy resources. Using harvest forecasts increases the achievable load and thus improves the utility of harvested energy.

Enforcing a minimum supercapacitor voltage V_crit above the harvester's cut-off voltage decreases the risk of depletion but does not affect utility in most cases. This is particularly important for forecasts using time slots and EWMA-filtering: Forecast errors and model inaccuracy (caused by the linearization in combination with long time slots) must be compensated. However, the load changes abruptly when V_c undershoots V_crit.

The results suggest that in a real deployment, depletion cannot be always prevented: Operating a node in its sleeping mode for a long time (hours or days) is likely to be impossible, since sensing and data forwarding tasks have to be performed. More precise harvest forecasts are needed to accomplish early detection of poor weather conditions, so that energy can be saved timely. Therefore, finding an improved method for harvest forecasting should be on the agenda of future sensor network research.

Performing load adaptations at the beginning of time slots, used for harvest forecasting, reduces adaptation overhead and improves load stability. For solar harvesting, the optimal choice is between 12 and 24 slots. Using fewer slots increases the risk of depletion, using more slots adds a certain randomness to forecasts (and adaptations), because slot values are too much affected by temporary weather influences, such as passing clouds.

6. PRACTICAL LOAD ADAPTATION FOR MULTI-HOP DATA COLLECTION

In the following, we present our motivation for practical load adaptation and the general approach. We introduce the ORiNoCo protocol, a low-power duty-cycling data-collection protocol, and describe a concrete method for online load adaptation.

6.1. Motivation and Objective

Combining state-of-charge assessment with the concept of energy policies enables load adaptation that minimizes the risk of depletion while it elevates average uniform load, as shown in Sect. 5. However, these results have been obtained through simulation. Because there is a gap between simulation and the real world, a practical evaluation in a real-world deployment is required. Moreover, simulation and its result focused on the perspective of a single node and did not cover the network aspect, which is of major relevance for wireless sensor networks. This section hence targets the following goals:

(1) Supporting the results on real hardware.
(2) Proving the practical interplay of harvest forecasts, predictive load adaptation, and actual energy consumption.
(3) Devising and analyzing a concept of practical load adaptation in a multi-hop sensor network deployment with the methods presented in this paper.

In the following, the general approach to reach these goals is presented. Details are provided subsequently in the section.

6.2. General Approach

One of the major application scenarios of wireless sensor networks is data collection. Due to topology changes, unstable wireless links, and uncertain consumption profiles of the nodes, it is difficult, if not impossible, to pre-set a radio duty cycle that prevents depletion and ensures sufficient throughput at the same time. For these reasons, radio duty-cycle adaptation in multi-hop sensor networks is an ideal candidate to analyze practical applicability of predictive load adaptation with energy policies.

Many well-known routing protocols for sensor networks (e.g., CTP [Gnawali et al. 2009]) produce traffic and consumption overhead due to packet exchange for link-quality assessment and route maintenance. Recently, opportunistic routing protocols (e.g., [Landsiedel et al. 2012; Unterschütz et al. 2012] without this deficiency have been developed. We will show how this novel class of routing protocols can be integrated with the concept of predictive load adaptation through local reckoning and without producing overhead in terms of packet exchange. For this purpose, we use the opportunistic receiver-initiated no-overhead collection (ORiNoCo) protocol from [Unterschütz et al. 2012]. The energy consumption of the afore-mentioned opportunistic routing protocols is mainly influenced by four factors:

(1) the rate of sending and receiving packets,
(2) the waiting time before packet transmission, called forwarding delay hereafter,
(3) the beacon rate of the integrated low-power MAC, and
(4) the size of packet bursts, i.e., a node sends multiple packets subsequently as a bulk.

The first two of these factors can only be (passively) measured by a sensor node online, whereas the second two are knobs for load adaptation. A detailed consumption model is presented in this section, and a scheme for practical load adaptation that ensures fair energy splitting for sending and receiving data is derived. The intention of this scheme is to prove the practical applicability of load adaptation with the concepts presented in this paper.

Finally, this adaptation scheme is put into practice through a case study of a twelve-node outdoor deployment. For this purpose, all components from this paper are implemented for TinyOS. A detailed analysis of the performance is conducted. In particular, the results of the algorithm for maximum policy-compliant load determination are compared with the actual consumption of the node and the average harvest. Node downtimes are analyzed and the results are compared to those of Sect. 5.

6.3. The ORiNoCo Protocol

For a better understanding of the load adaptation scheme, an explanation of the general concept behind ORiNoCo is provided.

6.3.1. Receiver-Initiated Media Access Control. ORiNoCo is based on the RI-MAC protocol [Sun et al. 2008], which aims at preserving energy by duty cycling the radio through what is called low-power probing.

All nodes in the network send periodic beacons to signal a general readiness to receive a data packet. A node willing to send data switches on its radio and waits for the beacon of the receiver. The sender transmits its data packet upon reception of this beacon after a small random back-off, if the channel is idle. Successful packet reception is acknowledged by the receiver with another beacon, containing the identifier (or address) of the sender. Having received the (acknowledging) beacon, the sender either switches off its radio, if there are no more packets left, or transmits an additional packet to the (same) receiver. The receiver node holds on for a short period T_{hld} and switches off its radio thereafter, if no additional data packet arrives. Hence, the value of T_{hld} also defines the maximum back-off. An example of a data transmission using the RI-MAC protocol is illustrated in Fig. 15.

Each node in the network periodically sends beacons. To decrease the chance of accidental and undesired node synchronization—which results in beacon collisions—inter-beacon times, or sleep

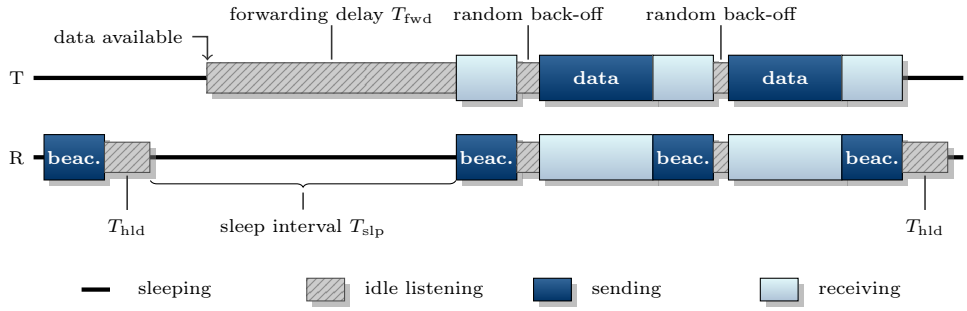


Fig. 15. Consecutive transmission of two data packets from sender T to receiver R with RI-MAC

intervals, are chosen randomly from the interval

$$[(1-\delta)\cdot T_{\text{slp}}, (1+\delta)\cdot T_{\text{slp}}] \qquad (0<\delta<1)\ . \tag{11}$$

The choice of T_{slp} is critical. On the one hand, a small value reduces the forwarding delay for the sender, but it increases consumption due to frequent beacon transmissions, and it provokes channel congestion in dense networks. On the other hand, a large value prevents congestion, but it increases the forwarding delay and energy consumption for the sender. An analysis of consumption w.r.t. T_{slp} is provided in Sect. 6.4.

6.3.2. Opportunistic Data Collection. RI-MAC only provides direct communication between sensor nodes. In most data-collection sensor networks, multi-hop communication towards the sink is required, so that an additional routing protocol has to be run on top of RI-MAC. ORiNoCo enhances RI-MAC to become a data-collection protocol.

To achieve this goal, each node stores a path metric ω and adds it to all beacons. The value of ω reflects the cost for delivering data to the sink via the sender of the beacon and enables the formation of a tree-like routing structure.

Data collection is enabled through the following protocol behavior. When a node T wants to transmit data, it listens on the channel until receiving a beacon from an arbitrary node R. Node T calculates the path weight $\omega_{\text{T,R}}$ using the cost metric function

$$\omega_{\text{T,R}} \leftarrow \omega_{\text{R}} + \upsilon_{\text{T,R}}\ . \tag{12}$$

Here, ω_{R} is the metric attached to the beacon sent by node R, and $\upsilon_{\text{T,R}}$ is a measure of the cost for sending a single packet to node R, where $\upsilon_{\text{T,R}}$ is determined by node T. In this section, the hop count metric is used, so that $\upsilon_{\text{T,R}} = 1$. If the condition

$$\omega_{\text{T,R}} \leq \omega_{\text{T}} \tag{13}$$

holds, node R is closer to the sink than node T. Only in this case, T transmits its data packet to node R and waits for the acknowledging beacon. Upon reception of that beacon, node T updates its weight ω_{T} with $\omega_{\text{T,R}}$—adapting to a path with low cost but poor connectivity is thus prevented. If node T receives no acknowledging beacon, it waits for the next beacon satisfying Eq. (13) from any node.

Initially, all nodes with exception of the sink assume $\omega = \infty$, i.e., there is no path to the sink. The sink S always maintains $\omega_{\text{S}} = 0$. Through weight updating by Eq. (12), a tree-like routing structure is constructed successively. Unlike using a traditional routing tree, nodes are not restricted to forward data to a single (parent) node only, but they may chose among all nodes satisfying Eq. (13). However, upon failure of a node or a changing topology, a sender T may be prevented from forwarding its data, if no receiver R satisfying Eq. (13) is in its range. In this case, node T resets its path weight (i.e., $\omega_{\text{T}} \leftarrow \infty$) after a pre-defined time period, e.g., a multiple of T_{slp}.

