



HAL
open science

Analyse des raisons de l'efficacité de ContikiMAC

Mathieu Michel, Quoitin Bruno

► **To cite this version:**

Mathieu Michel, Quoitin Bruno. Analyse des raisons de l'efficacité de ContikiMAC. ALGOTEL 2014 – 16èmes Rencontres Francophones sur les Aspects Algorithmiques des Télécommunications, Jun 2014, Le Bois-Plage-en-Ré, France. pp.1-4. hal-00984652

HAL Id: hal-00984652

<https://hal.science/hal-00984652>

Submitted on 28 Apr 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Analyse des raisons de l'efficacité de ContikiMAC

Mathieu MICHEL¹ and Bruno QUOITIN¹

¹UMONS, Service des réseaux et télécommunications (Mons - Belgique)

Cet article vise à mieux comprendre le gain de performance de ContikiMAC comparativement à X-MAC. ContikiMAC utilise une méthode de transmission consistant à répéter continuellement l'envoi d'un paquet complet jusqu'à réception d'un acquittement de la destination. X-MAC utilise au contraire une série de "strokes" de petites tailles utilisées pour annoncer la transmission. X-MAC utilise à priori moins de bande passante le rendant "intuitivement" moins coûteux. Dans l'optique de mieux comprendre les raisons de son efficacité, nous avons comparé ContikiMAC à une version mise à jour de X-MAC (visant à obtenir une comparaison plus objective). Leurs performances respectives ont été évaluées en termes de latence, de nombre de retransmissions, de *Packet Delivery Ratio* (PDR) et de *duty-cycle*. Ce type d'évaluation n'ayant, à notre connaissance, jamais été réalisé de façon détaillée. Les résultats obtenus montrent que la méthode de transmission de ContikiMAC combinée à une technique de réveil *fast-sleep* permettent d'obtenir un nombre de retransmissions réduit, une latence moins élevée, un *duty-cycle* plus de 3 fois moindre et un PDR supérieur.

Keywords: X-MAC, ContikiMAC, *collision avoidance*, *phaselock*, *fast-sleep*

1 Introduction

Ces dernières années de nombreux protocoles MAC à *duty-cycle* ont été développés pour répondre aux contraintes énergétiques des WSNs. En alternant périodiquement la radio de chaque noeud entre un état *off* et un état *on*, les protocoles à *duty-cycle* réduisent la consommation énergétique d'un noeud en évitant de garder sa radio allumée en cas d'absence de transmission le ciblant (*idle listening*). Particulièrement efficace dans les réseaux à faible bande passante [C⁺11], les protocoles à *duty-cycle* asynchrones se caractérisent par l'absence de synchronisation entre les périodes de réveil des noeuds.

De par leur disponibilité dans Contiki [con14], X-MAC [BYAH06] et ContikiMAC [Dun11] figurent parmi les plus connus et les plus utilisés. La littérature présente souvent ContikiMAC, utilisé par défaut dans Contiki, comme plus efficace en termes de consommation énergétique ou encore de latence [Pin12][Dun11]. Ceci pourrait paraître étonnant vu que la méthode de X-MAC consistant à annoncer une transmission de paquet par une série de "strokes" de petites tailles semble, à première vue, moins coûteuse que celle de ContikiMAC consistant à répéter la transmission de la trame complète jusqu'à sa réception. Pour mieux comprendre les raisons des meilleures performances de ContikiMAC, plusieurs expérimentations par simulation ont été réalisées. Afin de quantifier le gain de performance lié aux mécanismes d'envoi de paquets et de gestion du réveil de ContikiMAC, des versions mises-à-jour de X-MAC intégrant des fonctionnalités supplémentaires (présentes dans ContikiMAC) ont été comparées à ContikiMAC.

Cet article est structuré comme suit. La section 2 introduit les protocoles X-MAC et ContikiMAC. La section 3 présente la méthodologie employée et les modifications apportées à X-MAC. La section 4 reprend le résumé des résultats obtenus et une conclusion est donnée à la section 5.

2 X-MAC et ContikiMAC

X-MAC et ContikiMAC diffèrent principalement par la manière dont ils gèrent d'une part la transmission de paquets et d'autre part la période de réveil d'un noeud.

X-MAC précède chaque transmission de paquet par une série de *strokes* visant à alerter la destination. Celle-ci pourra alors notifier la source de sa disponibilité via un "strobe-ack" (Fig 1) ou se rendormir si la

transmission ne la concerne pas. ContikiMAC répète continuellement la transmission d'une trame complète jusqu'à ce que celle-ci soit acquittée par la destination (Fig 2).

