

Detection and Cooperative Communications for Deployment Sensor Networks



Bedr-Eddine Benaissa^{1,2}, Fedoua Lahfa¹, Khatir Naima², Giulio Lorenzini³, Mustafa Inc^{4,5,6*}, Younes Menni⁷

¹ Department of Computer sciences, Abou Bekr Belkaid University, Tlemcen 13000, Algeria

² Department of Technology, University Centre of Naama - Salhi Ahmed, P.O. Box 66, Naama 45000, Algeria

³ Department of Engineering and Architecture, University of Parma, Parco Area delle Scienze, 181/A, Parma 43124, Italy

⁴ Department of Computer Engineering, Biruni University, Istanbul 34025, Turkey

⁵ Department of Mathematics, Science Faculty, Firat University, Elazig 23119, Turkey

⁶ Department of Medical Research, China Medical University Hospital, China Medical University, Taichung 40402, Taiwan

⁷ Unit of Research on Materials and Renewable Energies, Faculty of Sciences, Department of Physics, Abou Bakr Belkaid University, P.O. Box 119, Tlemcen 13000, Algeria

Corresponding Author Email: minc@firat.edu.tr

<https://doi.org/10.18280/ts.380303>

ABSTRACT

Received: 15 April 2021

Accepted: 3 June 2021

Keywords:

Wireless Sensor Network (WSN), clustering, Received Signal Strength Indicator (RSSI), IoT routing protocol

In a sensor network, and more specifically with a single-hop deployment policy, sensor measurements contain a lot of redundancy either in the measurement dimensions of a single sensor, or between the measurement dimensions of different sensors due to of the spatial correlation either in the temporal dimension of the measurements. The goal is to reduce this redundancy by deploying fewer sensors, while ensuring high measurement accuracy and maximizing service life. The proposed method minimizes the complexity in terms of communication and calculation and maximizes the lifetime of the network based on an aggregation and consensus system to reduce the spatio-temporal dimension of the data captured and consequently the number of sensors deployed. The results show a visible performance compared to the standard method of transmission on the free platform of the COOJA/Contiki simulator allowing to simulate network connections of wireless sensors and to interact with them.

1. INTRODUCTION

Searching in the field of wireless communication has given rise to miniaturized and intelligent equipment with capture, calculation and wireless communication units in a single circuit, with reduced dimension, and with a reasonable cost. These equipments, commonly known as micro-sensors, have favored the idea of developing sensor networks based on the collaborative effort of a large number of nodes operating autonomously and communicating with each other via short-range transmissions in the purpose of reporting an event according to the field of application (Figure 1).



Figure 1. TMote Sky sensor node platform [1]

A sensor is able to reap data relating to its environment, process them, and if necessary, communicate them to neighboring sensors via a wireless medium. The deployment

of this type of device then forms a network of wireless sensors that can be used in a variety of areas: scientists, military, industrial, health, environment, etc. This type of network consists of a few hundred or even thousands of autonomous sensors, of extremely reduced size and that work with very limited resources in terms of memory, energy and treatment capacities and communicate wirelessly with others in their communication zones.

The design and implementation of WNS are influenced by a several constraints among them we quote [2]:

1. Environment,
2. Deployment of nodes,
3. Heterogeneity of nodes,
4. Failure tolerance,
5. Mobility,
6. Scaling,
7. Material constraints,
8. Transmission,
9. Energy consumption, and,
10. Quality of service.

In a WSN, total coverage of the area to be monitored is one of the most asked questions in research. Thus, due to the strong spatial correlation of the detection area, there is an intensive deployment of sensor nodes which often detect and relay similar or redundant information and leading to more complex data analysis and a significant loss of resources for capturing, processing and memorizing them and consequently causing

the nodes to over-consume their transmission energies. Among the solutions to this problem, the design of a hierarchical topology was implemented by dividing the nodes into several levels of responsibility. One of the most widely used methods is clustering, where the network is partitioned into groups called "clusters". A cluster is made up of a cluster-head and its members.

In this perspective, our contribution lays out an algorithm based on an aggregation and consensus system to reduce the dimension of the values captured. This new type of mean consensus leads clustering-based routing to analyse and classify detection data and extract data characteristics to infer similar data, then decrease storage space, transmission load, and power consumption energy and simplify the ongoing data mining process.

2. RELATED WORK

Depending on the number of nodes of the network and the surface of the cover field, some nodes cannot directly transmit their messages to the collector node. Thus, the collaboration between the nodes to guarantee this transmission becomes a requirement. In this way, the messages are relayed by the intermediate nodes by establishing multi-jumping paths between a remote source and the well. This process of routing messages from a source node of the network to a recipient node is called routing. Thus, for a sensor network to be effective, the setting up of a routing algorithm becomes inevitable [3].

2.1 Routing concepts

There are two classes of routing protocols: distance vector routing and link state routing, so a routing protocol can be proactive or reactive [4].

2.1.1 Distance vector routing

This protocol class is based on an exchange between neighboring nodes and the node sending information about the list of destinations available to it and the corresponding cost. Routing protocols based on the most famous distance vector for ad hoc networks are: DSR, DSDV6, AODV and the new RPL protocol.

2.1.2 Link state routing

This protocol is known under the name of 'Shortest Path First'. This routing is based on the information collected and disseminated periodically on the state of the links in the network to update its description base describing a general view of the topology of the network. The main routing protocols in ad hoc networks that belong to this class are as follows: TORA, OLSR, etc.).

2.1.3 Proactive / reactive routing protocol

The proactive protocol builds its routing table before the request is made. A system of periodic exchange of control packets is put in place so that each node can build the network topology in a distributed fashion. The main proactive protocols are OLSR, FSR. On the other hand, a node using the reactive protocol completely ignores the topology of its network it updates its routing table only after having received a routing request from a neighboring node.

2.2 Detection in wireless sensor network

The trivial meaning of the term 'detection' is the process of capturing physical quantities related to environmental conditions (movement, temperature, humidity, light, etc.). The processing is based on the compatibility of the data captured by deploying different algorithms, either by comparing the previous state with the current state or using a consensus in order to eliminate the outliers and transmit a single common value to the sink.

2.2.1 Detection methods

1. Detection aggregation: based on statistical tests such as Maximum, Average [5] or a discretized classification of data [6] to aggregate the values captured over time;

2. Spatial and temporal correlation: Periklis et al. [7] list all the sensors in good condition and use them as references to test the other sensors. The simulation results show that this method consumes a lot of landfill and congestion. The use of cluster architecture is adopted to reduce this overhead [8];

3. Average consensus: The algorithms based on this principle conform to calculate the average between the recorded measurements and distribute it between neighboring sensors [9]; and

4. Consensus of beliefs: this consensus [10] is an algorithm derived from Bayes rule where a set of sensors try to harmonize on the most probable classification of an event.

