ODMAC++: An IoT Communication Manager based on Energy Harvesting Prediction

Samuel Perez
École Polytechnique
91128 Palaiseau, France
samuel@perez.pro

Juan Antonio Cordero Fuertes
École Polytechnique,
91128 Palaiseau, France
juan-antonio.cordero-fuertes@polytechnique.edu

Marceau Coupechoux
LTCI, Telecom ParisTech
Univ. Paris-Saclay, Paris, France
marceau.coupechoux@telecom-paristech.fr

***Abstract*—In large low-power networks of battery-driven sensors, power outages are a major concern and communication rates have to be carefully designed in order to optimize energy consumption, network connectivity and sensors lifetime. In some IoT use cases, power can be supplied to sensors by way of renewable energy automatic harvesting (solar panels, etc.). Given the high variability of energy arrival processes, energy consumption in sensors, in particular caused by transmissions to the sink, has to be aligned with energy harvesting patterns, so as to maximize throughput while avoiding power outages that may arise when the battery is empty. This paper proposes ODMAC++, an extension to a well-known protocol for sensor transmission scheduling in a WSN. ODMAC++ relies on learning techniques to adapt sensors communication rate to energy harvesting patterns, and uses a beaconing mechanism whose frequency is adjusted based on past measurements on the harvested energy process. Simulations based on analytical energy arrival models and on real solar radiation measurements indicate that ODMAC++ is able to avoid power outages and to cope with battery limitation and energy variations due to variability in time.**

I. Introduction

The basic idea of Internet of Thing (IoT) can be summarized as follows: enable connectivity between physical objects (“things”) and the Internet, *e.g.* by way of wireless communication to a gateway (or *sink*). As these devices require energy and typically cannot be “plugged” into the electric grid, the problem of power supply soon arises. Renewable energy harvesting appears as an attractive solution in this context. By using energy collectors like solar panels or piezoelectric harvesters, connected objects can be supplied reliable power over time without any human interaction [1]. Renewable energy arrival processes are however usually highly variable in space and time, and sometimes unpredictable, so that energy-harvesting devices have to manage accurately their power consumption when they communicate.

Several MAC protocols for energy harvesting objects have been proposed in the literature. Unlike traditional MAC protocols, throughput and latency are not the only performance parameters – energy consumption also plays a crucial role. This paper addresses the case of Energy Harvesting Wireless Sensor Networks (EH-WSN), in which multiple sensors communicate with one sink. An overview of the main protocols designed for this model is presented in [2]. In [3], a probabilistic poll protocol is proposed: the sink broadcasts a contention probability that is used by all sensors to decide whether to transmit. If a sensor runs out of energy, it saves power for a few cycles before being able to transmit again. The contention probability is dynamically adjusted: it decreases in case of collisions and increases if the channel stays idle during a cycle. This protocol is fair, but slow to converge; thus it is not appropriate for fast-changing environments. DeepSleep [4] is a classical CSMA/CA protocol with sensor specific contention window sizes. When a sensor runs out of power, it goes into a *deep sleep* state so as to save energy. After a sleeping time, the contention window is reduced to compensate for the long delay induced by the sleep. This protocol lacks fairness, and it can have a long latency if several successive deep sleeps occur for a node. LEB-MAC [5] is a receiver-initiated protocol, which adapts the communication duty-cycle of the nodes to their available energy. ODMAC [6], similar to LEB-MAC, is an on-demand (*i.e.,* sensors are allowed to transmit only if the sink demands sensor transmissions) and energy-aware (duty-cycle of the devices is based on the energy available in their battery) MAC protocol. ODMAC, for which this paper proposes an extension, is greedy: when more energy is available, the throughput systematically increases, without anticipating possible power outages in the future.

None of these protocols addresses the issue of power outage. This can be however crucial in some key applications, *e.g.* military or critical infrastructure management. This paper proposes a new energy management approach to prevent power outage, based on ODMAC and denominated ODMAC++. ODMAC++ is a prediction-based communication manager: it explores the use of measurements of energy available to minimize the difference between power production and consumption. By anticipating excess and scarcity, such an approach would enable devices to consume at equal rates, increasing its reliability. Although only tested in specific conditions, ODMAC++ shows resiliency to quick power changes and can follow long term evolutions of energy arrival processes.

The paper is structured as follows: Section II describes the principles of ODMAC and ODMAC++, and introduces the ODMAC++ models for energy harvesting and power consumption. Section III presents four different energy management strategies – this Section includes the main contributions of the paper. In Section IV, performance of ODMAC++ is evaluated with simulations. Section V concludes this paper.

II. ODMAC++

A. ODMAC Protocol

This section describes ODMAC [6] operation in the case of N_s sensors transmitting packets to a single sink.

