# GACN: Self-clustering Genetic Algorithm for Constrained Networks

J. A. Martins, A. Mazayev, N. Correia, G. Schütz, A. Barradas

***Abstract*—Extending the lifespan of a Wireless Sensor Network is a complex problem that involves several factors, ranging from device hardware capacity (batteries, processing capabilities, radio efficiency) to the chosen software stack, which is often unaccounted for by previous approaches. This paper proposes a Genetic Algorithm-based clustering optimisation method for Constrained Networks that significantly improves previous state-of-the-art results, while accounting for the specificities of IETF CoRE standards for data transmission and specifically relying on CoRE Interfaces, which fit this purpose very well.**

***Index Terms*—Internet of Things, CoRE Interfaces, Energy Saving, Genetic Algorithm.**

## I. Introduction

Wireless sensor networks have a key role in many applications, and such trend is expected to grow. For this reason, better ways to reach and manage such devices and their data, at any time and place, are needed while having energy saving in mind for network lifetime increase. However, there is a high level of heterogeneity in *Wireless Sensor Networks* (WSNs) and different proprietary and non-proprietary solutions are available. The current trend to overcome this inconvenience is to move away from closed standards, embracing IP-based sensor networks and using common upper layer protocols [2]. This way WSNs can be more easily integrated into the *Internet of Things* (IoT).

For IoT to become a reality, the *Internet Engineering Task Force* (IETF) initiated the standardisation process of *IPv6 over Low-Power Wireless Area Networks* (6LoWPAN) [9]. Also within the IETF, the *Constrained RESTful Environments* (CoRE) working group has been focused on the development of the *Constrained Application Protocol* (CoAP), a Web application transfer protocol intended to provide RESTful services in constrained environments [11]. An important extension to CoAP, called Observe, has been proposed by Hartke [5] to give clients the ability to observe resource changes.

Besides the above-mentioned protocols, standards for resource discovery in constrained nodes are being developed in [10, 12]. With such standards, CoRE-based applications will be able to use Web discovery and linking in constrained environments, and flexible interfaces can be provided. More recently, a set of Link Format compatible *Interface Types* started to be standardised[1] by Shelby et al. [13].

This article proposes a self-adapting energy efficient approach for monitoring WSNs. Such approach is based on the use of CoRE Interfaces for the creation of *Collection* resources at locations that allow for an increase in network lifetime. There are several advantages to this approach, such as being able to model the battery state of nodes, radio transmission energy costs, and node processing efficiency or limitations. Also, the use of CoRE Interfaces allows for less control messages to be used, reducing energy consumption while being a standards-based solution.

The remainder of this article is organised as follows. In Section II, the ongoing standardisation process on CoRE Interfaces is discussed. In Section III, the state of the art on WSN routing is presented. Section IV discusses the self-adapting model used to improve network lifetime in WSN monitoring applications. Section V evaluates the performance of the model, while Section VI elaborates the conclusions.

## II. CoRE interfaces for resource design

Interface Types, proposed by Shelby et al. [13], allow a server to compose/organise resources, and a client to discover and determine how to consume them. A Collection is a resource representing one or more related resources. Collections allow resources to be organised for discovery, observation, and various forms of bulk interaction according to the Interface Type used. A Collection at node $A$ may include reference to resources available at other nodes, meaning that $A$ will have to register, at those nodes, for observation of these resources using COAP/Observe registration messages. A Collection is itself a resource and can be referenced by other collections.

CoRE Interfaces allow Collections to be built/updated dynamically which facilitates integration into applications. Such Collections can be built with the purpose of reducing energy consumption because of the following benefits: (a) Collections allow data to be aggregated and can be displaced according to the battery state of nodes and distance between nodes, influencing energy consumption; (b) Fewer control messages are required to set up resource observations since Collections incorporate multiple resources. Also, fewer notifications will travel on the network, reducing the control to sensor data ratio.

