

# Diffusion avisée après une collecte efficace de données par Intervalle de Confiance dans WSN basé sur l'IoT

Bedr-Eddine BENAÏSSA<sup>#1</sup>, Fedoua LAHFA<sup>#2</sup>

<sup>#</sup>Department of Computer sciences

Abou Bekr Belkaid University, Tlemcen, Algeria

<sup>1</sup>bbedro@hotmail.fr

<sup>2</sup>fedouadidi@yahoo.fr.com

**Résumé**— L'exploitation d'un réseau de capteurs soulève de nombreux problèmes et ce à plusieurs niveaux algorithmiques : localisation, déploiement, collecte de données, couverture et réduction de la consommation d'énergie des batteries, afin d'optimiser la durée de vie du réseau. Ce dernier point a suscité particulièrement l'attention des chercheurs. L'objet de cet article serait ainsi de considérer une nouvelle méthode d'optimisation d'énergie d'un nœud de capteurs en minimisant les fréquences de transmission des données mesurées vers la station de base. Le nœud (mote) possédant plusieurs capteurs mesurant différentes valeurs environnementales, notre travail reviendrait par conséquent à catégoriser les données captées dans des classes prédéfinies, numérotées au préalable, que nous nous nommerons « intervalles de confiance ». De cette manière chaque valeur captée sera rangée dans une classe et seul son numéro sera envoyé à la station de base, si (et seulement si) un changement de classe par rapport à la valeur précédente est observé. Les résultats montrent que la collecte de données par intervalle de confiance réduit considérablement l'énergie des capteurs des notes.

**Mots clés**— WSN; IoT; collection de données; durée de vie d'un capteur.

## I. INTRODUCTION

La modélisation de la consommation d'énergie dans les réseaux de capteurs est l'un des aspects les plus importants dans l'exploitation de ces réseaux, étant donné que l'énergie est l'une de ses ressources les plus vitales [1]. Diverses références ont été rapportées dans ce domaine qui proposent des modèles de consommation énergétique permettant d'étudier et d'évaluer les différents types de dispersion d'énergie dans ces réseaux. Diery NGOM, dans sa thèse, fait une synthèse de la question et y souligne différents états de consommation d'énergies [2]. Il affirme ainsi que la distance entre les nœuds crée des interférences qui induiront de fréquentes collisions, impliquant des retransmissions de paquets et par conséquent un complément d'énergie, d'autant plus que les nœuds peuvent aussi être composés de plusieurs capteurs et d'actionneurs [3]. Dans l'article « A Review of Different Energy Efficiency Techniques in Wireless Sensor Networks » [4], les auteurs soulignent que la raison essentielle de la consommation d'énergie d'un capteur est due au processus de communication. Une application en temps

réel entraîne obligatoirement que des données de mise-à-jour soient transmises au nœud collecteur dans un délai limité. Une des premières techniques utilisées pour réduire la quantité de données transférées est l'agrégation, selon le type des données étudiées et en utilisant la série temporelle ARMA [5]. D'autres études proposent un schéma de routage dynamique basé sur un anneau d'agrégation de données de corrélation (RBCDR). Dans ledit schéma, les données nodales sont acheminées dans un premier temps vers des anneaux à énergie abondante et par sauts minimaux ; dans un second temps toutes les agrégations de données sont traitées le long de l'anneau puis acheminées vers le puits, par le chemin le plus court [6]. Les auteurs de l'article [7] proposent, quant à eux, un système de collecte de données - MTDR visant à minimiser le délai de transmission dans les réseaux sans fil à faible coefficient d'utilisation. Ce contrôle de mise à jour est l'un des moyens capables de réduire les données transmises par les nœuds ; il influe directement sur la consommation d'énergie et par conséquent augmente la durée de vie du réseau [8].

## II. MOTIVATION

Dans les applications de réseaux de capteurs, il y a nécessairement collecte d'informations sur l'état d'un environnement afin de se prévenir des risques. Cela suppose donc que les données provenant de ces réseaux de capteurs aient un système de communication efficace et optimisé. L'optimalité, ici, signifie que le capteur doit pouvoir prendre une décision appropriée d'envoi ou non des données au collecteur. Dans les travaux d'Alduais et al. [8], les auteurs proposent de réduire le nombre de transmissions et diminuer la quantité de données, ce qui va prolonger la durée de vie du réseau. Ils introduisent ainsi la notion de changements relatifs et celles des différences relatives entre les valeurs mesurées à l'instant  $t$  du capteur et les valeurs déjà mesurées à l'instant  $(t-I)$  du même capteur. Dans le même temps, [8] expose leur approche, avec des résultats concurrents, pour pallier les travaux de [9] dont les auteurs proposent un algorithme d'agrégation, avec le principe qu'aucun des nœuds du capteur ne puisse envoyer réellement ses données que si la valeur  $Sv_{(t)}$  de l'état courant (Curent value measured by the Sensor) de la détection, est identique (algorithme ES : Exactly Same)

à la valeur du précédent état ( $Sv_{(t-1)}$ ). En d'autres termes, la valeur du changement absolu entre ( $Sv_{(t)}$ ) et ( $Sv_{(t-1)}$ ) est nulle.

Dans le présent article, nous faisons le point sur une application de prise de décision des transmissions de données captées qui introduit la notion « d'intervalle de confiance. » Cet algorithme décidera ainsi de déclencher ou non la transmission des données vers la station de base pour participer à l'optimisation, la plus efficace, de la transmission et, par conséquent, en minimiser l'énergie.

### III. ETAT DE L'ART

L'introduction de l'IoT a de toute évidence généré une somme importante de données, dont regorge le cloud computing lui-même. Par ailleurs, la demande d'objets intelligents, connectés (capteurs) et autonomes, permettant le confort et la sécurité domotique se fait de plus en plus importante. En effet, si la distance de transmission de l'objet « Smart » est grande, ce dernier devra élever sa puissance d'émission et par conséquent consommera plus d'énergie, et réduira sa propre durée de vie. Il convient donc de trouver des méthodes à même de surpasser ces contraintes. Beaucoup d'auteurs se sont en effet penchés sur la question ; par exemple [10]-[12] ont montré que l'agrégation des données est plus efficace pour les réseaux statiques, là où le taux de transmission de données est constant. Quant aux réseaux dynamiques, leur regroupement dépendra de leur degré de mobilité. Dans un modèle de données « Evènement-Réponse », le concepteur du réseau d'IoT, ne s'intéresse qu'à l'occurrence de l'apparition d'une annonce ou d'un ensemble d'annonces insolites. Par contre, dans le modèle de données orientées « Requête-Réponse », les capteurs ne signalent leurs résultats mesurés au nœud collecteur, qu'à la suite d'une requête manifestée par l'observateur (directement ou indirectement par l'intermédiaire d'autres capteurs). Ces approches peuvent toutefois se trouver dans un même réseau (modèle hybride). Une continuité améliorée de cette approche est présentée dans [13] ; PEGASIS y propose que chaque nœud puisse communiquer uniquement avec son proche voisin et transmettre tour à tour à la station de base, réduisant ainsi la quantité d'énergie dépensée.

