

Alma Mater Studiorum Università di Bologna  
Archivio istituzionale della ricerca

Dual-mode wake-up nodes for IoT monitoring applications: Measurements and algorithms

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

*Published Version:*

Dual-mode wake-up nodes for IoT monitoring applications: Measurements and algorithms / Bedogni, Luca; Bononi, Luciano; Canegallo, Roberto; Carbone, Fabio; Di Felice, Marco; Scarselli, Eleonora Franchi; Montori, Federico; Perilli, Luca; Cinotti, Tullio Salmon; Trotta, Angelo. - ELETTRONICO. - (2018), pp. 8422173.1-8422173.7. (Intervento presentato al convegno 2018 IEEE International Conference on Communications, ICC 2018 tenutosi a Kansas City, MO, USA nel 2018, 20 - 24 May)  
[10.1109/ICC.2018.8422173].

*Availability:*

This version is available at: <https://hdl.handle.net/11585/660300> since: 2019-02-05

*Published:*

DOI: <http://doi.org/10.1109/ICC.2018.8422173>

*Terms of use:*

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (<https://cris.unibo.it/>).  
When citing, please refer to the published version.

(Article begins on next page)

This is the post peer-review accepted manuscript of:

L. Bedogni, L. Bononi, R. Canegallo, F. Carbone, M. Di Felice, E. Franchi Scarselli, F. Montori, L. Perilli, T. Salmon Cinotti and A. Trotta, "Dual-Mode Wake-Up Nodes for IoT Monitoring Applications: Measurements and Algorithms", 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, 2018, pp. 1-7. doi: 10.1109/ICC.2018.8422173

The published version is available online at: <https://doi.org/10.1109/ICC.2018.8422173>

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

# Dual-Mode Wake-Up Nodes for IoT Monitoring Applications: Measurements and Algorithms

Luca Bedogni\*, Luciano Bononi\*, Roberto Canegallo<sup>†</sup>, Fabio Carbone<sup>†§</sup>, Marco Di Felice\*, Eleonora Franchi Scarselli<sup>†§</sup>, Federico Montori\*, Luca Perilli<sup>†§</sup>, Tullio Salmon Cinotti<sup>\*§</sup>, Angelo Trotta\*

\* Department of Computer Science and Engineering, University of Bologna, Italy

<sup>†</sup> Department of Electrical, Electronic, and Information Engineering, University of Bologna, Italy

<sup>§</sup> Advanced Research Center on Electronic Systems "Ercolo De Castro" - ARCES, University of Bologna, Italy

<sup>‡</sup> STMicroelectronics, Agrate Brianza, Italy

Emails: roberto.canegallo@st.com, {luca.bedogni4, luciano.bononi, fabio.carbone7, marco.difelice3, eleonora.franchi, federico.montori2, luca.perilli, tullio.salmoncinotti, angelo.trotta5}@unibo.it

**Abstract**—Internet of Things (IoT)-based monitoring applications usually involve large-scale deployments of battery-enabled sensor nodes providing measurements at regular intervals. In order to guarantee the service continuity over time, the energy-efficiency of the networked system should be maximized. In this paper, we address such issue via a combination of novel hardware/software solutions including new classes of Wake-up radio IoT Nodes (WuNs) and novel data- and hardware-driven network management algorithms. Three main contributions are provided. First, we present the design and prototype implementation of WuN nodes able to support two different energy-saving modes; such modes can be configured via software, and hence dynamically tuned. Second, we show by experimental measurements that the optimal policy strictly depends on the application requirements. Third, we move from the node design to the network design, and we devise proper orchestration algorithms that select both the optimal set of WuN to wake-up and the proper energy-saving mode for each WuN, so that the application lifetime is maximized, while the redundancy of correlated measurements is minimized. The proposed solutions are extensively evaluated via OMNeT++ simulations under different IoT scenarios and requirements of the monitoring applications.

## I. INTRODUCTION

At present, environmental and structural health monitoring constitute two broad application fields which might take advantage from the recent advances of the Internet of Things (IoT) [1]. Energy efficiency represents one of the main requirements specially in large-scale and wild environments, or in installations where sensors cannot be easily accessed. To address such issue, three main enabling technologies have been proposed, namely, duty cycling, data driven and hardware-based. The first approach has been largely investigated in Wireless Sensor Networks (WSNs), and consists in the alternation of two phases, i.e. the active phase and the sleep phase, during which the device is fully operative or in low-power mode [2]. Clearly, the duration of the sleep time becomes a crucial issue, and introduces a significant trade-off with the Quality-of-Service (QoS) of applications with stringent latency requirements. Moreover, in duty-cycled protocols, nodes should periodically wake-up to check the channel and exchange data, and this poses the need for synchronization strategies in distributed

scenarios [3]. Since the energy consumption of sensor devices is largely determined by the wireless radio transceiver, data-driven techniques aim at reducing the amount of transmissions performed by the nodes of the WSNs, e.g. avoiding redundant, duplicate or unnecessary data exchange. To this aim, data aggregation and compression schemes can be employed at the WSN border routers [4]; conversely, correlation-aware clustering solutions aim to identify sensor nodes that produce correlated data in the spatial or temporal domains [5] [6]. Among the hardware solutions, wake-up radio technology is a way for maximizing the energy efficiency and is based on the utilization of a near-zero consumption secondary transceiver which triggers the main radio interface when a specific signal is detected [7]. In the Passive configuration, the wake-up radio harvests energy from the RF signals, and hence introduces full-zero consumptions [8]. While there is plenty of research on the circuitry design of a Wake-Up receiver (WuRx) (e.g. [9] [10]), few works consider the integration of WuRx on a full operative IoT node, and, as next step, the operations of a network of IoT nodes enabled with wake-up functionalities. In this latter case, given the receiver-oriented nature of the communication, the energy efficiency of the network deployment strictly depends on the intelligent orchestrations of the wake-up operations; hence, proper scheduler components should be deployed [11].