2.2.2 Detection methods comparison

In Table 1, we present a classification of the protocols described in the previous section 2.1.

Table 1. Advantage and disadvantage of detection methods

Method	Advantage	Disadvantage
Detection aggregation	Reliable in the presence of faulty sensor readings.	Consumes any energy
Spatial and temporal correlation	Identifies sensors in good condition. For a tree architecture	Worn sensors. Communication and energy.
Consensus of the mean	No protocol. Low complexity algorithms.	Faulty sensors problem.
Consensus of beliefs	Classification model.	Complexities of calculations
Pair gossiping	Simple model. Scalability.	Converges slowly More iterations to converge.
Broadcast gossiping	Converges quickly. Scalability.	Consumes communications.
Gossip by triplet	Simplicity of calculations.	Controlled broadcast at 2 nodes only.
Gossip between neighbourhood	High performance Scalability.	Very high cost communication.

2.3 Simulator Cooja

Contiki OS Java Simulator is a sensor network emulator that can house a compiled program in order to test a scenario before loading it into flash memory of real nodes as in the TI MSP430 platform (Figure 2).

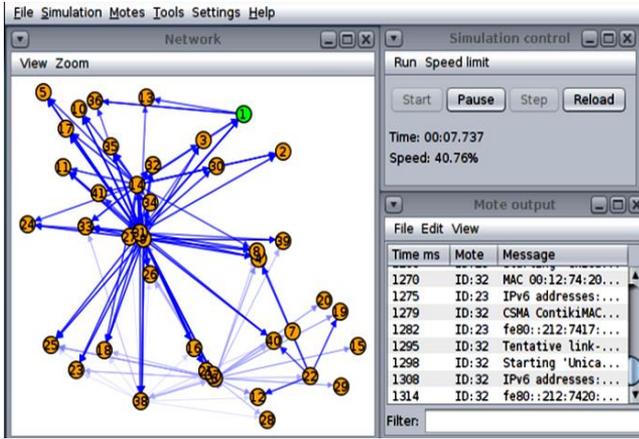


Figure 2. Interface graphique Cooja

2.4 Routing protocol - RPL

The basic principles of the protocol are [11, 12]:

1. Routing: the path discovery is based on the quality of the signal, the density of nodes and active path is represented by its costs;
2. Route selection: a reliable route relies on signal strength (RSSI);
3. Road quality criteria: signal strength (RSSI), number of hops, available energy and error rate (outliers);
4. Changing the path: the path initially traced can be corrected later depending on the strength of the signal and the total number of nodes visited in the event of failure of the main road; and
5. Constraint-based routing: The life of a node depends on the battery and diversity of node types and their characteristics (memory, location, collector, etc.).

2.5 Signal Strength (RSSI)

Received Signal Strength Indication informs us by a measurement on the intensity of the received signal Rx. It is an essential parameter for radiolocation applications, due to its simplicity and low cost, but its accuracy can be degraded by noise caused by stable, moving objects and obstacles or by diffraction or refraction of radio waves [13].

$$RSSI = -(10 * n * \log d + A) \quad (1)$$

n : Constant - signal propagation.

d : distance between the transmitter Tx and the receiver Rx.

A : Signal strength received for 1 meter distance.

So the distance is calculated from the RSSI:

$$d = 10^{((Abs(RSSI) - A) / 10 * n)} \quad (2)$$

According to Eq. (1), if A increases then RSSI increases. Thus the relation between the power of the transmitted signal Tx and the distance of the signal traveled " d " is exponential; but in reality this relationship is only as implicit in theory.

In this article we focus on a well-chosen configuration to collect values by exploiting the received signal power (RSSI).

3. PROPOSED WORK

Noting that a network of sensors is effective if is only if the

communication links are reliable. Alrawashdeh [13], underlines and experiments this theory on the implantable bodily applications in the bone, by applying a standard binding with a phase shift modulation (PSK).

In this paper, a detection method is proposed in order to capture physical quantities and transmit them to the base station (Sink) in a very short time, based on the processing time that each node can perform according to a scenario. Our system is based on the following three ordered tasks:

1. Detect physical quantities which are whole values of temperature and humidity taken at random and which change every 30 seconds.
2. Detect the sensor nodes neighboring the sensor node which wants to transmit these quantities.
3. Use the DODAG routing topology built through the RPL protocol.

DODAG (Destination Oriented Directed Acyclic Graph): A directed acyclic graph directed to a destination which is the root of the network (Sink).

It should be noted that throughout our simulation we will use the RSSI calculation formula given in Eq. (3) in order to ensure that the signal is sufficient to guarantee a reliable wireless connection. Said formula is tested outdoors giving acceptable results [14].

$$d = (-0.5894 * RSSI) - (5.6768) \quad (3)$$

3.1 Triple point method

This approach is based on a hybrid method, which combines the mean consensus algorithm and the triplet gossip algorithm in order to agree on the detected value (mean between pairs whose values are closer) to relay to the well (Figure 3).

At a time " t " of the clock, a cluster leader node N_i (CH), is randomly selected among all nodes (CHs). N_i selects two neighboring nodes N_j , N_k among all the nodes of its cluster respecting the condition: the elected nodes must have the closest values to the value of N_i respecting a proximity threshold quantified at the value S verifying for example the condition ($|VN_i - VN_j| < S$) between node N_i and node N_j .

An exception to this rule is made when one of the neighboring nodes is out of range, the triple gossip is then reduced to one chat per pair in this time slot.

As for the transmission cost, the notification from node N_i to nodes N_j and N_k is evaluated at one transmission, thanks to the wireless broadcast function.

Nodes N_j and N_k transmit their own local values $VN_j(t)$ and $VN_k(t)$ respectively to node N_i (no collision, node N_i can bring a consensus to its first notification). This step costs 2 transmissions.

Node N_i averages the values of the three locally selected nodes and updates its local estimate according to the Election1 algorithm:

Algorithm Election1

```
//A sensor is randomly activated N1 and chooses two closest
neighbors N2 and N3 among the others//
{v1, v2, v2 values detected by N1, N2 and N3 respectively}
min = abs(v1-v2)
If min > abs(v1-v3) then min = abs(v1-v3)
If min > abs(v2-v3) then min = abs(v2-v3)
If min = abs(v1-v2) then ElectedValue=(v1+v2)/2
If min = abs(v1-v3) then ElectedValue=(v1+v3)/2
```

```

If min = abs(v2-v3) then ElectedValue =(v2+v3)/2
Send (ElectedValue)
End

```

The triple point method summarizes the average value of three nodes per 3 radio transmissions in a gossip context.