The sink periodically broadcasts (with frequency f_{com}) a beacon indicating sensors that communication may start. If the channel stays idle for T_{IFS}, the sink goes back to a sleeping state. In order to reduce sensor energy consumption, sensors listen to the channel only when they have packets to send. On receiving a beacon, sensors compete to send their information using a classical backoff procedure with contention window and ACK/NACK from the sink. Depending on the energy available in the battery, the duty-cycle of sink and sensors may change according to a reference energy level. For sinks, the beaconing frequency may change. For sensors, the sensing frequency may change. For the case of multiple sinks, ODMAC includes an opportunistic forwarding mechanism. This mechanism has little interest, though, if devices are located in the same area and harvest from the same energy source (*e.g.* the sun), as in this case communication rates and battery states are similar for all sensors. This paper thus addresses the case of an homogenous network with a single sink.

ODMAC makes no assumption about the energy process and has no mechanism to avoid energy outage.

B. ODMAC++ Principles

ODMAC++ is an extension of ODMAC that includes: (i) a learning module that estimates the energy process statistics, and (ii) an energy management system that adapts the communication frequency to the energy statistics.

With these two elements, the main difference between ODMAC++ and ODMAC is that the former takes into account the periodicity of the energy process. Many power sources like wind, sun or tide indeed provide energy periodically, so that the average amount of energy per period becomes predictable. Typical periods duration of the energy process, T, is denominated *energy cycle* and is assumed to be known in advance in this protocol. For example, solar energy exhibits an energy cycle of $T = 24$ hours. The objective of ODMAC++ is to predict the amount of energy to be received, in order to prevent power outage: if sufficiently accurate, this prediction makes possible to plan a uniform consumption for high and low energy periods, with no energy loss.

In this paper, only the case of homogenous EH-WSN is considered[1]: sensors have the same battery size and power consumption. For simplicity, it also assumed that all devices harvest the same energy at all time. This approximates the case of equal sensors located in the same area, with relatively small variations in the state of batteries, and entails that all sensors have the same battery level at all times.

ODMAC++ operation can be described as follows:

- The energy cycle is divided into N slots of equal duration T_m, *i.e.,* $T = N \times T_m$, for sensor measurement purposes. Measurements about harvested energy are piggybacked in each packet sent to the sink, and stored by the sink stores in a temporary buffer. After T_m seconds, the mean value is stored in a prediction buffer and the temporary buffer is flushed. If no packets have been received from a device during T_m, it is assumed that the device ran out of energy and it should not be considered for predictions for the next energy cycle.
- ODMAC++ starts with a learning phase that can be launched, for instance, when deploying the nodes. The sink remains idle during N_i energy cycles (*i.e.,* during $N_i \times T$ time) in order to make measurements of harvested energy and deduce the average and standard deviation of the harvested energy for every slot.
- The sink starts sending beacons to the sensors at a specific frequency, $f_{com}(t)$. From this point, an energy management strategy is used to adapt this frequency $f_{com}(t)$ to the energy process estimated from the learning phase onwards. The frequency $f_{com}(t)$ is likely to be modified at the beginning of every energy cycle. Two strategies are proposed in Section II-C.
- As time goes on, the learning process continues and improves the accuracy of the estimation of the energy process. At the beginning of every energy cycle, the beacon frequency of the sink is adjusted based on new measurements. If the energy process is stationary, it converges to a stable value on the long term. To prevent energy outage with high probability, a margin is taken on the expected harvested energy during a slot based on the measured standard deviation. This makes ODMAC++ robust to uncertainties.
- In order to deal with long term changes (*e.g.* seasons for solar energy) and so with the non-stationarity of the observed process, the measurements are stored into a sliding window of duration T_W. This window should be large enough to get a reliable value on the average energy, but small enough to adapt to long term changes.

C. Energy and Throughput Models

This section describes the throughput and energy process models that are used for performance evaluation (see Section IV). For these models, time is slotted with a granularity Δt:

• *Simple model*. Devices harvest solar energy, which is globally periodic with small fluctuations due to weather changes. Instantaneous energy available is thus modeled as a function sinusoidal at day, and null (zero) at night, added to a standard Gaussian random variable $X_g \sim N(\mu = 0, \sigma = 1)$ to model weather variations; X_g is renewed every ω sec (*i.e.,* $X_g(t) \neq X_g(t + \omega)$).

$$P_a(t) = \max(0, \sin(2\pi \cdot t/T) + \sigma_e X_g(t)), \tag{1}$$

where $P_a(t)$ is the harvested power, t is the time in seconds and σ_e is the standard deviation of the process.

• *Real data*. Evaluation is also performed with solar power measurements. A dataset from the National Renewable Energy

[1]The case of heterogeneous EH and traffic pattern is left for future work.

Laboratory [7] is used; this provides 5-minutes ($\Delta t = 300$ sec) average solar radiation data, collected in Itta Bena, Mississippi from 1997 to 2005.