## III. State of the art

Several techniques have been developed to extend WSN longevity. Since Smaragdakis et al. [14] created the *Stable Election Protocol* (SEP) various extensions have been proposed, such as the *Developed Distributed Energy-Efficient Clustering* (DDEEC) [3] and *Threshold Sensitive Stable Election Protocol* (TSEP) [7], which categorise sensor nodes based on remaining energy and try to select cluster heads from the nodes with higher remaining energy.

J. A. Martins, A. Mazayev, N. Correia, G. Schütz and A. Barradas are with the Center for Electronic, Optoelectronic and Telecommunications (CEOT).

J. A. Martins and A. Mazayev are researchers (emails: {jamartins, amazayev}@ualg.pt), N. Correia and A. Barradas are with the Faculty of Science and Technology (emails: {ncorreia, abarra}@ualg.pt), and G. Schütz is with the Institute of Engineering (email: gschutz@ualg.pt); all at the University of Algarve, 8005-139 Faro, Portugal.

[1] Still an ongoing work.

The *Hybrid Energy Efficient Reactive* protocol (HEER) was developed by Javaid et al. [6], and selects cluster heads based on the ratio of the residual energy of nodes and average energy of the network.

Kumar et al. [8] proposed an *Energy Efficient Heterogeneous Clustered* (EEHC) scheme, and Tuah et al. [15] proposed an *Efficient Three-Level Energy* (ETLE) algorithm, which select cluster heads based on probabilities proportional to the residual energy of nodes.

Elhoseny et al. [4] developed a *Genetic Algorithm for Heterogeneous Networks* (GAHN) that creates clusters based on: residual energy of nodes, expected energy expenditure, distance to the gateway, and number of neighbour nodes.

Our approach is different from the above methods, as we incorporate the role of a cluster head into a *Collection Aggregator* (CA), on top of a standardised IETF CoRE framework.

## IV. The Model

We simulate a WSN architecture of 50 sensor nodes transmitting to a single gateway router. At each network round, sensor data is dynamically aggregated by Collections at elected CA nodes, having as goal the minimisation of energy consumption.

### A. Energy Model

*Radio Model:* The energy consumption of the wireless sensor nodes is modelled using the *first order radio model*, as described in [4]. This is a standard radio model which defines the consumed energy $E$ of a wireless sensor node, as the sum of the energies it requires to Acquire, Receive, Process and Transmit data bits[2]. Mathematically,

$$E \triangleq E_{\mathrm{A}}(b) + E_{\mathrm{R}}(b) + E_{\mathrm{P}}(b) + E_{\mathrm{T}}(b, d), \qquad (1)$$

where $E_{\mathrm{A}}(b)$ represents the required energy to Acquire $b$ bits of data, $E_{\mathrm{R}}(b)$ the energy to Receive $b$ bits of data, $E_{\mathrm{P}}(b)$ the energy to Process and aggregate $b$ bits of data, and $E_{\mathrm{T}}(b, d)$ the energy to Transmit $b$ bits of data at a distance $d$. In the network simulation detailed in this article, the terms will be:

$$E_{\mathrm{A}}(b) = 0 \qquad (2a)$$

$$E_{\mathrm{R}}(b) \triangleq b \cdot \mathrm{e_{elec}} \qquad (2b)$$

$$E_{\mathrm{P}}(b) \triangleq b \cdot \mathrm{e_{dpa}} \qquad (2c)$$

$$E_{\mathrm{T}}(b, d) = \begin{cases} E_{\mathrm{T_{mf}}}(b, d) \triangleq b \cdot \left(\mathrm{e_{elec}} + \mathrm{e_{mf}}\, d^2_{[s_1,s_2]}\right) & \text{if } d_{[s_1,s_2]} < \mathrm{d_0} \\ E_{\mathrm{T_{fsp}}}(b, d) \triangleq b \cdot \left(\mathrm{e_{elec}} + \mathrm{e_{fsp}}\, d^4_{[s_1,s_2]}\right) & \text{if } d_{[s_1,s_2]} \geq \mathrm{d_0}. \end{cases} \qquad (2d)$$