Following such thread, this paper shifts the perspective of energy efficiency from the single device to the IoT monitoring application, and addresses the problem of how to maximize the service lifetime while guaranteeing the QoS requirements. We consider a generic application which must query  $\lambda$  homogeneous sensor devices at a fixed rate. Differently from the previous works, we propose a unified hardware/data-driven approach, which combines the utilization of prototypal WuRx-enabled devices with novel algorithms that achieve optimal orchestrations of the wake-up operations, while minimizing the presence of redundant readings. More specifically, we provide three main research contributions in this paper:

- First, we describe the architecture and the implementation of a prototype IoT Wake-Up node (WuN), which

integrates the WuRx transceiver for both the near-zero (Active) and full-zero (Passive) configurations. In addition, our WuN supports two operative modes for the IoT node, namely the Switch (SW) mode which completely powers-off the main board, and the Lowpower (LP) mode, which leaves it in sleep state (see Figure 1). Both the modes can be activated via software, in a dynamic way.

- Second, we measure the energy consumption of the LP and SW modes for both Active/Passive WuRx, and we demonstrate that the optimal mode in terms of energy efficiency depends on the application rate, i.e. on how often the WuN should be woken up.
- Third, based on the experimental results, we propose a centralized orchestration framework which maximizes the lifetime of a network of WuNs. The proposed scheme works in two stage. First, a novel correlation-aware clustering algorithm splits the WuN set into subgroups so that the maximal intra-cluster correlation is minimized. Then, a cluster scheduling algorithm decides the WuNs to query at each slot, and the current operative mode (SW or LP) for the selected WuNs.

We propose a modular evaluation of the clustering and scheduling components via OMNeT++ simulations, and demonstrate the ability of our framework under different IoT scenarios and monitoring application requirements. The rest of the paper is structured as follows. Section II describes the architecture of the dual-mode WuN devices, and reports the energy measurements. Section III introduces the problem of lifetime maximization for a network of WuN devices. Section IV introduces our two-stage framework, which is evaluated in Section V. Future works are discussed in Section VI.

## II. SYSTEM AND MEASUREMENTS

We present here the design of a novel multi-mode Wake-Up node (WuN) for IoT applications. Details of the architecture are provided in Section II-A, while the implementation and the evaluation are described in Section II-B.

### A. WuN Architecture

A WuN device is composed of three main hardware blocks: a battery, a Wake-Up receiver (WuRx) module, and an IoT node. The latter includes a sensor, a radio device for data communication and a dedicated antenna. The WuRx circuit is in charge of detecting signals from the wake-up transmitter (WuTx), and of re-activating the IoT module accordingly. Based on the way the WuRx is powered, we distinguish between two classes of WuN [7]:

- **Active** WuN. The WuRx is powered by the internal battery, with a minimal energy consumption in idle state.
- **Passive** WuN. The WuRx harvests energy from an external RF illuminator, hence its consumption in idle state is completely zero.

Clearly, the Passive configuration constitutes the most effective solution in terms of energy saving [9] [10]; at the same time, both the WuRx and the WuTx must be located within an operative range which allows energy harvesting from the

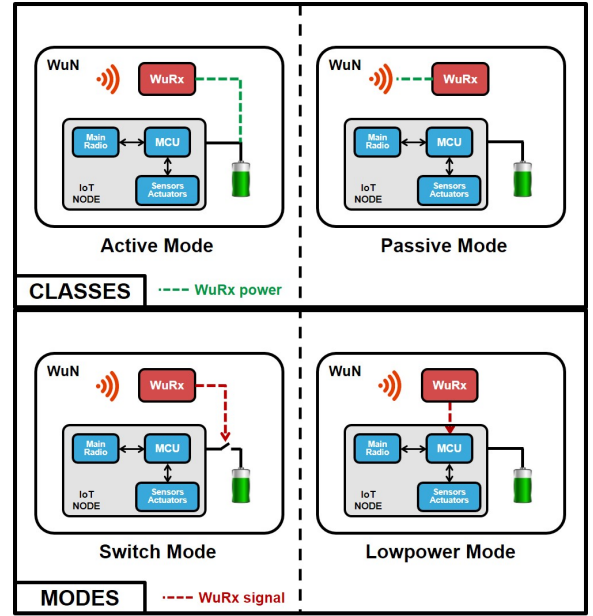


Fig. 1. The architecture of the WuN device, with the different classes/modes.

	Active		Passive	
	LP	SW	LP	SW
$E_{ON}$	$494\mu J$		$494\mu J$	
$E_{STB}$	$12\mu W \cdot t_{STB}$	$2\mu W \cdot t_{STB}$	$10\mu W \cdot t_{STB}$	0
$E_{BOOT}$	0	$7.5mJ$	0	$7.5mJ$

TABLE I  
AVERAGE ENERGY CONSUMPTION VALUES

received RF signal. Such range is significantly lower than the Active case, as investigated in the next Section. Regardless of the class, the proposed WuN architecture supports two additional operative modes, which differ on the way the IoT node is powered:

- **Switch mode** (SW). The power supply of the IoT node can be dynamically enabled/disabled by acting on a hardware switch. When the switch is off, the consumption in idle state ( $E_{STB}$ ) is zero; however, a significant overhead is induced at each boot phase, both in terms of energy ( $E_{BOOT}$ ) and time ( $t_{BOOT}$ ).
- **Lowpower mode** (LP). An output trigger connected to an interrupt MCU input is used to wake-up the IoT node from a sleep state. Since the hardware switch is always ON, the IoT node consumes an amount of energy  $E_{STB}$  proportional to the duration of the sleep period ( $t_{STB}$ ).

We highlight that classes and modes are orthogonal, hence leading to the four different configurations depicted in Figure 1. However, while the class depends on different *hardware* settings of the WuN, our current deployment allows the mode to be tunable via *software*, in order to meet the QoS requirements of the applications; the IoT node, depending on the output of the scheduling algorithm (see Section IV-B), sets the hardware switch in on/off state in order to put itself in LP/SW mode, respectively.