Power consumption is modeled considering two cases. When sensors and sinks are idle during a slot, they consume a constant energy $c_i\Delta t$, where c_i is a constant corresponding to the idle power consumption (in W). When communications take place, the energy consumed during a slot is proportional to the number of received beacons, *i.e.,* $c_{com} f_{com} \Delta t$, where c_{com} is a constant corresponding to the unit communication cost (in J). Energy consumption during a slot, $C(t)$, is therefore:

$$C(t) = (c_i + c_{com} \cdot f_{com}(t)) \cdot \Delta t, \tag{2}$$

Combining the simple harvesting energy model and the consumption model, the energy available in the battery, E_b, evolves in a timeslot $[t, t + \Delta t]$ as follows:

$$\frac{E_b(t + \Delta t) - E_b(t)}{\Delta t} = P_a(t) - \frac{C(t)}{\Delta t} \tag{3}$$

with $0 \leq E_b \leq E_{bmax}$, E_{bmax} being the battery size.

Sensors compete for the channel at every beacon transmission. It can be assumed, without loss of generality, that every sensor can send a single packet when receiving the beacon and that collisions are resolved well before the next beacon arrives. With these assumptions, the throughput at the sink, $\eta(t)$, is calculated in packets per second as $\eta(t) = N_s f_{com}(t)$.

The communication rate is modified depending on the energy management strategy, as detailed in Section III.

III. Energy Management Strategies

Four possible energy management strategies are presented in this section. For each of them, it is described how the communication frequency $f_{com}(t)$ can be controlled as a function of the available/forecasted harvested energy so as to maximize the throughput and/or avoid power outages.

A. Constant Frequency

This strategy serves as a benchmark for others and assumes a constant frequency, *i.e.,* $\forall t, f_{com}(t) = F$, where F is a constant. Many energy-harvesting receiver oriented protocols [3], [4] use this strategy.

B. Proportional Frequency

Communication frequency is set proportional to the amount of energy present in the battery: $f_{com}(t) = \alpha \cdot E_b(t)$, where α is the proportionality coefficient. This approach is a simplified version of the energy management system of ODMAC.

C. Predictive Freq. for Sufficiently Large Batteries (PF-SLB)

If batteries are sufficiently large, available energy can be estimated as the difference between harvested and consumed. The frequency is then derived, for each energy cycle, from Eqs. (2) and (3), so that the energy stored in the battery remains the same after an energy cycle T – *i.e.,* the communication is energy-neutral over T. Then, the optimal communication frequency, $f^*_{com}(t)$, is as follows:

$$f^*_{com}(t) = \frac{N \cdot T_m \sum_{k=0}^{N} E_a(t + k \cdot T_m) - c_i}{c_{com}} \tag{4}$$

In practice, the energy arrival process, E_a, is not deterministic; $E_a(t + kT_m)$ in every slot can only be estimated with the average value of measurements obtained during the learning phase and past cycles (*i.e.,* $E_a(t + k \cdot T_m)$ is replaced by $\mathbf{E}[E_a(t + k \cdot T_m)]$, where $\mathbf{E}$ is the measurements empirical average). In order to take into account measurements variability and reduce the probability of power outages, a suboptimal frequency value is used:

$$f_{com}(t) = \frac{N \cdot T_m \sum_{k=0}^{N} (\mathbf{E}[E_a(t + k \cdot T_m)] - \beta \cdot \sigma_k) - c_i}{c_{com}}, \tag{5}$$

where σ_k is the measured standard deviation of the energy process in slot k. The factor β allows to bound the probability of power outage (the higher, the less likely is a power outage, *e.g.* $\beta = 2$ implies 2.5% probability of outage for a Gaussian power distribution with average $\mathbf{E}[E_a(t + k \cdot T_m)]$).

This approach is simple and efficient, but is only appropriate if batteries are never (or seldom) filled (see Section III-E).

D. Predictive Frequency for Generic Batteries (PF-GB)

When batteries are often full, the difference between harvested and consumed energy overestimates the available energy. Instead, frequency is selected after simulating the energy profile in the battery during the next energy cycle T. Starting from the energy present at the beginning of the cycle, $E_b(t + k \cdot T_m)$ is computed by way of the Euler method: $E_b(t + (k+1)T_m) = E_b(t + k \cdot T_m) + dE_k$ where:

$$\begin{cases} dE_k = \min(V_k(f_{com}), E_{bmax} - E_b(t + k \cdot T_m)) \\ V_k(f) = \mathbf{E}[E_a(k \cdot T_m)] - \beta \cdot \sigma_k - (c_i + f \cdot c_{com}) \cdot T_m \end{cases}$$