The constants $\mathrm{e_{elec}}$, $\mathrm{e_{dpa}}$, $\mathrm{d_0}$, $\mathrm{e_{mf}}$, and $\mathrm{e_{fsp}}$ are detailed in Table I. The transmission circuitry of a sensor node has an energy cost based on the Euclidean distance $d_{[s_1,s_2]}$, between two arbitrary sensors $s_1$ and $s_2$:

- if $d_{[s_1,s_2]} < \mathrm{d_0}$ the wireless sensor transmits by Multipath Fading (mf), with an energy cost of $E_{T_{\mathrm{mf}}}(b, d)$;
- if $d_{[s_1,s_2]} \geq \mathrm{d_0}$ the sensor uses Free Space Propagation (fsp), with a cost of $E_{T_{\mathrm{fsp}}}(b, d)$.

TABLE I
Sensor parameters

| Parameters | | Values |
|---|---|---|
| Initial battery energy | | 0.5 J or 1.0 J |
| Transmission/Reception circuitry cost | $\mathrm{e_{elec}}$ | 50 nJ/bit |
| Data processing and aggregation cost | $\mathrm{e_{dpa}}$ | 5 nJ/bit |
| Transmission type switching distance | $\mathrm{d_0}$ | $\frac{\mathrm{e_{fsp}}}{\mathrm{e_{mf}}} = 87.7\,\mathrm{m}$ |
| Tr. cost using Multipath Fading ($d < \mathrm{d_0}$) | $\mathrm{e_{mf}}$ | 1.3 fJ/bit/m$^2$ |
| Tr. cost using Free Space Propagation ($d \geq \mathrm{d_0}$) | $\mathrm{e_{fsp}}$ | 10 pJ/bit/m$^2$ |
| Data processing efficiency* | | 100% or 200% |
| Observe register packet size | | 120 bits |
| Observe unregister packet size | | 96 bits |

* Sensors with higher data processing efficiency have reduced energy usage in relation to a normal sensor, for a similar workload.

*Expected Consumed Energy:* At each network round, the expected consumed energy $\hat{E}$ is computed for each sensor, depending on whether it is functioning as a regular sensor ($r$) or with the additional role of being a Collection Aggregator ($a$). Assuming equal $b$ bits of data are generated by each sensor, at each round, we define:

$$\hat{E}_r = E_{\mathrm{T}}\left(b, d_{[r,a]}\right) \qquad (3a)$$

$$\hat{E}_a \triangleq n_r E_{\mathrm{R}}(b) + (n_r + 1)\left[E_{\mathrm{P}}(b) + E_{\mathrm{T}}\left(b, d_{[a,g]}\right)\right] + E_{\mathrm{T_O}}, \qquad (3b)$$

where $d$ represents the Euclidean distance between nodes or between a CA and the gateway router $g$; for each CA, $n_r$ is the number of regular sensor nodes $r$ aggregated by it. For a regular sensor $r$, as shown in Eq. (3a), the expected consumed energy is just the transmission cost of its data payload to the nearest CA; for a CA, as defined in Eq. (3b), the expected energy cost depends on the number of regular child sensors $n_r$ that it has to receive data from, the cost of processing that data plus its own sensor data, the cost of transmitting all data to the gateway router $g$, and the cost of registering/unregistering observations of regular nodes ($E_{\mathrm{T_O}}$), as defined below.

*Energy Impact of Collections:* The nodes selected as CAs dynamically create or destroy collections at each round, as needed. Therefore, the following transmission energy costs ($E_{\mathrm{T_O}}$) must be considered when a CA changes its state (registers or unregisters itself) as an Observer of regular nodes:

- For registering, we use a default value of 15 bytes (120 bits). This is due to the 4 byte CoAP header, 8 bytes of a possible Token (the maximum size according to RFC 7252), and 3 bytes for the Observe option (the maximum size according to RFC 7641).
- For unregistering, CAs can issue observation deletes when the Observe option is removed from messages. So according to RFC 7252, 4 fixed bytes plus a maximum Token of 8 bytes, for a total of 12 bytes (96 bits).