Dans les travaux centrés sur les données, nous avons aussi le travail de [14] nommé EAD, représenté par un arbre de diffusion spécial, avec plusieurs feuilles, dans le but de faciliter le routage. Ce travail est repris par [15] où l'accent est mis sur la mise à jour du nœud-puits en faisant d'abord une agrégation des données provenant des différents capteurs sources. Les auteurs mettent en application deux approches : « Directed Diffusion, (DD) » et « Sensor Protocol for Information via Négociation (SPIN) ». D'autres solutions de minimisation d'énergie sont proposées. On peut citer : EAR, MECN, HEED, EAQSR... etc. [16].

Dans tous les travaux centrés sur les « données », une attention particulière est donnée à l'étude de [8]. En effet, la réduction du nombre de transmissions de données, par requêtes et par intervalle de temps, en vue de collecter périodiquement les mises à jours des données d'une zone sous observation, est un moyen efficace de réduire le On/Off

des capteurs et par conséquent accroître la durée de vie du réseau IoT.

Les auteurs de [9] proposent une agrégation de données vers le nœud collecteur ; un traitement est fait à ce niveau. Ils recourent à « l'égalité absolue » ; aussi, si la valeur de l'état actuel ( $Sv_{(t)}$ ) est « égale exactement » – au sens absolu – à la valeur de l'état précédent ( $Sv_{(t-1)}$ ) –précédemment enregistrée et sauvegardée dans un tampon du collecteur– il n'y aura pas transfert de données à l'observateur. [8] reprennent ces travaux et soulignent la même idée avec un algorithme amélioré : un nœud à plusieurs capteurs ne transmet ces données que s'il y a une « différence relative » entre la valeur mesurée à l'instant ( $t$ ) et la dernière valeur mesurée transmise à ( $t-1$ ).

Nous savons tous que la valeur des données est relative, selon que l'on se trouve du côté du réseau ou de celui de l'observateur. Ce dernier s'intéresse beaucoup plus aux changements de l'environnement surveillé qu'aux valeurs des données elles-mêmes. Par exemple, si les capteurs surveillent une zone de forêt pour prévenir le déclenchement du feu, l'observateur n'a pas besoin de voir les données numériques de tous les capteurs individuels, mais doit savoir s'il y a eu une augmentation notable de température indiquant le démarrage d'un incendie dans la zone surveillée. C'est cette question primordiale qui va déclencher des prises de décisions efficaces.

### IV. PRISE DE DECISION DANS UN NŒUD A PLUSIEURS CAPTEURS

La *différence réelle* entre deux grandeurs numériques,  $v_t$  et  $v_{t-1}$ , est présentée par la formule :  $\Delta = v_t - v_{t-1}$ . Cette dernière est dite *changement réel* lorsque la valeur  $v_{t-1}$  est une valeur de départ ou de référence à laquelle  $v_t$  est comparée. Par contre, si  $v_{t-1}$  n'est pas une valeur de référence comme décrite dans [8], la valeur de  $\Delta$  serait alors nommée « différence absolue », soit :  $|\Delta| = |v_t - v_{t-1}|$ , où l'ordre de  $v_t$ ,  $v_{t-1}$  n'a pas d'importance. Cependant, une autre observation suscite notre attention : à savoir que la différence absolue entre deux valeurs n'est pas toujours le bon moyen de comparer les nombres. Illustrons cela par l'exemple suivant : un capteur relève à l'instant ( $t$ ) une valeur discrète, mesurée ( $v_t$ ) alors qu'à l'instant ( $t-1$ ) il a déjà relevé la valeur  $v_{t-1}$ . Nous avons le tableau ci-après :

TABLEAU I  
ILLUSTRATION DE LA DIFFERENCE ABSOLUE

value \ Sensor	Sensor 1 $t$	Sensor 1 $t=t+1$	Sensor 1 $t=t+2$
$v_t$	35°C	50°C	65°C
$v_{t-1}$	20°C	35°C	50°C
$ \Delta  =  v_t - v_{t-1} $	15°C	15°C	15°C

Remarque:

Si un capteur entre dans le domaine de non-destruction, le retour à l'état de stabilité est impossible ; seul un nouveau calibrage manuel permettra de le remettre en état d'emploi.

Dans notre exemple du tableau 1, nous prenons en considération une certaine durée de remise en état initial du capteur, inférieur à la durée nécessaire à la stabilité du capteur, c-à-d un temps de passage du domaine de non détérioration au domaine nominal d'emploi.

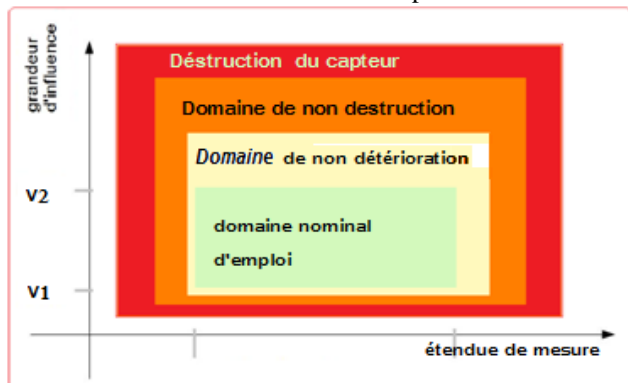


Fig. 1 Limites d'utilisation d'un capteur et étendue de mesure [17]

La différence absolue est égale à 15°C à l'état (t), (t+1) et à (t+2), cette différence n'étant pas significative. La question qui se pose serait : pourrions-nous dire, dans ce cas, qu'il y a un changement d'état ? Quand bien même cette différence serait-elle prise par rapport à un seuil donné ( $\beta \leq 20^\circ\text{C}$ ) pour déclencher une transmission, la réalité des choses n'est pas établie. Kim et al. [9] mettent en valeur leur théorie en se basant sur l'équation (1).

$$F_s = \begin{cases} 1 & \text{Si } A_c = |Sv_t - Sv_{t-1}| > \beta \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

Alduais et al. [8] reprennent le travail de [9], en remettant toutefois en question la différence absolue qui ne reflète pas, d'une manière générale, un changement d'état réel. Les auteurs de [8] s'appuient quant à eux sur la différence relative (Equation 2) et le changement relatif (Equation 3) pris par rapport à un coefficient de tolérance  $\beta$  (=seuil) comme différence maximale admise fixée par le concepteur. Aussi, énoncent-ils les équations (2) et (3) suivantes liées à la décision de mettre à jour le nœud collecteur ou la station de base :

$$F_s = \begin{cases} 1 & \text{Si } R_f = \frac{|Sv_t - Sv_{t-1}|}{Sv_{t-1}} > \beta \\ 0 & \text{Otherwise} \end{cases} \quad (2)$$

$$F_s = \begin{cases} 1 & \text{Si } R_f = \frac{|Sv_t - Sv_{t-1}|}{\frac{Sv_t + Sv_{t-1}}{2}} > \beta \\ 0 & \text{Otherwise} \end{cases} \quad (3)$$

La collecte des données à la demande - et en temps réel - est très importante dans un réseau IoT. L'approche [8] est souvent utilisée lorsque les deux nombres reflètent un changement dans une entité sous-jacente unique. Cette approche peut néanmoins susciter deux problèmes :

