Article:

Bozorgchenani, A. orcid.org/0000-0003-1360-6952, Disabato, S., Tarchi, D. et al. (1 more author) (2020) An energy harvesting solution for computation offloading in Fog Computing networks. Computer Communications, 160. pp. 577-587. ISSN 0140-3664

https://doi.org/10.1016/j.comcom.2020.06.032

An Energy Harvesting Solution for Computation Offloading in Fog Computing Networks*

Arash Bozorgchenani[a], Simone Disabato[b], Daniele Tarchi[a,*], Manuel Roveri[b]

[a] Department of Electrical, Electronic and Information Engineering, University of Bologna, Italy
[b] Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy

Abstract

Fog Computing is a promising networking paradigm enabling the nodes at the edge to share computational and storage resources. Being pervasively distributed, Fog Nodes are often battery powered and, for this reason, an efficient energy management should be considered to prolong network lifetime. In this paper, we introduce a smart energy management solution able to exploit information about the predicted harvested and consumed energy by Fog Nodes, equipped with small solar panels. The smart energy management is applied on a cluster based Fog Computing environment where computation offloading operations are performed. In the experimental section the effect of the smart energy management is explored in terms of network lifetime by considering variable battery size and Fog Nodes density in a realistic solar-panel harvesting-model and Fog Nodes setting.

Keywords: Fog Computing, Energy Harvesting, Computation Offloading, Clustering, Internet of Things.

1. Introduction

The last decades have been characterized by an increasing number of pervasive devices that have been deployed in the environment to support a wide range of applications, from environmental monitoring [1] and in-home automation [2] to Smart-cities [3] and Industry 4.0 [4]. Such devices, operating under the technological scenario of Internet-of-Things (IoT) [5], are generally characterized by constraints on memory, computation and energy consumption. Hence, information gathered from the environment by the IoT units are typically transmitted to the Cloud for storage and processing. Despite being very popular, this computational paradigm is not suitable when the latency in making a decision or activating a reaction is very strict (or at least constrained in time) or when the connection between the IoT devices and the Cloud is intermittent or limited in bandwidth.

To overcome these issues a new computational paradigm, called Fog Computing (FC) [6, 7], has been recently introduced. The novel approach brought by FC is the possibility to move computation as close as possible to the data generation source, i.e., the IoT devices, hence, providing a distributed framework of computing resources. This allows to acquire and process information locally, hence supporting prompt reactions/decisions and making FC units autonomous without requiring a permanent connection with the Cloud computing platform. For these reasons, FC enables novel application and technological scenarios allowing pervasive processing and analysis. Among these scenarios, the *computation offloading* is one of the most promising [8, 9, 10]. In computation offloading, a node has the possibility to offload, completely or partially, a processing task to be executed to another node. This would bring several advantages (e.g., the ability to exploit FC nodes characterized by larger energy availability) but, at the same time, introduces several challenges in terms of management complexity, energy distribution, nodes coordination, just to name a few.

The FC scenario under consideration is composed by a set of FC nodes, called Fog Nodes (FN), scattered in a given area. Such nodes are battery-powered, since we assume no or reduced connection to the electrical network. In such a scenario, we aim at prolonging the network lifetime, i.e., the time span a certain amount of nodes or the whole network runs out of energy, by introducing and managing energy harvesting mechanisms in the FC nodes. In particular, in this paper we are interested in prolonging the network lifetime of battery-powered FNs by leveraging the presence of energy harvesting devices (e.g., solar panels) when they offload data/code to be processed to other FNs. This could be particularly important for those applications in which the human intervention could be an issue. In particular we will focus on a distributed intensive processing scenario, e.g., distributed image processing, where the sensors are able to collect data and share

*This work has been partially supported by the project “GAUChO - A Green Adaptive Fog Computing and Networking Architecture” funded by the MIUR Progetti di Ricerca di Rilevante Interesse Nazionale (PRIN) Bando 2015 - grant 2015YPXH4W.

*Corresponding author

Email addresses: arash.bozorgchenani2@unibo.it (Arash Bozorgchenani), simone.disabato@polimi.it (Simone Disabato), daniele.tarchi@unibo.it (Daniele Tarchi), manuel.roveri@polimi.it (Manuel Roveri)

for processing among themselves.

Our goal is to exploit the prediction of the required and harvested energy to be spent/acquired by each FN over time to support the optimization of the offloading among FNs so as to prolong the network lifetime. The considered application scenario foresees the presence of pervasive FNs able to organize themselves into clusters. This organization has been demonstrated to be effective in terms of both distributed processing and energy saving, allowing also to reduce the overall delay [11, 12]. The novel contribution of this work can be summarized as:

- Designing a green FC environment where the FNs are equipped with solar panels and capable of recharging their batteries from the solar energy.
- A harvesting prediction method is proposed in order to avoid the network to go off in the future by predicting the energy consumption of the nodes based on their harvesting pattern.
- Differently from previous works, we have considered the energy management among the devices by changing the role of the nodes between requesting devices and computing devices, when necessary, in order to prolong the network lifetime.
- A clustering algorithm is considered based on the energy status of the devices, so that the cluster members in each cluster offload their tasks to the cluster head for computation, by taking into account both consumed and harvested energy.

The proposed energy management algorithm for computational offloading has been tested and evaluated in a synthetic scenario but with real data about the energy harvesting profiles in Northern Italy [13].

The rest of the paper is organized as follows. In Section 2, a literature review is given in the area of FC with a particular attention on energy harvesting applications and cluster organizations. In Section 3, the system model is described by focusing on the energy-related operations and their impact on the battery capacity. In Section 4, the prediction-based harvesting algorithm is detailed, and the clustering overlay is introduced, whereas the experimental results are given in Section 5. Finally, in Section 6 the conclusions are drawn.

2. Related Works

FC has been recently investigated by taking into account different aspects and points of view. Due to the importance of the energy issues in this area, many of the works have taken it into consideration.

An Energy-Efficient Computation Offloading (EECO) algorithm is introduced in [14] and relies on three main phases: classifying the nodes considering their energy and cost feature, prioritizing them by giving a higher offloading priority to the nodes which cannot meet the processing latency constraint, and the radio-resource allocation of the nodes considering the priority. The proposed EECO algorithm allows to decrease the energy consumption up to 18% w.r.t. computation without offloading. A heuristic offloading decision algorithm was proposed in [15] with the aim of maximizing system utility which considers task completion time and the FN energy consumption in a single server Mobile Edge Computing (MEC) scenario. The authors in [16] proposed energy-efficient offloading policies for transcoding tasks in a mobile cloud system. With the objective of minimizing the energy consumption while meeting the latency constraints, they introduced an online offloading algorithm which decides whether the task should be offloaded to the Cloud or executed locally. Task processing in [17] was based on a decision of either local processing or total offloading. The authors aimed at minimizing the local execution energy consumption for applications with strict deadline constraints. Energy consumption has also been targeted in [18] for an offloading approach. In this work, the authors targeted energy consumption and response time for offloading to the Cloud. In [19], the authors aim at maximizing the number of offloaded tasks for battery-charged nodes to reduce power consumption. In their proposed heuristic, they obtain the optimal transmission power and adjust the offloading decisions. The authors in [20], considered resource management and workload allocation in a fog-cloud environment. They have proposed learning based algorithms to balance the power consumption and reduce workload processing delay at the edge.

The solutions concerning energy harvesting, especially with solar panels, have been widely studied in literature. The authors in [21, 22] extensively studied all the technological details and challenges involving the adoption of solar panels (and/or other energy harvesting techniques). The authors in [23] studied the problem of unpredictability associated with renewal energies (including the solar one) to ensure the quality of a service in an edge computing system. The authors in [24] developed a real Wireless Sensor Network (WSN) with nodes capable of acquiring data at high frequencies and, at the same time, are equipped with solar panels. Similar works can be found in [25, 26, 27].

An almost novel approach in energy harvesting is wireless energy transfer, that has been considered in different scenarios. In [28], the authors designed a mechanism to assign tasks to Unmanned Aerial Vehicles by taking into account also the possibility to recharge them when they are at the base station. Similarly, [29] considered the energy harvesting through wireless when offloading tasks to the nodes of an Internet of Things network.

In [30], a dynamic algorithm is proposed, aiming at minimizing energy costs by leveraging dual energy sources, solar and grid power, to support the FNs. The Lyapunov technique has been considered to design algorithms in Cloud of Things system. In [31] the authors took into account the social relationships of energy harvesting devices for computation offloading to a fog base station, where they proposed a game theoretic approach with the objective of minimizing the execution cost.