### B. Genetic Model

At each transmission round, data collected by all sensors is transmitted towards the gateway router, along with some sensor monitoring information (i. e., the remaining battery energy). This information is then used by the gateway to search for the optimal network topology configuration for the next round, using a genetic algorithm.

[2] For the current simulation, the model is simplified by using an equal amount of data bits (packet sizes) for all terms.

*Genetic encoding:* The mode of operation for each sensor node is represented as an element (gene) of a binary chromosome vector: a *zero* specifies that a sensor will operate in regular mode (i. e., being observed), where a *one* specifies that the sensor will also operate as a CA, receiving data payloads (from observed members), aggregating those with its own data in a Collection, and transmitting it to the gateway. As sensors become inactive (e. g., due to an empty battery), their genes are set to *minus one*, exempting them from future optimisations.

*Fitness function:* The genetic algorithm tries to minimise the energy expenditure of the network, by iteratively maximising a fitness function $f$ at each network round $t$, where only alive sensors are considered. Assuming that $S$ is used to denote the alive sensor set (of size $n$), $f_t$ can be defined as:

$$f_t \triangleq \bar{\mathrm{m}}\left[\frac{E_{S_1}(t)}{E_{S_1}(0)}, \ldots, \frac{E_{S_n}(t)}{E_{S_n}(0)}\right] + \frac{\tilde{E}_S}{\hat{E}_S} + \frac{1}{\hat{D}_a} + \frac{1}{\|S\|}\sum_a G_a(\delta). \quad (4)$$

The function $\bar{\mathrm{m}}[\cdot]$ is the arithmetic mean of the terms, $\tilde{E}_S$ is the total energy cost if all data was to be directly transmitted to the gateway router, and $\hat{E}_S$ is the total expected consumed energy for the network configuration specified in the chromosome being tested, by summing all Eq. (3) results; $\hat{D}_a$ is the total Euclidean distance between all CAs and the gateway router; $G_a(\delta)$ is the *local node density* of each CA, which counts the number of regular sensors in the Euclidean $\delta$-vicinity of each CA, i. e., $D(a,r) \leq \delta$ (a value of $\delta = 20$ meters was chosen), normalised by the maximum number of possible sensors (the function $\|\cdot\|$ gives the set size). A relatively similar fitness function was used in [4].

*Genetic evolution:* The algorithm follows these steps before each network transmission round:

1) The optimisation starts with a random initial population of 30 chromosomes (network configurations), each with 50 genes (one for each sensor). A gene can be either *zero*, representing a regular node or *one* for a CA. If available, the best chromosome (i. e., the elite individual) from the last round is used and 29 new ones are generated randomly.
2) The fitness function $f_t$ is evaluated for each chromosome, using the data from the last round of the WSN and the proposed sensor configuration specified by each chromosome.
3) The most fit chromosomes are selected for crossover and mutation (the fractions are 0.8 and 0.01 respectively). The previous step is repeated for this new generation of chromosomes, and for every following generation.
4) The genetic algorithm stops evolving when either:
   (a) the fitness function converges;
   (b) a maximum number of generations is reached (30);
   (c) or after a maximum number of stall generations with no fitness improvement (10).
5) At the end of the optimisation round, the elite individual (i. e., the chromosome yielding the highest fitness value) represents the most energy-efficient network configuration, which is broadcast to all nodes, setting up the network for the next round of data gathering.

## V. Evaluation

The chosen simulation parameters are shown in Table II. At the start of each experiment, the individual sensors are randomly attributed different characteristics, such as a position $(x, y)$, initial energy (either 0.5J or 1.0J), data processing efficiency (either 100% or 200%), and capability of being used as a CA. Sensors with a high data processing efficiency use 50% less energy than a normal sensor, for doing the same work.