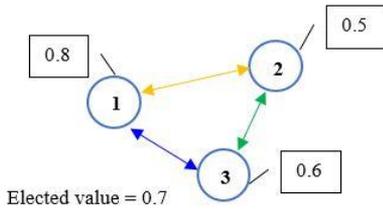


Figure 3. Example of the value selection to transmit

Note:

An execution of the algorithm consumes a maximum of: 6 subtraction operations and 6 comparison tests, 1 addition, 1 division, 2 assignments (benefit from negligible energy consumption).

3.2 Standard detection method

Standard transmission is based on the principle that the nodes can transmit their physical quantities captured to the neighbor node or to the sink node without going through a selection or gossip or consensus algorithm, but it uses the DODAG (Destination Oriented Directed Acyclic Graph) routing topology built by the RPL protocol and which is composed of a set of routes or paths connecting the sensor nodes of the network to the root sensor node (Sink).

The RPL (Routing Protocol for Low-power and Lossy Networks) is a proactive distance vector protocol: When a preferred parent has been selected, all traffic will be transferred via this preferred parent, as long as it is accessible, without any attempt at load balancing between the other available parent candidates. This benefit can drain the power of overloaded parents, prompting network failure and unreliability issues, as overloaded nodes are likely to die soon enough. Many articles explain how the RPL protocol works, for example the authors, Sobral et al. [15] which present in more detail the RPL protocol in Internet of Things applications.

4. SIMULATION AND RESULT ANALYSIS

The proposed methods were tested by a simulation on the CONTIKI / COOJA platform. For our simulation the mote selected is TmoteSky is a wireless sensor board from Moteiv [16]. This is an MSP430 based board with an 802.15.4 compatible CC2420 radio chip, 1megabyte external serial flash memory and two light sensors. TmoteSky port was integrated into the Contiki building system in March 2007. The functionality of the MSP430 F1611 module is detailed in the Texas Instruments MSP430x1xx Series User Guide available in Ref. [17].

We place the sink node (called mote1) at a fixed initial position. The three scenarios are tested respectively by applying the three methods by deploying several anchor, aggregator or simple nodes according to the chosen algorithm. Two parameters are recorded, the first one is the convergence time of the value captured at the sink and the second the

quantity of energy consumed.

The estimation of the energy consumption of the motes is based on the 'Energest' module of Contiki by evaluating certain states like CPU, LPM, Tx and Rx, in order to calculate the elapsed time at each sensor node in real time. This energy estimation 'Energest' module is called when the component is activated to produce a timestamp. The energy consumption of the nodes is calculated with the following equation [18]:

$$E_c = (E_v \times E_p \times V_t) / R_{t_second} \times R_t \quad (4)$$

E_c ≡ Energy consumption (milli-watt),

E_v ≡ Energy value,

E_p ≡ Electric power,

V_t ≡ Voltage,

R_t ≡ $R_{timer} = 10$ second, and

R_{t_second} ≡ Low frequency crystal frequency = 32768 Hz.

We underline that the value of $E_{nergest_Value}$ is the remainder of the difference between the number of ticks (ALL_CPU) at the current time 't2' and the previous time 't1'.

A single system tick is approximately 1 millisecond.

$$eff_c (\%) = \frac{\Delta T_x + \Delta R_x}{CPU_e + LPM_e} \quad (5)$$

eff_c ≡ Effective cycle (%),

ΔT_x ≡ Transmission energy,

ΔR_x ≡ Reception energy,

CPU_e ≡ CPU energy,

LPM_e ≡ Low Power Mode energy, and

LPM ≡ Low Power Mode, device inactive waiting for events.

4.1 Simulation with a standard detection method

The topology taken for this first method comprises 20 anchor mote and 01 Sink (mote1). These 20 motes can sense and transmit temperatures, humidity and light levels to the sink without any calculations and without requiring conditions.

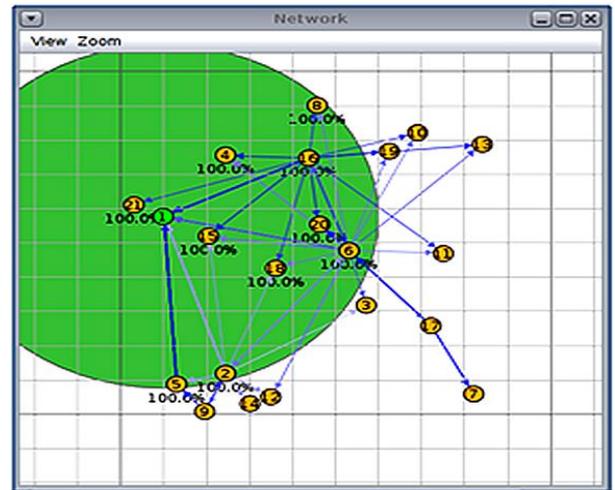


Figure 4. Scenario of a topology in standard detection

The effective cycle is the energy consumption expressed in milli-joule of a mote close to Sink (here mote 15) and another which is far from Sink but is in its coverage zone (mote 6) and a third mote is outside the Sink's coverage area (mote 7) after every 10 seconds and in a time interval between 0 and 200 seconds (Figure 4).

By setting the data display to 10 seconds and using the POWERTRACE tool from Contiki knowing that it is accurate to 94% compared to the real value.

1. Mote 15: The analysis made on the nearest mote gave us the following Table 2:

Table 2. Number of ticks for anchor mote 15

Time	ALL CPU	ALL LPM	ALL TX	ALL RX
0	14902	312578	3070	6686
10	34244	620853	6061	11922
20	52817	929943	6836	16040
30	64626	1245800	9056	18395
40	76011	1562087	9795	20365
50	94331	1871430	12145	24386
60	106827	2186599	14337	27172
70	118474	2502616	16545	29142
80	130156	2818601	18795	31168
90	148018	3128405	20045	33581
100	161477	3442793	22595	37137
110	173940	3757821	25047	39887
120	185565	4073865	27735	41857
130	197266	4389828	30051	43827
140	208931	4705830	32655	45797
150	220532	5021897	35140	47767
160	232200	5337896	37625	49737
170	244925	5652837	41045	51821
180	256569	5968862	43210	53791
190	268565	6284531	45140	56050
200	280938	6599823	47354	58723

By applying the Eq. (5), we obtain the following Table 3 of the radio effective cycle:

Table 3. Percent of effective cycle of node 15 radio

Time	Tx(%)	Rx(%)	Total(%)
10	0.91	0.02	0.93
20	0.24	0.01	0.25
30	0.68	0.01	0.68
40	0.23	0.01	0.23
50	0.72	0.01	0.73
60	0.67	0.01	0.68
70	0.67	0.01	0.68
80	0.69	0.01	0.69
90	0.38	0.01	0.39
100	0.78	0.01	0.79
110	0.75	0.01	0.76
120	0.82	0.01	0.83
130	0.71	0.01	0.71
140	0.79	0.01	0.80
150	0.76	0.01	0.76
160	0.76	0.01	0.76
170	1.04	0.01	1.05
180	0.66	0.01	0.67
190	0.59	0.01	0.60
200	0.68	0.01	0.68

i.e. $Tx- Eff_c = (6061-3070) / ((34244-14902) + (620853-312578)) = 0.009130 = 0.91\%$ (see Table 2 and Table 3).

After activating Power trace on Cooja and connecting its library and configuring its clock to 10 seconds in our source program, we obtain a set of values printed in the following table representing an estimate of the amount of energy consumed:

Table 4. Power consumption on TmoteSky 15

Time	CPU	LPM	Tx	Rx
10	0.088541	0.007338	0.533977	1.045027
20	0.085020	0.007357	0.138359	0.821891
30	0.054057	0.007519	0.396332	0.470023
40	0.052116	0.007529	0.131932	0.393182
50	0.083862	0.007364	0.419540	0.802531
60	0.057202	0.007502	0.391333	0.556044
70	0.053316	0.007522	0.394189	0.393182
80	0.053476	0.007522	0.401688	0.404359
90	0.081766	0.007374	0.223160	0.481599
100	0.061610	0.007484	0.455246	0.709724
110	0.057051	0.007499	0.437750	0.548859
120	0.053215	0.007523	0.479883	0.393182
130	0.053563	0.007521	0.413470	0.393182
140	0.053398	0.007522	0.464886	0.393182
150	0.053105	0.007524	0.443642	0.393182
160	0.053412	0.007522	0.443642	0.393182
170	0.058250	0.007497	0.610565	0.415935
180	0.053302	0.007523	0.386513	0.393182
190	0.054913	0.007514	0.344559	0.450862
200	0.056639	0.007505	0.395261	0.533491

Note: See datasheet for TmoteSky pp.9 [1] 1. Active courant at $V_{cc} = 500\mu A/s = 0.50mA/s$, Voltage = 3v and $1s = 32768$ clocks. 2. i.e. Energy consumed (CPU) = $(34244-14902) \times 0.50 \times 3 / (32768 \times 10) = 0,088541$ mw.

Table 4 displays a return of Tmote Sky values in four states in the form of a number of clock ticks.

- ALL_CPU: The total (high) CPU (CPU in active mode).
- ALL_LPM: The total number of ticks in an LPM state (low power mode).
- ALL_Tx: The total number of ticks in the Tx (Transmit) state.
- ALL_Rx: The number of ticks in the Rx (Receive) state.

We calculate the energy consumption using the formula of Eq. (5).

The standard model displays in Figure 5 the variation of the energy consumption in the four states mentioned previously. We can clearly see that the increase in energy concerns the transmission but above all the reception. Figure 6 confirms our assertion by summing the transmission and reception energies to represent the effective radio service cycle.

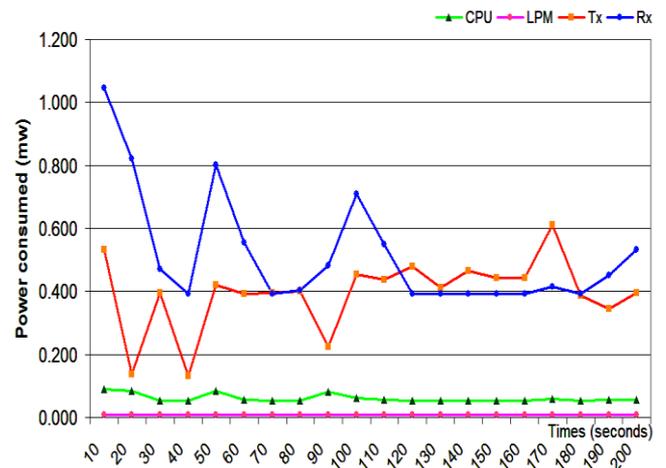


Figure 5. Skymote 15 power consumption in standard detection

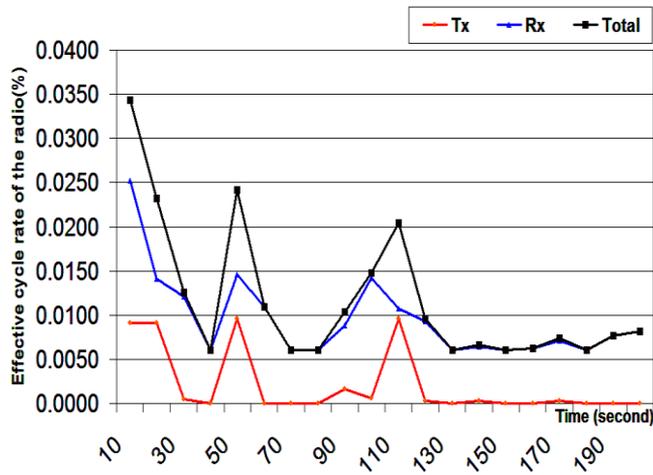


Figure 6. Effective cycle rate of the radio after every 10 seconds from mote 15 in standard detection