- 1 Si dans l'équation (2) du changement relatif,  $(Sv_{t-1}) = 0$  la fonction de sortie  $F_s$  ne sera pas définie. Il faut souligner que cette grandeur discrète  $(Sv_{t-1})$  peut être nulle du fait qu'elle n'est pas la valeur de référence fixée par l'observateur, et qu'elle varie aléatoirement dans le temps.  
Doit-on comprendre alors que le capteur est en état Off ou bien qu'il n'a rien décelé ?
- 2 Si  $(Sv_{t-1})$  et  $(Sv_t)$  sont de même valeur mais de signes opposés, le dénominateur  $(D_{Rf})$  de l'équation (3) vaut zéro [19],

$$\text{Si } Sv_t = -Sv_{t-1} \text{ alors } D_{Rf} = \frac{Sv_t + Sv_{t-1}}{2} = 0 \quad (4)$$

Doit-on comprendre, là encore, que le capteur est en état Off ou qu'il n'a détecté aucun changement d'état. Cela peut ne pas affecter le réseau IoT, à un certain degré de liberté ; mais si c'est le cas - par exemple en relevant à l'instant  $(t-1)$  une température  $Sv_{t-1} = +5^\circ\text{C}$  et à l'instant  $(t)$  une température  $Sv_t = -5^\circ\text{C}$  (cf. les valeurs de fonctionnement d'un nœud Waspote, Figure 2) [18] - nous serons face à une incertitude de résultat.

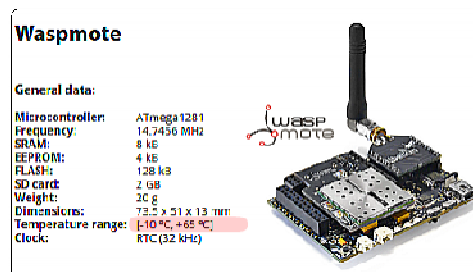


Fig. 2 Données générales de fonctionnement d'un nœud Waspote [18]

La solution présentée par l'auteur [19] à la question de la « différence relative » consisterait à remplacer le dénominateur de l'équation (3) par la moyenne des valeurs absolues de  $(Sv_t)$  et  $(Sv_{t-1})$  :

$$F_s = \begin{cases} 1 & \text{Si } R_f = \frac{|Sv_t - Sv_{t-1}|}{\frac{|Sv_t| + |Sv_{t-1}|}{2}} > \beta \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

#### V. APPROCHE PROPOSEE

Dans un réseau IoT, le déploiement de capteurs a pour but de surveiller une zone quasi- autonome, chaque capteur étant concerné par une sous-zone. Un mote spécial est chargée de rassembler les informations en provenance des capteurs et et d'en informer l'observateur humain si un phénomène inhabituel survient. Ainsi, minimiser la consommation d'énergie reviendrait à minimiser, entre autres actions, la quantité de données transmises dans le réseau. En effet, selon les études menées dans ce contexte, l'énergie consommée dans un nœud capteur serait due aux transmissions de données. De ce fait, notre approche consiste à agréger les données des différents capteurs en une information significative appelée « intervalle de confiance », en éliminant

la redondance par le recours à une nouvelle méthode basée sur la notion de tolérance (d'appartenance à des intervalles de confiance). Ceci résoudra nécessairement le problème d'excès de transmission de valeurs collectées, et par conséquent augmentera la durée de vie des capteurs.

Les modèles de Markov Cachés (Hidden Markov Models ou HMM) modélisent des phénomènes aléatoires composés d'un processus aléatoire de transition entre des états inobservables dans un premier niveau (les états cachés) et, à un second niveau, d'un autre processus aléatoire qui, dans chaque état, engendre des valeurs observables (appelées observations).

Notre idée repose sur le principe du niveau d'observation des HMM.

Nous construisons une discrétisation des données de transitions d'un ensemble de jour, nous relevons une classification par intervalle des données, et nous l'appliquons sur l'ensemble des données des jours à venir. De ce fait les valeurs seront vérifiées par rapport à un intervalle de données préétabli, dans les conditions normales. Si une valeur n'est pas classée dans ces intervalles de confiance, elle l'est donc dans une classe spéciale appelée « OutOff range ». L'application de la méthode « Exactly Same » sera appliquée pour comparer les appartenances aux classes. Ainsi, toutes les valeurs normales recueillies seront dans une classe spécifiée et connues à l'avance, et les valeurs d'alertes seront quant à elles représentées par la classe « 0000 ».

A noter que l'ensemble des tests est effectué sur la base de données des informations collectées par 54 capteurs déployés dans le laboratoire Intel Berkeley Research entre le 28 février et le 5 avril 2004. [20]

N°	Date	Heure	epoch	MoteID	Temp	Humidity	Lumière	Tension
1	28/02/2004	00:59:16.028	3	1	19.9884	37.0933	45.0800	2.6996
2	28/02/2004	01:03:16.334	11	1	19.3024	38.4629	45.0800	2.6874
3	28/02/2004	01:06:16.013	17	1	19.1652	38.8039	45.0800	2.6874
4	28/02/2004	01:06:46.778	18	1	19.1750	38.8379	45.0800	2.6996
5	28/02/2004	01:08:45.993	22	1	19.1456	38.9401	45.0800	2.6874
6	28/02/2004	01:09:22.324	23	1	19.1652	38.8720	45.0800	2.6874
7	28/02/2004	01:09:46.110	24	1	19.1652	38.8039	45.0800	2.6874
8	28/02/2004	01:10:16.679	25	1	19.1456	38.8379	45.0800	2.6996
9	28/02/2004	01:10:46.251	26	1	19.1456	38.8720	45.0800	2.6874
10	28/02/2004	01:11:46.941	28	1	19.1456	38.9401	45.0800	2.6996
11	28/02/2004	01:12:46.251	30	1	19.1358	38.9061	45.0800	2.6874
12	28/02/2004	01:14:16.631	33	1	19.1162	38.8039	45.0800	2.6996
13	28/02/2004	01:14:46.569	34	1	19.1162	38.8720	45.0800	2.6996
	28/02/2004	01:15:16.650	35	1	19.1064	39.0082	45.0800	2.6996
	28/02/2004	01:16:16.344	37	1	19.1064	38.8720	43.2400	2.6996
	28/02/2004	01:16:46.009	38	1	19.0966	38.8039	43.2400	2.6996
	28/02/2004	01:17:16.009	40	1	19.0966	38.7857	43.2400	2.6996
	28/02/2004	01:17:46.009	41	1	19.0868	38.8039	43.2400	2.6996
	28/02/2004	01:18:16.009	42	1	19.0672	38.9061	43.2400	2.6874
	28/02/2004	01:18:46.009	43	1	19.0672	38.8720	43.2400	2.6996

Fig. 3 Aperçu de la structure du DataSet d'Intel (2004)

### Hypothèse

Dans les conditions normales d'observation, les relevés de données issues des capteurs représentent les valeurs les plus fiables. Un intervalle de rangement de ces valeurs suffit ainsi à prendre connaissance de l'état de cet environnement. Par contre, au-delà de ces intervalles, les valeurs sont sujettes à conjecture.

Hypothèse 1 : Une discrétisation par intervalle de données agrégera les transitions en observation.

Hypothèse 2 : La décision de mise à jour est une fonction booléenne ( $F_s$ ) basée sur un intervalle de confiance ; on utilisera la méthode « Exactly same » des travaux de [5], sur des intervalles et non sur des valeurs observables :

- ( $F_s$ ) ne doit, en aucun cas, être indéterminée.
- Dans toutes les situations ( $F_s$ ) doit fournir une solution  $\in \{0,1\}$ .