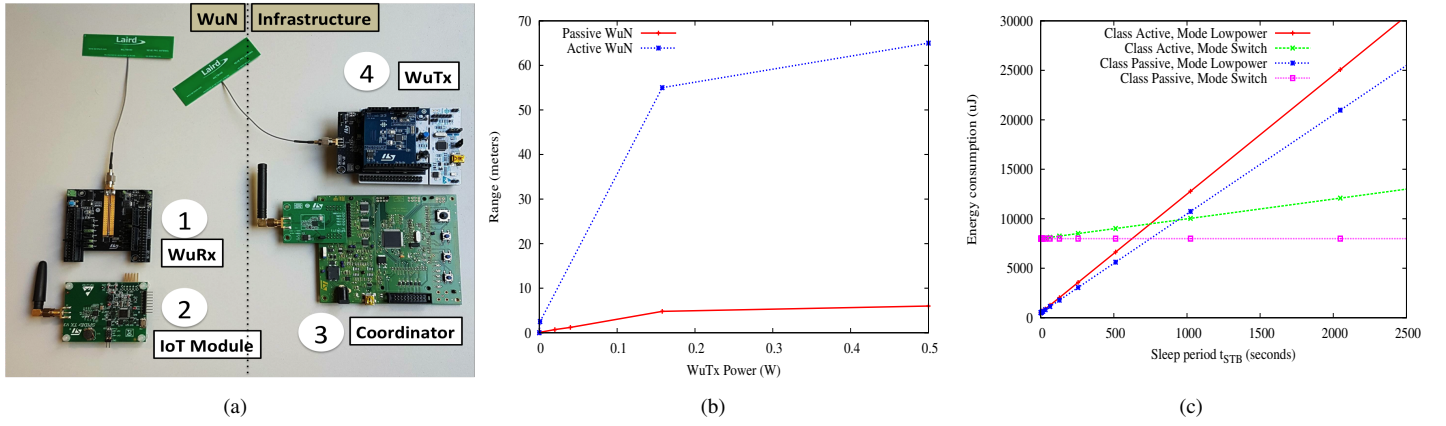


Fig. 2. The WuN deployed for the measurements is shown in Figure 2(a). The operative range of Active/Passive WuNs is depicted in Figure 2(b). The energy consumption for different classes/modes combination is shown in Figure 2(c), as a function of the  $t_{STB}$  length.

### B. Implementation and Measurements

We performed a prototype implementation of the proposed WuN architecture, by using the WuRx radio [12] [13] developed by STMicroelectronics. The WuRx is composed of a radio with a sensitivity of -18 dBm @ 868MHz in Passive Mode or -38 dBm @ 868MHz in Active Mode, an RF to DC energy transducer (RF energy harvester), an adjustable LDO, and an ultra-low power management unit. The IoT module includes an ultra-low power microcontroller (STM32L1<sup>1</sup>), a subGHz radio device for data communication (SPIRIT<sup>2</sup>), a battery and a temperature sensor (STTS751<sup>3</sup>). The microcontroller acquires the temperature data and implements the communication data link based on DASH7 standard in request/response mode. Figure 2(a) shows a picture of the WuN deployed and used for the measurements. Figure 2(b) depicts the maximal WuTx-WuRx distance at which it is possible to wake-up the WuN device with 100% probability of success; on the  $x$ -axis, we vary the transmitting power of WuTx, for both the Active and Passive classes. Not surprisingly, the Active class achieves a range which is 11 times greater than the Passive for the maximal power configuration (0.5W). However, the range increase is counterbalanced by the additional energy consumption of the Active class, for both the SW and LP modes, as reported in Table I. Here,  $E_{ON}$  refers to the energy consumption of the WuN device when a single wake-up/sensing/transmission sequence occurs, which is clearly not dependent on the class/mode in use. Finally, Figure 2(c) compares the four configurations in terms of average energy consumption on a duty-cycle (i.e. ON-OFF transition) when varying the sleep length ( $t_{STB}$ ). It is easy to notice that in both the Active/Passive classes, there is a threshold  $t_{STB}$  value after which the SW mode is always preferable to the LP mode. The threshold depends on how often the WuN is woken up; this demonstrates that, in a centralized deployment, the mode selection cannot be decoupled from the application selection.

### III. WUN NETWORK DEPLOYMENT: PROBLEM FORMULATION

In this Section, we move from the perspective of single WuN device, to the perspective of a WuN-based sensor network scenario. Based on the comments on Figure 2(c), we design optimal centralized scheduling strategies, which decide the proper operative mode of each WuN, so that user-decided network requirements are always met. To this aim, we model the network scenario as a couple  $\langle U, E \rangle$ , where:

- $U = \{u_0, u_1, \dots, u_{N-1}\}$  is the set of WuNs, placed at random locations of the scenario;
- $E = \{e_0, e_1, \dots, e_{m-1}\}$  is the set of WuTx irradiators, placed in a way that the scenario is fully covered. Let  $\nu : U \rightarrow E$  the function assigning each WuN  $u_i$  to the corresponding WuTx  $e_j$ .

We restrict our analysis to the case where all the WuNs are Passive devices, since this configuration maximizes the battery lifetime (see Table 1), although increasing the deployment costs. The optimal combination of Active and Passive WuNs achieving the best trade-off between energy efficiency and costs will be investigated as future study. We assume a slotted time model, with fixed slot length equal to  $t_{slot}$ . Let  $t_j$  denote the  $j$ -th time slot since the system boot. We model the functionality of a generic IoT monitoring applications with a fixed data-rate: exactly  $\lambda \leq N$  measurements must be gathered from the sensor network at each time slot, with  $\lambda$  being a user-defined parameter. Let  $C(t_j) \subseteq U$ ,  $|C(t_j)| = \lambda$  indicate the subset of WuNs selected for data reading at slot  $t_j$ , and  $\phi : U \times T \rightarrow \{0, 1\}$  be the related kernel function, so that  $\phi(u_i, t_j) = 1$  whether  $u_i \in C(t_j)$ , 0 otherwise. In order to reduce the data redundancy, which might be considered a wastage of energy [5] [6], we also require the  $\lambda$  measurements to be maximally uncorrelated. Let  $CR(u_i, u_j)$  be a proxy for the amount of correlation between the data readings performed by  $u_i$  and  $u_j$ . A straightforward approach is to consider the spatial correlation among the WuNs, i.e.  $CR(u_i, u_j) = \frac{1}{d(u_i, u_j)}$ . As a result, minimizing the correlation metric is equivalent to maximize the  $\lambda$ -coverage

