



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-memoires-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Evaluation de l'impact du mécanisme du « duty-cycle » sur la qualité de service

Serigne Makhtar Dioum

Université de Lorraine

Campus Sciences, BP 70239F - 54506 Vandoeuvre-les-Nancy Cedex, FRANCE

serigne-makhtar.dioum.1@etumail.uhp-nancy.fr

Résumé— La radio est le composant le plus gourmand en énergie pour un nœud capteur sans fil, diminuer le pourcentage de temps que celle-ci reste active en utilisant des périodes de sommeil permet de minimiser la consommation d'énergie du nœud et ainsi, augmenter la durée de vie du réseau de capteurs sans fil. Néanmoins, effectuer du routage sur ce type de réseau reste difficile à cause de l'indisponibilité des nœuds pendant leur phase de sommeil. De nombreux protocoles de routage ont été proposés pour les réseaux de capteurs sans fil, mais rare sont ceux qui prennent en compte des périodes de sommeil des nœuds. Ce type de politique de mise en veille du composant radio, appelée "low duty-cycle", est pourtant omniprésente dans de nombreux protocoles MAC tels que le B-MAC, X-MAC ou le ContikiMAC. Il est donc important de pouvoir évaluer l'impact d'un protocole MAC de ce type sur les performances du routage afin de fournir une certaine garantie de la QoS, notamment en ce qui concerne le délai de bout en bout. Nous évaluons dans cet article l'impact du duty-cycle sur le routage en utilisant le protocole RPL de l'IETF ROLL sur ContikiMAC et nous proposons un modèle de simulation permettant d'effectuer un choix judicieux de la valeur du duty-cycle qui va satisfaire les besoins en termes de qualité de service et d'économie d'énergie.

Mots-clés— Réseaux de capteurs sans fil; Routage; Duty-cycle; Evaluation de performance; Simulation; Consommation d'énergie.

I. INTRODUCTION

Les réseaux de capteurs sans fil sont aujourd'hui utilisés dans notre quotidien pour diverses applications notamment domotiques. L'avantage d'une telle utilisation reste le coût faible d'installation et de maintenance d'un système de réseaux de capteurs sans fil. Au-delà de son faible coût, un réseau de capteurs sans fil répond à un ensemble de critères qui favorisent son utilisation de plus en plus fréquente pour le contrôle et la commande de systèmes à savoir son observabilité, son monitorabilité, sa contrôlabilité et la facilité de son déploiement. Néanmoins, l'utilisation des réseaux de capteurs sans fil pour certaines applications peut être inadaptée à cause de la durée de vie limitée de la batterie des nœuds qui composent le réseau. Il est donc nécessaire de maximiser cette durée de vie tout en respectant les exigences de qualité de service pour étendre le domaine d'application des réseaux de capteurs sans fil.

Pour acheminer un paquet jusqu'à sa destination, le protocole de routage détermine le prochain saut dans un premier temps en utilisant une métrique de routage et des estimations de lien, puis dans un deuxième temps, le protocole MAC attend le réveil du prochain saut choisit afin qu'il puisse recevoir avec succès le paquet. Plusieurs travaux se focalisent sur ces deux étapes fondamentales pour proposer des politiques de minimisation de la consommation d'énergie et du délai.

Au niveau de la couche MAC, on diminue le pourcentage de temps que la radio reste allumée (duty-cycle) pour minimiser sa consommation d'énergie en utilisant des instants de réveil qui peuvent être indépendants ou synchronisés par rapport au instants de réveil des nœuds voisins. Avec la première approche, un émetteur qui ne connaît pas la prochaine date de réveil du récepteur, doit transmettre un préambule avant la transmission du paquet de donnée afin que le récepteur reste réveillé pour réceptionner le paquet. Alternativement, le récepteur peut transmettre des paquets sonde à chaque réveil pour signaler à un probable émetteur qu'il est réveillé. On peut citer quelques protocoles de couche MAC qui implémentent ce mécanisme de réveils opportunistes notamment le X-MAC [1], ContikiMAC [3], le B-MAC [23] et le RI-MAC [25].