L'objectif de la station de base est de s'informer des valeurs relevées dans le réseau à intervalles réguliers afin de détecter un éventuel changement. Sur les données issues du dataSet d'Intel, l'intervalle des relevés est de trente et une seconde (31s), le relevé d'une journée effectuée par l'équipe du laboratoire variant entre 1316 et 2800 valeurs.

Une perspective de transmettre ce nombre, pendant une année, consisterait donc à envoyer un nombre de valeurs d'environ  $\cong (3600/30) \times 24 \times 365 = 10\ 512\ 000$  valeurs. Ce nombre est calculé pour un seul paramètre : la température ; toutefois, si nous prenons en considération l'humidité, la lumière et la pression, la taille du paquet augmentera forcément, ce qui influera obligatoirement sur la durée de transmission.

Il en résulte que notre approche est de réduire la transmission des données en utilisant la méthode de discrétisation issue des statistiques descriptives, ce qui permettra de réunir des données captées sur des ensembles.



Fig. 4 Schéma résumant les statistiques descriptives

### Définition 1 :

Une série statistique se définit par une population et un caractère :

1. La population est l'ensemble des individus (ici : les capteurs)
2. Le caractère est la qualité étudiée chez ces individus. (Ici : température, humidité, lumière, tension, etc.). De plus, un caractère peut-être :
  - a) Qualitatif, (valeurs non numériques : couleur, forme, etc.).
  - b) Quantitatif : à savoir
    - Discret (nombre fini de valeurs naturelles)
    - Continu (infinité de valeurs) : notre cas d'études.

La discrétisation met en évidence la question du nombre de classes ( $k$ ) à créer. D'après la définition de [22], « son objectif est de perdre le moins d'information possible tout en retranscrivant au mieux l'organisation statistique des données ».

Nos données constituent de la sorte un caractère quantitatif continu. Ainsi, l'établissement du tableau de fréquences implique d'abord une répartition en classes de ces données. Cela nécessite de définir, en premier, le nombre de classes puis l'amplitude de chacune d'elles. En règle générale, le choix des classes doit être de même amplitude.

Il est à noter cependant qu'une distribution en fréquences est explicative si – et seulement si – chaque classe comprend un nombre suffisant de valeurs captées ( $v_i$ ). Plusieurs méthodes expérimentales permettent d'établir le nombre de classes pour un échantillon de taille ( $n$ ).

**Phase 1 : Classification**

**1<sup>ère</sup> étape : Identifier le nombre de classes (k)**

**Définition 2 :**

Le nombre de classes (k) et la formule qui y est associée dépendent du type de variables. Quel que soit le cas, deux règles s'imposent :

- Le nombre de classes est arrondi à l'entier le plus proche.
- Le nombre de classes doit s'inscrire dans l'intervalle [5,8].

Diverses formules empiriques existent afin de classer les données; ainsi pour les variables continues :

- Formule de Sturge :  $k = \text{Log}_2(n)$
- Formule de Sturge-Huntsberger :  $k = 1 + (10/3) \cdot \text{Log}_{10}(n)$
- Formule de Brooks-Carruthers :  $k = 5 \cdot \text{Log}_{10}(n)$
- Formule de Scott :  $k = (\text{valeur min} - \text{val max}) / 3,5 \cdot \sigma \cdot n^{(-1/5)}$  (cas de distribution asymétrique)

**Application**

Notre choix s'est porté sur la formule de Sturge-Huntsberger<sup>1</sup>[21], étant donné la distribution de l'échantillon élu :

$$k = 1 + (10/3) * \text{Log}_{10}(n) \quad (6)$$

Si nous prenons en exemple un échantillon de valeur n = 1000, le nombre de classe significatif serait :

$$k = 1 + 3.33 \cdot \text{Log}_{10}(1000) = 10.99 \approx 11 \text{ classes (en application de la définition 2).}$$

De cette formule, nous déduisons la méthode de classification en choisissant k = 5 classes pour un contrôle supervisé et une meilleure visibilité.

**2<sup>ème</sup> étape : Analyse descriptive de l'échantillon**

Notre échantillon est pris dans la base de données du laboratoire d'Intel Berkley. Un intervalle de confiance est calculé sur la base de 5 Classes (k=5) en déterminant à titre d'exemple, le niveau de risque associé au danger.

Nous relevons les valeurs minimale et maximale admises parmi les valeurs d'une journée jugées acceptables, (valeurs prises dans des conditions normales d'observation) : exemple :

- Min ( $v_{t_i}$ ) = 17,5090
- Max ( $v_{t_i}$ ) = 26,6328

Calculons l'écart par la formule de Sturge :  
 $E = \text{Max}(v_{t_i}) - \text{Min}(v_{t_i}) = 26,6328 - 17,1954 = 9.4374$   
 Calculons l'amplitude :  $A = E/k = 9.4374 / 5 = 1.8874$

Une analyse statistique descriptive des données de température de notre échantillon ( $v_{t_i}$ ), nous révèle ainsi les données quantitatives suivantes :

TABLEAU 2

STATISTIQUES DESCRIPTIVES DES INTERVALLES DE CONFIANCES – MOTE 1: SENSOR « TEMPERATURE »

Classes	Borne inf.	Borne Sup.	Effectif	Fréquence	Densité
1	[17.1954;	19.0829[	2575	0.291	0.154
2	[19.0829;	20.9704[	2106	0.238	0.126
3	[20.9704;	22.8579[	1687	0.191	0.101
4	[22.8579;	24.7454[	1618	0.183	0.097
5	[24.7454;	26.6328[	857	0.097	0.051

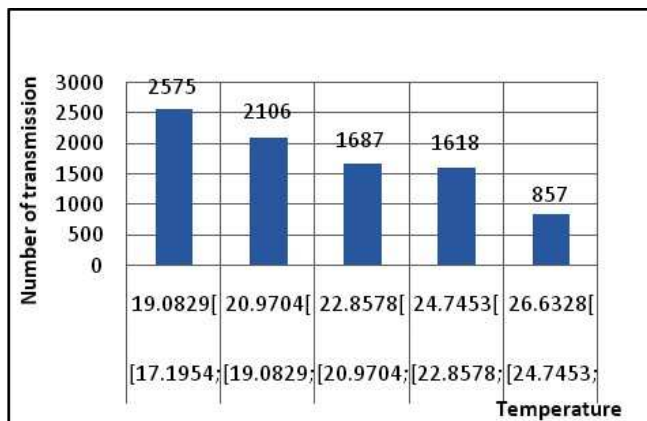


Fig. 5 Nombre de transmissions par « Intervalle de Confiance – C.I.» (Mote1 - Sensor1 : Température)

**3<sup>ème</sup> étape :**

De cette étape, les valeurs de température sont établies par intervalles de confiance (classes); exemple : Classe1 = [17,1954 ; 19,0829[, Classe1 = [19,0829; 20,9704[, etc. Sachant que l'on a recensé 5 classes, on met en marche les capteurs, et chaque température relevée (niveau caché) sera attribuée à la classe correspondante (niveau observable). De ce fait, si la valeur d'une température captée est cadrée dans une classe, le numéro de classe est transmis à la station de base pour une mise à jour. Si, par contre, la température relevée n'appartient à aucun intervalle de confiance, un niveau de classe « OutOff » (Out of range) sera transmis pour une mise à jour, soulignant ainsi un état d'alerte. De cette façon, la station de base pourra observer le mote et prévenir le niveau de danger, avant d'arriver au seuil de détérioration.