An approach having similarities with computation offloading, named in-network processing, has been also proposed in the last years [32, 33, 34, 35], where the focus is in WSNs, able to interact for cooperatively processing data generated by themselves. In particular, in [32, 35], energy harvesting is also considered. In this paper, differently from [32, 35] we are aiming at jointly exploiting the energy consumption and harvesting profiles, in order to dynamically adapt the cluster formation for coping with the offloading requests by the FNs composing the network.

Clustering in edge networking has also been proposed in some works. In [36] clustering was performed among the access points considering channel and caching status. A clustering algorithm was also proposed in [37] for the radio access points dealing with joint computation and communication resource allocation inside the cluster. The same problem has been also addressed in [38]. In this work, whenever a user has a packet to offload, the computing cluster is assigned to it. The main idea is to jointly compute clusters for all active users' requests simultaneously in order to distribute computation and communication resources among the users. A similar approach has also been proposed in [39]. A mobile edge computing clustering algorithm is also proposed in [40], which aims at maximizing the traffic handled inside the clusters and reduce the traffic going out to the core network data servers. Channel conditions of the IoT devices have been considered for creating clusters in [41], where they considered the IoT devices with best and worst channel condition to be placed in one clusters. In addition, the authors proposed a power allocation method for each cluster.

In [11], different clustering mechanisms have been studied considering the energy consumption of the nodes. Moreover, the impact of cluster updating has been investigated. A centralized and distributed architecture has been introduced in [12] where, in the centralized architecture, the nodes are clustered having the possibility of offloading to the edge devices. The aim of the work is to estimate the offloading portions to the available devices for computation.

After studying the previous works we have noticed that, to the best of our knowledge, most of the works have been focusing on the offloading decision (i.e., whether to offload or not), or where to offload in order to minimize the energy consumption. On the other hand, some works considered different parameters for cluster formation in different environments. However, no paper has studied network lifetime by considering a joint exploitation of energy harvesting and offloading decision.

3. System Model

In this work, we are focusing on a scenario composed of N FNs identified by the set $\mathcal{U} = \{u_1, \dots, u_i, \dots, u_N\}$. The generic ith FN is supposed to periodically generate some data to be processed, where l_i stands for the data unit generated by the ith FN and δ_{l_i} accounts for its memory occupation in Byte[1]. Each FN is supposed to have a certain processing capability η_c^i, for each $i = 1, \dots, N$, characterized by the floating point operations per second (FLOPS) it can handle; to this aim, we suppose that the l_ith data unit requests a certain amount of floating point operations equal to ω_{l_i}. The data processing operation impacts the energy consumption of the FN and, to this aim, we suppose that the power consumption for computing is P_c^i. Thus, the energy spent by the ith FN for processing l_i can be defined as:

$$E_{p,l_i}^i = P_c^i \frac{\omega_{l_i}}{\eta_c^i}. \tag{1}$$

Each FN is supposed to be equipped with an energy harvesting device. In particular, in this work, we have considered that the generic ith FN is equipped with a small solar panel able to harvest $E_h^i(t)$ Joule at the time instant t. The behaviour of $E_h^i(t)$ depends on several factors, among which, the FN location, the size of the solar panel, the time of the day, as well as the relative orientation between the solar panel and the sun can be mentioned. With respect to this, in this paper we have considered an energy harvesting model based on the data provided by the ENEA agency [13] that averages daily acquisitions from 1995 to 1999 in the city of Milan, Italy, whose details are given in Section 4.

The operative scenario is supposed to be an area having a dimension of A m^2, in which the FNs are scattered in random positions, where the generic ith FN is placed in (x_i, y_i). By resorting to the FC paradigm, we are supposing that data can be processed at the originating FN, or can be offloaded to any other surrounding FN for remote processing. We suppose that the FNs can communicate each other within a range R, i.e, any couple of nodes u_i and u_j can interact if and only if $d(u_i, u_j) \leq R$, where $d(\cdot,\cdot)$ is the Euclidean distance operator and R is the coverage range depending on the considered wireless technology. The transmission technology is assumed to be equal for all the FNs. This operation impacts on the power consumption of the FNs, where we suppose that the ith FN is characterized by a power P_{tx}^i and a power P_{rx}^i, consumed for transmitting and receiving, respectively.

Any FN can act in two different roles. The FN is an offloader (OR) when offloads all the generated tasks. The FN is an offloadee (OE) when gathers the tasks offloaded to be processed. The OR and the OE always interact

[1]The data can be also generated by external sensor nodes connected to the FN; however, from the processing point of view, this corresponds to have data generated directly by the FN itself.

for performing the offloading operation. By focusing on the generic ith FN acting as OR and the generic jth FN acting as OE, having defined the size of the l_ith data unit generated by the ith FN as δ_{l_i}, and the data rate of the link between the ith FN and the jth FN as r_{ij}, we can write the energy spent for transmitting the l_ith data unit as:

$$E^i_{tx,l_i} = P^i_{tx}\frac{\delta_{l_i}}{r_{ij}}. \tag{2}$$

Similarly, by focusing on the generic jth FN acting as OE with respect to the ith OR, we can write the energy spent for receiving the l_ith data unit to be processed as:

$$E^j_{rx,l_i} = P^j_{rx}\frac{\delta_{l_i}}{r_{ij}}. \tag{3}$$

In case the ith FN is not transmitting, receiving, or computing, the energy spent in idle mode corresponds to:

$$E^i_{id} = P^i_{id}t^i_{id} \tag{4}$$

where P^i_{id} is the power consumption in idle and t^i_{id} is the time interval in which the ith FN is in idle.

The objective of this study is designing an energy-aware solution able to prolong the network lifetime characterized as the amount of time the FC network can operate without the need to replace batteries.

To this aim we model the lifetime maximization problem on an infinite time horizon, as an equivalent problem characterized by the composition of consecutive time intervals. This allows to simplify the problem by focusing on minimizing the net energy consumption in each interval, considering both consumed and harvested energy. In the following, we focus on a generic time interval having length $\bar{t}$ seconds, in which the network energy consumption minimization process is performed. We emphasize that $\bar{t}$ tunes the trade-off between the reclustering period and the adaptiveness of the proposed algorithm. This trade-off must be tailored to the considered application scenario and the environment in which the system is operating. However, it has to be pointed out that the algorithm works by considering the energy level at the end of each reclustering period $\bar{t}$, so as to allow the selection of the higher energy nodes.

By supposing that the generic ith FN is generating N_i data units l_i to be processed within $\bar{t}$, the energy spent when acting as an OR can be written as:

$$\bar{E}^i_{OR} = N_iE^i_{tx,l_i} + P^i_{id}\left(\bar{t} - \frac{\delta_{l_i}}{r_{ij}}N_i\right) \tag{5}$$

corresponding to the sum of the energy spent for offloading the N_i generated data units plus the energy spent in idle; it is worth to be noticed that the energy spent in idle is evaluated on the time interval minus the time needed for transmitting the N_i tasks to be offloaded.

On the other side, when the jth FN is the OE, it spends energy for computing its own generated data plus the energy spent for both receiving and computing the data offloaded by the offloader FNs. In addition, the energy spent in idle is considered. This corresponds to an energy consumption equal to:

$$\bar{E}^j_{OE} = N_jE^j_{p,l_j} + \sum_{\substack{u_i\in\mathcal{U}_j \\ u_i\neq u_j}} N_i\left(E^j_{rx,l_i} + E^j_{p,l_i}\right) + P^j_{id}\bar{t}^j_{id} \tag{6}$$

where $\mathcal{U}_j = \{u_i|d(u_i,u_j)\leq R\}$ is the set of FNs that can communicate with the jth FN, being at a distance lower than the considered wireless coverage range R, and N_i identifies the data units generated by the ith FN offloaded to the jth FN. The idle time within the interval $\bar{t}$ for the ith FN can be written as:

$$\bar{t}^j_{id} \geq \bar{t} - N_j\frac{\omega_{l_j}}{\eta^j_c} - \sum_{u_i\in\mathcal{U}_j} N_i\left(\frac{\delta_{l_i}}{r_{ij}} + \frac{\omega_{l_i}}{\eta^j_c}\right) \tag{7}$$

calculated as the interval minus the time spent for receiving and computing. It is worth to be noticed that the computing and receiving actions can be done also in parallel; this means that when they are disjoint an equality should be considered, otherwise the inequality should be considered.