Avec l'approche de la synchronisation, les nœuds sont réveillés à des dates spécifiques, ainsi, l'émetteur sait à quelle date un potentiel récepteur se réveillera. Des exemples de mécanisme de « duty-cycle » avec synchronisation sont le S-MAC [26] et le TSMP [27]. Des techniques d'optimisations ont été proposées dans le cas où il n'y a pas de synchronisation, on peut noter la technique du verrouillage de phase dans le cas du ContikiMAC [3] qui consiste à retarder la transmission de paquet jusqu'à l'approche de la date de réveil du récepteur. Dans [2], une méthode d'optimisation présentée est d'utiliser des protocoles cotés récepteur comme fonction du duty-cycle des récepteurs et leur distance par rapport au nœud puits. L'analyse de ces protocoles a permis d'élaborer un modèle qui permet d'obtenir une valeur du duty-cycle qui minimise la consommation d'énergie pour une distance donnée [2].

Au niveau réseau, plusieurs protocoles sont proposés pour effectuer du routage dans les réseaux de capteurs sans fil où on a un ensemble de contraintes de ressources telles que la puissance de calcul, la taille de la mémoire, la quantité

d'énergie stockée et la bande passante de communication. Pour acheminer un paquet, une manière simple de procéder est d'utiliser le routage opportuniste comme le protocole ORW [17] qui consiste à faire une diffusion générale du paquet et ainsi utiliser comme prochain saut le nœud voisin qui se réveille en premier. Mais la plupart des protocoles de routage utilisent comme prochain saut un nœud voisin spécifique en se basant sur une métrique de routage comme les protocoles RPL [22], SAR [4], MCMP [4] et Eres-QoS [4], ce qui peut inclure un certain délai du fait que la couche MAC doit attendre le réveil du nœud choisit comme prochain saut pour transmettre le paquet. Il est donc nécessaire d'évaluer les performances du protocole de routage qui ne procède pas de manière opportuniste sur un réseau qui fonctionne avec le mécanisme du duty-cycle. Ces performances peuvent être évaluées en terme de :

- Délais des paquets.
- Tolérance aux fautes.
- Rapidité de mise à jour de la table de routage.
- Rapidité de propagation d'une route dans le réseau.
- Charge de trafic.

Beaucoup de travaux ont été menés pour évaluer les performances des protocoles de routage dans les réseaux de capteurs sans fil comme dans [6], [11] et [14] où une évaluation de performance du protocole RPL est effectuée. Toutefois, ces évaluations ne prennent pas en compte l'impact du choix des périodes de réveil et de sommeil des nœuds dans le réseau. Dans [1], les auteurs évaluent l'impact du choix de ces périodes sur le délai par saut des paquets, le taux de perte paquet et sur la consommation d'énergie. Ils proposent aussi un modèle permettant de choisir ces périodes de manière optimale par rapport à la charge de trafic de données et la densité du réseau.

Notre évaluation met en évidence l'impact du mécanisme du duty-cycle sur le délai de bout en bout, la consommation d'énergie du réseau et le comportement du protocole de routage par rapport à la maintenance de la topologie du réseau.

Le reste de cet article est structuré comme suit. La section II présente le mécanisme de ContikiMAC. La section III présente le protocole RPL et son principe de fonctionnement. La section IV évalue l'impact du mécanisme du « duty-cycle » sur, la consommation d'énergie, le délai de bout en bout des paquets et les paramètres du protocole RPL. La section V conclut cet article.

II. LE PROTOCOLE CONTIKIMAC

Le protocole ContikiMAC [3] a été développé par le Swedish Institute of Computer Science (SICS). Il permet à un nœud d'éteindre le composant radio et d'utiliser des réveils périodiques pour écouter le médium afin de détecter la transmission de paquets de ses voisins ou de vérifier s'il n'a pas de paquet en attente de transmission. Si une transmission de paquet est détectée pendant cette phase de réveil, le récepteur reste réveillé pour pouvoir recevoir le paquet lorsque celui-ci lui est destiné sinon il se remet à l'état sommeil. Lorsque le paquet est reçu avec succès, le récepteur envoie un acquittement. Pour transmettre un paquet, un

émetteur envoie de manière répétée (strobing) le paquet jusqu'à ce qu'il reçoit un acquittement du récepteur, sauf pour les paquets envoyés en diffusion générale. Ces derniers sont émis de manière répétée sur tout l'intervalle de réveil afin de s'assurer que tous les voisins l'ont reçu.