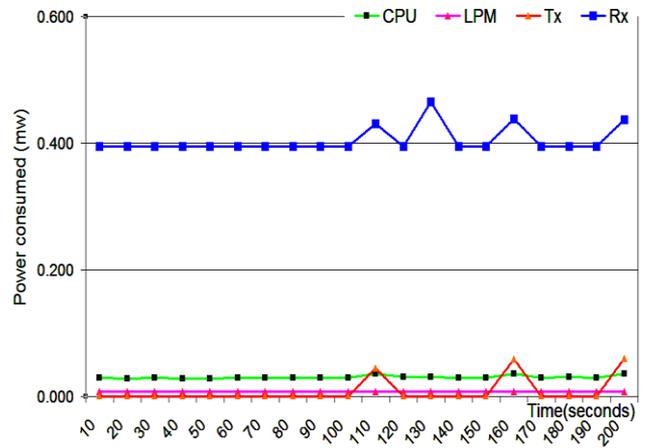


Figure 9. Skymote 7 power consumption in standard detection

2. Mote 6: The data of Mote 6 in number of captured ticks are displayed in the following graph:

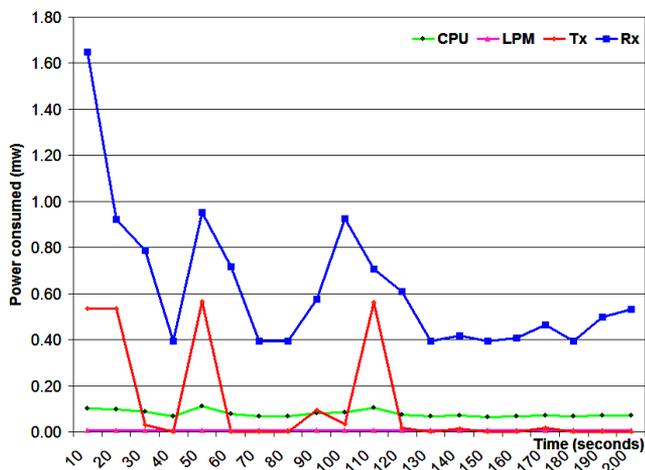


Figure 7. Skymote 6 power consumption in standard detection

As for Figure 5, Figure 7 represents the variation of the energy consumption in the four states (CPU, LPM, Tx, Rx) of the mote 6 in the standard model. Figure 8 shows the effective radio cycle in order to show the difference between emission and transmission.

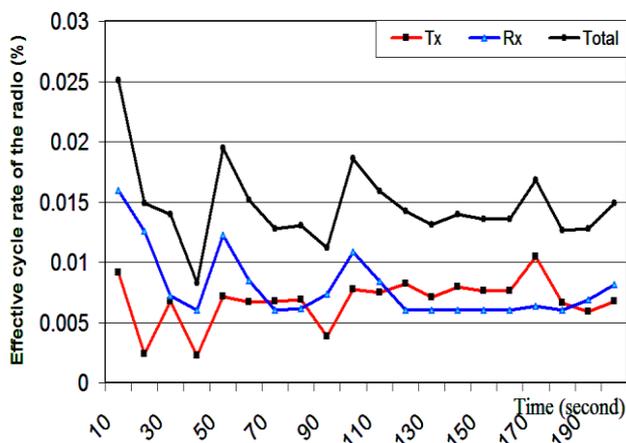


Figure 8. Effective cycle of the radio after every 10 seconds from mote 6 in standard detection

3. Mote 7: Now we are going to study the energetic behavior of mote 7.

Including its effective radio cycle in order to understand the influence of the distance of a mote from the sink in order to previously predict an optimal unfolding with collaboration the nodes between them. The data of word 7 in number of captured ticks are displayed in Figure 9. Just as Figures 5 and 7, Figure 9 shows the variation of the energy consumption in the four states (CPU, LPM, Tx, Rx) of the mote 7 in the standard model.

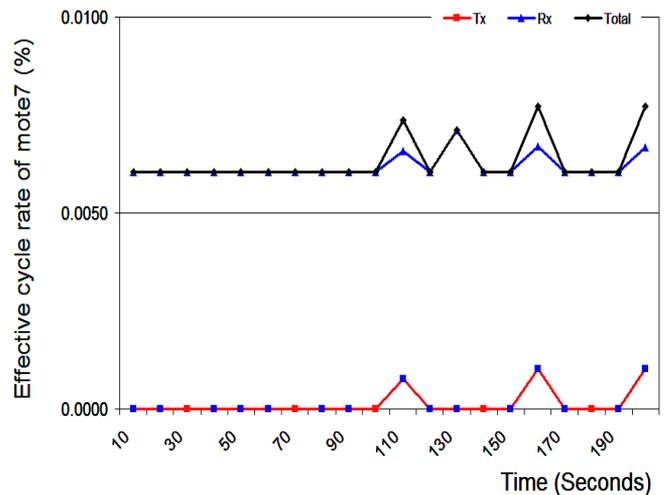


Figure 10. Effective cycle rate of the radio after every 10 seconds from mote7 in standard detection

By analyzing Figure 4 of the scenario of the topology of the standard detection model, we notice that the mote7 is outside the coverage of the sink on one side and that it has almost no neighbor. Figure 10 denotes a virtual absence of transmission but the mote6 is still listening.

4.2 Standard detection method discussion

Note that mote 15 and mote 6 are in the same coverage of the sink (mote1), on the other hand mote 7 is outside the coverage area of the sink.

It can be seen in Figure 11 that the energy consumption by the three motes 15, 6 and 7 is more marked on the motes in packet reception mode (Rx) (Figures 5, 7 and 9), this could be explained by the fact that the motes are always listening to the

transmission channel, and this is also proved by the results found by the effective cycle of each mote (high percentage of Rx compared to Tx).

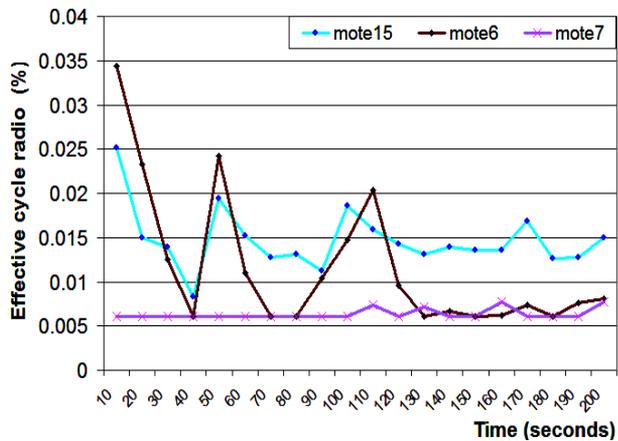


Figure 11. Effective cycle rate of the radio comparison for of the three motes, after every 10 seconds in standard detection

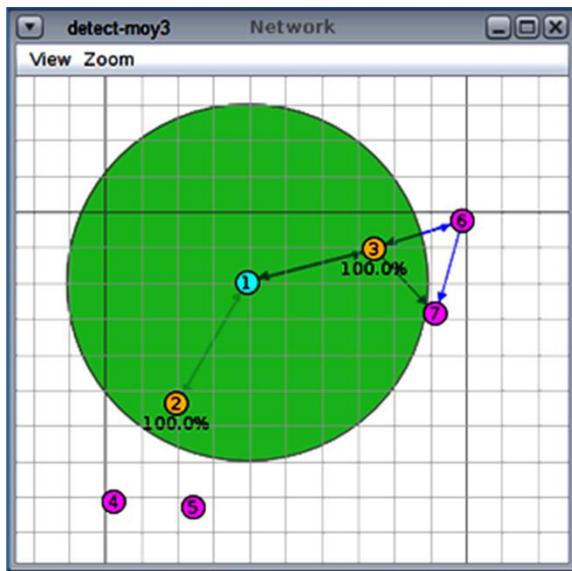


Figure 12. Scenario of a topology in the triple point method

The corresponding data on the eve of these three motes (mote 15, mote 6, mote 7) is represented by the parameter LPM which underlines an energy consumption. This means that the motes consume almost no power when they are in standby mode.