Frequency $f_{com}(t)$ is determined by using *recursive dichotomy*. Given a minimum precision ϵ and a range of suitable frequencies $[f_{\min} = 0, f_{max}]$, the recursion starts at $f_0 = \frac{1}{2}(f_{\min} + f_{\max})$, and the energy profile of the battery for f_0 is simulated for one cycle T. In an iteration i, with $f_i \in [f_{\min,i}, f_{\max,i}]$, if there is a slot k for which $E_b(k \cdot T_m) \leq 0$, the candidate frequency is too high and $f_{i+1} = \frac{1}{2}(f_{\min,i} + f_i)$; otherwise, it may be too low and next simulated frequency is $f_{i+1} = \frac{1}{2}(f_i + f_{\max,i})$. When the desired precision ϵ is achieved, 95% of the final frequency is returned, to keep a safety margin. In order to preserve energy-neutrality, the condition $E_b(t + N \cdot T_m) \geq \min\{E_b(t), E_0\}$ has to be respected, for some constant E_0 ($E_0 = \frac{1}{2} E_{bmax}$, experimental value).

E. Discussion

PF-SLB has lower computational complexity –and thus, lower energy consumption due to computation– than PF-GB, but shows a poorer performance when the battery is often full (or, equivalently, if the battery has small size). In this case, it is likely that the battery is unable to store a significant fraction of available energy – the fraction that was received when full, and therefore PF-SLB entails overconsumption, because it does not foresee potential energy capping. PF-GB allows to overcome this case, and is thus appropriate for small batteries and suitable for any size of batteries; the use of one or the other will thus depend on the device characteristics.

IV. Simulations

This section describes ODMAC++ performed simulations.

A. Simulation Settings

The simulation scenario consists of $N_s = 10$ sensors sending data to a single sink. Time is slotted (with slot length Δt) and the throughput within a slot is assumed constant. Results are presented for simulations based on energy arrival analytic models (Secs. IV-B, IV-C and IV-D), and based on real energy measurements (Sec. IV-E). Simulation parameters are shown in Table I; parameters in the right define the analytic model for the former.

TABLE I
Simulation Parameters.

Parameter	Value
Simulation	
N_s	10
Δt	60 sec, 300 sec
F	0.05, 0.2, 2 Hz
α	10^{-6}, $2 \cdot 10^{-5}$, 0.1
ODMAC++	
T_m	1 hour
ϵ	0.01
N_i	2
β	1

Parameter	Value
Energy and throughput model	
T	24 hours
σ_e	1/4
c_i	0.1 W
c_{com}	0.1 J
E_{bmax}	40000 J
ω	1 hour

It is assumed that (a) sensors have always a packet to transmit (full buffer assumption), (b) the sink has perfect knowledge of sensors battery state, and (c) all sensors are identical (in battery, power consumption and harvested energy).

Figures in this section depict, from top to bottom, energy arrival process (top, in W/m^2), stored energy (*i.e.,* the state of the battery) (middle, in J), and the achieved throughput (bottom, in packets per second). Represented throughput values are averages over timeslots of duration Δt ($\Delta t = 60$ sec for simulated data, $\Delta t = 300$ sec for measured solar data).

B. Constant Frequency

Figs. 1, 2 and 3 display the results of the constant frequency mechanism. Three main cases can be observed.

1) If the communication frequency is too low (Fig. 1), the battery is filled quickly, and a lot of energy is lost because of the maximum battery capacity. There is no outage in this situation. Over one week, with a frequency of $f_{com} = 0.5$ Hz, the throughput is $\eta = 42,9 \cdot 10^3$ packets per day.

2) If the frequency is "good" (Fig. 2), *i.e.,* the daily consumption approximately matches the amount of harvested energy, power outages can occur at night. Over one week, with a frequency of $f_{com} = 5$ Hz, the cumulated throughput is $\eta = 216 \cdot 10^3$ packets per day.

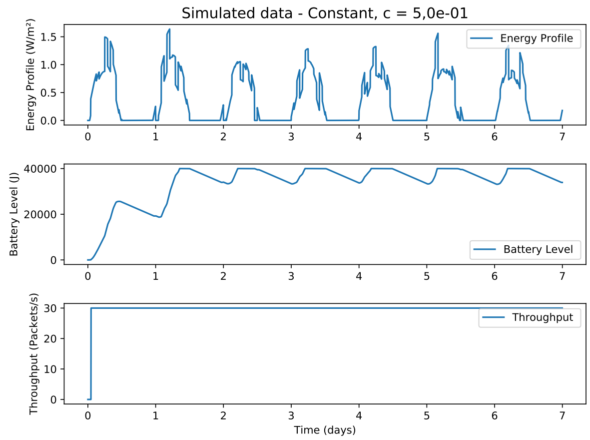

Fig. 1. Constant frequency: low frequency.