6.4. Energy Consumption and Load Adaptation

Load adaptation in sensor networks using low-power protocols, such as ORiNoCo, is usually achieved by adjusting the radio duty cycle ϕ. However, the duty cycle cannot be adjusted directly as outlined in Sect. 6.2.

Therefore, an understanding of the ORiNoCo consumption model and its parameters is required to allow for effectively and accurately adapting consumption. Such a model is elaborated in the following.

6.4.1. Parameters and Assumptions. To model energy consumption of ORiNoCo while keeping complexity low, the following assumptions and simplifications are made:

— There are three consumption states, namely the sleep state with a current consumption I_{slp}, the radio receive state with a current consumption I_{rx}, and the radio send state with a current consumption I_{tx}. Consumption during periods of idle listening is I_{rx}, thus equivalent to receiving. This simplification introduces only small errors on many sensor node platforms.
— The time T_{beac} needed for sending a beacon, the time T_{dat} needed for sending and receiving a data packet, and the hold time T_{hld} are constant and known. Back-offing takes an average of $T_{\mathrm{hld}}/2$.
— Times for switching the radio on and off, and for changing its state are neglected.
— The average inter-beacon time is T_{slp}. In fact, the introduction of the variation parameter δ increases the average inter-beacon slightly beyond T_{slp}—this is called the Hitchhiker's paradox (details are provided in [Unterschütz et al. 2012]). However, for values $\delta < 0.2$, the effect is negligible.
— Each node is aware of (or estimates) the rate κ of packets created by itself and the rate λ of incoming packets.
— Each node knows (or estimates) its average forwarding delay T_{fwd}. This is of particular relevance, because (i) each node may run an individual duty cycle, so that forwarding delays differ within the network, and because (ii) forwarding delay depends on the number of available parents.
— Each node maintains a packet queue. Packets are only forwarded, if the number of packets in the queue is at least $Q \geq 1$, a threshold smaller than the queue size.
— Packet loss is not modeled, since ORiNoCo does not rely on link-quality information intentionally. Such data will thus not be accessible by the nodes, and adding link-quality estimators would increase algorithmic overhead.

6.4.2. Consumption Model. A node running ORiNoCo quasi-periodically broadcasts beacons and sleeps for the remainder of time. After sending a beacon, the node waits an extra T_{hld} for incoming data. The basic operation induces the mean base consumption

$$\bar{I}_{\mathrm{base}} = \frac{T_{\mathrm{slp}} \cdot I_{\mathrm{slp}} + T_{\mathrm{beac}} \cdot I_{\mathrm{tx}} + T_{\mathrm{hld}} \cdot I_{\mathrm{rx}}}{T_{\mathrm{slp}} + T_{\mathrm{beac}} + T_{\mathrm{hld}}} \approx I_{\mathrm{slp}} + \frac{T_{\mathrm{beac}} \cdot I_{\mathrm{tx}} + T_{\mathrm{hld}} \cdot I_{\mathrm{rx}}}{T_{\mathrm{slp}} + T_{\mathrm{beac}} + T_{\mathrm{hld}}}\,, \tag{14}$$

assuming that $T_{\mathrm{slp}} \gg T_{\mathrm{beac}} + T_{\mathrm{hld}}$.

A single packet reception consists of sending the initial beacon, receiving the data packet, and sending an acknowledging beacon. Moreover, the receiver has to wait an additional $\frac{1}{2} \cdot T_{\mathrm{hld}}$ on average due to congestion avoidance (back-offing by the sender). Since the consumption of the initial beacon is accounted for by $\bar{I}_{\mathrm{base}}$, the mean current consumption for receiving packets is

$$\bar{I}_{\mathrm{recv}} = \lambda \cdot \left(\tfrac{1}{2} \cdot T_{\mathrm{hld}} \cdot I_{\mathrm{rx}} + T_{\mathrm{dat}} \cdot I_{\mathrm{rx}} + T_{\mathrm{beac}} \cdot I_{\mathrm{tx}}\right). \tag{15}$$

To send one or multiple data packets, the sender must wait for a beacon before starting the actual transmission. If Q consecutive data packets are sent, waiting is necessary only every Qth data packet. For each sent data packet, one acknowledging beacon is received. Congestion avoidance requires waiting for $T_{\mathrm{hld}}/2$ on average between beacon reception and packet transmission. Mean

consumption for data transmission is

$$\bar{I}_{\mathrm{send}} = (\kappa + \lambda) \cdot \left(\frac{T_{\mathrm{fwd}} + T_{\mathrm{beac}}}{Q} \cdot I_{\mathrm{rx}} + \tfrac{1}{2} \cdot T_{\mathrm{hld}} \cdot I_{\mathrm{rx}} + T_{\mathrm{dat}} \cdot I_{\mathrm{tx}} + T_{\mathrm{beac}} \cdot I_{\mathrm{rx}} \right) . \tag{16}$$

Overall consumption of ORiNoCo is the sum of the individual components

$$\bar{I}_{\mathrm{base}} + \bar{I}_{\mathrm{recv}} + \bar{I}_{\mathrm{send}} . \tag{17}$$

6.4.3. Load Adaptation. In an energy-harvesting sensor network, the goal of load adaptation is to align consumption and harvest. According to Sect. 4.2, consumption needs to match the maximum supported load I_{n}^{*}, i.e.:

$$I_{\mathrm{n}}^{*} \stackrel{!}{=} \bar{I}_{\mathrm{base}} + \bar{I}_{\mathrm{recv}} + \bar{I}_{\mathrm{send}} . \tag{18}$$

In this equation, consumption due to sensing and running the load adaptation algorithm is not considered, because it is low (in the considered scenario). In an application with high-power sensors, Eq. (18) can be modified, e.g., by subtracting the average sensor consumption from I_{n}^{*}.

According to Eqs. (14) to (16), energy consumption of the ORiNoCo protocol is influenced by the parameters T_{slp}, T_{fwd}, Q, κ, and λ, of which only T_{slp} and Q can be directly controlled by a node. Forwarding delay T_{fwd} is determined by the sleep interval of parent nodes and their number; and λ primarily depends on the topology but may be implicitly affected by the choice of T_{slp}. Changing the local packet rate κ is not considered. All other parameters in the equations mainly depend on the hardware and cannot be modified.

One simple approach is to choose a fixed value for Q (e.g., $Q = 1$ to reduce packet delay) and determine T_{slp} using Eq. (18). However, this approach bears one conceptual threat: When a node increases T_{slp} to reduce its own consumption, it amplifies the average forwarding delay T_{fwd} of its neighboring nodes that have a larger value of ω. The resulting consumption increase can only be compensated by these nodes through prolonging T_{slp}. An epidemic increase of T_{slp} results with its origin being nodes close to the sink, since those nodes have the heaviest network load and must therefore run at a lower duty cycle, i.e., larger values of T_{slp}. Therefore, consumption is mainly caused by waiting for packet transmission. Since large values of T_{slp} enlarge the average forwarding delay T_{fwd}, a single lost beacon has the potential to blow energy consumption. The latter is particularly critical in sparse networks, where only one forwarder may be available for a node.

Due to these observations, a different concept of load adaptation is followed in this section. It stems from the fact that each node in the network acts as a data forwarder—i.e., data is received from other nodes and data is sent towards the sink. Assuming that the network is generally delay-tolerant (which is the case in many monitoring applications), one reasonable option is to choose Q and T_{slp}, such that an equal amount of energy is spent for receiving and for sending, i.e.,

$$\bar{I}_{\mathrm{base}} + \bar{I}_{\mathrm{recv}} \stackrel{!}{=} \bar{I}_{\mathrm{send}} , \tag{19}$$

while additionally satisfying Eq. (18). We account $\bar{I}_{\mathrm{base}}$ as expenditure for reception, because beacons serve as reception invitations.

Here, T_{slp} is usually measured in milliseconds and ranges from a few tens to a few thousand milliseconds. In contrast, Q is unlikely to exceed a few tens due to the limited memory (RAM) resources of sensor nodes. Storing packets in the Flash or EEPROM memory is possible but not considered for simplicity reasons: Such an approach requires to alter the consumption model, because Flash and EEPROM access entail non-negligible additional consumption. However, since Q is less fine-grained than T_{slp}, it is beneficial to solve Eqs. (18) and (19) for Q first (and round the result to the ceiling to prevent $Q < 1$). Afterwards, T_{slp} can be deduced from Eq. (18) using the determined value of Q. A practical solution, assuming $I_{\mathrm{tx}} \approx I_{\mathrm{rx}}$, to this pair of equations yields

$$Q \leftarrow \left\lceil \frac{2 \cdot (T_{\mathrm{fwd}} + T_{\mathrm{beac}}) \cdot I_{\mathrm{rx}} \cdot (\kappa + \lambda)}{I_{\mathrm{n}}^{*} - I_{\mathrm{rx}} \cdot (\kappa + \lambda) \cdot (2 \cdot T_{\mathrm{beac}} + 2 \cdot T_{\mathrm{dat}} + T_{\mathrm{hld}})} \right\rceil \tag{20}$$