En vue de recevoir d'éventuelles transmissions, X-MAC va se réveiller à intervalles réguliers pour une période prédéfinie. Cette période peut être étendue si le noeud est impliqué dans une transmission à la fin de celle-ci. ContikiMAC sonde rapidement l'activité du médium à l'aide de deux CCAs (*Clear Channel Assessments*) successifs pour établir la présence ou non de trafic et ainsi d'avoir la possibilité de se rendormir de manière anticipée en cas d'absence de signes d'activité. Dans le cas contraire, une fois une trame complète recue, un noeud peut, sur base de l'adresse de destination, décider de se rendormir ou non. Cette méthodologie dite "fast-sleep" et permettant d'optimiser la durée de la période de réveil n'est pas intéressante à transposer dans X-MAC étant donné que l'intervalle de temps entre deux stobes est par défaut 4 fois supérieur à celui séparant deux transmissions successives d'un paquet par ContikiMAC. Ce qui impliquerait une augmentation conséquente du nombre de CCAs nécessaires pour sonder le médium durant un intervalle de temps suffisant à la détection d'une transmission.

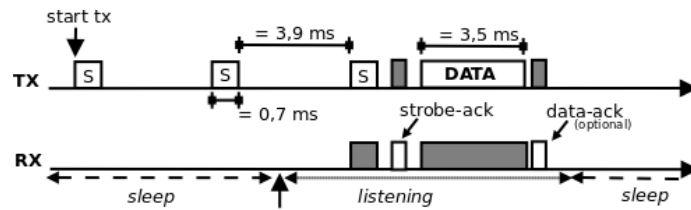


FIGURE 1: X-MAC précède l'envoi de paquets par une série de stobes.

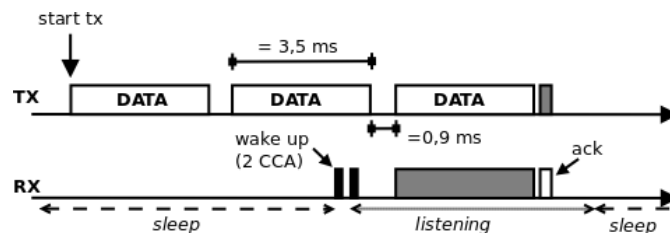


FIGURE 2: ContikiMAC répète l'envoi du paquet de manière continue jusqu'à la réception d'un ACK.

ContikiMAC diffère également par la présence de fonctionnalités supplémentaires ne lui étant pas spécifiques : une méthode de *collision avoidance* basée sur l'emploi de CCAs et un mécanisme de *phaselock* ([EHD04]) permettant d'apprendre les horaires de réveil d'un noeud.

3 Méthodologie

Dans le but de mieux comprendre les mécanismes conférant à ContikiMAC de meilleurs performances, des expérimentations ont été menées dans le simulateur Cooja [con14]. Celles-ci visent à évaluer ContikiMAC et X-MAC sur base des critères suivants : le nombre de retransmissions (dues à des collisions ou à l'absence d'ACK), la latence, le *Packet Delivery Ratio* (PDR) et le *duty-cycle*. Ce dernier permet de statuer sur la consommation énergétique, liée à l'utilisation de la radio, en évaluant le temps passé par un noeud dans les états *listen*, *RX* et *TX*.

Ces expérimentations ont été réalisées au moyen d'une application de type collection de données déployée sur une topologie de 40 noeuds positionnés aléatoirement avec une densité moyenne de 6 (+/- 4) voisins et utilisant le protocole RPL[W⁺12]. 5 noeuds sont désignés aléatoirement pour envoyer des paquets à destination du *sink*. Ces envois sont effectués de manière périodique à raison d'un paquet toutes les 15 secondes avec une limite maximale de 4 retransmissions (valeur par défaut de Contiki). Les configurations par défaut des protocoles X-MAC, ContikiMAC et RPL définies dans Contiki 2.6 ont été conservées ; le *duty-cycle* de base de X-MAC étant fixé à 5%. Les résultats sont lissés sur un ensemble de 100 répétitions, chacune impliquant un positionnement et des noeuds émetteurs différents.

De façon à se concentrer sur les mécanismes principaux de ContikiMAC et X-MAC il s'avère indispensable d'équilibrer les deux protocoles en implémentant dans X-MAC les mécanismes de *CCA collision avoidance* et de *phaselock*.

CCA collision avoidance Pour éviter les collisions, la version par défaut de X-MAC utilise un *flag* notifiant un noeud de la détection préalable d'une strobe indiquant qu'une transmission est en cours sur le médium. La limitation de cette technique est qu'un noeud en phase de sommeil ou occupé à transmettre ne peut activer ce *flag*. C'est pourquoi un système d'évitement de collisions, similaire à ContikiMAC, a été implémenté. En effectuant une suite de CCAs successifs, un noeud peut sonder le médium à la recherche de signes d'activité avant de commencer à émettre des stobres.