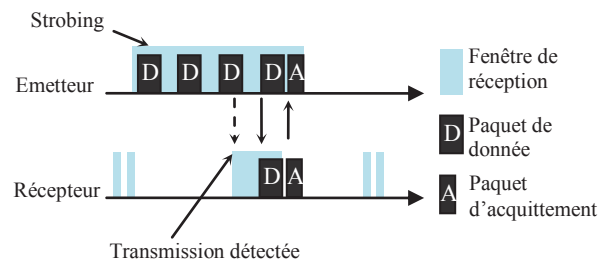


Fig. 1 Opération basique de transmission et réception de paquet entre deux nœuds avec le mécanisme du « duty-cycle » du protocole ContikiMAC [3].

L'objectif de ce mécanisme est donc d'endormir le nœud pendant une partie du temps afin d'économiser l'énergie embarquée et d'éviter les collisions qui peuvent subsister dans le réseau lorsque des nœuds proches émettent en même temps. La stratégie d'évaluation du canal de ce protocole repose sur des réveils, Channel Check Assessment (CCA), successifs du nœud. Chaque CCA est suivi d'une phase de sommeil.

Pour optimiser sa transmission, le protocole ContikiMAC utilise la technique du verrouillage de phase qui consiste à estimer la date de réveil du prochain saut pour une destination donnée à partir de la date de réception d'acquiescement et de retarder le moment d'émission de paquets jusqu'à l'approche de cette date de réveil. Cette technique permet de réduire la durée de la phase de strobing donc d'économiser de l'énergie et d'éviter la surcharge du réseau. Toujours dans l'objectif d'augmenter la durée de la période de sommeil, le protocole ContikiMAC utilise la technique du « sommeil-rapide » qui permet de distinguer une transmission de paquet et le bruit en procédant comme suit, lorsqu'une activité de la radio est détectée lors de l'évaluation du canal, la durée de cette activité est mesurée et le nœud retourne à l'état sommeil si la période de l'activité est :

- supérieure à la longueur maximale d'un paquet,
- suivie d'une période de silence plus longue que l'intervalle entre deux transmissions successives,
- suivie d'une période de silence de longueur correcte, suivie d'une activité, mais que le début d'un paquet n'est pas détecté.

III. LE PROTOCOLE RPL

RPL (routing protocol for low power and lossy network) [22] est un protocole de routage IPv6 basé sur le vecteur de distance. Il a été développé par le groupe de travail IETF ROLL pour les réseaux à faible consommation d'énergie déployés dans des environnements tels que les réseaux urbains,

les réseaux domotiques, les réseaux industriels [21]. Ce protocole est optimisé pour les réseaux de collecte de données qui sont basés sur trois types de trafics : Point-à-Point, Point-à-Multipoint et Multipoint-à-Point.

Le fonctionnement de RPL repose sur la construction d'un DODAG (Destination Oriented Directed Acyclic Graph) à partir du nœud source (puits). Le DODAG est construit à l'aide de messages DIO (DODAG Information Object) générés par le nœud source du DODAG. Un message DIO contient un ensemble d'information tel que l'ID du DODAG, l'instance de RPL, la version du DODAG, la fonction objective, le rang du nœud diffuseur du message. Ceci va permettre aux nœuds récepteurs du message de calculer leur rang. Plus le nœud est éloigné du nœud source, plus son rang augmente.