<sup>1</sup>Datasheet: <http://www.st.com/resource/en/datasheet/stm32l151c6.pdf>

<sup>2</sup>Datasheet: <http://www.st.com/resource/en/datasheet/spirit1.pdf>

<sup>3</sup>Datasheet: <http://www.st.com/resource/en/datasheet/stts751.pdf>

of the scenario. We use such formulation in the simulation analysis (Section V); however, we remark that our algorithmic solution is agnostic on the way the  $CR$  function is defined. We then define the correlation of a set of nodes  $C(t_j)$  as the maximum correlation between each couple, i.e.:

$$CR(C(t_j)) = \max CR(u_i, u_j), \forall u_i, u_j \in C(t_j) \quad (1)$$

Besides the WuNs and the WuTx devices, the scenario includes a coordinator node, which implements the scheduler algorithm, and is in charge of: (i) selecting exactly  $\lambda$  WuN to wake-up at each slot  $t_j$ , and (ii) deciding the next mode for the selected WuNs, choosing between SW or LP. All the coordinator actions are implemented via the WuTx devices. More formally, we introduce the state function  $S : U \times T \rightarrow \{0, 1\}$ , such that:  $S(u_i, t_j) = 1$  whether WuN  $u_i$  is in LP mode at time slot  $t_j$ , and  $S(u_i, t_j) = 0$  whether WuN  $u_i$  is in SW mode at time slot  $t_j$ . We denote with  $E(u_i, t_j)$  the residual energy of WuN  $u_i$  at slot  $t_j$ , with  $E(u_i, t_0) = E_{INIT}$ . Based on  $C(t_j)$  and  $S(u_i, t_j)$ , we update  $E(u_i, t_j)$ , at each time slot  $t_j$  and for each WuN  $u_i$  as follows:

$$\begin{aligned} E(u_i, t_j) &= E(u_i, t_{j-1}) \\ &- \phi(u_i, t_j) \cdot [E_{ON} + E_{BOOT} \cdot (1 - S(u_i, t_{j-1}))] \\ &- S(u_i, t_j) \cdot E_{STB} \end{aligned}$$

Finally, we introduce the alive function  $A : U \times T \rightarrow \{0, 1\}$ , such that  $A(u_i, t_j) = 1$  iff  $E(u_i, t_j) > 0$ , i.e. WuN  $u_i$  has still some residual energy at slot  $t_j$ . Vice versa,  $A(u_i, t_j) = 0$ .

**Problem formulation.** Given the scenario  $\langle U, E \rangle$  and the application parameter  $\lambda$ , we want to determine the selection set  $C(t_j)$  and the state function  $S(u_i, t_j)$  at each slot  $t_j$ , so that the application lifetime  $L$  is maximized, subject to:

- 1)  $\sum_{0 \leq i < n} \phi(u_i, t_j) = \lambda \forall t_j \in T$ , i.e. the coordinator gathers exactly  $\lambda$  measurements at each slot.
- 2)  $\min(\max(CR(t_j)))$ , i.e. the maximum correlation among the sensor readings performed during the system life cycle is minimized.

The application lifetime is defined as the maximum  $t_{final}$  slot such that  $\lambda$  measurements can be gathered from the network, i.e. there are still at least  $\lambda$  WuNs alive, i.e.:

$$L = \max_j s.t. \sum_{0 \leq i < n} A(u_i, t_j) \geq \lambda \quad (2)$$

#### IV. WUN NETWORK MANAGEMENT FRAMEWORK

We split the scheduling problem into two parts. First, we partition the sensor set  $U$  in  $k = \lfloor \frac{|U|}{\lambda} \rfloor$  maximally uncorrelated clusters, according to the CR metric previously introduced. Let  $G_0, G_1, \dots, G_{k-1}$  be the clusters, such that:  $G_i \subseteq U, G_i \cap G_j = \emptyset \forall G_i, G_j$  and  $\bigcup G_i = U$ . Moreover, all the clusters are balanced and contain exactly  $\lambda$  sensors, except for the first cluster which might contain  $\lambda + r$  elements, with  $r = N\% \lambda$ . Then, we schedule the clusters in a sequential order; once one node in cluster  $G_i$  (or  $r + 1$  nodes in cluster  $G_0$ ) runs out of battery, we move to the next cluster  $G_{i+1}$ . The clustering algorithm is detailed in Section IV-A. The optimal scheduler is presented in Section IV-B.

---

#### Algorithm 1: Clustering algorithm

---

```

1: procedure Cluster ( $U, \lambda$ )
2:  $k \leftarrow \lfloor \frac{|U|}{\lambda} \rfloor$ 
3: for  $i = 0$  to  $k - 1$  do
4:    $G_i = \emptyset$ 
5: end for
6: randomize  $U$  in  $(G_0, G_1, \dots, G_{k-1})$ 
7: while (true) do
8:    $(u_i, u_j, G_l) \leftarrow \underset{u_i, u_j \in G_l, u_i \neq u_j, 1 \leq l \leq k}{\operatorname{argmax}} (CR(u_i, u_j))$ 
9:   for all  $G_p \neq G_l$  do
10:    for all  $u_q \in G_p$  do
11:      if (positiveSwap( $CR(u_i, u_j), u_i, G_l, u_q, G_p$ )) then
12:         $G_p \leftarrow G_p \setminus \{u_q\} \cup \{u_i\}$ 
13:         $G_l \leftarrow G_l \setminus \{u_i\} \cup \{u_q\}$ 
14:        continue while
15:      else if (positiveSwap( $CR(u_i, u_j), u_j, G_l, u_q, G_p$ )) then
16:         $G_p \leftarrow G_p \setminus \{u_q\} \cup \{u_j\}$ 
17:         $G_l \leftarrow G_l \setminus \{u_j\} \cup \{u_q\}$ 
18:        continue while
19:      end if
20:    end for
21:  end for
22:  break while
23: end while
24: return  $(G_0, G_1, \dots, G_{k-1})$ 
25:
26: procedure positiveSwap ( $c_{max}, u_i, G_l, u_q, G_p$ )
27:  $cost_{i,p} \leftarrow \max_{u_j \in G_p, u_j \neq u_q} (CR(u_i, u_j))$ 
28:  $cost_{q,l} \leftarrow \max_{u_j \in G_l, u_j \neq u_i} (CR(u_q, u_j))$ 
29: if ( $cost_{i,p} < c_{max}$ ) and ( $cost_{q,l} < c_{max}$ ) then
30:   return (true)
31: else
32:   return (false)
33: end if

```

---

#### A. Clustering algorithm

The aim of the clustering algorithm is to minimize the maximal correlation among two distinct WuNs belonging to the same cluster. More formally, the objective function of the clustering process, i.e.  $f_G : G_0, G_1, \dots, G_{k-1} \rightarrow [0..1]$  is:

$$f_G(G_0, \dots, G_{k-1}) = \min(\max\{CR(G_0), \dots, CR(G_{k-1})\}) \quad (3)$$

The problem can be considered an instance of the min-max clustering [14], which is computationally difficult (NP-hard); in addition, it includes the constraint on the load-balancing among clusters, which has not been considered in the literature. For this reason, we provide a novel, heuristic solution whose approximation bounds are analyzed in Section V. The clustering operations are detailed by Algorithm 1.