According to Figure 11, the use of the radio is more accentuated when the motes are on the cover of the sink, but it is much more when it approaches the latter. This could be explained by the fact that mote 15 is more likely to relay data to the sink than others. This deduction is also verified by analyzing the amount of energy consumed by CPU during processing. For example, mote 15 prints (Figure 5) an average usage between [0.3 mw - 0.5 mw].

An important remark is also verified from these results is the fact that when a mote (here mote6) has several neighbors, the energy consumed at reception increases considerably (Figure 6), this can be verified with mote 7 which has only one neighbor (mote17).

So, for an optimum deployment of the sensors, it is necessary to avoid the concentration of the nodes around a single mote because this could induce an excessive use of

energy and arrive at its loss and a re-structuring of the topology or to permanently lose the network of sensors wireless.

4.3 Simulation with the triple point method

The configuration chosen is given by the following Figure 12. The simulation of our method is based on 7 motes. One sink, 4 standard catch motes and two aggregator motes responsible for cooperation to send a single value, according to a consensus, to the Sink.

Standard motes 4, 5, 6, 7 detect and transmit the values of temperature, humidity and light in broadcast. Motes 4 and 5 aggregate their data to word 2 and words 6 and 7 to word 3. Motes 2 and 3 are the aggregators.

Our method is based on aggregators to optimize radio transmission and overcome the problems posed by the standard configuration.

Once the simulation is triggered, the POWERTRACE tool gives the following results in Table 5:

1. Mote 2: On this mote we note the number of ticks after every 10 seconds:

Table 5. Number of ticks for a mote 2

Time	ALL CPU	ALL LPM	ALL TX	ALL RX
0	15584	312155	5765	3422
10	31396	623959	9080	7586
20	44257	938711	12237	10370
30	52227	1258393	12317	13175
40	58490	1579789	12317	15155
50	72257	1893695	15468	18589
60	79606	2213977	15548	20708
70	85543	2535618	15548	22688
80	91483	2857264	15548	24668
90	97890	3178514	15548	26648
100	111115	3492951	18705	29548
110	118191	3813542	18705	32537
120	126278	4133033	18875	34795
130	132433	4454461	18875	36938
140	138398	4776082	18875	38918
150	144656	5097483	18875	40898
160	150987	5418815	18875	43092
170	157191	5740282	18875	45072
180	163593	6061546	18875	47290
190	169786	6383022	18875	49270
200	176559	6703892	18965	51391

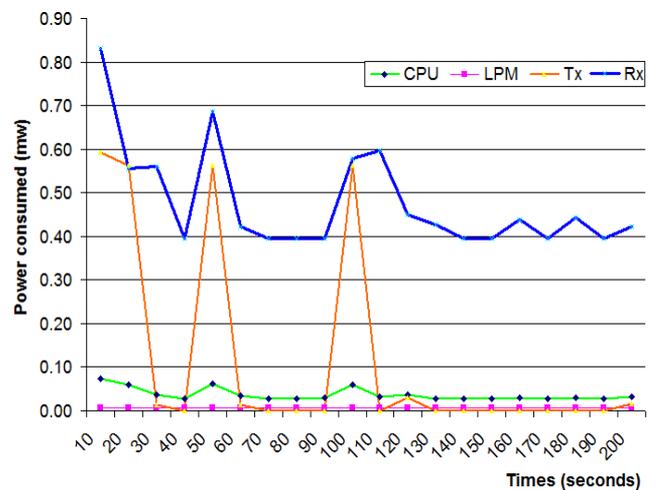


Figure 13. Skymote 2 power consumption in the triple point method

By respecting the same approach of the standard model but using our algorithm this time, we obtain Table 5 representing the number of ticks in the four states (CPU, LPM, Tx, Rx) for the mote 2. Next, we calculate the energy consumption using the formula of the Eq. (4).

Next, we calculate the energy consumption using the formula of the Eq. (5) and after a graphical representation we obtain Figure 13 which visualizes the energy consumption by our triple point method. This curve highlights an increase in energy consumption around the radio (Tx, Rx).

Likewise, Figure 14 underlines a use much of the transmission than that of reception.

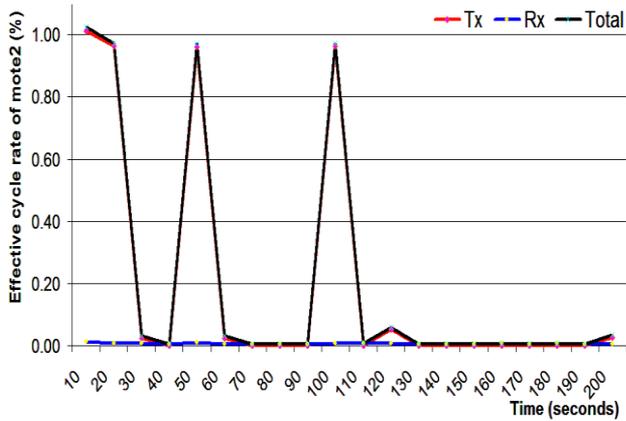


Figure 14. Percent of number of ticks for mote 2 in the triple point method

2. Mote 3: On this aggregator mote we note also the number of ticks after every 10 seconds:

Similarly to Mote 2 and by using the POWERTRACE tool, the data collected by the Mote 3 aggregator is shown in Figure 15 marking the energy consumed in transmission and reception as well as by the processor.