Le rôle du DODAG est de minimiser le coût pour atteindre le nœud source à partir de n'importe quel nœud dans le réseau. Le routage vers la source du DODAG est basé sur les valeurs du rang, un nœud choisit comme parent le voisin avec le rang le plus petit lorsqu'il envoie un paquet vers la source du DODAG. Pour maintenir le DODAG, les nœuds dans un réseau RPL échangent périodiquement des messages DIO. La transmission de ces messages est régulée par un temporisateur appelé « trickle timer » [28] pour éliminer les messages de contrôle redondants. Dans l'algorithme de ce temporisateur, le temps est divisé en une infinité d'intervalles de taille I . Le nœud peut transmettre un nouveau message DIO à n'importe quel instant dans la seconde moitié de chaque intervalle. A la fin de chaque intervalle, I est doublée jusqu'à M fois à partir d'une valeur minimale I_m et donc sa taille est comprise dans l'intervalle $[I_m, 2^M I_m]$. Lorsqu'un problème est détecté, perte de parent par exemple, le temporisateur est remis à la valeur I_m . La transmission de message DIO est aussi conditionnée par un seuil δ_{RC} , le nombre de messages de contrôle émis depuis le début de l'intervalle ne doit pas dépasser ce seuil. Cet échange périodique de message DIO permet de faire le routage vers la source du DODAG. Un autre type de message échangé est le DAO (Destination Advertisement Object) qui est envoyé en unicast vers la source du DODAG par chaque nœud dans le réseau. Le contenu des messages DAO permet d'effectuer du routage vers le bas, c'est-à-dire que la source du DODAG connaît les chemins pour atteindre n'importe quel nœud du réseau. Un autre type de message RPL est le DAO ACK (Destination Advertisement Object Acknowledgement) qui est optionnel.

Un nœud qui rejoint le réseau peut envoyer en multicast un message DIS (DODAG Information Solicitation) pour déclencher l'émission de messages DIO par ses voisins qui sont déjà dans le réseau RPL pour intégrer un DODAG. Ceci est donc une alternative à l'attente de réception de messages DIO.

IV. EVALUATION DE PERFORMANCE

Nous évaluons l'impact du mécanisme du « duty-cycle » sur le délai de bout en bout des paquets, sur la consommation d'énergie du réseau et sur le temps de configuration du réseau.

A. Environnement de simulation

Nous utilisons le simulateur Cooja [24] pour l'évaluation. Ce simulateur est disponible dans Contiki qui est un système d'exploitation destiné aux réseaux de capteurs sans fil. Ce système d'exploitation dispose du protocole ContikiRPL qui est une implémentation fiable du protocole RPL [22] par le Swedish Institute of Computer Science (SICS) qui a aussi développé et implémenté le protocole ContikiMAC. Nous utilisons la plateforme sky qui est une émulation des nœuds capteurs TelosB. Le modèle d'énergie utilisé par le système d'exploitation Contiki prend en compte la consommation d'énergie des états écoute, transmission et repos du composant radio.

Nous faisons l'évaluation sur un réseau composé de 48 nœuds sources et un nœud puits déployé dans un environnement de 200m x 200m.

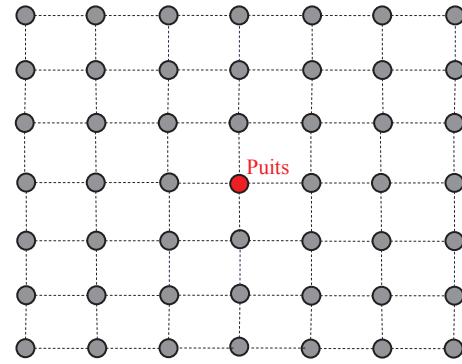


Fig. 2 Topologie du réseau avec 48 nœuds sources et un puits.

B. Paramètres de simulation

Le débit d'émission de paquets de données de taille 77 octets est fixé à un paquet par minute pour tous les nœuds sources, chaque nœud sources génère 100 paquets de données dans le réseau.

Le choix des paramètres du protocole ContikiMAC est effectué en respectant les recommandations de [3].

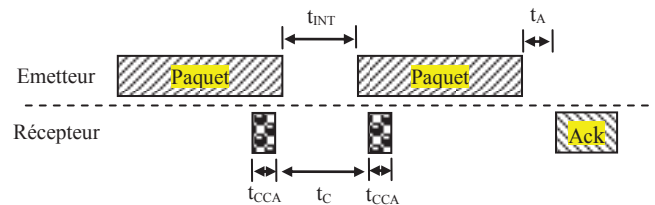


Fig. 3 Paramètres temporels du protocole ContikiMAC

- t_{INT} : l'intervalle de temps entre chaque transmission de paquet durant le strobing est fixée à 0.4ms.