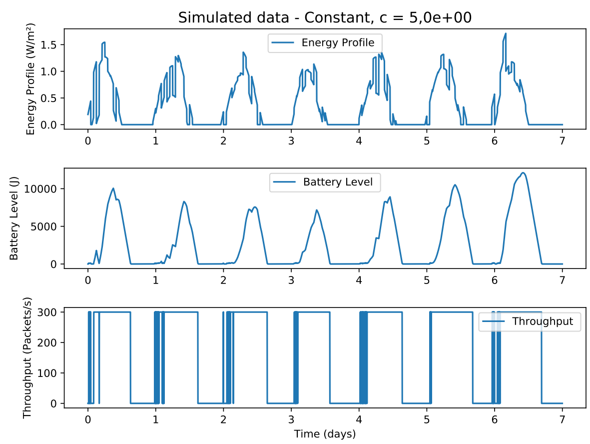

Fig. 2. Constant frequency: intermediate frequency.

3) If the frequency is too high (Fig. 3), the situation is unstable. When there is energy available, the battery is quickly exhausted. Then the device stays down, the battery slightly recharges and the cycle restarts. Over one week, with $f_{com} = 20$ Hz, the cumulated throughput is $\eta = 245 \cdot 10^3$ packets per day but the system also exhibits a lot of outage situations.

If the frequency is constant, power outage can occur quite often. This solution ensures a constant communication frequency when there is sufficient power. Since the frequency is not adaptable, a calibration should be performed at the beginning to avoid having too low or too high frequency. Of course, this simulation does not take into account possible power recovery mechanisms of protocols such as Probabilistic Poll or DeepSleep. However, rather than prevent outage, these mechanisms would only decrease instability in the third case, without changing the overall conclusion.

C. Proportional Frequency

Figs. 4, 5 and 6 display the performance of a proportional frequency strategy, and illustrate the impact of the proportionality coefficient, α, on the system performance (see Section IV-C). Recall that α could be considered as the reference energy parameter in ODMAC.

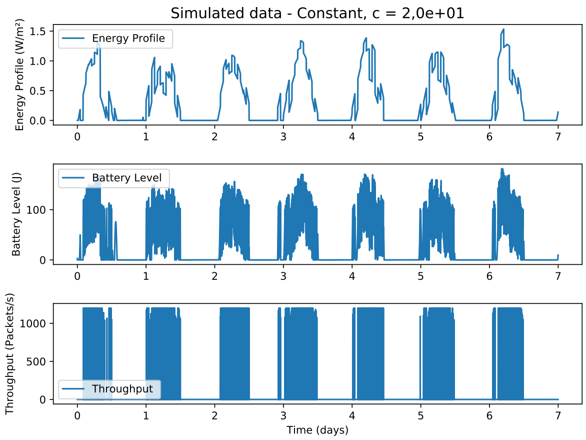

Fig. 3. Constant frequency: high frequency.

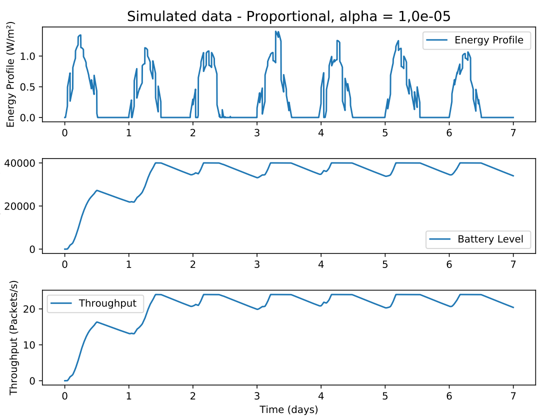

Fig. 4. Proportional frequency: $\alpha = 1/10^5$.

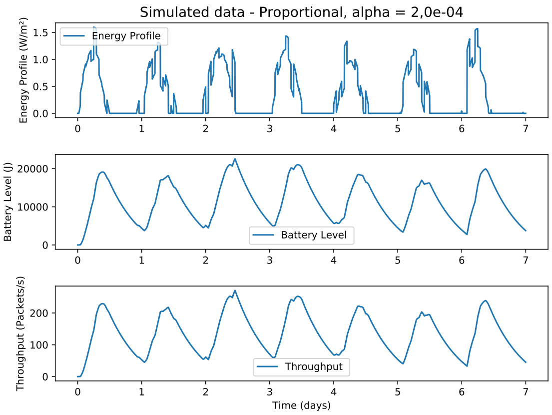

Fig. 5. Proportional frequency: $\alpha = 1/(5 \cdot 10^3)$.

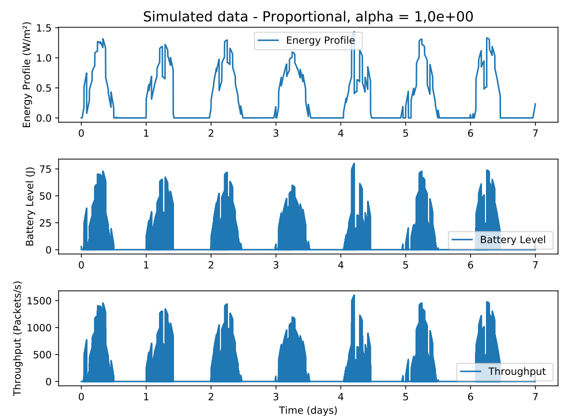

Fig. 6. Proportional frequency: $\alpha = 1$.