We assume that  $|U| \geq \lambda$ . First, each WuN  $u_i \in U$  is randomly assigned to a cluster, so that at the end of the *randomize* procedure, there are  $(k - 1)$  clusters with  $\lambda$  WuNs and one cluster with at least  $\lambda$  WuNs. Then, the algorithm follows an iterative procedure, so that the worst link between any two nodes  $u_i, u_j \in C_l$  (line 8) is selected, and compared against any other node  $u_q \in C_p \neq C_l$  (lines 11-19). A cluster swap between  $u_q$  and  $u_i$  or  $u_j$  is performed in case the *positiveSwap* procedure (lines 26-33) returns **true**, which indicates that the swap operation produces a decrease of the objective function  $f_G(G_0, G_1, \dots, G_{k-1})$ . The algorithm ends when it converges to a stable assignment of WuNs to cluster, i.e. no other swap operations are performed (line 22). It is easy to notice that, at the each iteration, the link with maximal correlation



is computed ( $O(n^2)$ ). Hence, the overall complexity of the algorithm is  $O(\text{num}_{iter} \cdot n^2)$ . We now prove that the algorithm always converges to a solution, and that  $\text{num}_{iter} = O(n^2)$ .

**Theorem 1.** *The maximum number of iterations performed by Algorithm 1 is  $O(n^2)$ .*

*Proof.* Let  $CR(G_i, t)$  be the correlation value of cluster  $G_i$  at iteration  $t$  of Algorithm 1, and  $\text{maxCR}(t) = \text{max}_{0 \leq i < k-1} CR(G_i, t)$ . Let  $(u_i, u_j)$  be the link selected at iteration  $t$  and line 8, with  $u_i \in G_l$ ,  $u_j \in G_l$ , so that  $CR(u_i, u_j) = \text{maxCR}(t)$ . If a swap occurs, then by construction,  $\text{maxCR}(t+1) < \text{maxCR}(t)$ , i.e. the correlation decreases at each iteration. As a result, the same link  $(u_i, u_j)$  can be selected only once by the Algorithm. Since the number of links is  $O(n^2)$ , this implies that the number of iterations cannot exceed such value in the worst case.  $\square$

### B. Cluster Scheduling algorithm

The algorithm employs two different policies, named ROUND\_ROBIN and GREEDY. The ROUND\_ROBIN policy is used only for cluster  $G_0$  in case  $r > 0$ , while the GREEDY policy is adopted for all the other clusters. If  $r = 0$ , then the GREEDY policy is adopted also for cluster  $G_0$ . The operations of the proposed scheduler are detailed by Algorithm 2.

#### Algorithm 2: Scheduling algorithm

```

1: procedure Schedule ( $U, \lambda$ )
2: Set  $k = \lfloor \frac{N}{\lambda} \rfloor$ ,  $r = N \% \lambda$ ,  $\text{time} = 0$ 
3: Create groups  $G_0, G_1, \dots, G_{k-1}$  as in Section IV-A
4: Order group  $G_0 = \{u_{0,1}, u_{0,2}, \dots, u_{0,|\lambda+r-1|}\}$  based on ids
5: Set  $S(u_{0,j}, t_{\text{time}}) = 1$ ,  $\forall 0 \leq j < |\lambda|$ 
6: for  $i = 0$  to  $k-1$  do
7:   if  $i == 0$  and  $r > 0$  then
8:     policy  $\rightarrow$  ROUND_ROBIN
9:   else
10:    policy  $\rightarrow$  GREEDY
11:   end if
12:   if policy=ROUND_ROBIN then
13:     Compute  $ST$  as in Equation 4
14:     for  $j = 0$  to  $|G_0| - 1$  do
15:       Build set  $G'_0 = \{u_{0,j}, u_{0,j+1}, \dots, u_{0,(j+\lambda-1)\%(\lambda+r)}\}$ 
16:       for  $k = 0$  to  $ST - 1$  do
17:         Set  $\phi(u_{0,j}, t_{\text{time}}) = \text{getMode}() \forall u_{0,j} \in G'_0$ 
18:         Set  $S(u_{0,j}, t_{\text{time}}) = 1 \forall u_{0,j} \in G'_0$ 
19:         Update  $\text{time} = \text{time} + 1$ 
20:       end for
21:     end for
22:   while  $A(t_{\text{time}}, u_{0,j}) == 1 \forall u_{0,j} \in G'_0$  do
23:     Set  $\phi(u_{0,j}, t_{\text{time}}) = \text{getMode}() \forall u_{0,j} \in G'_0$ 
24:     Update  $\text{time} = \text{time} + 1$ 
25:   end while
26:   else
27:     while  $A(t_{\text{time}}, u_{i,j}) == 1 \forall u_{i,j} \in G_i$  do
28:       Set  $\phi(u_{i,j}, t_{\text{time}}) = \text{getMode}() \forall u_{i,j} \in G_i$ 
29:       Set  $S(u_{i,j}, t_{\text{time}}) = 1 \forall u_{i,j} \in G_i$ 
30:       Update  $\text{time} = \text{time} + 1$ 
31:     end while
32:   end if
33: end for
34: return  $(\phi, S)$ 
35:
36: procedure getMode ()
37: if  $t_{\text{slot}} \cdot E_{STB} \leq E_{BOOT}$  then
38:   return 1 // mode LP
39: else
40:   return 0 // mode SW
41: end if

```

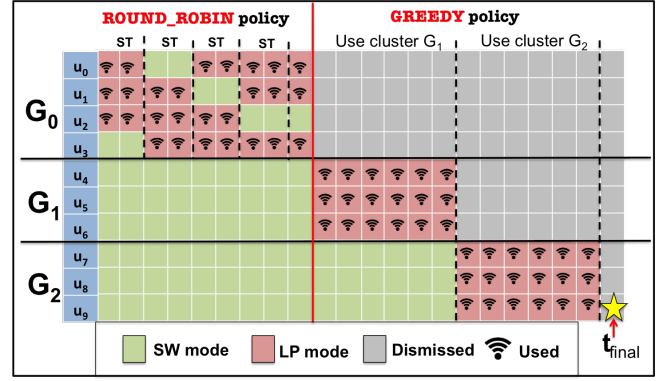


Fig. 3. The scheduler operations, for  $N = 10$ ,  $k = 3$ ,  $ST = 2$ .