- t_{CCA} : la durée d'évaluation du canal requise pour vérifier s'il n'y a pas de paquet en attente de transmission ou de réception. La valeur de ce paramètre est donnée par la documentation du driver CC2420 de la radio et est fixée à 0.192ms.
- t_C : l'intervalle entre chaque CCA est fixée à 0.5ms.
- t_A : le temps entre la réception de paquet et la transmission de l'acquittement est de 0.192 ms.

Aucun mécanisme d'optimisation du protocole ContikiMAC n'est utilisé.

Au niveau du routage, le choix de la métrique est assez important, elle doit refléter l'impact du choix de la valeur des périodes de réveil et de sommeil des nœuds. Nous avons choisi d'utiliser le taux de retransmission (ETX) comme métrique de routage. La valeur des paramètres du « trickle timer » sont fixés comme suit : $I_m = 4.1s$, $M = 8$ et $\delta_{RC} = 10$.

A. Consommation d'énergie

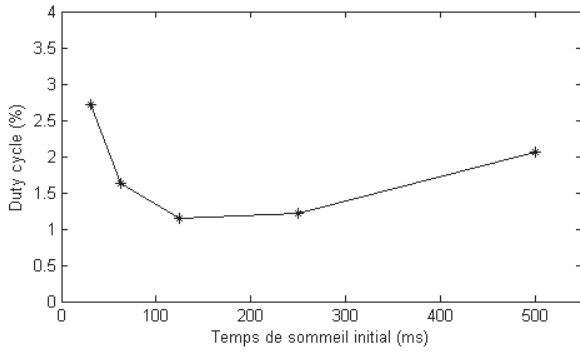


Fig. 4 Duty cycle moyen de tous les nœuds en fonction du temps de sommeil initial.

Augmenter le temps de sommeil initial des nœuds ne permet pas toujours de minimiser le pourcentage du temps que la radio reste à l'état réveil. En effet, lorsqu'un récepteur utilise des périodes de réveil et de sommeil comme politique d'économie d'énergie, un probable émetteur transmet le paquet de manière répétée jusqu'à la réception d'un acquittement et donc plus la période de sommeil est grande, plus la durée de l'activité de la radio augmente. Ceci peut se constater sur la figure 4.

En se basant sur la figure 1, on peut exprimer la quantité d'énergie E_{Tx} requise pour transmettre un paquet dans le cas du protocole ContikiMAC pour une communication point à point. Celle-ci est donnée par l'expression suivante:

- N_p : le nombre de paquet émis durant la phase de strobing.
- t_{RxACK} : le temps de réception de l'acquittement.
- t_{TxDATA} : le temps d'émission d'un paquet de donnée.
- P_{Rx} et P_{Tx} : respectivement la puissance requise pour recevoir et pour transmettre.

- R_r et R_s : les périodes de réveil et de sommeil du récepteur.

Soit S , la durée de la phase de strobing et I le nombre d'itérations de l'écoute du strobing requis:

$$S = t_{CCA} + N_p t_{INT}$$

Il est vérifié dans [1] que I peut s'exprimer comme suit :

$$I = \left(\frac{1}{\left(\frac{R_r - S}{R_r + R_s} \right)} \right)$$

$$E_{Tx} = [(S + t_A + t_{RxACK})P_{Rx} + N_p P_{Tx} t_{TxDATA}] * I$$

Dans le cas d'une diffusion générale, les paquets émis ne sont pas acquittés [3]. Au lieu d'un acquittement, l'émetteur transmet le paquet de manière répétée sur toute la durée de l'intervalle de réveil et donc l'énergie requise ne dépend que de la durée de cet intervalle. Il est donc nécessaire de diminuer le nombre de paquets émis en diffusion générale tels que les messages DIO et DIS du protocole RPL.