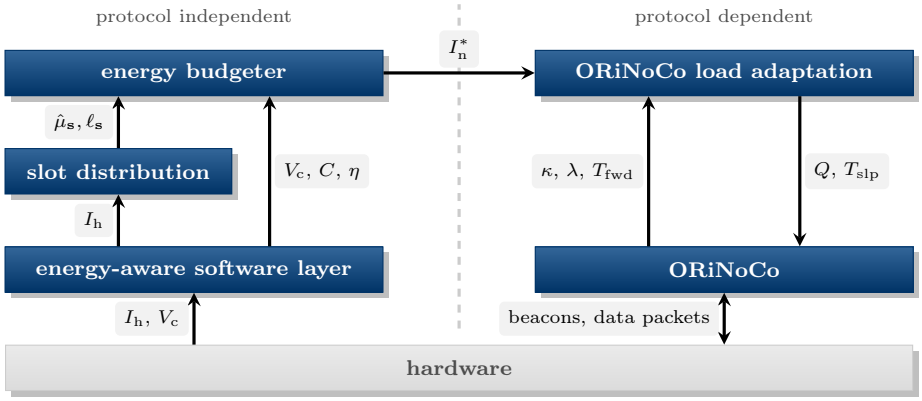


Fig. 16. System components and data flow of the integration of ORiNoCo and predictive load adaptation

and

$$T_\mathrm{slp} \leftarrow \frac{I_\mathrm{rx} \cdot (T_\mathrm{beac} + T_\mathrm{hld})}{I_\mathrm{n}^* - I_\mathrm{slp} - \bar{I}_\mathrm{recv} - \bar{I}_\mathrm{send}} - (T_\mathrm{beac} + T_\mathrm{hld}) \tag{21}$$

In both equations, the denominator may become zero or negative. This case requires individual handling, e.g., by assigning pre-defined values to T_slp and Q, because I_n^* is smaller than the minimum load (for sending and receiving packets).

6.5. Integration of ORiNoCo and Predictive Load Adaptation

The integration of ORiNoCo and predictive load adaptation consists of several software components that are described in this section. Figure 16 provides an overview of the components and the data flow. All software was developed for TinyOS. The binary program image for an Iris node has a size of 32.5 kB and uses 1.7 kB RAM (not including the packet queue); thus sparing sufficient resources.

The energy-aware software layer measures and provides the values of V_c, I_h, and I_n. Samples of all sensors are taken periodically. Value updates are reported to registered software components through TinyOS events; in addition, they can be accessed on demand. The calibrated capacity C and the regulator efficiency η are also provided.

A day is split into S time slots by the slot distribution. It stores and provides the values of slot-wise harvest forecasts with a forecast horizon of one day (cycle). The average value $\bar{I}_\mathrm{h}$ since the beginning of the current slot $\mathbf{s}$ is calculated from the periodic samples of I_h provided by the energy-aware software layer. When a slot $\mathbf{s}$ elapses, the final value of $\bar{I}_\mathrm{h} = \mu_\mathbf{s}$ is used to update $\bar{\mu}_\mathbf{s}$ by EWMA filtering. The initial value $\bar{\mu}_\mathbf{s} = 0$ is replaced on the first day with the value of $\mu_\mathbf{s}$ to overcome the slowness of the EWMA filtering process for values of α around 0.7 and larger.

The energy budgeter corresponds to the implementation of the algorithm for determining the maximum supported load in Sect. 4.4.4. The values of V_c, C, and η are obtained from the energy-aware software layer. Slot value forecasts are provided by the slot distribution. The prediction horizon is one day. Energy policies are implemented as individual, exchangeable software components.

The ORiNoCo protocol consists of several components that replace most parts of the TinyOS radio stack. It was implemented for the Iris sensor node platform. A packet queue of configurable length is used to buffer incoming packets and those generated by the node. As soon as the queue length exceeds Q (and if no packet reception is pending), packet transmission is initiated. It is only stopped, when the packet queue is empty. This implies that both sending and receiving are blocking (each other). Changes of the parameters Q and T_slp are enabled through a programming interface. To provide the necessary parameters for load adaptation, a traffic monitor estimates κ, λ, and T_fwd with an EWMA filter with a filter coefficient of 0.95, i.e., new values are weighted with a factor of 0.05 to decrease the sensitivity to outliers.

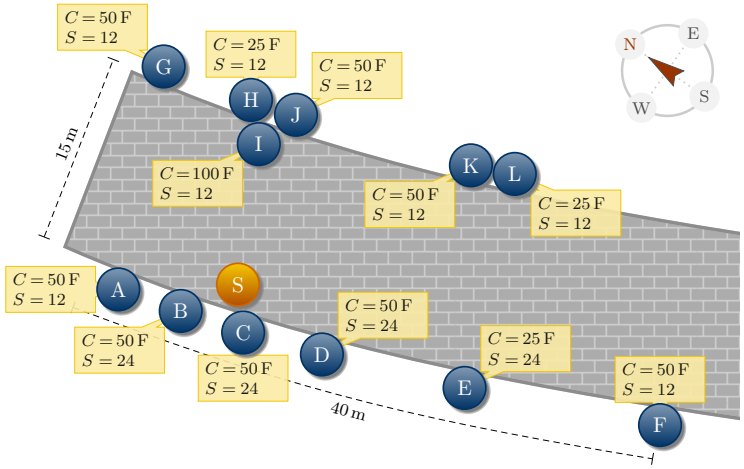


Fig. 17. Node deployment: positioning and configuration

Load adaptation is started upon expiration of a slot by capturing the corresponding event. First, the maximum supported load I_{n}^{*} is obtained from the energy budgeter. Thereafter, new values of Q and T_{slp} are determined as described in Sect. 6.4.3. Required parameters are obtained from the ORiNoCo interface. To avoid queue congestion—which requires to drop packets—load adaptation prevents Q from exceeding a configurable value. Configurable upper and lower boundaries of T_{slp} are also enforced during load adaptation (cf. Sect. 6.3.1).

7. EVALUATION

Next, we present the evaluation methodology and results of our real-world case study. We show that predictive load adaptation enables precise consumption adaptation in multi-hop data collection networks, where consumption is influenced by external and changing factors such as the network topology. The case study confirms that harvest forecasts increase node activity. Load adaptation automatically balances the packet flow, so that nodes with low energy resources have to handle less network traffic.

7.1. Evaluation Methodology

7.1.1. Deployment. The field test consisted of twelve sensor nodes equipped with the harvester prototype and one sink. The latter was connected to and powered by a PC via USB. All nodes but two were placed on the outside window sills of the university building. Plastic boxes protected the nodes from rain. To maintain sufficient harvest, the solar cells were mounted on top of the boxes. One of the nodes and the sink were placed inside the building to maintain network connectivity (direct communication from one side of the building to the other was not possible). However, the solar cell of the indoor node was installed on the outside window sill. Figure 17 shows the arrangement of nodes around the building. Deployment was divided into several phases. In each of these phases, a subset of the nodes was deployed; the network grew and the topology changed over time. The intention of this approach was to simulate changes in network topology and to check the adaptation performance in this situation.

7.1.2. Setup and Parameters. All nodes (except the sink) were equipped with supercapacitors from 25 to 100 F (nominal capacity), see Fig. 17. They ran the software introduced in Sect. 6.5. Capacity was calibrated after booting.

Samples of I_{h} were taken every 3 s and samples of V_{c} were obtained every 5 s. The slot distribution consisted of 12 to 24 slots as shown in Fig. 17, and harvest forecasts were equivalent to the

Table III. Packet types used in the field test. Harvest forecasts are split into multiple packets for $S > 12$

Type	Payload	Size (byte)	Interval
Energy and ambient report	V_c, I_h, consumption, light, temperature	26	3 min
ORiNoCo statistics	κ, λ, T_fwd	10	10 min
ORiNoCo packet statistics	sent, received, and beacon counters	36	15 min
Configuration	C, V_b	3	3 h
Load adaptation	Q, T_slp, I_n^*, κ, λ, T_fwd	13	once per slot
Harvest forecast	S, S tuples ($\hat{\mu}$, ℓ)	38	1 d

EWMA-filtered slot values, cf. Sect. 2.2. Slot values were smoothed with $\alpha = 0.8$ in compliance with the results from Sect. 5. Slots were not synchronized among the nodes, because each node was started immediately at its deployment, i.e., at an arbitrary time within the overall experiment duration. The maximum-power-point policy $\mathcal{P}_\mathrm{MPP}$ with $V_\mathrm{crit} = 1.2\,\mathrm{V}$ and $V_\mathrm{mpp} = 2.5\,\mathrm{V}$ was chosen. To avoid abrupt decreases of the load, the current voltage V_0 was not used for policy evaluation (cf. Sect. 5.2.4). The forecast horizon was one day.

The queue of ORiNoCo could hold at most 30 packets, which ensured a sufficient amount of RAM for the program stack. A packet history (storing the packet origin and sequence number) of length 30 was used to filter out duplicates. Received beacons with too low signal strength (RSSI values) were discarded to reduce packet loss due to low link quality (the actual value is hardware specific and was determined empirically). After waiting for a beacon for $6 \cdot T_\mathrm{fwd}$, the path weight was reset. The variation parameter in Eq. (11) was $\delta = 0.1$, and the hold time was $T_\mathrm{hld} = 8\,\mathrm{ms}$.