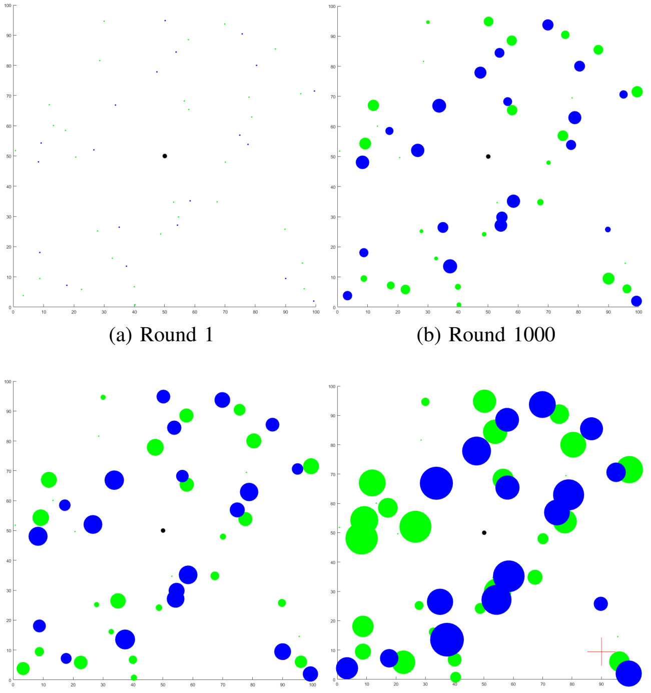
TABLE II
Network simulation parameters

| Parameters | Values |
|---|---|
| Simulation area (square shaped) | $100\,\mathrm{m}^2$ |
| Number of gateway routers | 1 (at centre) |
| Number of sensor nodes | 50 |
| Sensor node payload ($b$) | 400 bits |
| Percentage of sensors with higher initial energy | 10% |
| Percentage of higher *data processing efficiency* sensors | 10% |
| Percentage of sensors able to serve as CAs | 90% |
| Delta-vicinity distance for sensor groups ($\delta$) | 20 m |

TABLE III
First and last node to die by network rounds

| Avg. | DDEEC | ETLE | HEER | TSEP | ERP | GAHN | **GACN** |
|---|---|---|---|---|---|---|---|
| FND | 1100 | 1514 | 1789 | 1986 | 2010 | 2690 | **5678** |
| LND | 8900 | 6904 | 6150 | 7640 | 9200 | 10,400 | **24,999** |

Note: Values from previously proposed algorithms were obtained from [4].



(a) Round 1

(b) Round 1000

(c) Round 2000

(d) Round 6000

Fig. 1. Sensor roles by rounds. Each filled circle represent a sensor node, with a radius proportional to the number of times it was selected as a CA. The gateway router is represented in black (at the middle). For each round, green circles represent normal sensors and blue circles represent CAs (20, 21, 20 and 17 respectively); red crosses represent dead sensors.

We have simulated a WSN with 50 randomly distributed sensors inside a 100 by 100 meters area, with one gateway router at the centre. Network transmissions were simulated by rounds, where each sensor transmitted a data packet of 400 bytes per round, towards the gateway.

Table III compares our method, named *Genetic Algorithm for Constrained Networks* (GACN), on the average number of rounds until the *First Node Death* (FND) and the *Last Node*