The amount of energy harvested in a certain time interval with length $\bar{t}$ at the time t can be defined as:

$$\bar{E}^i_h(t,\bar{t}) = \int_t^{t+\bar{t}} E^i_h(\tau)d\tau \tag{8}$$

where $E^i_h(\tau)$ is the harvested energy at time instant τ. It is worth to be noticed that the harvested energy depends on both the starting time t and the interval duration $\bar{t}$.

In order to have a better view of the consumption and harvesting patterns in each interval for an offloader node and an offloadee node, in **Figs. 1** and **2** the energy patterns have been plotted by considering that the offloader generates two tasks to be computed by the offloadee. **Fig. 1** represents the power consumption levels for both the OR and OE nodes within one interval; it is possible to notice the different phases during which the FNs can be in transmission, reception, computation or idle. It is possible to notice that while the OE is computing the tasks, the OR is in idle. **Fig. 2**, on the other hand, depicts the time behaviour of the remaining energy level of both OR and OE nodes within the same interval. It is possible to notice that the OR is gaining more from the harvested energy due to a higher idling with respect to the OE, that instead spends a higher amount of time in receiving and computing when the harvested energy is not enough to cover the reception and processing phases. In the depicted example, both nodes are harvesting during the whole interval, however, only during the idle the FNs are able to positively recharge their batteries.

The previous considerations bring to the observation that nodes selected for assuming the OE role consumes more energy than the nodes selected for assuming the OR role. It should, however, be noted that it is a general assumption that offloading is not always beneficial, while its

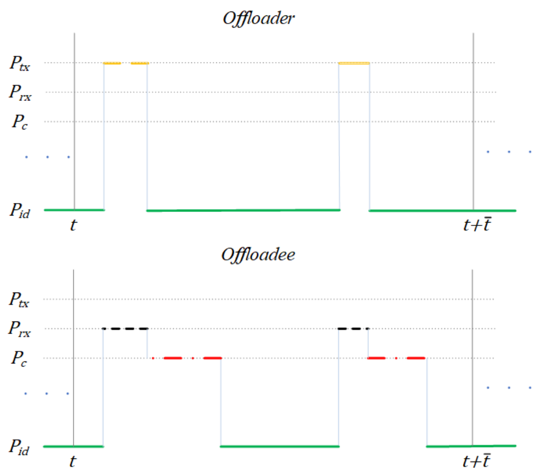


Figure 1: Power consumption levels for offloader and offloadee in an interval

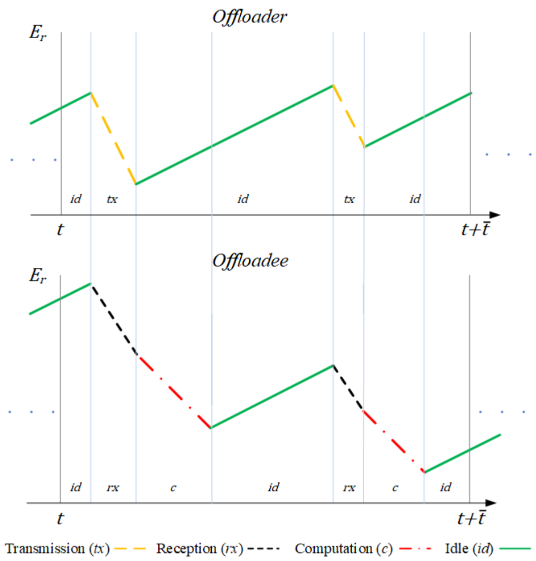


Figure 2: Remaining Energy profiles for offloader and offloadee in an interval

usefulness depends on some parameters, e.g., the task size or computational capability of the devices. However, in our scenario where there is an energy-constrained computation offloading application, the OEs consume more energy consumption than the ORs; hence this drives to the idea that for prolonging the network lifetime in such computation offloading environment, the nodes role should be dynamically selected based on the remained energy.

The FNs are supposed to have a rechargeable battery that can store a maximum amount of energy E_{bc}. The energy stored by the ith FN at a reference starting time $t = 0$ can be modeled as:

$$E_r^i(0) = \gamma_i(0)E_{bc}, \tag{9}$$

where $\gamma_i(0)$ is a parameter in the range $[0, 1]$ modeling the portion of energy in the battery at the time instant $t = 0$; in particular through the parameter $\gamma_i(0)$ it is possible to model scenarios where the FNs are characterized by different energy levels at the beginning of the working operations.

Finally, it is possible to derive the energy stored by the FNs at the end of any interval by exploiting (5), (6) and (8):

$$E_r^i(t + \bar{t}) = E_r^i(t) - \bar{E}_{OR}^i + \bar{E}_h^i(t, \bar{t}), \text{ if } u_i \text{ is OR} \tag{10a}$$

$$E_r^j(t + \bar{t}) = E_r^j(t) - \bar{E}_{OE}^j + \bar{E}_h^j(t, \bar{t}), \text{ if } u_j \text{ is OE} \tag{10b}$$

where

$$E_r^i(t) \leq E_{bc}, \qquad \forall t, \ \forall u_i$$

standing that the remaining energy is always upper bounded by the battery capacity, supposed to be the same for all FNs.

The goal is to design an energy sustainable procedure able to select at each time interval the role of each FN (i.e., offloader or offloadee) with the aim of maximizing their life-time and, hence, having a self-sustainable network.

4. Problem Formulation and Proposed Solution

The aim of this section is to introduce a procedure able to jointly consider the energy consumed and harvested with respect to the role (i.e., OE or OR) that each FN can have. In particular, by gaining from appropriate models a computation offloading procedure exploiting the predicted harvested and consumed energy levels is considered. The proposed procedure is able to exploit information about the predicted harvested energy as well as the consumed and the available energy by FNs to increase the FN lifetime. To this aim, the FN role selection, between the OR and the OE roles, is performed with the aim of avoiding that the FNs are running out of energy during the next interval.

The daily acquisition curve for the generic ith FN can be modelled as a Gaussian with mean μ_i (i.e., the acquisition peak), for each FN $u_i \in \mathcal{U}$, and standard deviation σ_i defined in a way that the interval $(\mu_i - 3\sigma_i, \mu_i + 3\sigma_i)$ covers the sunshine hours of the considered month [24] (e.g., 9 hours and 15 minutes in January and 15 hours and 40 minutes in June in the city of Milan, Italy). Hence, (8) can be rewritten as:

$$\bar{E}_h^i(t, \bar{t}) = \int_t^{t+\bar{t}} E_h^i(\tau)d\tau = \int_t^{t+\bar{t}} \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(\tau-\mu_i)^2}{\sigma_i^2}} d\tau \tag{11}$$

for each node $u_i \in \mathcal{U}$.

Examples of energy harvesting profiles are given in **Fig. 3**, while the way to define μ_i and σ_i are clarified in the Section 5. As expected, the sunshine daily hours change during the year and the daily peak is different for each FN with a discrepancy of at most 30 minutes from the 13:00.

If considering the worst case scenario in terms of energy consumption, i.e., an FN acting the role of an OE and at

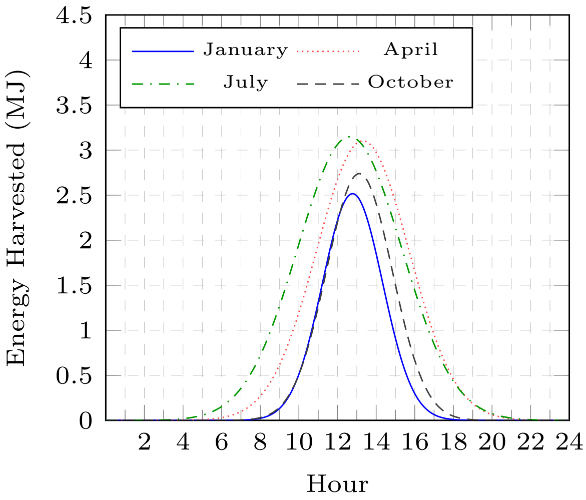


Figure 3: The energy harvested over different hours of a day for four different months of the year, i.e., January, April, July, October.

every time instant in the interval receiving and processing tasks[2], it is possible to predict the upper bound of the consumed energy by the ith FN in the interval $\bar{t}$ as:

$$\tilde{E}_c^i = \left(P_{rx}^i + P_c^i\right)\bar{t} \tag{12}$$

Hence, by exploiting (10), we can define the predicted remaining energy of a generic ith FN at the end of the νth interval as:

$$\tilde{E}_r^i\left((\nu+1)\bar{t}\right) = E_r^i(t_\nu) - \tilde{E}_c^i + \bar{E}_h^i(t_\nu, \bar{t}). \tag{13}$$

where $t_\nu = \nu\bar{t}$ for simplifying the notation.