**Remarque**

Le nombre de classes (k) peut être augmenté, ce qui élargira l'étendue des valeurs captées.

**Phase 2 : Transmission et mise à jour des données**

**Etape1. Structure du paquet transmis**

Les données sont la partie la plus importante dans la transmission d'un capteur et le nœud capteur waspmote peut supporter jusqu'à 16 différents capteurs en même temps (voir caractéristiques). Pour ce type de mote, nous pourrions réserver 16 champs de données. Aussi, étant donné l'entête du paquet, si nous avons (n) capteurs nous aurons (n) champs de données en plus (Fig. 5).

<sup>1</sup> La règle de Sturges : formule mathématique proposée par Herbert Sturges (1882-1958).

Notre paquet de données sera donc le même que le paquet décrit par le groupe du laboratoire Intel [20], mais en substituant un champ « N° classe » de chaque type de caractère du mote au champ des valeurs des capteurs. Le paquet se représentera comme suit :

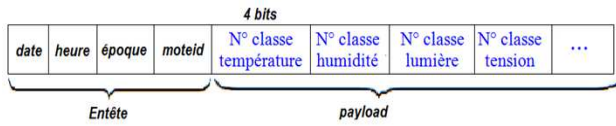


Fig. 6 Structure d'un paquet du Laboratoire Intel modifiée [20]

On peut en déduire que pour un mote de 4 capteurs, nous avons besoin de  $4 \text{ bits} \times 4 = 16 \text{ bits}$ .

- Le champ « époque » est la période de capture de ces grandeurs, c'est un numéro de séquence croissant de façon monotone à partir de chaque mote.
- Le champ « Moteid » : Un nombre séquentiel de capteurs.
- Le champ N° classe contient (m) bits et dépend du nombre de classe k ;  
 Ici,  $k = 5 = 2m$ , ceci donne  $m = 3 \text{ bits}$  pour codifier 8 classes;

TABLEAU 3

REPRESENTATION BINAIRE DU CONTENU DES CHAMPS

Nbr. Classes	N° classe température	Min ( $v_i$ )	Max ( $v_i$ )
0	0000	"OutOff"	
1	0001	[17.1954;	19.0829 [
2	0010	[19.0829;	20.9704 [
3	0011	[20.9704;	22.8579 [
4	0100	[22.8579;	24.7454 [
5	0101	[24.7454;	26.6328 [
7	0110	Réservés pour d'autres classes	
8	0111		

Connaissant l'amplitude de la température ( $A = 1.8874$ ) dans le tableau ci-dessus, il est possible de rajouter d'autres classes (voir tableau 5) ; on pourra alors choisir le nombre de bits y correspondant ; par exemple pour  $m = 4$  on peut avoir plus de 8 classes :  $2^4 = 16 \text{ classes}$ .

TABLEAU 4

EXTENSION DES CLASSES EN UTILISANT L'AMPLITUDE  $A = 1.8874$

N° classe température	Min ( $v_i$ )	Max ( $v_i$ )
0110	[26.6328;	28.5202 [
0111	[28.5202;	30.4076 [
1000	[30.4076;	32.2950 [
1001	[32.2950;	34.1824 [
1010	[34.1824;	36.0698 [
1011	[36.0698;	37.9572 [
1100	[37.9572;	39.8446 [
1101	[39.8446;	41.7320 [
1110	[41.7320;	43.6194 [
1111	[43.6194;	45.5068 [

**NB.** Il reste que les classes ajoutées sont prises dans les conditions d'acceptabilité des valeurs captées.

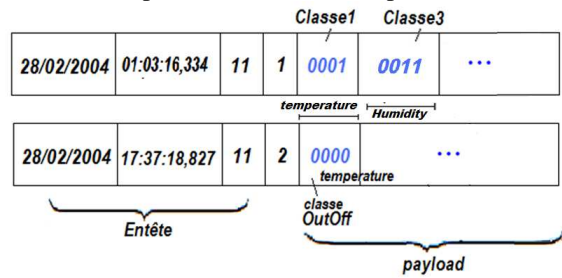


Fig. 7 Exemple de paquets avec le caractère température

**Remarque :**

Dans notre cas d'étude, le choix des intervalles de confiance (au lieu des valeurs) contribue pleinement à réduire la taille du paquet : pour 4 capteurs par mote, nous aurons 16 bits de données. Ceci entrainera nécessairement une diminution des délais de transmission et de propagation dudit paquet, et par conséquent l'énergie consommée pour une transmission.

**Etape 2. Algorithme de mise à jour**

- Algorithme mono-sensor

```

Inputs: classes(k), Sv(t), Sv(t-1), A //k=1..5; A : amplitude = 1,8874;
Outputs: Fs ; // Fs: Binary function to send data or Not send
Begin
  Set Fs ← 0
  Set Ts ← OFF
  Set Classe(t-1) ← the last classed Sensor (Sj) value // Classe(t-1) : Class sensor at time "t-1"
  Read Sv(t) // Sv(t) a sensor value at time t.
  // j : a counter of (k) classes
  For j = 1 to k=5
    If Sv(t) in classe(j) then
      If classe(t-1) = classe (j) then // "Exactly-Same" comparison class method,
        Set Fs ← 0 // already in class
      Else
        Set Fs ← 1
      End if
    Else
      Set Classe(t-1) ← "0000" // "OutOff" range class - ALARM -
      Set Fs ← 1
    End if
  End For
  // sending data
  IF Fs = 1 then
    RF-Transmit ← ON // sending a new class of data
  Else
    RF-Transmit ← OFF
  End if
End
    
```

**Etape 3. Réalisation de l'interface**

Nous avons mis en place un programme nommé « Sensing » illustrant les quatre algorithmes décrits précédemment, à savoir « Exactly same - ES », « Absolute Difference - AD », « Relative Difference - RD » et naturellement notre algorithme « Confidence Interval - CI ».

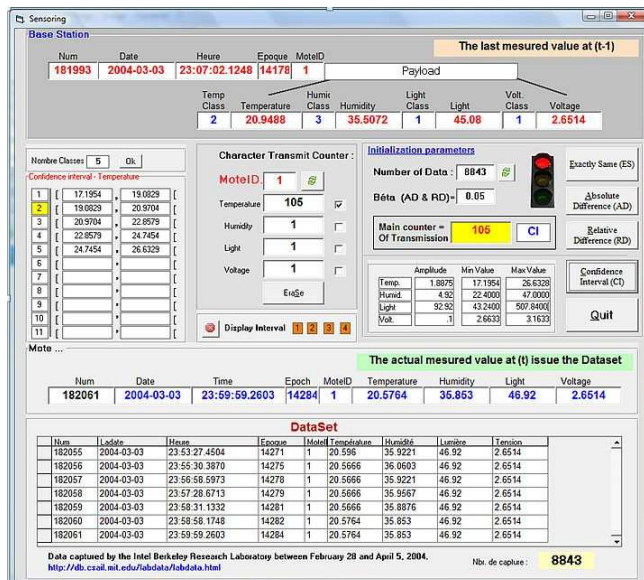


Fig. 8 Interface du programme « Sensing »

VI. EVALUATION DES PERFORMANCES

L'évaluation de l'algorithme est effectuée sur un échantillon de données d'Intel Berkeley Research Lab (IBRL) [20]. L'expérience est prise au hasard sur plusieurs notes. Ainsi on en a retenu 6 (sur 54) : soit les notes 1, 4, 7, 9, 18, 20 pour la présentation, mais les tests ont été effectués sur l'ensemble et les résultats ont été très satisfaisants.