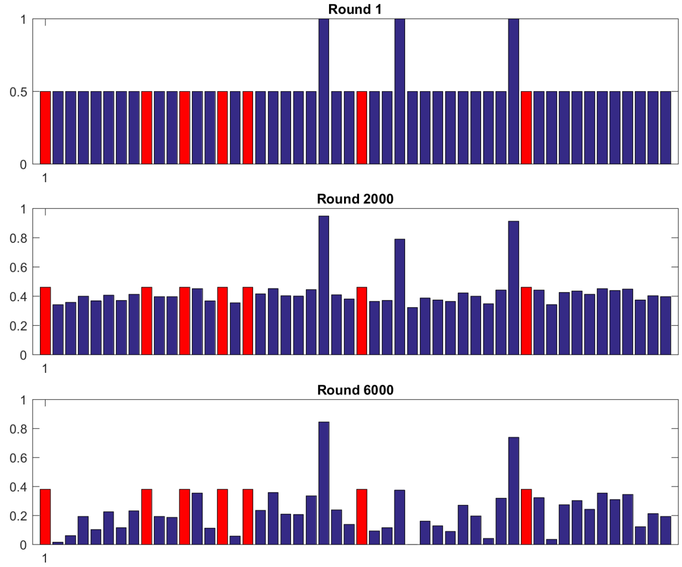


Fig. 2. Remaining energy *(y–axis)*, in Joule, at each node *(x–axis)*, for the 1st (top), 2000th (middle) and 6000th (bottom) rounds. Red nodes are unable to serve as CAs.

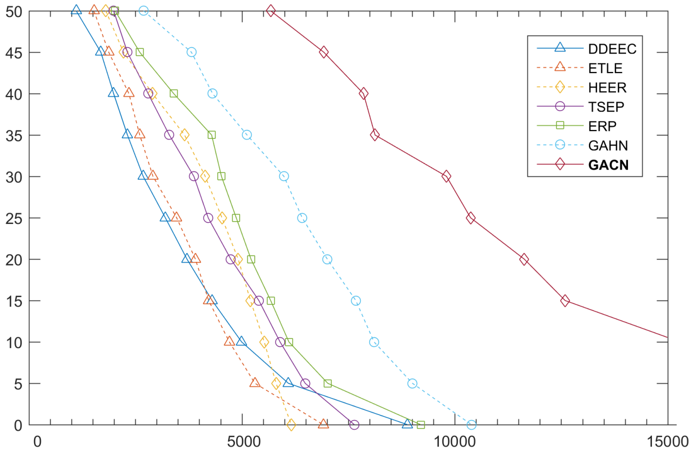


Fig. 3. Number of alive nodes *(y–axis)* by network rounds *(x–axis)*, for different methods. Values from previously proposed algorithms were obtained from [4].

*Death* (LND), with six state-of-the-art methods, which include HEER [6], TSEP [7], DDEEC [3], ETLE [15], ERP [1] and GAHN [4] (averages were done over ten different simulations for each method and were reported in [4]). In comparison, our method has the best results, with FND only happening at the 5678th round and LND at the 24,999th round. These are very good results, even accounting for the fact that we use an approach based on CoRE Interfaces and Collection resources, which require CA nodes to register and unregister observations when their roles change.

Figure 1 shows the spatial position and the CA selection frequency of the simulated nodes, Figure 2 shows the battery levels of nodes at specific rounds and Figure 3 represents the number of alive nodes throughout the network life. In comparison with a state-of-the-art GA-based solution [4], Figure 1 shows a much more uniform distribution of the CA role through time, by various nodes, which is a key factor for increasing node lifetime. This is mainly attributed to the efficiency of the fitness function chosen.

The algorithm was implemented in Matlab, taking around 30s to compute each network round (about one second per GA generation). We expect this time to be significantly improved upon further optimisation, to around 0.5-0.6s for a complete round, similarly to [4]. The algorithm almost always converges before or at the 30 generation limit.

## VI. Conclusions and Future Work

Network clustering is an efficient way to increase the longevity of WSNs, especially if clusters are determined dynamically and adapt to the network state. However, it is not easy to find the optimal cluster configuration, as it involves defining the best predictors of higher resource efficiency. Still, most of the literature does not mind about protocols choices, a relevant issue in real-world WSNs. Here we focus on standard IETF CoRE protocols: CoAP and Observe. CoRE Interfaces are used to dynamically build/destroy Collections with the purpose of maximising network lifetime. Our model, GACN, accounts for energy costs associated with this dynamic. Our model showed very good results, surpassing other state-of-the-art methods. As future work, one direction worth exploring is the usage of GACN with mobile sensing nodes.