If  $t_{\text{slot}} \cdot E_{STB} \geq E_{BOOT}$ , the SW mode is always preferable, and hence the mode selection is trivial. Otherwise, the rationale of the ROUND\_ROBIN policy is to make each WuN stay in SW mode for the longest number of consecutive time slots, in order to minimize the overhead of the boot phase ( $E_{BOOT}$ ). To this purpose, the scheduler orders the elements in  $G_0$  based on their id (line 4), and selects the first  $\lambda$  nodes (line 18); let  $G'_0 = \{u'_{0,1}, u'_{0,2}, \dots, u'_{0,\lambda-1}\}$  be the first subset. Each WuN  $u' \in G'_0$  is used for  $ST$  consecutive slots, in LP mode (line 17-20). The  $ST$  value is computed so that all the WuNs in  $G_0$  are used for the same, maximal amount of time, i.e.:

$$ST = \left\lfloor \frac{E_{INIT} - E_{BOOT}}{\lambda \cdot (E_{STB} + E_{ON})} \right\rfloor \quad (4)$$

After  $ST$  iterations, the group  $G'_0$  is updated on a rolling base (line 15), i.e. removing the head element  $u'_j$ , setting it to SW mode, and adding to  $G'_0$  the element next to the current tail, in a circular way, i.e.  $u'_{(j+1+\lambda)\%(\lambda+r)}$ . Exactly  $|G_0|$  iterations are executed, so that all the WuNs belonging to  $G_0$  are used for  $\lambda \cdot ST$  slots. At the end of the loop (line 23), exactly  $\lambda$  nodes are in LP mode (the ones belonging to the current  $G'_0$  set), and are used till completely discharged (line 23-26). Next, the GREEDY policy is executed on the remaining clusters. Its behaviour is straight-forward: since the application model requires  $\lambda$  measurements at each slot, all the WuNs in  $G_i$  are switched from the SW mode to the LP mode, and used at each slot, till the first WuN runs out of battery (lines 30-34). In such case, the scheduler moves to the next cluster  $G_{i+1}$  (line 6), till  $i$  is equal to  $k$ . The graphic behaviour on a specific use-case (with  $N = 10$ ,  $k = 3$ ,  $ST = 3$ ) is depicted in Figure 3. We now prove the optimality of the proposed scheduler.

**Theorem 2.** *Given the set of clusters  $G_0, G_1, \dots, G_{k-1}$ , Algorithm 2 (Algo2) maximizes the network lifetime  $t_{\text{final}}$ .*

*Proof.* Let  $t_{\text{max}}^{\text{Algo2}}(u_i)$  be the number of slots where node  $u_i$  is active (i.e. contributes to the sensing function) in Algo2, i.e.  $t_{\text{max}}^{\text{Algo2}}(u_i) = \sum_{0 \leq j \leq t_{\text{final}}} (\phi(u_i, t_j))$ . The maximum number of slots where each node can be active is:  $t_{\text{max}}^* = \left\lfloor \frac{E_{INIT} - E_{BOOT}}{E_{STB} + E_{ON}} \right\rfloor$ . It's easy to notice that if  $r = N \% \lambda = 0$ , each node  $u_i$  contributes to the sensing for exactly  $t_{\text{max}}^{\text{Algo2}}(u_i) = t_{\text{max}}^*$  slots, hence the lifetime  $t_{\text{final}} = \frac{N}{\lambda} \cdot t_{\text{max}}^*$

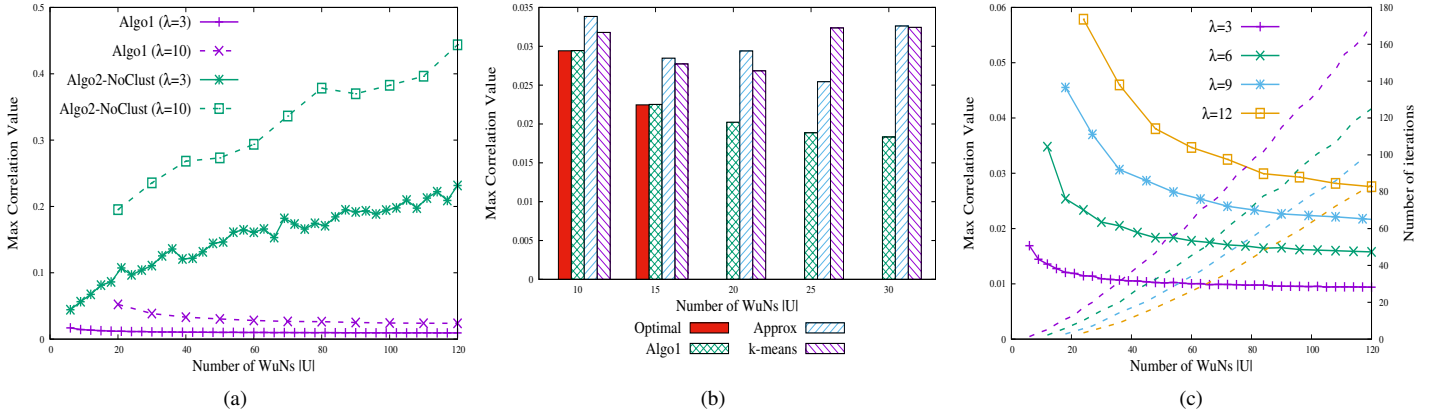


Fig. 4. The gain of *Algo1* compared to a no-clustering solution is depicted in Figure 4(a). The correlation metric of *Algo1* and other clustering schemes is depicted in Figure 4(b). The correlation metric for the *Algo1* for different values of  $|U|$  and  $\lambda$  is depicted in Figure 4(c).