Load adaptation used $I_\mathrm{slp} = 40\,\mu\mathrm{A}$ and $I_\mathrm{rx} = I_\mathrm{tx} = 19.5\,\mathrm{mA}$ based on the consumption profile of common sensor nodes [Haas et al. 2012]. The value of I_slp is the sum of the node's sleeping consumption and the harvester's self-consumption. Consumption $I_\mathrm{rx} = I_\mathrm{tx}$ is the sum of an idle microprocessor, an active radio, and an illuminated red LED (which was used to receive an optical feedback of the ORiNoCo activity). During adaptation, $1 \le Q \le 15$ was enforced to prevent packet drops—i.e., packet loss—if a node runs at maximum Q and receives data from another node also running at maximum Q. To avoid beacon collisions and restrict both packet and forwarding delay, $125\,\mathrm{ms} \le T_\mathrm{slp} \le 5\,\mathrm{s}$ was ensured during adaptation. The initial values after booting were $T_\mathrm{slp} = 500\,\mathrm{ms}$ and $Q = 1$.

The sink ran a modified version of the TinyOS base station application. We replaced the default MAC layer with ORiNoCo running at a fixed $T_\mathrm{slp} = 250\,\mathrm{ms}$. Packets received by the sink were forwarded to a PC via the serial line, where all data was logged.

7.1.3. Network Traffic. Each node in the network periodically created data packets to simulate network load and to collect evaluation data at the same time. Table III lists the types and frequencies of packets created by each node during the field test. In addition to the payload, each packet contained a data sequence number, a routing sequence number, a hop counter, the first seven hops (8 bit node addresses), a time stamp, a delay counter, and the typical TinyOS MAC header. This packet meta information added another 29 B to the packet size. Note that path information (hop counter and hops, accounting for 8 B) was only added for the following evaluation.

7.1.4. Metrics and Methodology. The evaluation of the deployment is based on the logged packets generated by the nodes in the network. Data significance is checked by assessing the number of received and expected packets, i.e., the loss rate is determined. Network statistics are obtained from the routing path information included in the packets.

Node consumption I_n is derived from the consumption readings of the energy-aware software layer. It is compared to the results of the algorithm determining the maximum policy-compliant load I_n^* in terms of their mean value over experiment runtime. To show the improvement of predictive load adaptation versus the depletion-safe variant without harvest forecasts (cf. Sect. 5.1.2), the

Table IV. Deployment and network statistics

node	A	B	C	D	E	F	G	H	I	J	K	L
Deployment statistics												
deployment (d	20.2	29.1	29.1	29.1	26.1	25.1	25.1	25.1	25.1	25.1	18.4	20.4
downtime (%)	—	—	3.3	—	1.9	0.6	—	—	—	—	—	—
depletions	0	0	1	0	1	2	0	0	0	0	0	0
Data packets statistics												
created (1 000)	14.9	21.9	21.2	21.9	18.6	18.7	18.6	18.6	18.6	18.6	13.6	15.1
lost	10	163	136	183	25	30	5	7	130	6	8	17
lost (‰)	0.7	7.4	6.4	8.4	1.3	1.6	0.3	0.4	7.0	0.3	0.6	1.1
avg. path length	2	1	1	1	2	2	3	2	1	2	3	3

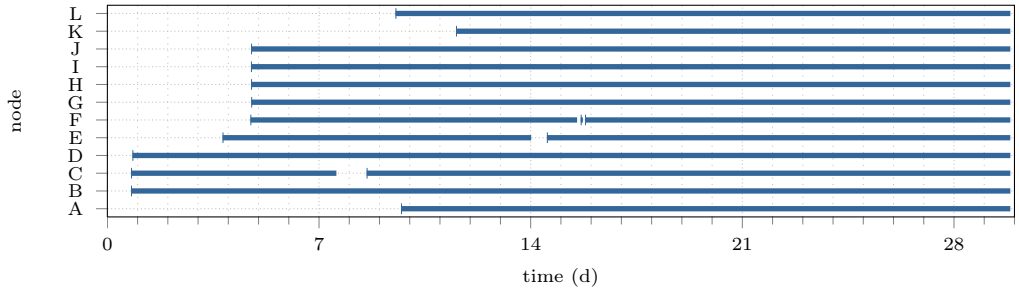


Fig. 18. Deployment time (blue bars) and node downtimes (interruptions of the bars)

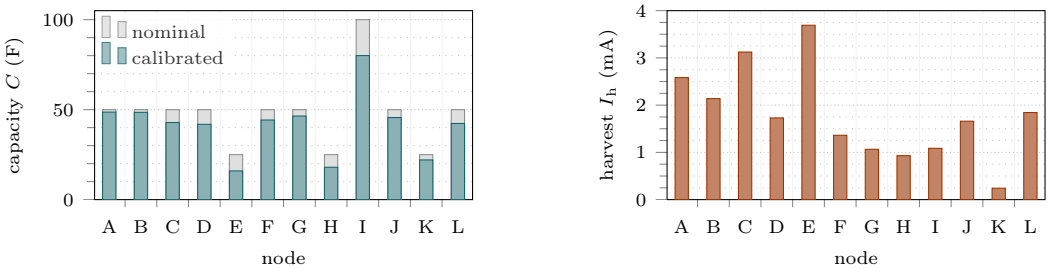
maximum $\mathcal{P}_{\mathrm{DS}}$-compliant load was obtained for $V_0 = V_{\max}$, $V_t = V_{\mathrm{crit}}$, $\Delta t = 24\,\mathrm{h}$. The maximum load at the intermediate voltage $V_0 = 2\,\mathrm{V}$ was also determined for comparison.

Relative and absolute errors between actual consumption I_{n} and maximum policy-compliant load I_{n}^* were obtained with a resolution of 3 min. Load adaptation deltas ΔI_{n}^* (cf. Sect. 5.1.3) are displayed by box plots with median, extreme values, and quantiles specified in the plot captions. Traces of the energy state are used to illustrate notable behavior and to explain results in more detail. Here, all plots have a resolution of 15 min for improved display. All traces show the supercapacitor voltage V_{c}, the maximum policy-compliant load I_{n}^*, the actual consumption I_{n} reported by the software consumption tracker, and the harvest I_{h}.

7.2. Analysis and Evaluation Results

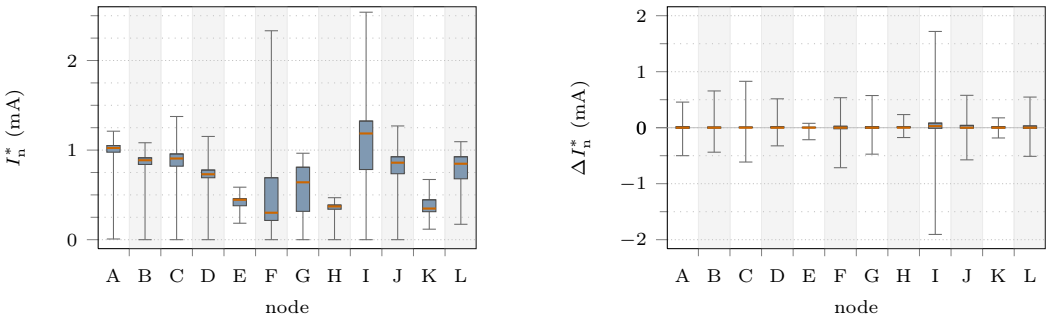
We start with an evaluation of general network statistics. It follows an analysis of the maximum policy-compliant load and actual consumption. A detailed analysis of energy traces is carried out, and the adaptation of ORiNoCo parameters is investigated.

7.2.1. Deployment and Network Statistics. Network statistics of all nodes are summarized in Table IV, and Fig. 18 shows the deployment time of the individual nodes (interruptions indicate times of depletion). It shows that downtimes were particularly low, and depletion occurred only four times in total, of which the two for node F were due to a problem with the solar current sensor software and node reprogramming (see below). The network had an intended depth of up to three hops. Packet loss (not accounting for lost packets due to node downtime) was below 0.9% in all cases and mainly occurred for nodes close to the sink. The large amount of total data, the low loss rate, and the path length variation support the expressiveness of the following evaluation.



(a) capacity (b) harvest

Fig. 19. (a) Comparison of nominal and calibrated capacity and (b) mean harvest



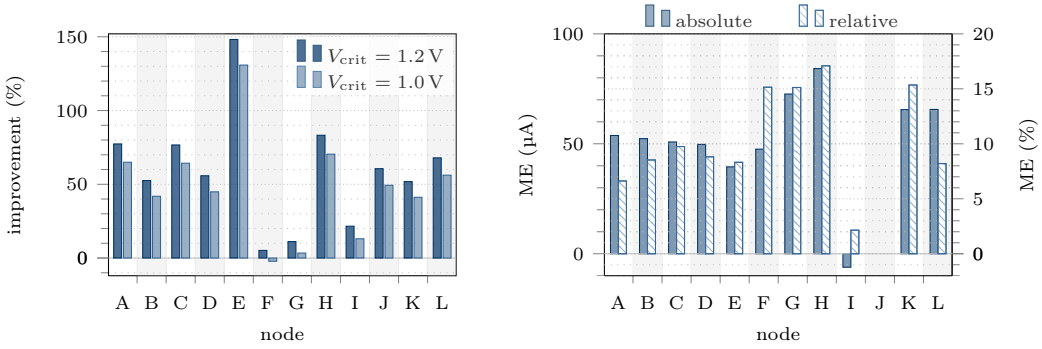
(a) maximum load (b) adaptation deltas

Fig. 20. Distribution of (a) maximum load I_n^* and (b) adaptation deltas ΔI_n^*. Box plots show medians, quartiles, and extreme values

However, the following problems were encountered during the evaluation. Firstly, the values produced by the consumption tracker of node J are invalid due to a loose cable. Node K falsely reported zero harvest most of the time, because its reference light sensor was shielded by a sticker (node label) on its plastic box. Node I suffered from a similar problem, because it was placed inside the building. Although its light sensor was directly pointed (and close) to the window, zero harvest was reported in the early morning and late afternoon while V_c was slightly increasing. Node F was reprogrammed after 15 days due to a mis-configured software image.