Since the goal is to select the role of each FN based on its stored energy, we assume that it is possible to dynamically split the FN set into two subsets at the beginning of each interval:

- $\mathcal{U}_{HE}(t_\nu)$, the FNs having a high amount of energy and being able to support the processing of the other FNs. We will refer to them as High Energy FNs.
- $\mathcal{U}_{LE}(t_\nu)$ the FNs whose remaining energy is low and possibly not sufficient for completing the foreseen processing. We will refer to them as Low Energy FNs.

In particular, at each interval, the energy level of each FN is estimated by resorting to the predicted remaining energy defined in (13).

In order to assign the FNs to the two sets we define a two-step procedure. At the *first step* we define:

$$\tilde{\mathcal{U}}_{HE}(t_\nu) = \left\{u_i | \tilde{E}_r^i\left((\nu+1)\bar{t}\right) \geq 0\right\},$$

[2]It is worth to be noticed that transmitting/receiving can be done in parallel with respect to processing.

as the set of the candidate High Energy FNs at time instant t_ν. The FNs whose predicted remaining energy at the end of the νth interval is positive are considered as the High Energy potential FNs. Indeed, the FNs not belonging to $\tilde{\mathcal{U}}_{HE}$ cannot guarantee to be able to process all the potentially required tasks in the next interval.

At the *second step* we perform the energy-aware FN classification by using a threshold based approach. The FNs in $\tilde{\mathcal{U}}_{HE}(t_\nu)$ having a remaining energy higher than a given threshold θ_{E_r} are selected as High Energy nodes and the rest as Low Energy nodes. We will see in the next section that, to be effective, threshold θ_{E_r} is not a-priori fixed but it is adapted to the current energy level values of the nodes thanks to a quantile-based approach.

More specifically, we can define the two sets $\mathcal{U}_{LE}(t_\nu)$ and $\mathcal{U}_{HE}(t_\nu)$ as follows:

$$\mathcal{U}_{HE}(t_\nu) = \left\{u_i \in \tilde{\mathcal{U}}_{HE}(t_\nu) | E_r^i(t_\nu) \geq \theta_{E_r}\right\} \tag{14}$$

$$\mathcal{U}_{LE}(t_\nu) = \left\{u_i \in \tilde{\mathcal{U}}_{HE}(t_\nu) | E_r^i(t_\nu) < \theta_{E_r}\right\} \cup \left\{u_i \notin \tilde{\mathcal{U}}_{HE}(t_\nu)\right\} \tag{15}$$

This rule allows to select the FNs having the highest energy level as OE nodes and capable of performing more energy hungry operations, while the rest, selected as Low Energy nodes, could work as OR nodes.

Algorithm 1 Prediction-based Energy Harvesting Scheme

```
1: Input: Ẽ_r^i((ν+1)t̄) ∀u_i ∈ U
2: Output: U_HE and U_LE
3: for all u_i ∈ U do
4:    if Ẽ_r^i((ν+1)t̄) ≥ 0 then
5:       Ũ_HE ← u_i
6:    else
7:       U_LE ← u_i
8:    end if
9: end for
10: for all u_i ∈ Ũ_HE do
11:    if E_r^i(t_ν) ≥ θ_{E_r} then
12:       U_HE ← u_i
13:    else
14:       U_LE ← u_i
15:    end if
16: end for
```

The pseudo-code of the Prediction-based Energy Harvesting procedure is shown in **Algorithm 1**, where the input is the estimated remaining energy of the FNs calculated as in (13), and the output is the two subsets $\mathcal{U}_{HE}$ and $\mathcal{U}_{LE}$ (Lines 1-2). In this algorithm, $\tilde{E}_r^i\left((\nu+1)\bar{t}\right)$ of all FNs is considered as a metric for selecting the role of the FNs. The FNs, having the estimated energy higher than zero at the end of the time interval, are considered as potential High Energy nodes able to work as OE while the remaining nodes are assumed to be the ORs due to their high probability of running out of energy before the end of the interval. Then, among the FNs in $\tilde{\mathcal{U}}_{HE}$, a threshold is considered so that the FNs with the remaining energy higher than the threshold are inserted in $\mathcal{U}_{HE}$, otherwise they are inserted in $\mathcal{U}_{LE}$ (Lines 10-16). In the end, the OE FNs are selected among those FNs having sufficient estimated energy level at the end of the time interval.

As a benchmark for evaluating the effectiveness of the proposed energy harvesting solution, a simple energy harvesting scheme is also considered where no prediction is performed. The FNs having a remaining energy level higher than a given threshold θ_{E_r} are selected as OE and the rest as ORs. This rule allows to select the FNs having the highest energy level as the recipient of the more energy hungry operations. In this case, it is possible to define $\mathcal{U}^H_{HE}(t_\nu)$ and $\mathcal{U}^H_{LE}(t_\nu)$ as:

$$\mathcal{U}^H_{HE}(t_\nu) = \left\{u_i | E^i_r(t_\nu) \geq \theta_{E_r}\right\}$$
$$\mathcal{U}^H_{LE}(t_\nu) = \left\{u_i | E^i_r(t_\nu) < \theta_{E_r}\right\}$$

4.1. Clustering-based communication overlay

The FN classification based on their energy level allows to implement an energy-based communication overlay able to take into account the FN energy level and, at the same time, to allow optimizing the energy consumption. To this aim, by relying also on previous studies, we resort on a cluster organization, considered as a feasible network structure for energy-efficient Fog Network design [11, 12].

The idea at the basis of the clustering scheme is that the FNs can transmit or receive the tasks to be processed depending on their remaining energy level. To this aim the clustering scheme allows to organize the FNs in a way that the High Energy FNs can act as Cluster Heads (CHs), being able to receive and process the tasks, while the Low Energy FNs can act as Cluster Members (CMs) that offload the data to be processed for saving energy. In addition, the cluster formation algorithm allows to assist those nodes with a lower amount of energy, considering them with a higher priority.

Each cluster is composed of one CH and several CMs. The CHs are selected among the High Energy FNs, starting from the FN with the highest remaining energy, i.e., $u_i \in \mathcal{U}_{HE}$ such that $\max_{u_i}\left\{E^i_r(t_\nu)\right\}$. For each selected CH, the CMs are selected among the Low Energy FN whose distance w.r.t. the CH is lower than the FNs coverage range R, prioritizing the FNs with the lowest remaining energy level. The CH and CM selection for the cluster formation is performed such that the CHs with the highest energy are connected to the CMs with the lowest energy in order to balance the clusters in terms of remaining energy. Hence, we can define a generic ith cluster with M FNs, i.e., having one CH and $M-1$ CMs, as:

$$\mathcal{C}_i = \left\{u_i \in \mathcal{U}_{HE} | \max_{u_i}\left\{E^i_r(t_\nu)\right\}\right\} \cup \left\{u_j \in \mathcal{U}_{LE} | \min_{u_j}\left(E^j_r(t_\nu)\right), d(u_i, u_j) \leq R, |\mathcal{C}_i| \leq M\right\} \tag{16}$$

In the following, we will refer to FNs acting as CH as those nodes processing tasks received from the FNs acting as CMs. After having classified the FNs based on the Prediction-based Harvesting Scheme, the clustering procedure is performed by considering $\mathcal{U}_{LE}$ and $\mathcal{U}_{HE}$ sets as an input, while the output is the cluster list with the belonging FNs.