Three main cases can be highlighted:

1) When α is too small (Fig. 4), a lot of energy is not harvested because of the capacity of the battery. The consumption is low and so is the throughput. Over one week, with $\alpha = 1/10^5$, the cumulated throughput is $\eta = 29.9 \cdot 10^3$ packets/day.

2) An intermediate state is reached for higher values of α (Fig. 5). In this case, almost all the energy available is consumed, without outages or losses. This choice of α is acceptable, however, the throughput is completely unbalanced between day and night. Over one week, with $\alpha = 1/(5 \cdot 10^3)$, the average throughput is $\eta = 175 \cdot 10^3$ packets per day, an acceptable value.

3) When α is too high, the unstable situation experienced with constant frequency is observed (Fig. 6): the battery switches repeatedly from on to off at a high rate. In simulations, the throughput is still high, but it may not be useful in reality, because of the unstability and the high proportion of outage situations. Over one week, with $\alpha = 1$, the cumulated throughput is $\eta = 270 \cdot 10^3$ packets per day.

From these results, it can be observed that better performance can be achieved if the frequency is proportional to the harvested energy. This is due to the fact that the device can adapt to the harvested energy, which is not the case with a constant frequency. However, the choice of α remains crucial and should be calibrated. Moreover, this system is greedy, and does not save any power during the day to use at night, leading to energy outages.

D. Predictive Frequency: ODMAC++

Figs. 7 and 8 present the results of ODMAC++ with the sunlight analytical model introduced in Section II-C. After two days of measurements, the sink starts to broadcast beacons. The two proposed implementations for ODMAC++, predictive frequency with sufficiently large batteries (PF-SLB) and with generic batteries (PF-GB), are evaluated.

1) ODMAC++ with Sufficiently Large Batteries (PF-SLB): In Fig. 7, the equilibrium duty-cycle is reached rapidly. On a period of 28 days, the cumulated throughput is $4.4 \cdot 10^6$, or $\eta = 160 \cdot 10^3$ packets per day. The throughput in ODMAC++ is higher than in proportional frequency with no outage.

Even in case of long scarcity, no power outage occurs. This implementation is simple and the throughput is stable over time. However, the battery level is variable, and the size of the battery can drastically change the behaviour of the algorithm. In case of very small batteries, the quantity of storable energy is limited, so that the devices can use less energy than what is predicted. This causes over-optimistic

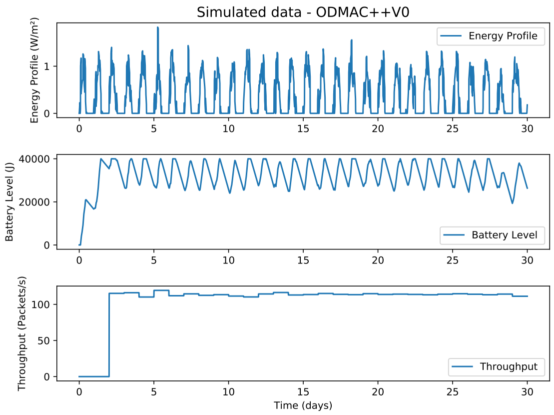

Fig. 7. ODMAC++ for sufficiently large batteries (PF-SLB).

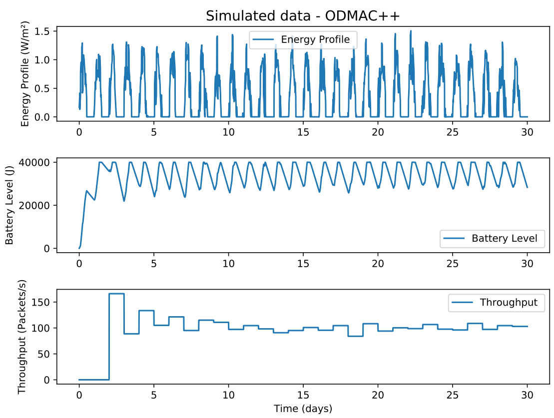

Fig. 8. ODMAC++ with generic batteries (PF-GB).

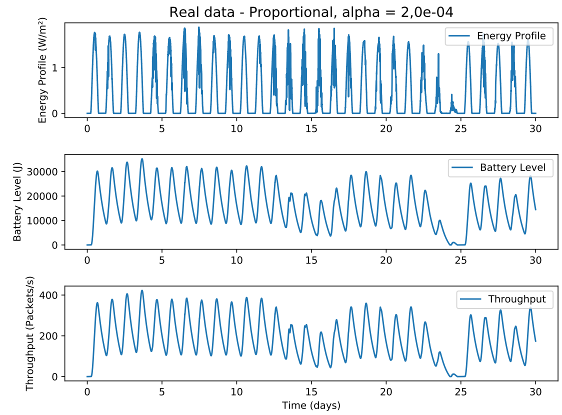

Fig. 9. Real data : proportional frequency.