The actual capacity and mean harvest influence the performance of predictive load adaptation, as shown in Sect. 5. Figure 19a compares the results of capacity calibration with the nominal values. In some cases, e.g., for nodes A and B, both values are essentially equal. In other cases, there is a large error; e.g, node E has a measured capacity of only 16 F, which is 64% of the nominal value only. Node I has only 80 % of its nominal 100 F. Average harvest is different on the two sides of the building and varies from node to node due to different solar cell orientation. This is evident from Fig. 19b. The low capacity of node E is paired with the highest average harvest, whereas the large capacity of node I meets a low average harvest of only 1 mA.

7.2.2. Maximum Policy-Compliant Load. Analogous to the evaluation in Sect. 5, Fig. 20 shows the distribution of I_n^* and ΔI_n^*. The median of I_n^* ranges from 0.3 mA (node F) to 1.2 mA (node I), see Fig. 20a. Generally, its value reflects the supercapacitor capacity; the low value of node F, compared to nodes with equal capacity, is caused by the erroneous readings of the solar current in the first half of the experiment. Moreover, node G has a relatively low median value of I_n^*, which corresponds to its low harvest. The median values meet the expectations from Sect. 5.2.3. Deviation



(a) improvement over non-forecasting

(b) consumption vs. scheduled load

Fig. 21. (a) Per cent improvement of mean I_{n}^* from the field test with maximum achievable load using $\mathcal{P}_{\text{DS}}$ without harvest forecasts, and (b) mean error (ME) of actual consumption I_{n} and scheduled load I_{n}^* (values for node J are omitted due to wrong consumption values, see Sect. 7.2.1)

was smaller in the field test, because harvest was lower and more stable (compared to the trace used in the simulation). Only for nodes F and G variation is comparable to the simulation results; here, the distance of upper and lower quartile exceeds 0.25 mA. The relatively large variation of node I is small compared to simulation.

Load adaptation deltas, shown in Fig. 20b, are small. For all nodes, the median plus upper and lower quartiles are essentially zero. The range of extreme values is well below the range of I_{n}^* values, indicating that adaptations from highest to lowest load and vice versa did not occur.

In compliance with Sect. 5.2.3, an influence of the number of slots cannot be observed, e.g., the median loads of nodes A and B exhibit a similar relation to their average harvest. Both nodes have almost the same capacity, where node A uses 12 slots and node B uses 24 slots. This finding supports all previous results, stating that 12 slots give an appropriate trade-off between forecast precision and memory consumption. Moreover, fewer slots require fewer adaptation steps. This conclusion is backed by the distribution of ΔI_{n}^*, which does not change from $S = 12$ to $S = 24$.

7.2.3. Benefits from Harvest Forecasts. Simulative evaluation in Sect. 5 has shown that load adaptation profits from harvest forecasts, i.e., the mean load is higher. The field test supports this finding: Figure 21a shows that all nodes run a higher average load than the maximum achievable value without harvest forecasts ($\mathcal{P}_{\text{DS}}$ with $V_{\text{crit}} = 1.2\,\text{V}$). Note that these maximum values correspond to the extreme values shown in Fig. 9a in Sect. 5.2.1. The comparison therefore shows a lower bound on the actually achievable improvement. According to the simulation results, the average profit is likely to rise by an extra 25%, because supercapacitor voltage decreases during the night. The figure additionally shows the same comparison for $V_{\text{crit}} = 1.0\,\text{V}$ (only used without forecasts); in the absence of forecasts, choosing a smaller value of V_{crit} is possible while not significantly increasing the risk of depletion. In this case, the improvement is only 10 per cent points smaller.

7.3. Actual Consumption

The load adaptation scheme from Sect. 6.4.3 reconfigures ORiNoCo to consume the maximum policy-compliant load I_{n}^*. Model simplifications and topology changes, affecting the parameters used for identifying the new configuration, cause a difference between actual consumption I_{n} and I_{n}^*. Figure 21b shows the absolute and relative ME values for all nodes. Absolute values in Fig. 21b are between 40 and $85\,\mu\text{A}$ for all nodes but node I, which corresponds to a relative error of 8% to $17\,\%$. According to existing research in [Hurni et al. 2011], this error is in the range of comparing real (measured) consumption with the results of a software tracker. A significant deviation of I_{n} from I_{n}^* can thus not be concluded. All error values but one are positive, indicating

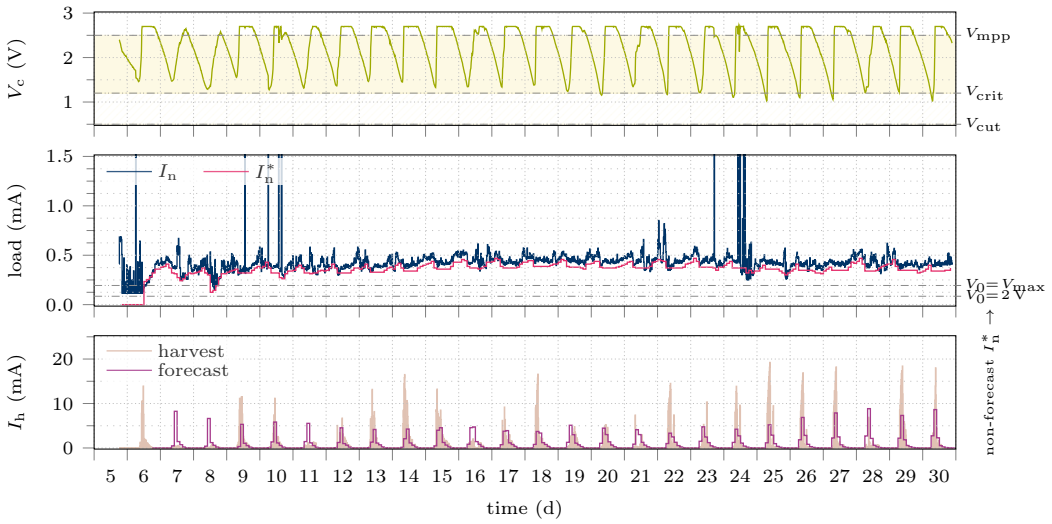


Fig. 22. Energy course of node H in the East of the building with average path weight of $\omega = 2$. Harvest forecasts are based on $S = 12$ slots and it is equipped with a small supercapacitor (nominal 25 F)

a larger consumption than configured on average. Possible influences are the neglected consumption of periodic sensor readings, a difference between actual and measured capacity, and the actual regulator efficiency. A difference between the average queue length and Q also changes consumption; this may happen, e.g., when a node receives more than Q packets consecutively. Topology changes, affecting the parameters κ and λ, are unlikely sources of error, as they would generally cause consumption changes in either direction.

7.4. Detailed Analysis of Energy Traces

To analyze the previous results in more detail, energy courses of two nodes are presented and discussed.

7.4.1. Medium Load and Small Capacity. Figure 22 shows the energy course of node H. Its harvest is high in a short period during the morning and low after noon. Harvest varies from day to day with several poor days. After its deployment on day 5, the node operates at a low consumption of around 0.2 mA. Fluctuating consumption is caused by changing routes and varying traffic, because several nodes were deployed at the same time. The node was deployed close to sunset, so that harvest forecasts were zero and, therefore, $I_n^* \approx 0$ mA until the next day (it is impossible to satisfy $\mathcal{P}_{MPP}$ with zero harvest forecast and $V_0 < V_{mpp}$). During day 6, I_n^* increases due to the improvement of harvest forecasts. The figure reveals that I_n follows the course of I_n^* closely throughout the remaining time but is always slightly in excess. However, the small difference does not cause changes in I_n^*. This implies that actual V_c meets the predicted course. The general stability of I_n^* indicates that $S = 12$ is an adequate choice for enabling stable and depletion-safe operation.

General deviations of I_n from I_n^* within time slots are due to changing values of λ and T_{fwd}, since all nodes in the network adapt their ORiNoCo configuration, affecting the estimated parameters of remote nodes. The node increases the value of I_n^* during night times. Since the harvest forecast does not change in the night, this means that V_c is larger than predicted. The reasons are manifold, e.g., consumption could be smaller than planned (which is yet unlikely due to the consumption trace), capacity could be underestimated, or the regulator efficiency could be better than configured. The few peak consumptions, e.g., on day 10, stem from missed beacons after initiating packet transmission: ORiNoCo requires a transmitting node to wait for an appropriate beacon. This results in long forwarding delays of several seconds.