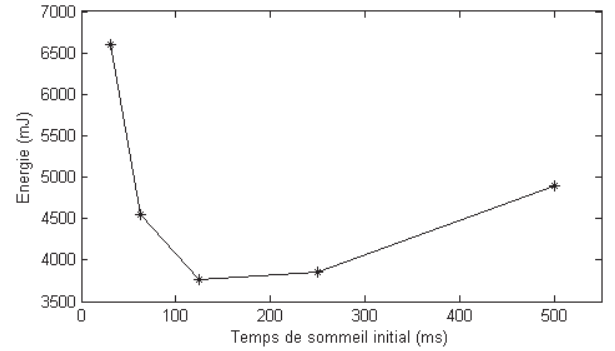


Fig. 5 Consommation d'énergie moyenne dans le réseau.

B. Délai de bout en bout

Dans [1], les auteurs ont exprimé le délai pour un saut dans un réseau qui utilise le mécanisme du duty-cycle comme suit:

$$d_{1saut} = \left(\frac{1}{\left(\frac{R_r - S}{R_r + R_s} \right)} \right) (S + t_{RxACK}) + t_{TxDATA}$$

Soit un nœud i le prochain saut du nœud j , en considérant un nombre n de nœuds qui composent un chemin, on peut exprimer le délai de bout en bout D des paquets tel que :

$$D = \sum_{j=1}^n \left[\left(\frac{1}{\left(\frac{R_{ir} - S_j}{R_{ir} + R_{is}} \right)} \right) (S_j + t_{RxACK}) + t_{TxDATA} + t_{fj} \right]$$

- t_{fj} : le temps d'attente dans la file de transmission du nœud j.

La figure 6 montre d'une part l'impact de la période de sommeil sur le délai et d'autre part le délai induit par le temps d'attente des paquets dans la file de transmission dû à la charge de trafic.

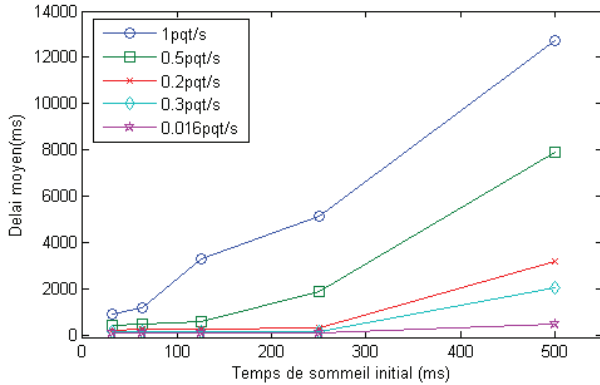


Fig. 6 Délai moyen de bout en bout des paquets.

C. Temps de configuration

Nous évaluons ici le temps de configuration du DODAG, c'est-à-dire le temps mis pour que les tables de routage vers la source du DODAG et vers les nœuds du réseau convergent. Ce temps est déterminé à partir de la date d'émission du premier message DIO par la source du DODAG.

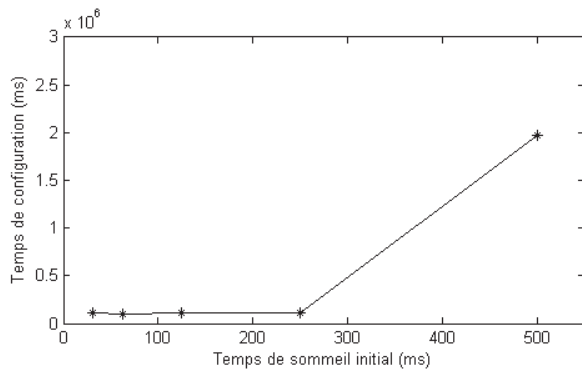


Fig. 7 Temps de configuration en fonction du temps de sommeil initial.

Une construction rapide du DODAG d'un réseau de capteurs sans fil permet de minimiser les pertes de paquets qui peuvent subsister au démarrage. Tous les paquets générés par

un nœud avant que celui-ci intègre un DODAG ne seront pas émis. Le nœud peut émettre en diffusion générale des messages DIS pour solliciter la transmission de message DIO de ces voisins, ce qui lui permettra d'intégrer le DODAG. Comme on le voit sur la figure 7, la période de sommeil peut retarder la réception de message DIO, ce qui fait que le nœud émet des messages DIS donc augmente la charge de trafic du protocole RPL comme on peut le voir sur la figure 8.