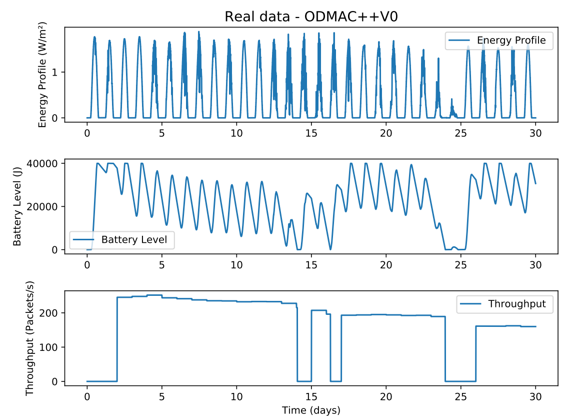

Fig. 10. Real data : ODMAC++ for sufficiently large batteries (PF-SLB).

predictions and fast power outage. Those issues motivate the design and implementation of a second solution.

2) ODMAC++ with Generic Batteries (PF-GB): In Fig. 8, the throughput over 28 days is $175 \cdot 10^3$ packets per day. The results show that the throughput becomes more variable, but the battery level is also more stable.

E. Experiments with Real Data

ODMAC++ performance is also tested with real data [7]. The data covers solar measurements during september, october and november 2005. To simulate a real sensor in this environment, the effective energy received, $E_a(t)$, is computed as $E_a(t) = P_{solar}(t) \cdot S \cdot \gamma$, where $P_{solar}(t)$ is the available power, S the surface of the solar panel, and γ the conversion rate of the solar panel. In the simulation, selected values are $S = 10^{-2}\ m^2$ and $\gamma = 0.2$.

The results on this data are shown in Fig. 9, 10 and 11 for proportional frequency, ODMAC++ PF-SLB and PF-GB implementations respectively. The results show that ODMAC++ achieves good performance on real data and is less variable than a simple proportional frequency. The cumulated throughput for 28 working days is $8.2 \cdot 10^6$ packets for proportional frequency with $\alpha = 1/5000$, $7.4 \cdot 10^6$ packets for ODMAC++ PF-SLB (sufficiently large batteries), and $7.6 \cdot 10^6$ packets for ODMAC++ PF-GB (generic-batteries) implementation. The implementation for generic batteries achieves better throughput. However, its real advantage is that it can deal with small-sized batteries. Fig. 12 shows the throughput results with a 4 times smaller battery than in the previous case. The PF-GB implementation shows almost no outage whereas the implementation for sufficiently large batteries is very unstable.

1) Sliding window: A sliding window was implemented to take into account long term evolutions, like seasons. This sliding window has two goals. First, it allows dealing with non-stationary energy processes: for example, it does not make sense to use summer data during winter. If the size of the sliding window has the same order of magnitude as the non-stationary process, the current average energy will be much more accurate. However, the smaller the sliding window, the bigger the standard deviation, so there is a trade-off to cope with that is left for further work. Second, the long term evolutions make standard deviation increase with time. The sliding window keeps standard deviation to a reasonable value.

Results with a sliding window of size 15 days with ODMAC++ PF-GB are shown in Fig. 13 and 14. Performance can change dramatically with or without sliding window. Maintaining a small sliding window makes the system adapt much better during winter time, avoiding many power outages.

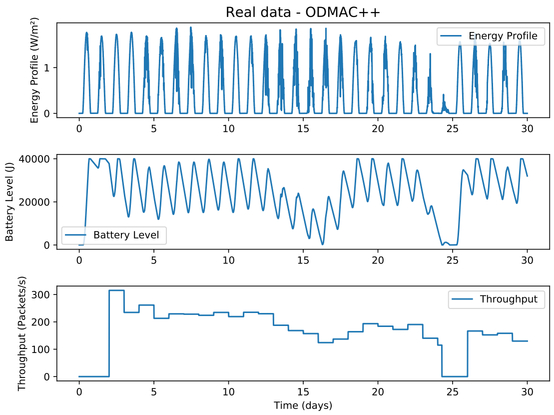

Fig. 11. Real data : ODMAC++ with generic batteries (PF-GB).