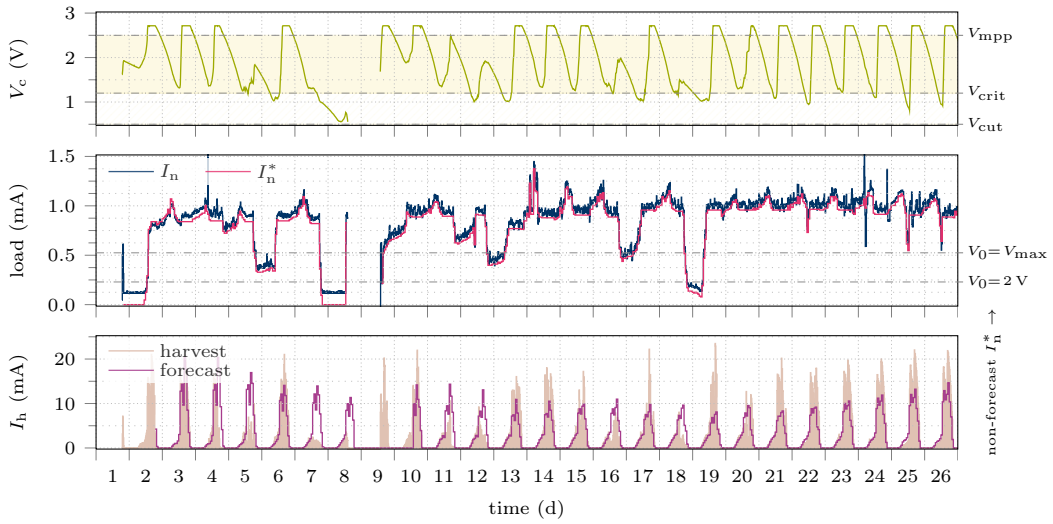
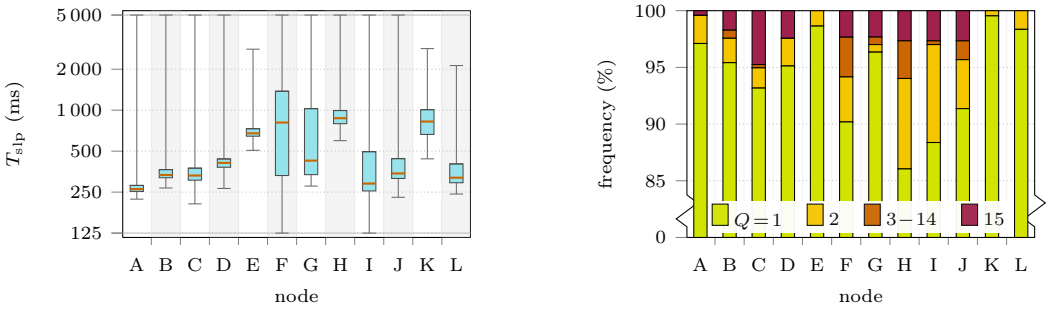


Fig. 23. Energy course of node C in the West of the building. The node has a capacity of (nominal) 50 F

Supercapacitor voltage V_c covers the full range down to V_{crit}, which is only undercut on few days. This implies that the algorithm determining I_n^* works fine in practice and gains from harvest forecasts. The latter is backed by the trace of I_n, which is above the maximum achievable value of I_n^* using $\mathcal{P}_{DS}$ without forecasting. Only after two consecutive days of poor harvest (on days 7 and 8), I_n^* drops for a few hours. The effect is yet limited due to the choice of $V_{mpp} = 2.5\,V$, which permits policy compliance for smaller harvest and larger loads compared to $V_{mpp} = V_{max}$. Moreover, not including V_0 in policy evaluation avoids abrupt drops of I_n^* after day 24 for the short periods of $V_c < V_{crit}$. A small capacity and short periods of high harvest lead to long times of $V_c = V_{max}$, which is evident from the figure. This is caused by the demand for a uniform load in combination with the policy requirement $V_c > V_{crit}$. While energy is generally wasted, simplicity and stability of load adaptation are gained.

7.4.2. Causes of Depletion. The previous traces convey the benefits of predictive load adaptation and the practical applicability. As reported in Sect. 5, depletion may not be completely preventable. To analyze the causes, Fig. 23 shows the energy course of node C, which depleted around noon of day 8. In general, the trace exhibits similar aspects as the previous ones, yet a combination of influencing factors caused depletion. Firstly, high harvest on the first days leads to overestimated energy intake on day 7. In consequence, V_c drops to a level slightly above V_{crit}, where it is trapped closely before sunset: Because the forecast is to good, the node believes it can still satisfy $\mathcal{P}_{MPP}$. In particular, the voltage prediction yields $V_c > V_{crit}$ at all times within the 24 h prediction horizon. This misjudgment leads to $V_c < V_{crit}$ and $I_n^* \approx 0\,mA$ shortly after sunset. Yet, actual consumption I_n is larger than this target value, because Q and T_{slp} have upper bounds (cf. Sect. 7.1.2) and define a minimum consumption w.r.t Sect. 6.4.2. Despite the mismatch of I_n^* and I_n, the node does not deplete during the night. Closely after noon of day 8, V_c approaches V_{crit}, so that the voltage prediction, caused by the minorly revised and hence still too good harvest forecast, satisfies $\mathcal{P}_{MPP}$ with $I_n^* \approx 0.9\,mA$ (the level of the previous day). Since supercapacitor voltage V_c is low, the regulator supply current I_r exceeds I_h, cf. Eq. (2), and results in depletion.

The major cause of depletion is the false harvest forecast. To prevent depletion, improved methods for harvest forecasting are required. Until such methods are available, an energy policy with an additional voltage threshold for the current voltage V_0 should be used (cf. Sect. 5.2.4).



(a) sleep interval (b) queue threshold

Fig. 24. (a) Distribution of T_{slp} in semi-logarithmic scale and (b) frequency of occurrence of Q-levels

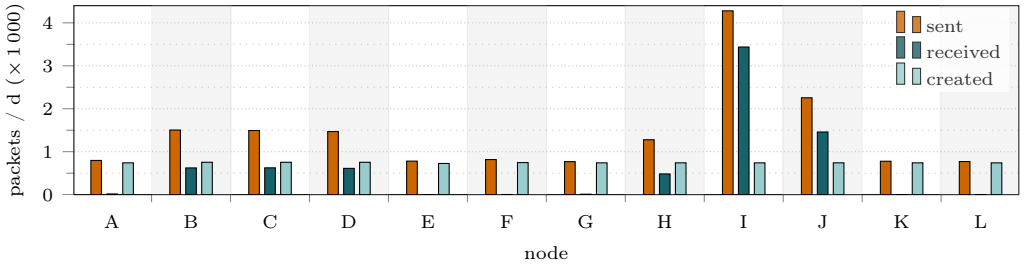


Fig. 25. Data creation and traffic statistics of all nodes

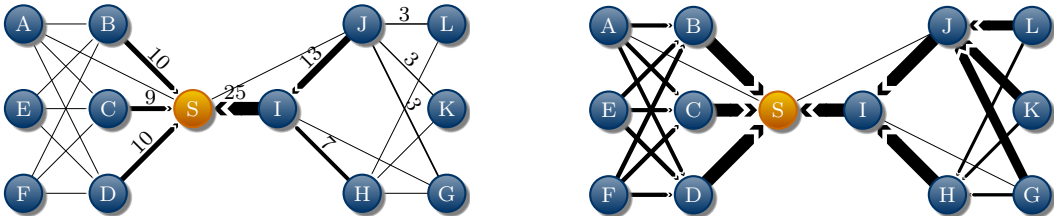
In addition to the explained weakness, the trace of node C shows another issue. After day 21, supercapacitor voltage V_{c} underruns V_{crit} almost every day despite the stable harvest pattern. This may be caused by, e.g., an overestimated capacity or the low resolution of voltage predictions. The specific cause could not be determined from the recorded data. Yet, the observation shows that parameter recalibration—meaning particularly capacity recalibration—are mandatory in a real deployment.

7.5. ORiNoCo Parameter Adaptation

In addition to the feasibility study of the predictive load adaptation algorithm, the suggested method for ORiNoCo reconfiguration was assessed.

The distribution of T_{slp} in Fig. 24a is generally inverse to that of I_{n}^{*}. As expected, the ability to manage a higher load allows a node to sleep less. The queue threshold Q took its minimum value ($Q = 1$) between 86% and 99% of the time, as shown in Fig. 24b. Values larger than 2 occurred only at the beginning of the experiment (when the expected harvest was zero and V_{c} approached V_{crit}) and on subsequent days with poor harvesting conditions. Although changes of Q occurred infrequently, they were required on some days, thus supporting the argument in Sect. 6.4.3. The achieved energy savings—increasing Q from 1 to 2 cuts sending energy expenditure roughly into half—indicates its usefulness over only adjusting the sleep interval T_{slp}. Moreover, doubling Q rather than doubling T_{slp} achieves the same effect while not affecting the consumption of other nodes, cf. Sect. 6.4.3.

Result variation stresses the difficulty of choosing a single pair of values T_{slp} and Q that works for all nodes. Guaranteeing depletion-safe operation implies low harvest utility with a static setup, if it is possible to find such setup prior to deployment at all.



(a) absolute link usage (b) link usage per sender

Fig. 26. Link statistics: (a) overall utilization and (b) traffic split per sender. Line with scales with frequency, only links with at least 0.1% utilization are shown for clearness

The daily average number of created, received, and sent data packets is shown in Fig. 25. Nodes B through D exhibit a similar packet load: they have equally sized capacitors and forward the data of nodes A, E, and F to the sink in equal shares. Node H and J, in contrast, have close positions but different packet loads. Load adaptation has relieved the burden of node H, which has a smaller supercapacitor and a smaller mean I_{n}^{*}. Node I had the highest packet load of all nodes, because it was the only connection between the sink and nodes on the East side of the building.