Phaselock Par défaut l'implémentation de X-MAC propose une *encounter optimization*[†] permettant d'apprendre l'horaire de réveil d'un noeud et d'ainsi réduire l'occupation du médium et la consommation énergétique en diminuant le nombre de stobres précédant la transmission du paquet. Cependant ce mécanisme place le noeud dans une attente bloquante jusqu'à ce qu'il puisse transmettre, lui ôtant toute possibilité de traiter d'autres tâches. Ce mécanisme a été remplacé par un "phaselock" similaire à ContikiMAC permettant à un noeud d'utiliser un timer lui indiquant à quel moment commencer la transmission de sa série de stobres sans le mettre en attente bloquante.

L'ajout de ces deux fonctionnalités mène à la création d'une version "améliorée" de X-MAC : **X-MAC-CP**[MQ14]. Celle-ci sera comparée avec la version "basique" de **X-MAC** et **ContikiMAC**. Afin de quantifier le gain de performance propre à chaque fonctionnalité, deux versions supplémentaires dans lesquelles ces mécanismes ont été implémentés séparément sont également évaluées : **X-MAC-C** (*CCA collision avoidance*) et **X-MAC-P** (*phaselock*).

4 Résultats

Les résultats obtenus[‡] sont présentés en Tableau 1. Il apparaît que **(1) ContikiMAC nécessite moitié moins de retransmissions que X-MAC** et ce pour une latence plus faible et un PDR plus élevé. En diminuant presque de moitié le nombre de retransmissions, X-MAC-CP permet de réduire la latence de X-MAC d'environ 10%. Cette diminution du nombre de retransmissions est obtenue d'une part par une réduction de l'utilisation du médium (confirmée par un temps plus faible passé en mode TX) due à l'emploi du *phaselock* et du *CCA collision avoidance*. Notons que la réduction de l'utilisation du médium, dans le cas de X-MAC-P, permet d'éviter de nombreuses collisions potentielles mais est impuissante quand il s'agit d'une collision entre deux transmissions destinées à des noeuds ayant des horaires de réveil similaires.

Comparé à X-MAC-C, l'introduction du *phaselock* dans X-MAC-P et X-MAC-CP engendre un gain d'énergie au détriment d'une légère augmentation de la latence. La raison de cette augmentation est qu'un noeud recevant un strobe-ack d'une destination en fin de période d'éveil est incapable de déterminer le début de cette période et réglera donc son *phaselock* sur le moment de réception du strobe-ack. Cette "mauvaise" calibration inévitable du *phaselock* implique un délai supplémentaire et inutile avant de démarrer la transmission. De ce fait une éventuelle transmission risque d'être retardée jusqu'au prochain rendez vous, entraînant un délai plus ou moins important. Celui-ci aurait été évité en l'absence de *phaselock*.

(2) ContikiMAC obtient une latence plus faible que celle de X-MAC-CP. Cette diminution est obtenue par l'optimisation *fast-sleep* qui en diminuant l'*idle listening* réduit la probabilité d'une réponse distante du début de la période de réveil et mitige ainsi le problème de calibration du *phaselock*. D'autre part, le *sink* étant en écoute constante, un paquet lui étant directement adressé sera acquitté dès la réception d'une seule trame par ContikiMAC, là où X-MAC nécessite au minimum une strobe en plus de la trame.

Etant donné que X-MAC utilise la réception d'un strobe-ack comme signal de la fin de la procédure de transmission, une retransmission sera nécessaire si le paquet data n'est pas acquitté, là où **(3) ContikiMAC permet d'obtenir une meilleure fiabilité** (cf PDR) en n'interrompant le processus de transmission qu'une fois le paquet acquitté. Ceci explique le nombre de retransmissions plus élevé de X-MAC-CP. Un paquet ne pouvant être retransmis qu'un nombre limité de fois et le PDR étant lié aux nombres de retransmissions sont les raisons expliquant le meilleur PDR de ContikiMAC comparé aux différentes versions de X-MAC.