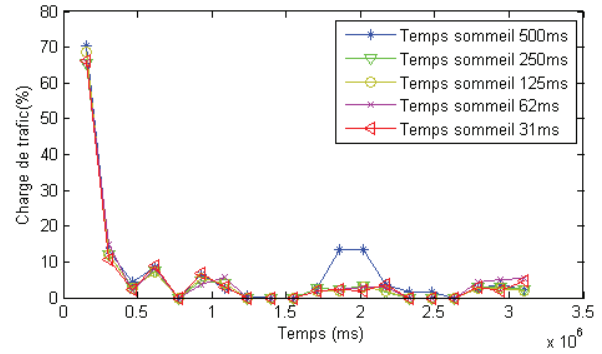


Fig. 8 Charge de trafic du protocole RPL pour différents temps de sommeil.

V. CONCLUSIONS

Dans cet article nous avons présenté l'évaluation de l'impact du mécanisme du duty-cycle sur la qualité de service. Les résultats montrent que le fait d'endormir les nœuds dans un réseau de capteurs sans fil introduit un certain délai lors de l'acheminement des paquets et peut augmenter considérablement le temps de configuration du réseau. Cette période de sommeil ne permet pas toujours d'économiser de l'énergie lorsqu'il n'y a pas de mécanisme d'optimisation pour diminuer la durée du strobing.

REFERENCES

- [1] M. Buettner, G. V. Yee, E. Anderson, R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks", *Sensys'06*, November 1-3, 2006, Boulder, Colorado, USA.
- [2] Y. Zhang, C. Feng, I. Demirkol, W. B. Heinzelman, "Energy-Efficient Duty Cycle Assignment for Receiver-Based Convergecast in Wireless Sensor Networks", *Global Telecommunications Conference (GLOBECOM 2010)*, 2010 IEEE, 6-10 Dec. 2010, Page 1 – 5.
- [3] Adam Dunkels, "The ContikiMAC Radio Duty Cycling Protocol", SICS Report, 2011.
- [4] A. Ghaffari, H. R. Banaeian, "QoS-Based Routing Protocols for Wireless Sensor Networks: A Survey", *World Applied Sciences Journal* 14(6): 866-875, 2011.
- [5] A. Paradisi, M.P. Uwase, J. Tiberghien, K. Steenhaut, J. M. Dricot, "Didactic simulators for understanding routing protocols in wireless sensor networks", *ICERI2011 Proceedings*, 4th International Conference of Education, Research and Innovations, 14-16 November, 2011, Madrid, Espagne, Page 5807-5814.
- [6] I. E. Radoi, "Performance Evaluation of Routing Protocols in Lossy Links for Smart Building Networks", <http://tools.ietf.org/html/draft-tripathi-roll-rpl-simulation-08>.
- [7] Shu Du, Amit Kumar Saha, David B. Johnson, "RMAC: A Routing-Enhanced Duty-Cycle MAC Protocol for Wireless Sensor Networks", *26th IEEE International Conference on Computer Communications*, 6-12 May 2007, Page 1478 – 1486.