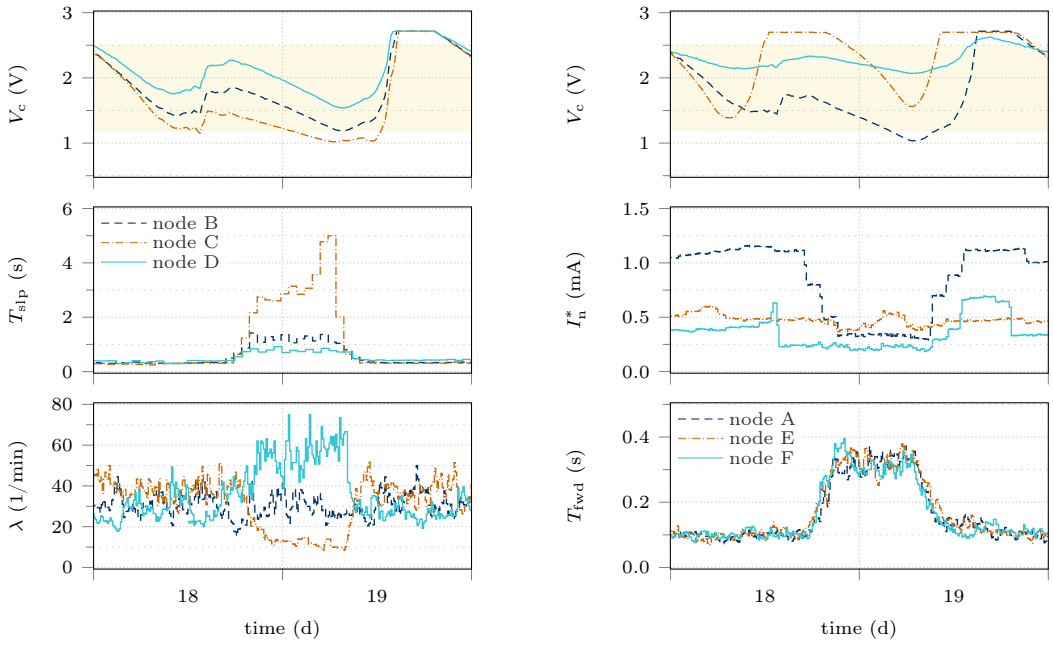
These results are reconfirmed by the link load statistics in Fig. 26. Node I sent 25% of all data packets in the network, whereas nodes B through D were able to share the load of their sub-network almost equally. Figure 26a proves that nodes with a hop count of $\omega = 2$ received a share of data packets from nodes with $\omega = 3$ (on the same side of the building) corresponding to their capacity relations. Figure 26b emphasizes the opportunistic concept of ORiNoCo: Nodes spread their packets to all remote nodes with a smaller value of ω. Combined with predictive load adaptation, the routing load is automatically balanced w.r.t. the energy situation by adapting the value of T_{slp}.

In addition to long-term packet-flow balancing, the combination of ORiNoCo with load adaptation enables automatic short-term packet flow balancing without the need of control traffic. Figure 27 illustrates such a case. Poor weather conditions leave nodes B through D with $V_{\mathrm{c}} < V_{\mathrm{mpp}}$ in the afternoon of day 18 in Fig. 27a—the voltage of node C is particularly low. Load adaptation forces nodes to sleep longer (T_{slp} is increased). While T_{slp} of nodes B and D is roughly doubled, node C sleeps six to ten times longer than before. In consequence, node C receives less data (its λ is decreased by a factor of four). The remaining packets have to be handled by nodes B and D. However, their actual consumption is generally not affected, because I_{n}^{*} does not depend on λ and load adaptation reconfigures ORiNoCo for the new situation. The chance of packet loss due to depletion of a node with a non-empty packet buffer is decreased, since packets are routed by nodes in a better energy situation. In addition, nodes closer to depletion forward less data and their risk of depletion is reduced, because the probability of missing beacons—causing high consumption over an extended and unforeseeable period of time—is reduced (cf. Sect. 7.4.1).

The same situation does not threat other nodes—in particular, those forwarding packets to nodes B through D—because they adapt to the new situation. Figure 27b shows the behavior of nodes A, E, and F. These nodes quickly identify the increased forwarding delay T_{fwd}. As nodes forward their data to any (parent) node with a smaller value of ω, these nodes experience the same increase of T_{fwd}. Despite the changes, the maximum load I_{n}^{*} stays relatively stable. Since the harvest forecast does not change after sunset (when slots and EWMA-based forecasts are used), this observation indicates that the nodes' consumption is maintained when reconfiguring ORiNoCo and well-aligned with the maximum policy-compliant load.

7.6. Summary and Discussion

A real-world case study in a multi-hop sensor network was conducted. Twelve sensor nodes periodically created data packets that were forwarded to a single sink using the ORiNoCo protocol. They were powered by the harvester prototype and traced their energy state. Predictive load adap-



(a) parent nodes (b) child nodes

Fig. 27. Automatic packet-flow balancing in the tree-like routing structure: (a) parent nodes with hop count $\omega = 1$ and (b) child nodes with $\omega = 2$

tation was achieved through harvest forecasts and the maximum-power-point energy policy, where the ORiNoCo protocol was reconfigured dynamically and online to meet the maximum policy-compliant load. A concrete method for this reconfiguration was devised and discussed.

A four-week field test was conducted and evaluated to show the practicability of the approaches pursued in this paper and to assess the quality of combining a simple method of load adaptation with a sensor network collection protocol of low overhead w.r.t. control traffic. In particular, the following major results were obtained.

— The presented consumption model of the ORiNoCo protocol showed to be a valid ground for practical load adaptation. Nodes were generally able to adjust their load with low error compared to the maximum policy-compliant load. Moreover, the field test has shown that load adaptation is not restricted to a single, isolated node, but also works in a multi-hop network. Here, load adaptation has effectively and implicitly balanced the packet flow w.r.t. available energy resources of individual nodes.
— The concept of energy policies works well in practice. All nodes were enabled to run at a relatively uniform load on subsequent days with constant weather and harvest conditions. Only on days with low harvest, abrupt load adaptations were required. The average achievable load of the nodes was improved by up to 150% when using harvest forecasts. This improvement yet came at an increased risk of depletion. A way to circumvent this problem was discussed.
— Predictive load adaptation exhibits the behavior expected from simulation results. This observation revalidates that the devised energy-flow model gives a sufficiently accurate picture in combination with the employed methods of energy assessment. Moreover, the expressiveness of simulation results in this paper is stressed.

In conclusion, it has been shown that the devised concept of predictive load adaptation with energy policies and the method for practical consumption adaptation of a multi-hop sensor network data-collection protocol, have been effectively put into practice. Weaknesses were identified, and possible solution strategies were pointed out.

8. CONCLUSION

We developed a novel method for online load adaptation for energy-harvesting sensor nodes. It consists of two building blocks. Firstly, the energy-flow model of the harvester prototype was turned into an algorithm for predicting the prospective energy reserve of the supercapacitor in terms of its terminal voltage for a given consumption and a given harvest forecast. Secondly, the concept of energy policies was introduced. An energy policy defines predicates to enforce properties of operation style of a sensor node, e.g., to enforce depletion-safety. The combination of these two components enables to identify the maximum policy-compliant load (or consumption). While this concept is generally tailored to harvesters using a supercapacitor as energy buffer, it can be transferred to other hardware. Two energy policies were suggested, of which the maximum-power-point policy yielded the highest achievable consumption at the smallest risk of depletion. In this context, the influence of capacity and forecast errors was studied through simulation. Larger capacities (of $50\,\mathrm{F}$ and above) completely prevented depletion and increased the achievable load. With the constraint of accomplishing uniform consumption, less energy was wasted in periods of harvest excess.

A real-world case study was conducted through a multi-hop sensor network. Twelve nodes were equipped with the harvester prototype and executed the online load adaptation algorithm. Multi-hop data collection was achieved with the ORiNoCo protocol. The maximum policy-compliant load was employed to adjust the consumption of ORiNoCo. This was achieved by modeling the consumption of ORiNoCo and adjusting its duty cycle and data forwarding policy. With a four-week deployment, the results of the previous simulation were confirmed, and the practicability of the algorithm was attested on real hardware. In particular, all nodes were able to adjust their consumption to avoid depletion in most cases while elevating the average load on days with high harvest.

The devised method and its final integration with the ORiNoCo protocol constitute an out-of-the-box software architecture for near-perpetual data collection. It is compatible with energy-harvesting power supplies equipped with a supercapacitor and an energy source with a cyclic harvest pattern. This combination of hardware and software keeps manufacturing, deployment, and maintenance costs little. The low complexity and small parameter set enables facile and practical setup of multi-hop sense-and-send applications by non-experts. This is an important step towards enabling mass-market sensor network deployments.

In the course of evaluation, we identified pointers to future work. Increasing forecast quality has the potential to completely wipe out the risk of depletion while improving the achievable load. Here, integrating reduced weather forecasts, possibly only consisting of cloud-cover information, will improve forecast accuracy. The concept of energy policies and predictive load adaptation may profit from longer forecast horizons in terms of more uniform and predictable routing load. This would have a positive impact on large-scale networks, where unexpected or abrupt changes of the routing load may amplify routing-related consumption to a degree that leads to depletion. A current surplus at times of a full energy buffer is currently not used. Methods to utilize this unused energy without affecting the consumption of other nodes should be devised and investigated. Practical load adaptation—e.g., of a low-power MAC, such as ORiNoCo—could be refined by feeding the actual consumption into the adaptation process. Expanding its operational area to delay-constrained networks also constitutes a research challenge of practical relevance. Finally, we plan to compare our approach with existing research.