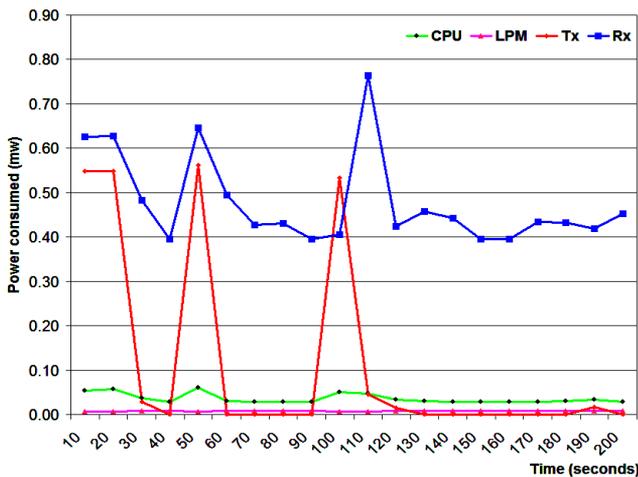


Figure 15. Skymote 3 power consumption in the triple point method

The energy consumption using the formula of the Eq. (5) and after a graphical representation we obtain Figure 16 which visualizes the energy consumption by our triple point method. This curve highlights an increase in energy consumption around the radio (Tx, Rx). Figure 16 displays the actual cycle of radio use and gives a graphical representation of high listening use versus transmission which appears occasional.

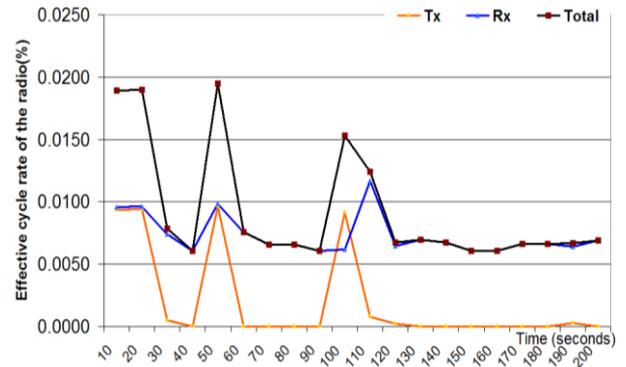


Figure 16. Percent of number of ticks for mote 3 in the triple point method

3. Mote 4: This is a simple sensor outside the sink coverage area and we want to follow its behavior and energy consumption throughout the process with the triple point method. Table 6 note also the number of ticks after every 10 seconds.

Figures 17 and 18 show the energy consumption of the main modules in Module 4 using the triple point method, respectively. The observation is highlighted on the receive (Rx) mode in front and the transmission thereafter.

Table 6. Number of ticks for a mote 4

Time	ALL CPU	ALL LPM	ALL TX	ALL RX
0	9403	318366	2992	3044
10	23207	632223	7426	6281
20	28738	954356	7426	8527
30	39955	1270802	10658	10943
40	45199	1593220	10658	12923
50	56917	1909170	13971	15644
60	62611	2231064	13971	17624
70	68188	2553069	13971	19604
80	73776	2875067	13971	21584
90	79342	3197086	13971	23564
100	91118	3512923	17060	26005
110	101343	3830365	20053	28421
120	107040	4152255	20053	30401
130	112700	4474244	20053	32605
140	118168	4796442	20053	34585
150	123620	5118657	20053	36565
160	133746	5436194	22820	38591
170	139037	5758573	22820	40571
180	149221	6076023	25590	42598
190	154507	6398399	25590	44578
200	159742	6720829	25590	46558

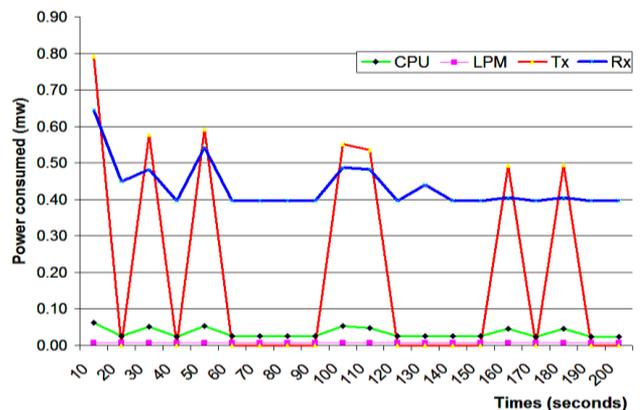


Figure 17. Skymote 4 power consumption in the triple point method

4.4 Triple point method discussion

According to the above results, we notice that the energy consumption by the three motes when these motes are in transmission (TX) or reception (RX) mode is greater than that if these same motes are in calculation (CPU) or standby (LPM). This means that most of the energy is often consumed in transmitting and receiving, and it is consumed more in receiving than in transmitting.

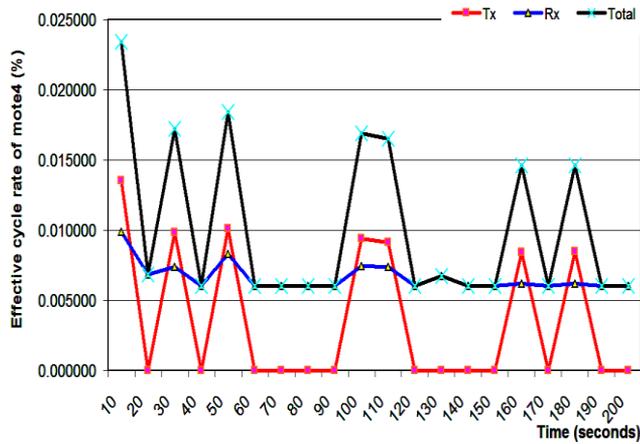


Figure 18. Percent of number of ticks for mote 4 in the triple point method

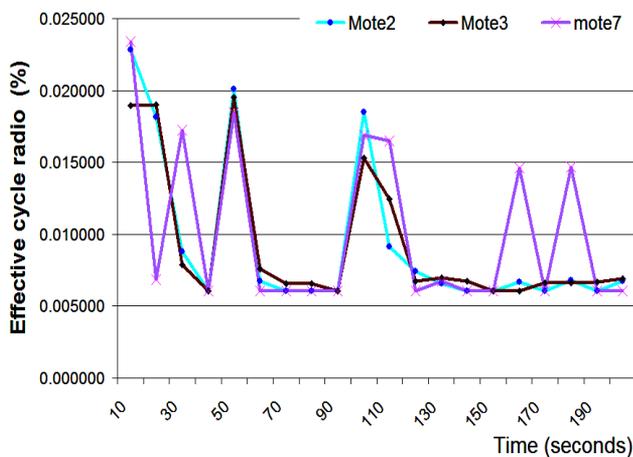


Figure 19. Effective cycle rate of the radio comparison for of the three motes, after every 10 seconds in in the triple point method

In Figure 19, we compare the two motes 2 and 3 which serve as aggregators for a mote 4, we note that the latter consumes a lot of energy than the aggregators and this is explained by the transmissions of temperatures, humidity rates and humidity rates broadcast light every 10 seconds.