Notons que tous les notes comportent quatre (04) capteurs, et que les tests ont été pris en considérant toutes les combinaisons possibles des capteurs.

Les scénarios de tests sont représentés dans le tableau ci-dessous ; on choisira les dénominations suivantes :

Temp. (T): Température; Humid. (H): Humidité; Light (L) : Lumière; Volt. (V) : Voltage

TABLEAU 5

RESULTATS OBTENUS APRES L'EXECUTION DU PROGRAMME « SENSING » POUR LES NOTES 1, 4, 7, 9, 18, 20

<b>MoteID</b>	1
<b>DataSet</b>	8843 paquets

(a)

Parameters	Algorithm	ES	AD	RD	CI
Temp. (T)		6679	1706	62	105
Humid. (H)		6596	4498	54	91
Light. (L)		1592	1592	777	163
Volt. (V)		2916	12	1	1
Temp.+ Humid.		8147	4718	80	193
Temp.+ Light.		6894	2678	802	261
Temp.+ Volt.		7363	1706	62	105
Humid.+ Light		6922	5095	794	249
Humid.+ Volt.		7365	4498	54	249
Light. + Volt.		4000	1592	777	163
Temp. + Humid. + Light		8207	5249	802	345
Temp.+Light+Volt		7512	2678	802	261
Humid. + Light + Volt		7575	5095	794	249
Temp.+Humid.+Light+Volt		8411	5249	802	345

<b>MoteID</b>	4
<b>DataSet</b>	8754 paquets

(b)

Parameters	Algorithm	ES	AD	RD	CI
Temp. (T)		6109	1291	51	67
Humid. (H)		6380	4275	51	96
Light. (L)		1043	1043	587	113
Volt. (V)		3053	9	9	2
Temp.+ Humid.		7893	4411	69	153
Temp.+ Light.		6316	1954	597	175
Temp.+ Volt.		7021	1291	51	68
Humid.+ Light		6595	4713	602	204
Humid.+ Volt.		7235	4275	51	204
Light. + Volt.		3692	1044	587	114
Temp. + Humid. + Light		7952	4821	602	259
Temp.+Light+Volt		7149	1954	597	176
Humid. + Light + Volt		7364	4713	602	205
Temp.+Humid.+Light+Volt		8234	4821	602	260

<b>MoteID</b>	7
<b>DataSet</b>	11969 paquets

(c)

Parameters	Algorithm	ES	AD	RD	CI
Temp. (T)		8128	1334	48	77
Humid. (H)		8180	5246	47	80
Light. (L)		1414	1414	754	109
Volt. (V)		4390	7	7	1
Temp.+ Humid.		10676	5423	64	153
Temp.+ Light.		6894	2678	802	184
Temp.+ Volt.		7363	1706	62	77
Humid.+ Light		6922	5095	794	187
Humid.+ Volt.		7365	4498	54	187
Light. + Volt.		4000	1592	777	109
Temp. + Humid. + Light		8207	6083	770	259
Temp.+Light+Volt		9814	2332	763	184
Humid. + Light + Volt		9792	5969	765	187
Temp.+Humid.+Light+Volt		11238	6083	770	259

<b>MoteID</b>	9
<b>DataSet</b>	9113 paquets

(d)

Parameters	Algorithm	ES	AD	RD	CI
Temp. (T)		6659	1371	59	78
Humid. (H)		6334	4298	57	118
Light. (L)		1343	1343	848	57
Volt. (V)		2378	10	1	260
Temp.+ Humid.		8264	4482	77	193
Temp.+ Light.		6906	2276	872	134
Temp.+ Volt.		7291	1371	59	334
Humid.+ Light		6690	4925	872	171
Humid.+ Volt.		7076	4298	57	171
Light. + Volt.		3362	1345	848	316
Temp. + Humid. + Light		8342	5042	884	246
Temp.+Light+Volt		7470	2276	872	390
Humid.+Light+Volt		7336	4925	872	429
Temp.+Humid.+Light+Volt		8529	5042	884	502

<b>MoteID</b>	18
<b>DataSet</b>	8923 paquets

(e)

Parameters	Algorithm	ES	AD	RD	CI
Temp. (T)		6082	1007	54	47
Humid. (H)		5734	3361	41	63
Light. (L)		1668	1600	914	161
Volt. (V)		2371	12	3	2
Temp.+ Humid.		7786	3493	63	108
Temp.+ Light.		6582	2239	939	205
Temp.+ Volt.		6821	1009	55	48
Humid.+ Light		6272	4262	936	221
Humid.+ Volt.		6583	3361	42	221
Light. + Volt.		3604	1603	914	162
Temp. + Humid. + Light		7986	4359	942	264
Temp.+Light+Volt		8093	3493	64	264
Humid.+Light+Volt		6999	4262	936	222
Temp.+Humid.+Light+Volt		8250	4359	942	265

<b>MoteID</b>	20
<b>DataSet</b>	9582 paquets

(f)

Parameters	Algorithm	ES	AD	RD	CI
Temp. (T)		6831	1393	64	84
Humid. (H)		6510	4171	73	137
Light. (L)		1823	1823	1266	111
Volt. (V)		2125	17	10	12
Temp.+ Humid.		8467	4297	84	217
Temp.+ Light.		7256	2735	1290	194
Temp.+ Volt.		7415	1393	70	93
Humid.+ Light		6957	4953	1288	246
Humid.+ Volt.		7191	4171	75	246
Light. + Volt.		3561	1825	1255	122
Temp. + Humid. + Light		8603	5053	1295	326
Temp.+Light+Volt		7749	2735	1290	203
Humid.+Light+Volt		7543	4953	1288	255
Temp.+Humid.+Light+Volt		8820	5053	1295	335

Nous représenterons ci-après les graphiques du tableau 5, afin d'avoir une comparaison claire des résultats obtenus dans les 6 motes avec les différents algorithmes comprenant toutes les combinaisons possibles entre les capteurs d'un mote.