En ce qui concerne la consommation énergétique, un noeud sous X-MAC passe un temps en écoute (*listen*) plus de 4 fois supérieur à ContikiMAC qui réduit considérablement la période d'éveil d'un noeud

†. A notre connaissance cette amélioration n'a pas été évaluée

‡. Un tech-report reprend des résultats supplémentaires similaires liés à l'évaluation d'une variation de la densité et du trafic [MQ14]

Protocole	ContikiMAC	X-MAC	X-MAC-CP	X-MAC-C	X-MAC-P
ReTX count	380 (+/-127)	794 (+/-351)	400 (+/-155)	466 (+/-191)	406 (+/-172)
Latency (s)	1.10 (+/-0.85)	1.25 (+/- 1.50)	1.17 (+/- 1.07)	1.03 (+/- 1.06)	1.14 (+/- 1.02)
PDR (%)	68.9	64.9	67.4	66.6	66.2
LISTEN Mode (%)	1.39 (+/-0.39)	6.78 (+/-1.37)	5.98 (+/-0.58)	6.51 (+/-1.21)	6.13 (+/-0.67)
TX Mode (%)	0.34 (+/-0.17)	0.31 (+/-0.31)	0.16 (+/-0.12)	0.28 (+/-0.27)	0.18 (+/-0.13)
RX Mode (%)	0.06 (+/-0.06)	0.18 (+/-0.11)	0.15 (+/-0.10)	0.16 (+/-0.10)	0.16 (+/-0.11)

TABLE 1: Nombre de retransmissions, latence *end-to-end*, PDR et Duty Cycle moyens (+écart-type)

grâce au mécanisme *fast-sleep*. Cette optimisation de la période de réveil est la raison principale de la plus faible consommation énergétique de ContikiMAC. En effet, du côté du temps moyen de transmission, la méthode employée par ContikiMAC se révèle plus "coûteuse" que X-MAC. L'introduction du *phaselock*, réduisant l'utilisation du médium, permet à X-MAC-P d'améliorer ce résultat. Ce dernier est encore meilleur dans X-MAC-CP qui combine le *phaselock* au *CCA collision avoidance*. Cependant ContikiMAC affiche un temps de réception inférieur principalement dû à l'optimisation *fast-sleep* qui en réduisant la période de réveil d'un noeud diminue les risques d'*overhearing*. En cas d'*overhearing*, un noeud sous X-MAC ne se rendormira qu'après l'identification de la destination ciblée par la transmission, ce qui nécessite au minimum la réception d'une strobe. Malgré une méthode de transmission plus coûteuse, **(4) ContikiMAC obtient donc un *duty-cycle* total (1.79) plus de 3 fois inférieur à celui de X-MAC-CP (6.29).**

5 Conclusion

Cette étude démontre que les meilleures performances de ContikiMAC ne sont pas uniquement dues à l'adjonction d'une technique de *CCA collision avoidance* et d'un mécanisme de *phaselock*. Alors que la méthode d'envoi de trames complètes par transmissions successives bien que plus coûteuse d'un point de vue énergétique se montre au final plus fiable et plus rapide, l'apport d'un réveil de type *fast-sleep*, inadéquat avec X-MAC, a un impact bénéfique sur la consommation énergétique mais aussi sur l'efficacité du *phaselock*. Notons cependant que l'implémentation de ces mécanismes dans X-MAC résulte en une nette amélioration des performances et peut se révéler intéressante dans le cadre de topologies ou des protocoles de type LPL se voient confrontés à un problème de "false wakeup" (tel que décrit par [SHL13]) dû à un *cca threshold*, inadapté à la topologie, forçant un noeud à se réveiller inutilement.

Références

- [BYAH06] M. Buettner, G. Yee, E. Anderson, and R. Han. X-mac : A short preamble mac protocol for duty-cycles wireless sensor networks. In *Technical Report CU-CS-1008-96*, May 2006.
- [C⁺11] C. Cano et al. Low energy operation in wsns : A survey of preamble sampling mac protocols. In *Comput. Netw*, doi :10.1016/j.comnet.2011.06.022, 2011.
- [con14] <https://github.com/contiki-os/contiki/wiki>, 2014.
- [Dun11] A. Dunkels. The contikimac radio duty cycling protocol. In *SICS Technical Report T2011 :13*, 2011.
- [EHD04] A. El-Hoiydi and J.-D. Decotignie. Wisemac : An ultra lowpower mac protocol for the down-link of infrastructure wireless sensor networks. In *9th ISCC*, 2004.
- [MQ14] M. Michel and B. Quoitin. Technical report : Contikimac vs x-mac performance analysis, 2014.
- [Pin12] C. Pinola. Evaluating the performance of synchronous and asynchronous media access control protocols in the contiki operating system. In *Technical Report 2012 :12*, 2012.
- [SHL13] M. Sha, G. Hackmann, and C. Lu. Energy-efficient low power listening for wireless sensor networks in noisy environments. In *IPSN 13, Philadelphia, Pennsylvania, USA*, 2013.
- [W⁺12] T. Winter et al. Rpl : Ipv6 routing protocol for low-power and lossy networks. In *Internet Engineering Task Force, RFC 6550*, March 2012.