4.5 Comparison of the two methods

The standard protocol of the RPL Cooja makes it possible to relay the value captured from one mote to another in a normal way. This method is simple and does not require any calculation. But the values captured are not always reliable and can cause a false alarm and a waste of time and resources for an audit. We have suggested a new method based on a reliability consensus governed by an aggregator node. It is based on a collector which calculates the average, at most, of

the first three values received, which then sends them to the Sink; this one we named it "the method of the triple points". This perspective ensures data reliability on one side and network perpetuity on the other side.

By taking the two plots of the effective radio cycles (Figures 11 and 19) side by side, we can draw the following conclusion:

(i) The maximum energy consumption by the standard method is 3.5% compared to our method which is 2.5%.

(ii) In our method the average consumption is around 1.1% compared to the standard method which exceeds 1.5% in energy.

(iii) It can also be noted that listening to the Rx reception channel is expensive in energy, but Tx transmission is more occasional.

(iv) In the standard method, mote 7 is not covered by the sink, so it is not overly stressed (no neighbor) if its reactivity will visibly increase the energy curve.

We notice that the standard RPL protocol gives very good results in response time but in power consumption it is by far the best. Our solution called 'triple points' is obviously the best solution in energy consumption and therefore the perpetuity of the network.

The standard method underlines a more interesting convergence time than our method that is surely due to our program for processing the average consensus but on the other hand it consumes more energy than ours.

5. CONCLUSIONS

The objective outlined in this article is the study of detection and cooperation for a good deployment of sensor networks. Detection can be summed up in two points: the first, detecting an event, but this operation is fortuitous because it is the pure function of the sensor. Agree on a consensus of the value to be transmitted in order to avoid false alarms on one side and to minimize network saturation by relaying the same values captured.

Our contribution was to improve the standard method of packet relay at the base station by other methods taking into account response time, data reliability and network lifetime. A method has been developed: Triple point method.

Our expectations thus verified, the simulations prove that the aggregation and the use of a cluster-head (aggregator) gives very good results in terms of network energy consumption and indeed a more interesting longevity. The trade-off is therefore emphasized: if the network relies on response time only then the standard method is the preferred one. But if we are talking about energy consumption and data reliability in order to avoid false alarms our triple point method is the best.

REFERENCES

- [1] Tmote, S. (2006). Ultra low power IEEE 802.15. 4 compliant wireless sensor module. <http://www.moteiv.com>
- [2] Al-Karaki, J.N., Kamal, A.E. (2004). Routing techniques in wireless sensor networks: A survey. *IEEE Wireless Communications*, 11(6): 6-28. <https://doi.org/10.1109/MWC.2004.1368893>
- [3] Wang, F., Hu, H. (2019). An energy-efficient unequal

- clustering routing algorithm for wireless sensor network. *Revue d'Intelligence Artificielle*, 33(3): 249-254. <https://doi.org/10.18280/ria.330311>
- [4] Yamparala, R., Perumal, B. (2019). Secure data transmission with effective routing method using group key management techniques-A survey. *Journal Européen des Systèmes Automatisés*, 52(3): 253-256. <https://doi.org/10.18280/jesa.520305>
- [5] Gorbachov, V., Sytnikov, D., Ryabov, O., Batiaa, A.K., Ponomarenko, O. (2020). Dimension reduction for network systems using structure model aggregation. *International Journal of Design & Nature and Ecodynamics*, 15(1): 13-23. <https://doi.org/10.18280/ijdne.150103>
- [6] Benaissa, B.E., Lahfa, F. (2019). Diffusion avisée après une collecte efficace de données par Intervalle de Confiance dans WSN basé sur l'IoT. *International Journal of Scientific Research & Engineering Technology (IJSET)*, 7(1): 6-15.
- [7] Periklis, N., Evangelia, P., Ozdemir, A. Barshan, B., Megalooikonomou, V. (2017). Investigation of sensor placement for accurate fall detection. In: Perego P., Andreoni G., Rizzo G. (eds) *Wireless Mobile Communication and Healthcare. MobiHealth 2016. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol 192. Springer, Cham. https://doi.org/10.1007/978-3-319-58877-3_30
- [8] Khan, S.M., Khan, W.M., Faraz, F.U., Khan, S.M. (2018). Incremental voting based spectrum sensing model for cognitive radio networks. *Review of Computer Engineering Studies*, 5(2): 27-33. <https://doi.org/10.18280/rces.050201>
- [9] Guyeux, C., Haddad, M., Hakem, M., Lagacherie, M. (2020). Efficient distributed average consensus in wireless sensor networks. *Computer Communications*, 150: 115-121. <https://doi.org/10.1016/j.comcom.2019.11.006>
- [10] Seam R.M., McCourt, M.J. (2020). Distributed estimation of an uncertain environment using belief consensus and measurement sharing. *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1362-1367. <https://doi.org/10.1109/SMC42975.2020.9283404>
- [11] Winter, T., Thubert, P., Brandt, A., Hui, J.W., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, J.P., Alexander, R. (2012). RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. *rfc*, 6550: 1-157.
- [12] Ren, J., Huang, S., Song, W., Han, J. (2019). A novel indoor positioning algorithm for wireless sensor network based on received signal strength indicator filtering and improved Taylor series expansion. *Traitement du Signal*, 36(1): 103-108. <https://doi.org/10.18280/TS.360113>
- [13] Alrawashdeh, R. (2018). Path loss estimation for bone implantable applications. *Jordanian Journal of Computers and Information Technology (JJCIT)*, 4(2): 94-101. <https://doi.org/10.5455/jjcit.71-1515153943>
- [14] Javaid, R., Qureshi, R., Enam, R.N. (2015). RSSI based node localization using trilateration in wireless sensor network. *Bahria University Journal of Information & Communication Technologies (BUJICT)*, 8(2): 58-64.
- [15] Sobral, J.V.V., Rodrigues, J.J.P.C., Rabêlo, R.A.L., Al-Muhtadi, J., Korotaev, V. (2019). Routing protocols for low power and lossy networks in Internet of Things applications. *Sensors*, 19(9): 2144. <https://doi.org/10.3390/s19092144>
- [16] http://contiki.sourceforge.net/docs/2.6/a01784.html#_details, accessed on Jan. 04, 2021.
- [17] Instruments, T. (2010). Msp430f5438 datasheet. Reference SLAS655B.
- [18] Contiki OS: Using Powertrace to estimate power consumption. [Online]. Available: <http://thingschat.blogspot.mk/2015/04/contikios-using-powertrace-and.html>, accessed on Feb. 25, 2021.