- [8] Richard Draves, Jitendra Padhye, Brian Zill, "Comparison of Routing Metrics for Static Multi-Hop Wireless Networks", conference on Applications, technologies, architectures, and protocols for computer communications, Pages 133-144, New York, USA, 2004.
- [9] Nicolas Tsiftes, Joakim Eriksson, Niclas Finne, Fredrik Österlind, Joel Höglund, Adam Dunkels, "A Framework for Low-Power IPv6 Routing Simulation, Experimentation, and Evaluation", ACM SIGCOMM 2010 conference, Pages 479-480, New York, USA 2010.
- [10] Bogdan Pavković, Fabrice Theoleyre, Andrzej Duda, "Multipath Opportunistic RPL Routing over IEEE 802.15.4", 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems, Pages 179-186, New York, USA 2011.
- [11] Yibo CHEN, Jean-Pierre CHANET, Kun Mean HOU, "RPL Routing Protocol a Case Study: Precision Agriculture", First China-France Workshop on Future Computing Technology (CF-WoFUCT 2012), Page 6, 2012.
- [12] Adam Dunkels, Joakim Eriksson, Niclas Finne, Nicolas Tsiftes, "Powertrace: Network-level Power Profiling for Low-power Wireless Networks", March 2011 SICS Technical Report T2011:05 ISSN 1100-3154.
- [13] Nicolas Tsiftes, Joakim Eriksson, and Adam Dunkels, "Low-Power Wireless IPv6 Routing with ContikiRPL", 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, Pages 406-407, New York, USA, 2010.
- [14] Leila Ben Saad, Cedric Chauvenet, Bernard Tourancheau, "Simulation of the RPL Routing Protocol for IPv6 Sensor Networks: two cases studies", International Conference on Sensor Technologies and Applications SENSORCOMM 2011, 2011.
- [15] Simon Duquennoy, Fredrik Österlind, Adam Dunkels, "Lossy Links, Low Power, High Throughput", 9th ACM Conference on Embedded Networked Sensor Systems, Pages 12-25, New York, USA, 2011.
- [16] Matthias Kovatsch, Simon Duquennoy, Adam Dunkels, "A Low-Power CoAP for Contiki", IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS), Page 855 - 860, 2011.
- [17] Olaf Landsiedel, Euhanna Ghadimi, Simon Duquennoy, Mikael Johansson, "Low Power, Low Delay: Opportunistic Routing meets Duty Cycling", 11th international conference on Information Processing in Sensor Networks, Pages 185-196, New York, USA, 2012.
- [18] Suman Nath, Phillip B. Gibbons, "Communicating via Fireflies: Geographic Routing on Duty-Cycled Sensors", 6th international conference on Information processing in sensor networks, Pages 440-449, New York, USA, 2007.
- [19] Nicola Accettura, Carlos Aguado Camacho, Luigi Alfredo Grieco, Gennaro Boggia, and Pietro Camarda, "Performance Assessment and Tuning Rules for Low-power and Lossy Stacks", 9th IEEE International Workshop on Factory Communication Systems (WFCS), Page 71 - 74, 2012.
- [20] A. Dunkels et al., "Contiki-a lightweight and flexible operating system for tiny networked sensors", 29th Annual IEEE International Conference on Local Computer Networks, Page 455 - 462, 2004.
- [21] P. Levis, A. Tavakoli, and S. Dawson-Haggerty, "Overview of existing routing protocols for low power and lossy networks", <http://tools.ietf.org/html/draft-levis-roll-overview-protocols-00>.
- [22] T. Winter and P. Thubert, "RPL: IPv6 routing protocol for low power and lossy networks", IETF, Draft, 2010.
- [23] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks". International Conference on Embedded Networked Sensor Systems (ACM SenSys), Baltimore, MD, USA, 2004.
- [24] F. Österlind, A. Dunkels, J. Eriksson, T. Voigt, "Cross-level sensor network simulation with COOJA", in Proc. Of LCN, 2006, Page 641-648.
- [25] Yanjun Sun, Omer Gurewitz, David B. Johnson "RI-MAC: A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks", 6th ACM conference on Embedded network sensor systems, Pages 1-14, New York, USA, 2008.
- [26] W. Ye, J. Heidemann, and D. Estrin. "An Energy-Efficient MAC Protocol for Wireless Sensor Networks". In Proceedings of the IEEE Conference on Computer Communications (INFOCOM), New York, NY, USA, June 2002.
- [27] K. Pister and L. Doherty, "TSMP: Time Synchronized Mesh Protocol". In Proceedings of the IASTED International Symposium on Distributed Sensor Networks (DSN08), Orlando, Florida, USA, November 2008.
- [28] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The trickle algorithm", IETF, Draft, 2010.
- [29] J. Vasseur, M. Kim, K. Pister, N. Dejean, and D. Barthel, "Routing metrics used for path calculation in low power and lossy networks", draft-ietf-roll-routing-metrics-10, Internet Engineering Task Force, Apr. 2011, <http://tools.ietf.org/html/draftietf-roll-routing-metrics-19>.
- [30] N. Tsiftes, J. Eriksson, and A. Dunkels, "Low-power wireless IPv6 routing with ContikiRPL", in Int Conf on Information Processing in Sensor Networks. ACM/IEEE, 2010.