Algorithm 2 Clustering Scheme

1: **Input:** $\mathcal{U}_{HE}$, $\mathcal{U}_{LE}$, $E^i_r(t_\nu)\ \forall u_i \in \mathcal{U}$
2: **Output:** $\mathcal{C}_i \forall i$
3: **while** $\mathcal{U}_{HE} \neq \emptyset$ **do**
4: Select $u_i \in \mathcal{U}_{HE} | \max_{u_i}\left\{E^i_r(t_\nu)\right\}$
5: $\mathcal{C}_i \leftarrow u_i$
6: **for all** $u_j \in \mathcal{U}_{LE}$ **do**
7: **if** $d(u_i, u_j) \leq R$ **then**
8: $\hat{\mathcal{C}}_i \leftarrow u_j$;
9: **end if**
10: **end for**
11: **while** $|\mathcal{C}_i| < M$ **do**
12: $\mathcal{C}_i \leftarrow u_j | \min_{u_j}(E^j_r(t_\nu))\quad \forall u_j \in \hat{\mathcal{C}}_i$;
13: remove u_j from $\mathcal{U}_{LE}$ and $\hat{\mathcal{C}}_i$;
14: **end while**
15: remove u_i from $\mathcal{U}_{HE}$
16: **end while**
17: **if** $\mathcal{U}_{LE} \neq \emptyset$ **then**
18: **for** each $u_j \in \mathcal{U}_{LE}$ **do**
19: $\mathcal{C}_j \leftarrow u_j$;
20: remove u_j from $\mathcal{U}_{LE}$
21: **end for**
22: **end if**

The pseudo-code of the clustering scheme is shown in **Algorithm 2**, where the lists of High Energy FNs and Low Energy FNs and the remaining energy level of all FNs are the input, while the output is the clusters list $\mathcal{C}_i$ (Lines 1-2). Since the goal is to minimize the energy shortage, the clustering algorithm starts from the High Energy set, defined by **Algorithm 1**, where starting from the ith FN having the highest remaining energy, the ith cluster $\mathcal{C}_i$ is created with the ith FN as the CH (Lines 4-5). By considering the energy at the end of the reclustering period, the selection of the higher energy node is performed for ensuring that the selected HE nodes have a sufficient amount of energy for processing the nearby nodes' tasks. All the FNs belonging to the Low Energy set are then considered, and among them, those with a distance with respect to the selected CH lower than the coverage range are selected and put in a temporary cluster $\hat{\mathcal{C}}_i$ composed by the candidates CMs of the ith CH, as long as the coverage range condition is respected (Lines 6-10). Among the candidates CMs, the $(M-1)$ FNs with the lowest remaining energy level are put in the cluster $\mathcal{C}_i$ and removed by both $\mathcal{U}_{LE}$ and $\hat{\mathcal{C}}_i$ sets (Lines 11-14). The bound M has been introduced for taking into account both communications and computing multiple access limitations, so that any cluster cannot have more than M FNs. Moreover, setting a bound on the maximum number of FN per cluster allows to indirectly balance the cluster load between different CHs. The selection of the lowest energy level FNs allows to go in the direction of increasing the network lifetime by limiting the power consumption of the FNs with lower energy values. The selected High Energy FNs are removed from their lists once they are assigned (Line 15). In the end, the remaining FNs in $\mathcal{U}_{LE}$ are considered as isolated nodes and hence performing as CHs of a cluster with no other FNs (Lines 17-22).

At the beginning of the following interval, both the Prediction-based Harvesting Scheme and the Clustering Scheme are performed for updating the cluster sets $\mathcal{C}_i$ in order to take into account the real amount of energy spent by each FN.

5. Experimental Results

The computer simulations are performed in Matlab for two possible spans of the daily time, as depicted in **Table 1**. We have considered two different starting times and sunshine duration in order to observe the behavior of the network in two different situations. The simulation parameters are listed in **Table 2**, whose values are consistent with similar papers in the literature [11, 12, 42].

We hypothesize a squared area A equal to 200×200 m^2, with a variable number of FNs equal to 200, 500, or 700 randomly placed in the area with uniform distribution. The FNs generate data units with a Binomial distribution with p=0.1 and n=10 per reclustering period. All FNs are supposed to have the same computational capability and battery capacity; however, in order to have a different initial energy, we have considered that the initial value of the remaining energy of the ith FN is $\gamma_i(0)E_{bc}$, where $\gamma_i(0)$ is uniformly distributed in the range $[0.7, 1]$, with different values for different FNs. The interval $\bar{t}$ in which the procedure is performed has been set equal to 50 s; this value has been defined by balancing the need to adapt quickly to the network changes and the management cost. Indeed, there is a trade-off between the reclustering period and the adaptiveness of the proposed algorithm, where shorter periods allow to have a faster reaction to the energy level changes but requiring a higher number of control message exchanges, while longer periods reduce the control message exchange at the cost of a reduced reactivity. The value considered in this work, 50 s, is consistent with the numerical results in [11], where the effect of reclustering intervals on lifetime and energy consumption of the nodes have been studied. Finally, it has to be pointed out that the HE nodes selection ensures that CH nodes have always a positive energy level by the end of the reclustering period.

The energy threshold θ_{E_r} has been defined by resorting to a 3-quantile classification performed on the energy level distribution of all the FNs. By using a 3-quantile function it is possible to classify the nodes based on the distribution of their remaining energy. At each interval, the FNs energy levels are evaluated, and then, the FNs are classified in one of the two groups depending on their residual energy value. The upper quantile index, used as threshold, is defined as:

$$\theta_{E_r} = E_r \quad \text{such that} \quad F_{E_r}(E_r) = \rho \tag{17}$$

where ρ for the upper 3-quantile index is 2/3, and $F_{E_r}()$ represents the cumulative distribution function of the remaining energy of all the FNs [12].

The solar panel is supposed to have the size equal to 25 cm^2, and is south-oriented (i.e., the Azimuth degree is zero), and has an inclination of 36 degrees w.r.t the horizontal ground. Moreover, it is assumed that there are no obstacles, and the ground reflection coefficient is set to 0.20, i.e., the value of stones/rubbles, a value in between the land (0.14), the asphalt (0.10), the roofs (0.13) and the dark buildings (0.27), representing a typical urban landscape with a few clear buildings that have a higher reflection coefficient (0.60).

The parameters of the Guassian curve modelling the energy acquired by the solar panel, as described in Section 4, are the peak time μ_i uniformly distributed between 12.30 and 13.30, and the variance σ_i such that $(\mu_i - 3\cdot\sigma_i, \mu_i + 3\cdot\sigma_i)$ covers the sunshine hours of the given month.

It is noteworthy to point out that the previous equation is valid if and only if the interval $\bar{t}$ is less than 24 hours. However, if one is interested to analyze acquisition periods greater than a day, the Gaussian curve is repeated with a period of 24 hours, hence assuming that the month does not change[3].

For the simulation results, we have considered three different approaches:

- Harvesting and Prediction (*H&P*): This is the approach presented in Section 4, where the FNs are able to harvest and predict their energy level in the next clustering period with the parameters listed in **Table 3**.
- Harvesting (*H*): This is a benchmark Harvesting clustering scheme outlined in Section 4 where, differently form the Harvesting and Prediction, we suppose that the energy balancing prediction in the following interval is not performed, while the FNs are able to harvest energy by using a solar panel.
- No Harvesting (*NH*): The FNs are not equipped with a solar panel and energy cannot be harvested. This is the benchmark approach presented in [11].

The NH approach is considered as a benchmark to see the impact of the harvesting in the network lifetime. In the NH approach **Algorithm 1** is used by considering that $E_h^i = 0$ for all FNs. On the other side, in the harvesting

[3]If needed, for periods significantly longer than a day, it is possible to define more realistic mechanisms. An example sets the sunshine hours equal to the monthly mean only in the first fifteenth day and then progressively changes that value toward the previous/subsequent month's mean

Table 1: Scenario Definition

Scenario	Starting Time	Length
1	10:00	4 hours
2	noon	12 hours

Table 2: Simulation Parameters

Parameter	Value
Dimension	200×200 m^2
Data Unit Size (δ_{l_i})	5 MB
Normalized Data Rate @ 1 m ($\bar{r}_{ij}$)	20 Mbit/s
Battery Capacity (E_{bc})	5000 J
FNs' Coverage Range (R)	25 m
Data Unit Operations (ω_{l_i})	50 GFLOP
Max Number of FNs per Cluster (M)	5
FN Computation Capability (η_c^j)	12 GFLOPS
FN Computation Power (P_c^i)	0.9 W
FN Idle Power (P_{id}^i)	0.3 W
FN Transmission Power (P_{tx}^i)	1.3 W
FN Reception Power (P_{rx}^i)	1.1 W

Table 3: Energy Harvesting Parameters

Parameter	Value
Solar Panel Area	25 cm^2
Azimuth	0 deg (South-oriented)
Inclination w.r.t. Ground	36 deg
Ground Reflection Coefficient	0.20
Obstacles Occlusion	None
Sun Peak Time Hour	13.00 (±30 minutes)
Sunshine Hours (December–June)	8h52 – 15h42

Table 4: Percentage of performance gain of *H&P* approach in terms of network lifetime for 14 hours of sunshine

	200 FN		500 FN		700 FN	
Bench. \ Scen.	10 am	noon	10 am	noon	10 am	noon
NH	55%	37%	92%	65%	119%	74%
H	24%	8%	17%	13%	17%	14%

approach, the FNs are equipped with solar panel and are able to harvest. **Algorithm 2** has been considered for the clustering procedure and (10) for the remaining energy of the FNs considering the amount of energy they have harvested. Finally, *H&P* approach considers **Algorithm 1** and **Algorithm 2** for the clustering procedure and (13) for calculating the remaining energy of the FNs. Details about energy harvesting parameters are given in **Table 3**.