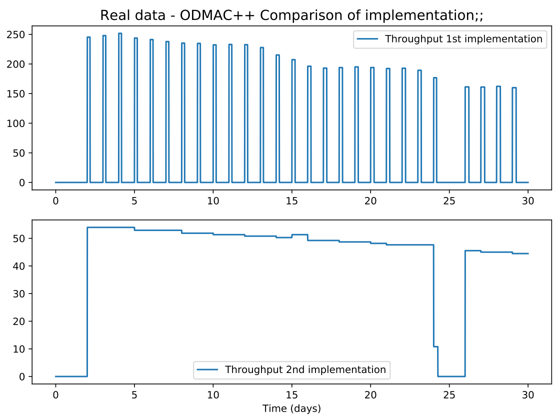

Fig. 12. Small battery case with ODMAC++.

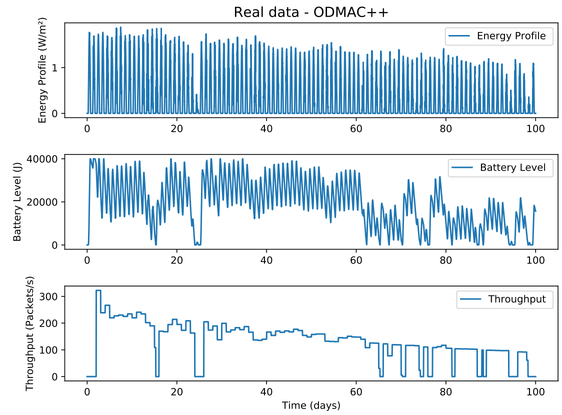

Fig. 13. ODMAC++ PF-GB without sliding window.

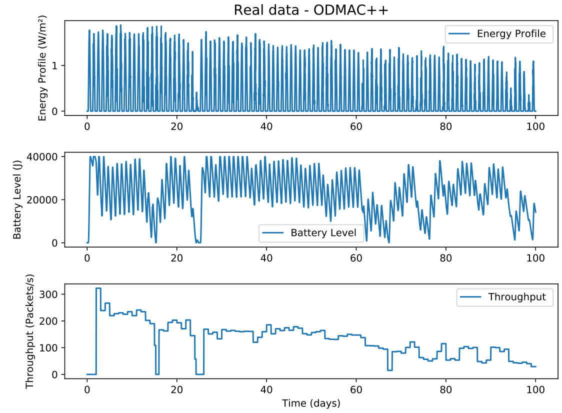

Fig. 14. ODMAC++ PF-GB with sliding window.

V. Conclusion

This paper presents ODMAC++, an improvement of the On Demand MAC protocol (ODMAC) for Energy Harvesting WSNs. ODMAC++ learns energy harvesting patterns from sensor statistics and dynamically adapts the frequency of the beacon sent by the sink to allow sensors to transmit. ODMAC++ objective is to maximize the beacon frequency while avoiding power outages. The paper proposes two possible implementations of ODMAC++ predictive frequency (PF). PF-SLB is energy-neutral over the cycle and computationally simple, but assumes that the battery size is large with respect to the amount of harvested energy. PF-GB is slightly more complex, but avoids power outages even with small batteries. Long term changes in the statistics of the energy process can be handled by way of a measurement sliding window. As ODMAC++ achieves good reliability, it could be used for critical applications, *e.g.* military ones. The assumption that all devices harvest the same energy is simplistic, and future work should generalize ODMAC++ for less constrained networks.

Acknowledgements

This work was supported in part by the Cisco-Polytechnique Chaire Internet of Everything (http://www.Internet-of-Everything.fr/).

References

[1] W. K. G. Seah, Z. A. Eu, and H. P. Tan, "Wireless sensor networks powered by ambient energy harvesting (wsn-heap) - survey and challenges," in *2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology*, pp. 1–5, May 2009.

[2] P. Ramezani and M. R. Pakravan, "Overview of mac protocols for energy harvesting wireless sensor networks," in *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 2032–2037, Aug 2015.

[3] Z. A. Eu, H.-P. Tan, and W. K. Seah, "Design and performance analysis of {MAC} schemes for wireless sensor networks powered by ambient energy harvesting," *Ad Hoc Networks*, vol. 9, no. 3, pp. 300 – 323, 2011.

[4] H.-H. Lin, H.-Y. Wei, and R. Vannithamby, "Deepsleep: Ieee 802.11 enhancement for energy-harvesting machine-to-machine communications," in *2012 IEEE Global Communications Conference (GLOBECOM)*, pp. 5231–5236, Dec 2012.

[5] H. I. Liu, W. J. He, and W. K. Seah, "Leb-mac: Load and energy balancing mac protocol for energy harvesting powered wireless sensor networks," in *2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 584–591, Dec 2014.

[6] X. Fafoutis and N. Dragoni, "Odmac: An on-demand mac protocol for energy harvesting - wireless sensor networks," in *Proceedings of the 8th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, PE-WASUN '11, (New York, NY, USA), pp. 49–56, ACM, 2011.

[7] C. N. F. R. Energy, "Solar energy resource data, mississipi valley state university." http://rredc.nrel.gov/solar/new_data/confrrm/mv/, 2005.