REFERENCES

Mustafa Ali, Bashir Al-Hashimi, Joaquin Recas Piorno, and David Atienza. 2010. Evaluation and Design Exploration of Solar Harvested-Energy Prediction Algorithm. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '10)*.

Carlo Bergonzini, Davide Brunelli, and Luca Benini. 2010. Comparison of Energy Intake Prediction Algorithms for Systems Powered by Photovoltaic Harvesters. *Microelectronics Journal* (Nov. 2010).

Davide Brunelli, Clemens Moser, Lothar Thiele, and Luca Benini. 2009. Design of a Solar-Harvesting Circuit for Batteryless Embedded Systems. *IEEE Transactions on Circuits and Systems I (TCAS-I): Regular Papers* 56, 11 (Nov. 2009).

Conrad Electronics. 2008. Product Specification: Conrad Miniatur Solar Cell (Type YH-39X35). http://www.produktinfo.conrad.com/datenblaetter/175000-199999/191308-da-01-en-SOLARZELLE_4V_35_MA.pdf. (2008). Last accessed: 2012/04/23.

Anand Eswaran, Anthony Rowe, and Raj Rajkumar. 2005. Nano-RK: An Energy-Aware Resource-Centric Operating System for Sensor Networks. In *Proceedings of the 26th IEEE Real-Time Systems Symposium (RTSS '05)*.

Kai-Wei Fan, Zizhan Zheng, and Prasun Sinha. 2008. Steady and Fair Rate Allocation for Rechargeable Sensors in Perpetual Sensor Networks. In *Proceedings of the 8th ACM International Conference on Networked Sensing Systems (SenSys '08)*.

Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. 2009. Collection Tree Protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys '09)*.

Christian Haas, Viktor Stöhr, and Joachim Wilke. 2012. Realistic Simulation of Energy Consumption in Wireless Sensor Networks. In *Proceedings of the 9th European Conference on Wireless Sensor Networks (EWSN '12)*.

Jason Hsu, Sadaf Zahedi, Aman Kansal, Mani Srivastava, and Vijay Raghunathan. 2006. Adaptive Duty Cycling for Energy Harvesting Systems. In *Proceedings of the 12th International Symposium on Low Power Electronics and Design (ISLPED '06)*.

Philipp Hurni, Torsten Braun, Benjamin Nyffenegger, and Anton Hergenröder. 2011. On the Accuracy of Software-based Energy Estimation Techniques. In *Proceedings of the 8th European Conference on Wireless Sensor Networks (EWSN '11)*.

Xiaofan Jiang, Joseph Polastre, and David Culler. 2005. Perpetual Environmentally Powered Sensor Networks. In *Proceedings of the 4th ACM/IEEE International Symposium on Information Processing in Sensor Networks (IPSN '05)*.

Aman Kansal, Jason Hsu, Sadaf Zahedi, and Mani Srivastava. 2007. Power Management in Energy Harvesting Sensor Networks. *Transactions on Embedded Computing Systems (TECS)* (Sept. 2007).

Vasileios Kyriatzis, Nicholas Samaras, Pavlos Stavroulakis, Haifa Takruri-Rizk, and Stergios Tzortzios. 2007. Enviromote: A New Solar-Harvesting Platform Prototype for Wireless Sensor Networks. In *Proceedings of the 18th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '07)*.

Olaf Landsiedel, Euhanna Ghadimi, Simon Duquennoy, and Mikael Johansson. 2012. Low Power, Low Delay: Opportunistic Routing meets Duty Cycling. In *Proceedings of the 11th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN '12)*.

Jiakang Lu and Kamin Whitehouse. 2012. SunCast: Fine-grained Prediction of Natural Sunlight Levels for Improved Daylight Harvesting. In *Proceedings of the 11th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN '12)*.

Clemens Moser, Davide Brunelli, Lothar Thiele, and Luca Benini. 2007. Real-Time Scheduling for Energy Harvesting Sensor Nodes. *Real-Time Systems* 37, 3 (Dec. 2007).

Clemens Moser, Lothar Thiele, Davide Brunelli, and Luca Benini. 2008. Robust and Low Complexity Rate Control for Solar Powered Sensors. In *Proceedings of the Conference on Design, Automation and Test in Europe (DATE '08)*.

Clemens Moser, Lothar Thiele, Davide Brunelli, and Luca Benini. 2010. Adaptive Power Management for Environmentally Powered Systems. *Transactions on Computers (TC)* 59, 4 (April 2010).

Chulsung Park and Pai Chou. 2006. AmbiMax: Autonomous Energy Harvesting Platform for Multi-Supply Wireless Sensor Nodes. In *Proceedings of the 3rd Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '06)*.

Vijay Raghunathan, Aman Kansal, Jason Hsu, Jonathan Friedman, and Mani Srivastava. 2005. Design Considerations for Solar Energy Harvesting Wireless Embedded Systems. In *Proceedings of the 4th ACM/IEEE International Symposium on Information Processing in Sensor Networks (IPSN '05)*.

Joaquin Recas Piorno, Carlo Bergonzini, David Atienza, and Tajana Simunic Rosing. 2009. Prediction and Management in Energy Harvested Wireless Sensor Nodes. In *Proceedings of the 1st International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (VITAE '09)*.

Christian Renner, Florian Meier, and Volker Turau. 2012a. Holistic Online Energy Assessment: Feasibility and Practical Application. In *Proceedings of the 9th IEEE International Conference on Networked Sensing Systems (INSS '12)*.

Christian Renner, Florian Meier, and Volker Turau. 2012b. Policies for Predictive Energy Management with Supercapacitors. In *Proceedings of the 8th IEEE International Workshop on Sensor Networks and Systems for Pervasive Computing (PerSeNS '12)*.

Christian Renner and Volker Turau. 2010. CapLibrate: Self-Calibration of an Energy Harvesting Power Supply with Supercapacitors. In *Workshop Proceedings of the 23th International Conference on Architecture of Computing Systems (ARCS '10)*.

Christian Renner and Volker Turau. 2012a. Adaptive Energy-Harvest Profiling to Enhance Depletion-Safe Operation and Efficient Task Scheduling. *Sustainable Computing: Informatics and Systems* 2, 1 (March 2012).

Christian Renner and Volker Turau. 2012b. State-of-Charge Assessment for Supercap-Powered Sensor Nodes: Keep it Simple Stupid!. In *Proceedings of the International Workshop on Algorithms and Concepts for Networked Sensing Systems Powered by Energy Harvesters (EnHaNSS '12)*.

Navin Sharma, Jeremy Gummeson, David Irwin, and Prashant Shenoy. 2010. Cloudy Computing: Leveraging Weather Forecasts in Energy Harvesting Sensor Systems. In *Proceedings of the 7th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '10)*.

Navin Sharma, Pranshu Sharma, David Irwin, and Prashant Shenoy. 2011. Predicting Solar Generation from Weather Forecasts Using Machine Learning. In *Proceedings of the 2nd IEEE International Conference on Smart Grid Communications (SmartGridComm '11)*.

Farhan Simjee and Pai Chou. 2006. Everlast: Long-Life, Supercapacitor-Operated Wireless Sensor Node. In *Proceedings of the 12th International Symposium on Low Power Electronics and Design (ISLPED '06)*.

Yanjun Sun, Omer Gurewitz, and David Johnson. 2008. RI-MAC: A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks. In *Proceedings of the 8th ACM International Conference on Networked Sensing Systems (SenSys '08)*.

Texas Instruments. 2009. Datasheet: TPS 61220. http://focus.ti.com/lit/ds/symlink/tps61220.pdf. (2009). Last accessed: 2012/04/01.

Stefan Unterschütz, Christian Renner, and Volker Turau. 2012. Opportunistic, Receiver-Initiated Data-Collection Protocol. In *Proceedings of the 9th European Conference on Wireless Sensor Networks (EWSN '12)*.

US Department of Energy. 2011. Solar irradiance recordings. http://www.nrel.gov/midc. (2011). Last accessed: 2012/05/14.

Christopher Vigorito, Deepak Ganesan, and Andrew Barto. 2007. Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks. In *Proceedings of the 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '07)*.

Alex Weddell, Geoff Merrett, Tom Kazmierski, and Bashir Al-Hashimi. 2011. Accurate Supercapacitor Modeling for Energy-Harvesting Wireless Sensor Nodes. *Transactions on Circuits and Systems II (TCAS-II): Express Briefs* 58 (Dec. 2011). Issue 12.

Bo Zhang, Robert Simon, and Hakan Aydin. 2010. Energy Management for Time-Critical Energy Harvesting Wireless Sensor Networks. In *Proceedings of the 12th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS '10)*.

Bo Zhang, Robert Simon, and Hakan Aydin. 2011. Harvesting Aware Energy Management for Time-Critical Wireless Sensor Networks with Joint Voltage and Modulation Scaling. *Transactions on Industrial Informatics (TII)* PP, 99 (2011).

Ting Zhu, Ziguo Zhong, Yu Gu, Tian He, and Zhi-Li Zhang. 2009. Leakage-Aware Energy Synchronization for Wireless Sensor Networks. In *Proceedings of the 7th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '09)*.

Received March 2013; revised November 2013; accepted January 2014