is maximized. If, instead, the fractional part  $r > 0$ , *Algo2* activates the ROUND\_ROBIN mode. In this case, only  $\lambda$  nodes of  $G_0$  are active for  $t_{max}^*$  slots, while  $r$  nodes are active for  $\lambda \cdot ST \leq t_{max}^*$  slots (line 15-22 of *Algo2*). Let  $R \subset G_0$ ,  $|R| = r$  be the set of nodes with  $t_{max}^{Algo2}(u_i \in R) = \lambda \cdot ST$ , and  $L \subset G_0$ ,  $|L| = \lambda$  be the subset of  $\lambda$  nodes with  $t_{max}^{Algo2}(u_j \in L) = t_{max}^*$ . By absurd, let us assume the existence of another scheduler algorithm *Algo\**, different from *Algo2*, which maximizes the system lifetime. This means that *Algo\** activates node  $u_i \in R$  with  $t_{max}^{Algo*}(u_i) > t_{max}^{Algo2}(u_i)$ . However, this is impossible because  $t_{max}^* - (\lambda \cdot ST) < \lambda$ . Hence, *Algo\** must be equal to *Algo2*.  $\square$

## V. SIMULATION RESULTS

In this Section, we propose a modular evaluation of our framework under different IoT scenarios and application requirements. First, in Section V-A, we demonstrate that Algorithm 1 is able to group WuNs into clusters by minimizing the maximum intra-cluster correlation. Then, in Section V-B, we show that Algorithm 2 is able to maximize the application lifetime thanks to the intelligent scheduler of WuN activations and mode selections. The evaluation has been conducted via OMNeT++ simulations, deploying new models for the irradiators and for the energy consumption of the WuNs.

### A. Cluster Analysis

We consider a network area of  $200 \times 200$  m<sup>2</sup>, with irradiators placed according to an hexagonal pattern so that the full scenario is covered. The correlation function  $CR(u_i, u_j)$  is defined as the inverse of the node distance, i.e.  $CR(u_i, u_j) = \frac{1}{d(u_i, u_j)}$ . As a result, the goal of the cluster algorithm is to maximally space the WuNs in each group. Figure 4(a) demonstrates the gain of Algorithm 1 (*Algo1*) compared to the case where clustering step is not executed (*Algo2-NoClust*), and the WuNs are scheduled in round-robin order, choosing exactly  $\lambda$  WuNs at each slot. Two configurations of the  $\lambda$  parameter are considered for each algorithm, i.e.  $\lambda = 3$  and  $\lambda = 10$ . In both cases, *Algo1* greatly outperforms the *NoClust* scheme, and shows a decreasing trend of the correlation metric when  $|U|$

increases. In Figure 4(b) we expand the previous analysis, by comparing *Algo1* against three other algorithms with  $\lambda = 5$ :

- *Optimal*, which provides the optimal allocation of WuNs to clusters minimizing the correlation metric. Given the NP-hardness of the problem, the optimal solution is computed by generating all possible allocations; this is clearly not feasible for large values of  $|U|$ .
- *Approx*, which is the min-max clustering Algorithm defined in [14], plus an additional greedy round which aims at balancing the number of WuNs on each cluster, since this feature is not supported by the original scheme.
- *K-Means*, which adapts the well-known clustering algorithm to our problem; the centroid distance reflects the maximum correlation within each cluster.

From Figure 4(b), we can notice that: (i) no solutions could be determined for the *Optimal* Algorithm when  $|U| \geq 20$  due to the excessive computation load; (ii) *Algo1* performs equally to *Optimal* Algorithm when  $|U| < 20$ , and outperforms the Approx and K-Means schemes for  $|U| \geq 20$ ; (iii) *Algo1* reduces the correlation factor of more than 90% for  $|U| = 30$ . Figure 4(c) shows the correlation metric of *Algo1*, for different values of  $|U|$  (on the x-axis) and  $\lambda$  (different lines). On the same graph, we show on the y2 axis the number of iterations performed by our Algorithm before convergence (dotted lines). Increasing the  $\lambda$  parameter reduces the number of created clusters (equal to  $\lfloor \frac{|U|}{\lambda} \rfloor$ ), and hence produces also an increment of the correlation factor; at the same time, the algorithm converges faster since there are less changes for WuNs exchanges between different groups.

### B. Scheduler Analysis

We investigate here the energy efficiency of the scheduler described by Algorithm 2 (*Algo2*). To this aim, we model the energy discharge operations of the WuNs by using the measured energy consumption values of Table I. We consider a Lithium-battery of 2V and 300 mAh, with a self-discharge equal to 8% per month. Figures 5(a), 5(b) and 5(c) depict the system lifetime, in number of days, computed as  $t_{slot} \cdot t_{final}$ . We compare *Algo2* against three different schedulers:



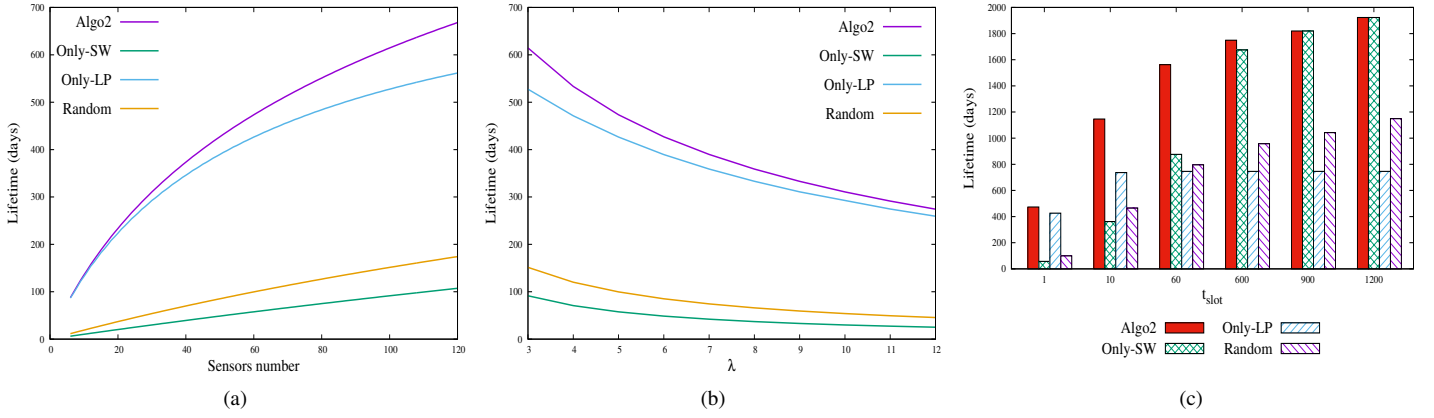


Fig. 5. The system lifetime as a function of the number of WuNs (Figure 5(a)), of the  $\lambda$  parameter (Figure 5(b)) and of the length of  $t_{slot}$  (Figure 5(c)).