		MoteID = 1			
		8843			
		Type of algorithm applied			
Sensor on Mote:	DataSet	ES	AD	RD	CI
1- Temperature	Temp. (T)	6679	1706	62	105
	Humid. (H)	6596	4498	54	91

Total number of data captured (Without any algorithm)

Number of data transmitted with the "ES" algorithm

Number of data transmitted by applying the "CI" algorithm

Fig. 9 Détail sur la lecture des tables

**Remarque importante :**

Lorsque l'un des capteurs du mote a des relevés assez élevés (exemple : le capteur de lumière), l'écart des algorithmes de la Différence Relative « RD » et ceux de la

Différence Absolue « AD » n'est pas important. Ceci est d'autant plus vrai lorsque l'on sait qu'une valeur de 1 lux correspond au clair de lune, 400 lux à un bureau lumineux et 100 000 lux au plein soleil [20]. (Voir les histogrammes ci-dessous)

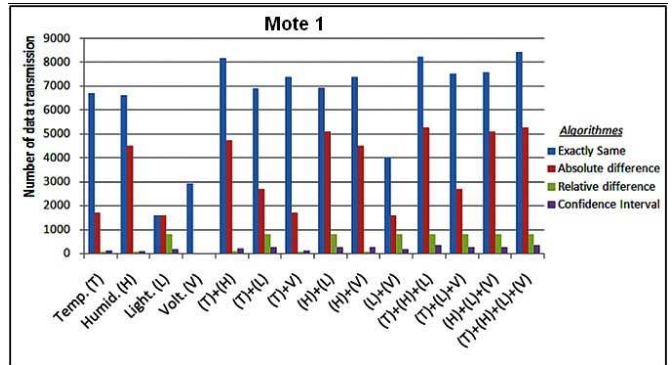


Fig. 10 Comparaison des résultats du Mote 1 avec les 4 capteurs.

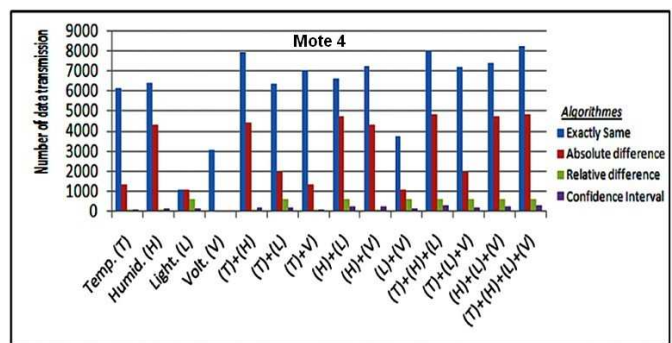


Fig. 11 Comparaison des résultats du Mote 4 avec les 4 capteurs

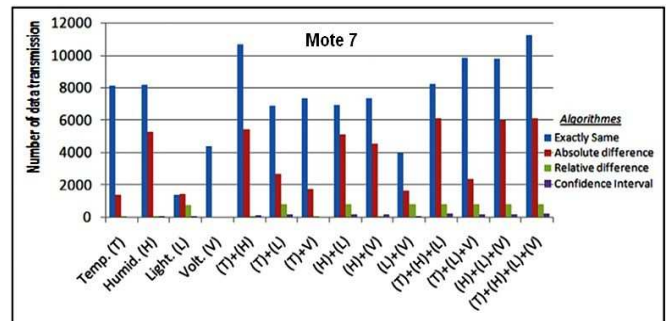


Fig. 12 Comparaison des résultats du Mote 7 avec les 4 capteurs

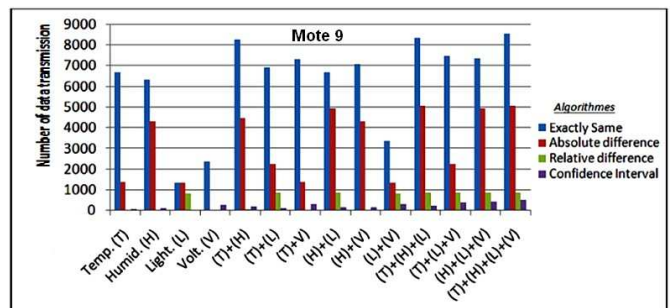


Fig. 13 Comparaison des résultats du Mote 9 avec les 4 capteurs



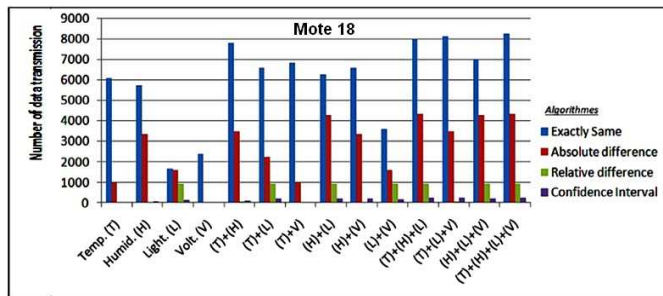


Fig. 14 Comparaison des résultats du Mote 18 avec les 4 capteurs

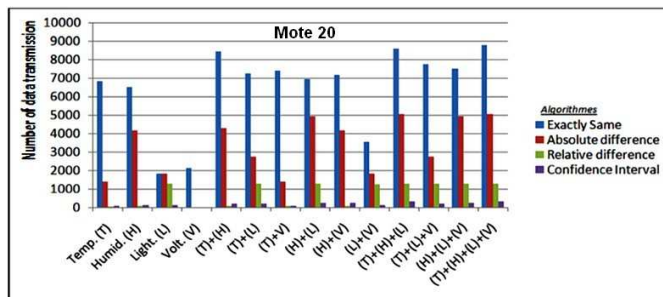


Fig. 15 Comparaison des résultats du Mote 20 avec les 4 capteurs

Le tableau ci-dessous engendre les valeurs issues du tableau 6 avec des motes comprenant les 4 capteurs réunis, déployés sur une architecture basée sur l’IoT, illustrant ainsi la consommation d’énergie pendant une durée de deux mois. Ces valeurs calculées se présentent à titre d’exemple :

Mote 1 (avec l’algorithme « ES ») : nous enregistrons 8411 transmissions, soit :

$$E = 8411 \times 59,2 \times 2 = 9,958624 \times 10^5$$

TABEAU 6

CONSOMMATION D’ENERGIE PAR MOTE

Algo. / Motes	ES	AD	RD	CI
Mote 1	9,958624	3,107408	0,474784	0,204
Mote 4	4,874528	2,854032	0,356384	0,154
Mote 7	6,652896	3,601136	0,45584	0,153
Mote 9	10,098336	5,969728	1,046656	0,594
Mote 18	9,768	5,161056	1,115328	0,314
Mote 20	10,44288	5,982752	1,53328	0,397

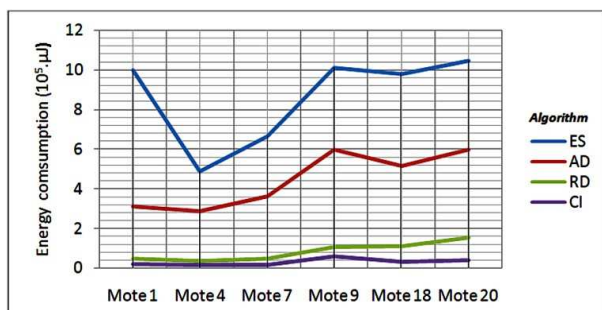


Fig. 16 Energie consommée sur chaque mote avec application des 4 algorithmes

L’algorithme « Sensoring », basé sur les intervalles de confiances, affiche une grande performance dans les résultats et par conséquent une économie d’énergie beaucoup plus importante. Pour ce faire, laissons la valeur de la taille de l’entête inchangée et reprenons la valeur du coût d’envoi d’un octet à 59,2 μJ (comme défini dans [8], calculé pour le mote MICA2Dot) ; ainsi la taille des données obtenues (payload) vaudra : 2 octets = 2 x 8 = 16 bits pour les 4 capteurs d’un mote (voir Fig.5), face aux 64bits de [8]. Cette taille influera largement et positivement sur la durée d’émission du paquet pour une mise à jour.