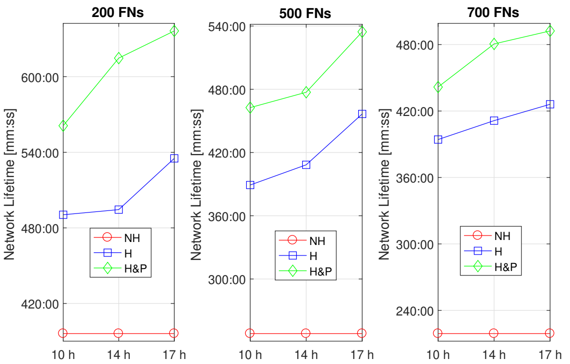
In the experimental results we are interested in observing the impact of harvesting solutions on the lifetime of the network measured as the amount of time the FNs deplete their battery [43]. In particular, in order to show how the *H&P* solution outperforms the others, we are conducting some experiments in the network where FNs have different battery capacities.

In order to see the pattern of the nodes going off, we have performed an experiment, plotting the lifetime expressed in minutes and seconds when all the FNs deplete their energy; the results are shown in **Fig. 4** for two different starting points, different number of FNs and sunshine hours [4]. We have labeled the curves with *NH*, *H*, and *H&P*, which refer, respectively, to the No harvesting, Harvesting and *H&P* based approaches. As seen in both **Fig. 4a** and **Fig. 4b**, in which all the FNs are supposed to have the same battery capacity ($E_{bc} = 5000$ J), the *H&P* allows to extend the FNs lifetime, as expected. It can also be seen that the *H&P* approach allows to extend the lifetime with respect to the simple harvesting approach thanks to the ability to predict the harvested energy. The other point to be highlighted is that network lifetime in harvesting solutions in **Fig. 4a** is longer than that in **Fig. 4b** and that is due to the higher amount of harvested energy starting at 10:00, owing to the presence of the harvesting peak around noon. Finally, we can see the effect of different sunshine hours; as seen even in the shortest duration of presence of sunshine, the *H&P* approach outperforms the other benchmarks. In both scenarios, the NH approach performs in a similar way since no harvesting is considered. **Table 4** presents the percentage of performance gain of the *H&P* approach w.r.t. the benchmarks in terms of network lifetime for the 14 hours scenario. It should be noted that, for the rest of the simulation results, we will fix the sunshine hours to 14 hours, representing a middle case in terms of energy harvesting.

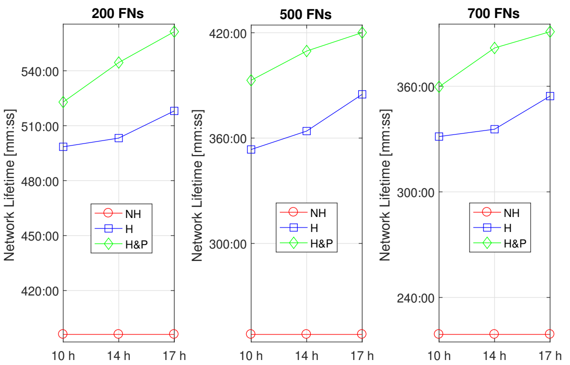
In order to identify when the harvesting based solutions have a higher impact on the network lifetime, we are here investigating the impact of the battery capacity on the network lifetime by considering the two different scenarios defined in **Table 1**. In particular, the following figures show the cumulative function of the day hour in which the FNs are going off.

In **Fig. 5**, we have compared the results in case of FNs performing *NH* with a battery capacity $E_{bc} = 5000$ J while the battery capacity for *H* and *H&P* has been set

[4] It is worth to be noticed that different sunshine hours are considered for mapping the behaviour of different seasons, where in Milan, Italy, there are 10 hours sunshine in December, 17 hours in June and 14 in April/September

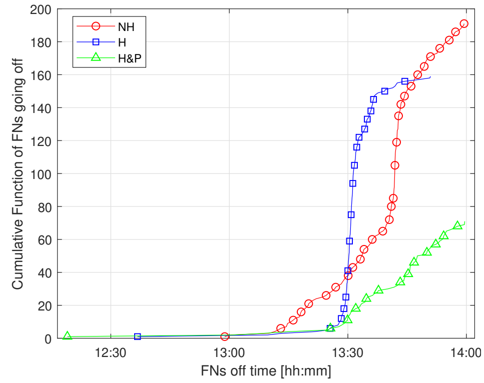


(a) Starting at 10 am

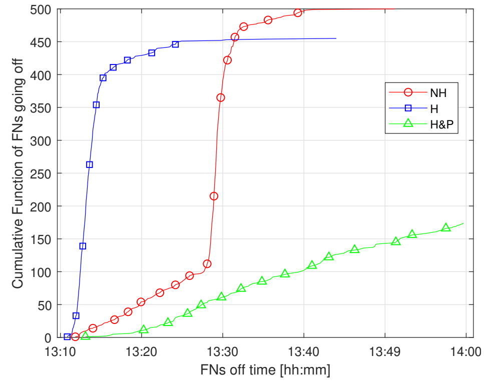


(b) Starting at noon

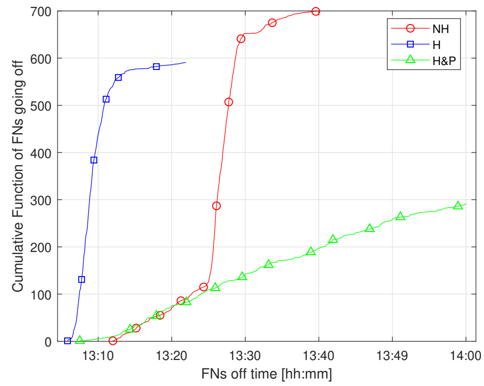
Figure 4: Network Life time of the FNs going off with $E_{bc} = 5000\,\mathrm{J}$ for the 3 approaches with different sunshine hours.



(a) 200 FNs



(b) 500 FNs



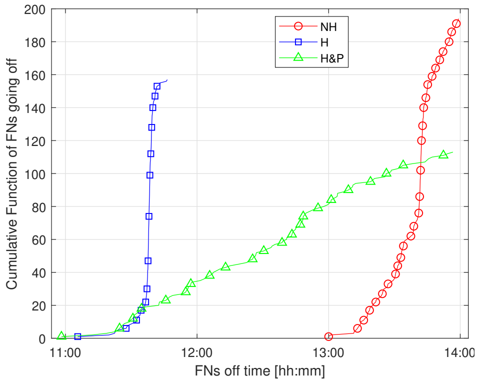
(c) 700 FNs

Figure 5: FNs off time in Scenario 1 with $E_{bc} = 2000\,\mathrm{J}$

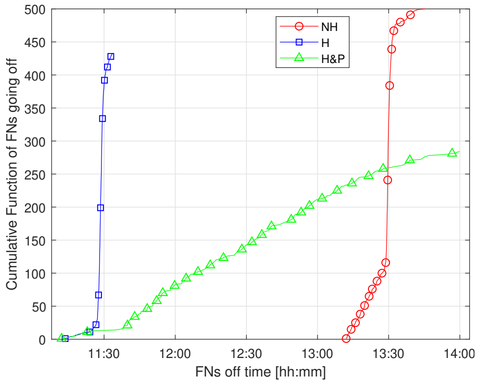
to $E_{bc} = 2000\,\mathrm{J}$. It is possible to note that even by reducing the battery capacity to less than one half of the battery capacity used for the *NH* approach, the nodes show a similar behaviour in terms of lifetime, demonstrating the effectiveness of the harvesting in the network lifetime. This is particularly important showing that if we are able to implement the harvesting solution in the IoT scenario we can reduce the battery size, and consequently its cost, while having a similar behaviour in terms of lifetime. The *H&P* approach allows to reduce even more the number of nodes depleting their energy. Furthermore, it can be seen that when the number of FNs increases, the lifetime decreases; this is due to the fact that when the network is composed by a higher number of FNs, also the clusters are composed by a higher number of nodes reflecting in an increased interaction among FNs and as a result a higher energy consumption. The other interesting point is that the lifetime behaviour in the *H&P* approach for all the FNs scenarios is changing with a lower slope; this is due to the prediction mechanism in which the energy consumption among the FNs is balanced thus reducing the effect of re-clustering.