- *Only-LP*: this scheme implements a greedy WuN selection, by choosing at each slot the  $\lambda$  nodes having the highest residual energy; moreover, all the WuNs are used in LP mode only, i.e. mode selection is not implemented.
- *Only-SW*: this scheme is similar to the previous case, except that all the WuNs are used in SW mode.
- *Random*: this scheme implements random selection -at each slot- of the  $\lambda$  WuNs to active; the next mode of each WuN is also chosen randomly between LP and SW.

Figure 5(a) depicts the lifetime of the four algorithms when varying the number of WuNs  $|U|$ , for  $t_{slot}=1$  second and  $\lambda=3$ . Always keeping in WuNs in LP mode is more efficient than using the SW mode, since the energy overhead induced by the boot phase is avoided. *Algo2* outperforms all the other schemes: the performance gain increases with the size of the network, and is equal to +15% (345 days more) for the case with  $|U| = 120$ . The same trend can be observed in Figure 5(b), where we vary the  $\lambda$  parameter, while keeping constant the number of WuNs (i.e.  $|U| = 60$ ). Finally, Figure 5(c) shows the lifetime when varying the  $t_{slot}$  duration, for  $|U|=60$  and  $\lambda=3$ . We can notice that, for short values of  $t_{slot}$ , the LP mode is more convenient than the SW mode; vice versa, when  $t_{slot} \geq 900$  seconds, completely switching off the nodes (i.e. using the *Only-SW* scheme) constitutes the most effective approach. The *Algo2* scheme provides the optimal performance under any configuration of  $t_{slot}$ . When  $t_{slot} \geq 750$  seconds, the *Algo2* scheme employs the SW mode only (since condition at line 40 of Algorithm 2 is not satisfied) and hence performs equally to the *Only-SW* scheme.

## VI. CONCLUSION AND FUTURE WORKS

In this paper, we have addressed the problem of maximizing the lifetime of IoT monitoring applications by using Wake-up IoT nodes (WuNs). First, we have presented the design, implementation and evaluation of novel WuN devices enabling dual energy-saving modes. Then, moving from the single device to a network of WuNs, we have presented centralized orchestration algorithms so that the application lifetime is maximized while the number of redundant measurements is

minimized. The viability of the proposed hardware/software solutions has been evaluated via measurements and OMNeT++ simulations. Future works include: the evaluation of the WuN scheduler in a test-bed and enhancements of the algorithm for the case of event-driven monitoring applications.

## ACKNOWLEDGEMENTS

This work was supported by the ALMAIDEA UniBO Grant Senior Project: “BeeDrones: environmental monitoring systems based on ultra-low power sensor nodes and aerial communications” and by the “Smart Water Management Platform” (SWAMP) Project, H2020-EUB-2017, Project id: 777112.

## REFERENCES

- [1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of Things: A Survey on enabling technologies, protocols, and applications. *IEEE Communication Survey and Tutorials*, 17(4), pp. 2347-2376, 2015.
- [2] R. C. Carrano, D. Passos, L. C. S. Magalhaes, and C. V. N. Albuquerque. Survey and taxonomy of duty cycling mechanisms in wireless sensor networks. *IEEE Communication Survey and Tutorials*, 16(1), pp. 181-194, 2014.
- [3] A. D’Elia, L. Perilli, F. Viola, L. Roffia, F. Antoniazzi, R. Canegallo and T.S. Cinotti. A self-powered WSN for energy efficient heat distribution. *Proc. of IEEE SAS*, Catania, Italy, 2016.
- [4] S. Patten, B. Krishnamachari and R. Govindan. The impact of spatial correlation on routing with compression in wireless sensor networks. *ACM Transactions on Sensor Networks*, 4(4), 2008.
- [5] F. Wang, S. Wu K. Wang and X. Hu. Energy-efficient clustering using correlation and random update based on data change rate for wireless sensor networks. *IEEE Sensors Journal*, 16(13), pp. 5471-5480, 2016.
- [6] M.-H. Tsai and Y.-M. Huang. A sub-clustering algorithm based on spatial data correlation for energy conservation in wireless sensor networks. *Sensors*, 14(1), pp. 21859-21871, 2014.
- [7] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato and D. Brunelli. Ultra low power Wake-Up radios: A hardware and networking survey. *IEEE Communications Surveys and Tutorials*, pp. 1-42, 2017.
- [8] J. Rinne, J. Keskinen, P. R. Berger, D. Lupo and M. Valkama. Viability bounds of M2M communication using energy-harvesting and passive wake-up radio. *IEEE Access*, pp. 1-9, 2017.
- [9] D. Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli and C. Petrioli. Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems. *Proc. of IEEE INFOCOM*, Hong Kong, 2015.
- [10] M. Del Prete, D. Masotti, A. Costanzo, M. Magno and L. Benini. A dual-band Wake-up radio for ultra-low power wireless sensor networks. *Proc. of IEEE WiSNet*, Austin, USA, 2016.
- [11] M. Iwata, S. Tang and S. Obana. Sink-based centralized transmission scheduling by using asymmetric communication and Wake-up radio. *Proc. of IEEE WCNC*, San Francisco, USA, 2017.
- [12] R. La Rosa. RF remotely-powered integrated system to nullify standby power consumption in electrical appliances. *Proc. of IEEE IECON*, Florence, Italy, 2016.
- [13] R. Guerra, R. La Rosa. An RF-Powered FSK/ASK Receiver for Remotely Controlled Systems. *Proc. of IEEE RFIC*, San Francisco, USA, 2016.
- [14] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(1), pp. 293-306, 1985.