## Acknowledgements

This work was developed within the CEOT (Center for Electronic, Optoelectronic and Telecommunications), and supported by the UID/MULTI/00631/2013 project of the Portuguese FCT (Foundation for Science and Technology).

## References

[1] Attea, B. A. and Khalil, E. A. (2012). A New Evolutionary Based Routing Protocol for Clustered Heterogeneous Wireless Sensor Networks. *Appl. Soft Comput.*, 12(7):1950–1957.

[2] Correia, N., Sacramento, D., and Schutz, G. (2016). Dynamic Aggregation and Scheduling in CoAP/Observe Based Wireless Sensor Networks. *IEEE Internet Things J.*, PP(99).

[3] Elbhiri, B., Rachid, S., El Fkihi, S., and Aboutajdine, D. (2010). Developed Distributed Energy-Efficient Clustering (DDEEC) for heterogeneous wireless sensor networks. In *2010 5th Int. Symp. I/V Commun. Mob. Networks, ISIVC 2010*, pages 1–4. IEEE.

[4] Elhoseny, M., Yuan, X., Yu, Z., Mao, C., El-Minir, H. K., and Riad, A. M. (2015). Balancing Energy Consumption in Heterogeneous Wireless Sensor Networks Using Genetic Algorithm. *IEEE Com. Lett.*, 19(12):2194–2197.

[5] Hartke, K. (2015). Observing Resources in the Constrained Application Protocol (CoAP). *IETF*, RFC 7641.

[6] Javaid, N., Mohammad, S. N., Latif, K., Qasim, U., Khan, Z. A., and Khan, M. A. (2013). HEER: Hybrid Energy Efficient Reactive Protocol for Wireless Sensor Networks. In *Electron. Commun. Photonics Conf. (SIECPC), 2013 Saudi Int.*, pages 1–4. IEEE.

[7] Kashaf, A., Javaid, N., Khan, Z. a., and Khan, I. a. (2012). TSEP: Threshold-Sensitive Stable Election Protocol for WSNs. *Front. Inf. Technol. (FIT), 2012 10th Int. Conf.*, pages 164–168.

[8] Kumar, D., Aseri, T. C., and Patel, R. B. (2009). EEHC: Energy Efficient Heterogeneous Clustered Scheme for Wireless Sensor Networks. *Comput. Commun.*, 32(4):662–667.

[9] Kushalnagar, N., Montenegro, G., and Schumacher, C. (2007). IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. *IETF*, RFC 4919.

[10] Shelby, Z. (2012). Constrained RESTful Environments (CoRE) Link Format. *IETF*, RFC 6690.

[11] Shelby, Z., Hartke, K., and Bormann, C. (2014). The Constrained Application Protocol (CoAP). *IETF*, RFC 7252.

[12] Shelby, Z., Koster, M., Bormann, C., and der Stok, P. V. (2016). CoRE Resource Directory. *IETF*, CoRE Draft(07).

[13] Shelby, Z., Vial, M., and Koster, M. (2015). Reusable Interface Definitions for Constrained RESTful Environments. *IETF*, CoRE Draft(04).

[14] Smaragdakis, G., Matta, I., and Bestavros, A. (2004). SEP: A Stable Election Protocol for Clustered Heterogeneous Wireless Sensor Networks. *Int. Work. Sens. Actor Netw. Protoc. Appl. (SANPA 2004)*, pages 1–11.

[15] Tuah, N., Ismail, M., and Jumari, K. (2011). Energy Efficient Algorithm for Heterogeneous Wireless Sensor Network. In *Proc. Int. Conf. Control Syst. Comput. Eng.*, pages 92–96. IEEE.