L’analyse des résultats confirme donc l’hypothèse que nous avons proposée, à savoir que le programme « Intervalle de confiance » consomme moins d’énergie, comparé aux autres algorithmes. Cette performance peut être exploitée pour prolonger au maximum la durée de vie d’un réseau basé sur l’IoT.

### VII. CONCLUSION & PERSPECTIVE

La collecte d’informations étant une opération capitale dans un réseau IoT, la durée de vie de ce dernier dépend essentiellement de l’énergie de ses capteurs. Toutes les recherches y afférant montrent que la consommation de cette énergie est tributaire, en premier lieu, de la transmission des données et, en second lieu, du traitement de ces données.

Dans la présente étude, notre but a été de cerner la question d’émission des données vers la station de base pour une éventuelle mise à jour. Plusieurs approches ont été élaborées dans ce sens parmi lesquelles la perspective de l’« Intervalle de confiance », inspirée du modèle de Markov cachée, nous a semblé la plus efficiente. Celle-ci repose sur la théorie des niveaux « observables » : les intervalles de confiances, appelés également classes et les niveaux « cachés » qui sont les valeurs effectives captées.

La méthode utilisée met alors en évidence un certain nombre d’intervalles (ou classes) prédéfinies par la station de base. Ces intervalles sont une discrétisation des données dans les conditions d’acceptabilité des valeurs captées, l’amplitude étant la même dans l’ensemble des dites classes. Le nombre de classes ainsi fixé, l’alerte est donnée au-delà de ce seuil (comme défini à la section 5).

Dans notre approche, nous avons intervenu sur trois (03) niveaux :

- Niveau 1 : Diversité des capteurs sur un seul mote,
- Niveau 2 : Taille du paquet.
- Niveau 3 : Emission,

La taille du paquet est réduite du fait qu’elle contient seulement le numéro de l’intervalle de confiance, exemple trois (03) bits suffisent pour coder 8 classes.

L’application, ainsi réalisée, a confirmé notre hypothèse et le nombre de transmissions de données est réduit au maximum. Une perspective d’étude serait de voir quelles valeurs seraient transmises au-delà des intervalles de confiance, en déployant une méthode hybride y associant l’algorithme « Relative difference » par exemple.

## REFERENCES

- [1] K. Manikandan, P. Kanmani, and M. Munnira Sulthana, "Energy Efficient Algorithms for Wireless Sensor Network", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 1, January 2015
- [2] Diery NGOM, "Optimisation de la durée de vie dans les réseaux de capteurs Sans fil sous contraintes de couverture et de connectivité", these, University of Upper Alsace, France, 2016.
- [3] M. Bourgoaoui, HBA. Sethom and IS. Belkhodja, "Speed/position sensor fault tolerant control in adjustable speed drives-A review", ISA Trans: 64, pp. 269–284. Sep. 2016.
- [4] K. Amandeep and K. Kamaljit, "A Review of Different Energy Efficiency Techniques in Wireless Sensor Networks"; International Journal of Advanced Research in Computer Science and Software Engineering; ISSN: 2277 128X, Vol. 5, Issue 6, June 2015.
- [5] A. Ghaddar, T. Razafindralambo, I. Simplot-Ryl, D. Simplot-Ryl, S. Tawbi and A. Hijazi, "Investigating Data Similarity and Estimation Through Spatio-Temporal Correlation to Enhance Energy Efficiency in WSNs". Ad Hoc & Sensor Wireless Networks, PKP Publishing Services Network, vol. 16, Issue 4, pp. 273–295,
- [6] L. Jiang, A. Liu, Y. Hu, and Z. Chen, "Lifetime maximization through dynamic ring-based routing scheme for correlated data collecting in WSNs," Computers & Electrical Engineering, vol. 41, pp. 191–215. Jan. 2015
- [7] He. Hua, Xu Zhen and Lei Yang, "Delay-aware data collecting protocol for low-duty-cycle wireless sensor networks", IET Networks, vol. 7, Issue: 1, pp. 44–49. Jan. 2018
- [8] N.A.M. Alduais, J. Abdullah, A. Jamil and L. Audah, "An efficient data collection and dissemination for IOT based WSN". In: IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), pp. 1–6. Oct. 2016
- [9] J.Y. Kim, C.H. Chu, and S.M. Shin, "ISSAQ: An Integrated Sensing Systems for Real-Time Indoor Air Quality Monitoring", IEEE Sensors Journal, vol. 14, Issue 12, pp. 4230-4243. Dec. 2014.
- [10] X. Tang and J. Xu, "Adaptive data collection strategies for lifetime-constrained wireless sensor networks". Parallel and Distributed Systems, IEEE Transactions, vol. 19, Issue 6, pp. 721–734. Juin 2008.
- [11] W. Heinzelman, "Application-specific protocol architectures for wireless networks," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, 2000.
- [12] W.B. Heinzelman, A.P. Chandrakasan and H. Balakrishnan, "Application-Specific Protocol Architectures for Wireless Networks", In IEEE Transactions on Wireless Communications, vol. 1, Issue 4, pp. 660–670. Dec. 2002.
- [13] S. Lindsey and C. Raghavendra, "PEGASIS Power-Efficient Gathering in Sensor Information Systems,"IEEE Aerospace Conference Proceedings, vol. 3, pp. 1125–1130. March 2002.
- [14] A. Boukerche, X. Cheng and J. Linus: Energy-Aware Data-Centric Routing in Microsensor Networks. MSWIM '03, Proceedings of the 6th ACM International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems, pp. 42–49. Sept. 2003.
- [15] May Mon Khaing, and Tun Min Naing, "Energy Aware Data-Centric Routing in Wireless Sensor Network", International Conference on Advances in Engineering and Technology (ICAET'2014), Singapore, pp. 110-114. March 2014.
- [16] A. BOUDRIES, "Maintien de la Connectivité dans les Réseaux Ad hoc sans l", Thesis, (published). Ferhat Abbas Université, Sétif 1, Algeria, 2014.
- [17] (2017), Wikiversity website, "Capteur : Introduction ", [Online].Available: <https://fr.wikiversity.org/wiki/Capteur/Introduction> last Update. 01/08/2017. Accessed : 27- Jan- 2018.
- [18] (2016), Libelium website. Waspnote, Data sheet | Libelium: Document Version: v7.1 - 10/2016, [Online]. Available: <http://www.libelium.com/development/waspnote/documentation/> Accessed: 27- 01- 2018.
- [19] (2018), The Wikipedia website. "Relative change and difference", [Online].Available: [https://en.wikipedia.org/wiki/Relative\\_change\\_and\\_difference](https://en.wikipedia.org/wiki/Relative_change_and_difference) Accessed: 27- 01- 2018.
- [20] (2014), Intel Lab Data website, Last modified: 06/2014, [Online]. Available: <http://db.csail.mit.edu/labdata/labdata.html>. Accessed: 2018- 02- 13
- [21] Prabhanjan N. Tattar, Suresh Ramaiah, BG Manjunath, "A Course in Statistics with R", Willey, Book, 2016.
- [22] (2017), The Magrit website. "Discretiser sur combien de classes ? ". [Online]. Available: <https://magrit.hypotheses.org/author/bmirou>. Accessed: 2018- 02- 13