In **Fig. 6** we have further reduced the battery capacity for FNs with *H* and *H&P* by setting $E_{bc} = 1000\,\mathrm{J}$ while in case of non harvesting $E_{bc} = 5000\,\mathrm{J}$. It is possible to notice that in this case the harvested energy is not sufficient, hence the FNs consume their energy earlier than the *NH* approach in which the nodes are supposed to have a battery capacity of five times larger. However, it is worth to be noticed that a reduced number of FNs still have a certain amount of energy when the nodes with *NH* are already out. We emphasize the effect of the *H&P* solution, that allows to extend even more the nodes lifetime for about one half of the nodes with respect to the situation of *NH* with a battery capacity five time larger.

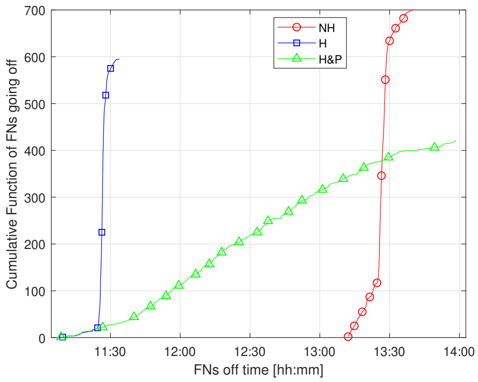
As a final result, we have considered the opposite case in Scenario 2 when the FNs operating the harvesting approach have a battery capacity $E_{bc} = 5000\,\mathrm{J}$, while the



(a) 200 FNs

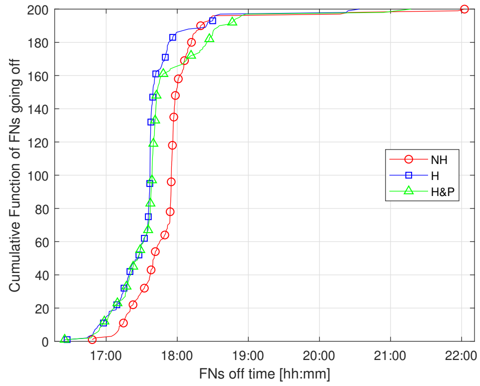


(b) 500 FNs

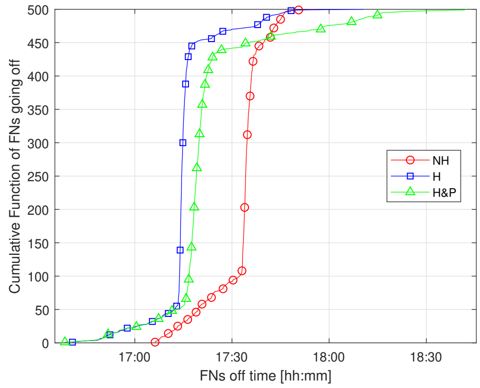


(c) 700 FNs

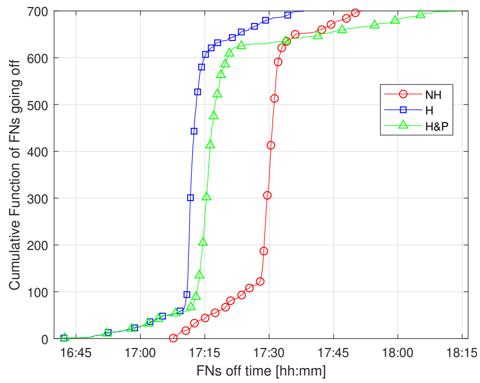
Figure 6: FNs off time in Scenario 1 with $E_{bc} = 1000\,\mathrm{J}$



(a) 200 FNs



(b) 500 FNs



(c) 700 FNs

Figure 7: FNs off time in Scenario 2 with $E_{bc} = 8000\,\mathrm{J}$

FNs performing the non harvesting approach have a battery capacity $E_{bc} = 8000\,\mathrm{J}$. In **Fig. 7**, the time instant in which the FNs are going off is depicted. It is possible to notice that the three solutions have quite the same performance. However, thanks to the prediction property, FNs in the *H&P* solution consume their energy later than the other two cases. Moreover, it is worth noticing that the FNs operating with harvesting solution have a smaller battery capacity confirming the effectiveness of the proposed approach.

6. Conclusions

In this work, a smart energy management procedure is proposed for extending the network lifetime of a fog computing network. To this aim, a clustering mechanism at the network edge is introduced for a computation offloading scenario. We have considered a harvesting scenario where the FNs are able to harvest energy exploiting on-board solar panels. We have mathematically modeled the energy consumption and harvesting in the fog networks. Later, we proposed a prediction method in which the FNs can be selected for composing a clustered architecture to be used for task computation. In order to have a balanced network in terms of energy consumption, the CH and CM selection in the clustering solution is performed by considering the FNs consumption and harvesting energy amount. By analyzing different battery capacity and node density in a realistic solar-panel harvesting model, in the experimental results, we have shown how effective the *H&P* solution is in terms of network lifetime increase. As a future work we will consider a distributed approach of the proposed solution, which is currently meant to operate in a centralized way. Such a distributed approach would allow to manage even large-scale distributed systems.

References

[1] G. Mois, T. Sanislav, S. C. Folea, A cyber-physical system for environmental monitoring, IEEE Transactions on Instrumentation and Measurement 65 (6) (2016) 1463–1471. doi:10.1109/TIM.2016.2526669.

[2] Y. Meng, W. Zhang, H. Zhu, X. S. Shen, Securing consumer IoT in the smart home: Architecture, challenges, and countermeasures, IEEE Wireless Communications 25 (6) (2018) 53–59. doi:10.1109/MWC.2017.1800100.

[3] O. Andrisano, I. Bartolini, P. Bellavista, A. Boeri, L. Bononi, A. Borghetti, A. Brath, G. E. Corazza, A. Corradi, S. de Miranda, F. Fava, L. Foschini, G. Leoni, D. Longo, M. Milano, F. Napolitano, C. A. Nucci, G. Pasolini, M. Patella, T. Salmon Cinotti, D. Tarchi, F. Ubertini, D. Vigo, The need of multidisciplinary approaches and engineering tools for the development and implementation of the smart city paradigm, Proceedings of the IEEE 106 (4) (2018) 738–760. doi:10.1109/JPROC.2018.2812836.

[4] F. Griffiths, M. Ooi, The fourth industrial revolution - Industry 4.0 and IoT [Trends in Future I&M], IEEE Instrumentation and Measurement Magazine 21 (6) (2018) 29–43. doi:10.1109/MIM.2018.8573590.

[5] V. Petrov, K. Mikhaylov, D. Moltchanov, S. Andreev, G. Fodor, J. Torsner, H. Yanikomeroglu, M. Juntti, Y. Koucheryavy, When IoT keeps people in the loop: A path towards a new global utility, IEEE Communications Magazine 57 (1) (2019) 114–121. doi:10.1109/MCOM.2018.1700018.

[6] M. Chiang, T. Zhang, Fog and IoT: An overview of research opportunities, IEEE Internet of Things Journal 3 (6) (2016) 854–864.

[7] B. Omoniwa, R. Hussain, M. A. Javed, S. H. Bouk, S. A. Malik, Fog/Edge computing-based IoT (FECIoT): Architecture, applications, and research issues, IEEE Internet of Things Journal 6 (3) (2019) 4118–4149. doi:10.1109/JIOT.2018.2875544.

[8] D. Mazza, D. Tarchi, G. E. Corazza, A unified urban mobile cloud computing offloading mechanism for smart cities, IEEE Communications Magazine 55 (3) (2017) 30–37. doi:10.1109/MCOM.2017.1600247CM.

[9] L. Liu, Z. Chang, X. Guo, S. Mao, T. Ristaniemi, Multiobjective optimization for computation offloading in fog computing, IEEE Internet of Things Journal 5 (1) (2018) 283–294. doi:10.1109/JIOT.2017.2780236.

[10] M. Mukherjee, S. Kumar, Q. Zhang, R. Matam, C. X. Mavromoustakis, Y. Lv, G. Mastorakis, Task data offloading and resource allocation in fog computing with multitask delay guarantee, IEEE Access 7 (2019) 152911–152918. doi:10.1109/ACCESS.2019.2941741.

[11] A. Bozorgchenani, D. Tarchi, G. E. Corazza, An energy-aware offloading clustering approach (EAOCA) in fog computing, in: 2017 International Symposium on Wireless Communication Systems (ISWCS), Bologna, Italy, 2017, pp. 390–395. doi:10.1109/ISWCS.2017.8108146.

[12] A. Bozorgchenani, D. Tarchi, G. E. Corazza, Centralized and distributed architectures for energy and delay efficient fog network based edge computing services, IEEE Trans. Green Commun. Netw. 3 (1) (2019) 250–263. doi:10.1109/TGCN.2018.2885443.

[13] Atlante Italiano della Radiazione Solare (Apr. 2019). URL `http://www.solaritaly.enea.it/CalcComune/Definizioni.php`

[14] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, X. Peng, L. Pan, S. Maharjan, Y. Zhang, Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks, IEEE Access 4 (2016) 5896–5907. doi:10.1109/ACCESS.2016.2597169.

[15] X. Lyu, H. Tian, C. Sengul, P. Zhang, Multiuser joint task offloading and resource optimization in proximate clouds, IEEE Transactions on Vehicular Technology 66 (4) (2017) 3435–3447. doi:10.1109/TVT.2016.2593486.

[16] W. Zhang, Y. Wen, H. H. Chen, Toward transcoding as a service: energy-efficient offloading policy for green mobile cloud, IEEE Network 28 (6) (2014) 67–73. doi:10.1109/MNET.2014.6963807.

[17] W. Zhang, Y. Wen, K. Guan, D. Kilper, H. Luo, D. O. Wu, Energy-optimal mobile cloud computing under stochastic wireless channel, IEEE Transactions on Wireless Communications 12 (9) (2013) 4569–4581. doi:10.1109/TWC.2013.072513.121842.

[18] Y. D. Lin, E. T. H. Chu, Y. C. Lai, T. J. Huang, Time and energy aware computation offloading in handheld devices to coprocessors and clouds, IEEE Systems Journal 9 (2) (2015) 393–405. doi:10.1109/JSYST.2013.2289556.

[19] Y. Liao, L. Shou, Q. Yu, Q. Ai, Q. Liu, Joint offloading decision and resource allocation for mobile edge computing enabled networks, Computer Communications 154 (2020) 361–369. doi:10.1016/j.comcom.2020.02.071.

[20] M. Abbasi, M. Yaghoobikia, M. Rafiee, A. Jolfaei, M. R. Khosravi, Efficient resource management and workload allocation in fog–cloud computing paradigm in IoT using learning classifier systems, Computer Communications 153 (2020) 217–228. doi:10.1016/j.comcom.2020.02.017.

[21] V. Raghunathan, A. Kansal, J. Hsu, J. Friedman, M. Srivastava, Design considerations for solar energy harvesting wireless embedded systems, in: IPSN 2005. Fourth International Symposium on Information Processing in Sensor Networks, 2005, Boise, ID, USA, 2005, pp. 457–462. doi:10.1109/IPSN.2005.1440973.

[22] W. K. G. Seah, Z. A. Eu, 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, Aalborg, Denmark, 2009. doi:10.1109/WIRELESSVITAE.2009.5172411.

[23] J. Xu, L. Chen, S. Ren, Online learning for offloading and autoscaling in energy harvesting mobile edge computing, IEEE Transactions on Cognitive Communications and Networking 3 (3) (2017) 361–373. doi:10.1109/TCCN.2017.2725277.

[24] C. Alippi, R. Camplani, C. Galperti, A. Marullo, M. Roveri, A high-frequency sampling monitoring system for environmental and structural applications, ACM Transactions on Sensor Networks 9 (4) (2013) 41:1–41:32. doi:10.1145/2489253.2489258.

[25] C. Alippi, R. Camplani, C. Galperti, M. Roveri, A robust, adaptive, solar-powered wsn framework for aquatic environmental monitoring, IEEE Sensors Journal 11 (1) (2011) 45–55. doi:10.1109/JSEN.2010.2051539.

[26] W. Hu, N. Bulusu, C. T. Chou, S. Jha, A. Taylor, V. N. Tran, Design and evaluation of a hybrid sensor network for cane toad monitoring, ACM Transactions on Sensor Networks 5 (1) (2009) 4:1–4:28. doi:10.1145/1464420.1464424.

[27] R. Pinciroli, M. Gribaudo, M. Roveri, G. Serazzi, Capacity planning of fog computing infrastructures for smart monitoring, in: S. Balsamo, A. Marin, E. Vicario (Eds.), New Frontiers in Quantitative Methods in Informatics, Springer International Publishing, Cham, Switzerland, 2018, pp. 72–81. doi:10.1007/978-3-319-91632-3_6.

[28] Q. Liu, M. Li, J. Yang, J. Lv, K. Hwang, M. S. Hossain, G. Muhammad, Joint power and time allocation in energy har-

vesting of UAV operating system, Computer Communications 150 (2020) 811 – 817. doi:10.1016/j.comcom.2019.12.009.

[29] C. Li, W. Chen, J. Tang, Y. Luo, Radio and computing resource allocation with energy harvesting devices in mobile edge computing environment, Computer Communications 145 (2019) 193 – 202. doi:10.1016/j.comcom.2019.06.001.

[30] Y. Nan, W. Li, W. Bao, F. C. Delicato, P. F. Pires, Y. Dou, A. Y. Zomaya, Adaptive energy-aware computation offloading for cloud of things systems, IEEE Access 5 (2017) 23947–23957. doi:10.1109/ACCESS.2017.2766165.

[31] L. Liu, Z. Chang, X. Guo, Socially aware dynamic computation offloading scheme for fog computing system with energy harvesting devices, IEEE Internet of Things Journal 5 (3) (2018) 1869–1879. doi:10.1109/JIOT.2018.2816682.

[32] C. Kulatunga, K. Bhargava, D. Vimalajeewa, S. Ivanov, Cooperative in-network computation in energy harvesting device clouds, Sustainable Computing: Informatics and Systems 16 (2017) 106 – 116. doi:10.1016/j.suscom.2017.10.006.

[33] Q. Ju, G. Sun, H. Li, Y. Zhang, Latency-aware in-network computing for internet of battery-less things, in: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), Chicago, IL, USA, 2018. doi:10.1109/VTCFall.2018.8690883.

[34] Q. Ju, G. Sun, H. Li, Y. Zhang, Collaborative in-network processing for internet of battery-less things, IEEE Internet of Things Journal 6 (3) (2019) 5184–5195. doi:10.1109/JIOT.2019.2899022.

[35] S. Yang, Y. Tahir, P. Chen, A. Marshall, J. McCann, Distributed optimization in energy harvesting sensor networks with dynamic in-network data processing, in: IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 2016. doi:10.1109/INFOCOM.2016.7524475.

[36] M. Tao, E. Chen, H. Zhou, W. Yu, Content-centric sparse multicast beamforming for cache-enabled cloud RAN, IEEE Transactions on Wireless Communications 15 (9) (2016) 6118–6131. doi:10.1109/TWC.2016.2578922.

[37] J. Oueis, E. Calvanese Strinati, S. Barbarossa, Distributed mobile cloud computing: A multi-user clustering solution, in: 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 2016. doi:10.1109/ICC.2016.7511046.

[38] J. Oueis, E. Calvanese Strinati, S. Sardellitti, S. Barbarossa, Small cell clustering for efficient distributed fog computing: A multi-user case, in: 2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall), Boston, MA, USA, 2015. doi:10.1109/VTCFall.2015.7391144.

[39] J. Oueis, E. Calvanese Strinati, S. Barbarossa, The fog balancing: Load distribution for small cell cloud computing, in: 2015 IEEE 81st Vehicular Technology Conference (VTC Spring), Glasgow, UK, 2015. doi:10.1109/VTCSpring.2015.7146129.

[40] M. Bouet, V. Conan, Mobile edge computing resources optimization: A geo-clustering approach, IEEE Transactions on Network and Service Management 15 (2) (2018) 787–796. doi:10.1109/TNSM.2018.2816263.

[41] X. Shao, C. Yang, D. Chen, N. Zhao, F. R. Yu, Dynamic IoT device clustering and energy management with hybrid NOMA systems, IEEE Transactions on Industrial Informatics 14 (10) (2018) 4622–4630. doi:10.1109/TII.2018.2856776.

[42] A. Bozorgchenani, F. Mashhadi, D. Tarchi, S. Salinas Monroy, Multi-objective computation sharing in energy and delay constrained mobile edge computing environments, IEEE Transactions on Mobile Computing (May 2020). doi:10.1109/TMC.2020.2994232.

[43] H. Yetgin, K. T. K. Cheung, M. El-Hajjar, L. H. Hanzo, A survey of network lifetime maximization techniques in wireless sensor networks, IEEE Communications Surveys and Tutorials 19 (2) (2017) 828–854. doi:10.1109/COMST.2017.2650979.