

# Protocoles auto-adaptatifs energie-traffic pour les reseaux de capteurs sans fil

Ye-Qiong Song

► **To cite this version:**

Ye-Qiong Song. Protocoles auto-adaptatifs energie-traffic pour les reseaux de capteurs sans fil. 8èmes journées francophones Mobilité et Ubiquité (Ubimob 2012), Philippe Roose (LIUPPA, IUT de Bayonne/UPPA) & Nadine Couture (LaBRI, ESTIA), Jun 2012, Anglet, France. hal-00745167

**HAL Id: hal-00745167**

**<https://hal.inria.fr/hal-00745167>**

Submitted on 24 Oct 2012

**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.

# Protocoles auto-adaptatifs énergie-traffic pour les réseaux de capteurs sans fil

Ye-Qiong SONG, Université de Lorraine - LORIA

*Résumé : La plupart des protocoles conçus pour les réseaux de capteurs sans fil se reposent sur le principe de « duty-cycle » afin d'économiser l'énergie. Cette approche peut poser des problèmes de performances quand le réseau doit transporter des données de quantité importante sur une courte période (c'est à dire une rafale de données). Dans ce papier nous traitons deux problèmes importants : comment ajuster le « duty-cycle » en fonction de la variation du trafic et comment transporter plus efficacement des rafales de données afin d'assurer une certaine qualité de service tout en minimisant la consommation de l'énergie. Nous faisons d'abord une analyse des principaux protocoles à « duty-cycle » existants et discuter des problèmes restants, puis présentons nos deux protocoles Queue-MAC et S-CoSenS qui résolvent efficacement ces deux problèmes posés.*

*Mots-clés : Réseau de capteurs, auto-adaptation, protocoles MAC et routage, énergie, qualité de service.*

## 1 Introduction

La technologie actuelle des réseaux de capteurs suppose que les nœuds de capteurs sont alimentés par des batteries. Economiser l'énergie est donc une préoccupation omniprésente dans toutes les conceptions de protocoles MAC et routage pour les réseaux de capteurs sans fil. Mettre en veille prolongée la partie transmission/réception radio est sans doute le moyen le plus efficace car avec les composants actuels, elle est la partie qui consomme le plus d'énergie [1].

Suivant cette approche, une famille de protocoles MAC à faible cycle actif (duty-cycle) a été conçue. Deux états sont associés à la partie radio d'un nœud : active et endormie (ou éteinte). Un cycle, d'une période fixe ou variable, est composé d'une période active durant laquelle un nœud peut transmettre et recevoir, et une période de sommeil où la partie radio est éteinte. Le cycle actif (duty-cycle) est défini comme la proportion de la période active sur la durée totale d'un cycle (période active + période de sommeil).

Par rapport à un réseau dont les nœuds sont actifs en permanence, il est clair qu'un réseau fonctionnant avec un faible, voire très faible cycle actif (low duty-cycle et ultra-low duty-cycle) permet d'économiser l'énergie et par conséquent prolonger la durée de vie du réseau en la multipliant approximativement par l'inverse de sa valeur de « duty-cycle ». Mais il offre un débit très limité. Prenons l'exemple du standard IEEE802.15.4 [2] avec son débit nominal maximal de 250 Kbit/s. Un réseau de cycle actif de 1% (ce qui est classique) ne peut qu'offrir au maximum 2,5 Kbit/s. Si ce faible débit peut satisfaire au besoin du transfert périodique de données de capteurs de faible volume et/ou de remonter des événements rares, en revanche, il est difficile de transmettre des données en rafale, de volume important et nécessitant un acheminement rapide, voire en temps réel.

Pourtant, bon nombre d'applications des réseaux de capteurs ont effectivement besoin à la fois de la haute réactivité du réseau et de l'efficacité énergétique. Prenons l'exemple d'un scénario de la détection et du suivi en temps réel de l'intrusion (que nous avons défini comme un scénario d'applications dans le projet ANR QUASIMODO : <http://quasimodo.loria.fr/>). La plupart du temps, le réseau peut fonctionner avec un cycle actif très faible pour simplement assurer que les nœuds de capteurs sont toujours en vie. Dès qu'une intrusion est détectée (probablement par plusieurs capteurs à cause du déploiement dense), l'événement doit être reporté rapidement à la station de surveillance et aux nœuds mobiles (robots) afin d'assurer le suivi en temps réel et l'interception de l'intrus. Ce scénario génère typiquement des rafales de trafic importantes à transporter par le réseau avec une certaine qualité de service (e.g. dans un délai raisonnable lié à la dynamique de l'application, ici le suivi en temps réel de l'intrus).

En général, économiser l'énergie et fournir la qualité de service, notée QoS (e.g. large bande passante, faible délai, grande fiabilité), sont deux objectifs antagonistes car l'endormissement des nœuds réduit la bande passante et augmente le délai de bout en bout dans un réseau multi-sauts. Dans la littérature, on peut trouver deux approches : l'une est sous l'appellation de « QoS-aware MAC ou QoS-aware routing » dont le principe est de considérer la QoS comme une contrainte et on cherche à optimiser l'énergie ; l'autre formule le problème comme un problème d'optimisation multi-objectifs et applique des techniques heuristiques pour trouver un compromis.

Dans nos travaux de recherche, nous adoptons une approche plus simple mais aussi plus efficace. Nous considérons qu'économiser l'énergie et fournir la QoS correspondent à deux phases différentes du même réseau. Comme l'exemple du scénario précédent, lorsqu'il n'y a pas ou peu d'événement (donc peu de trafic), nous cherchons à économiser l'énergie. En revanche, quand le réseau reçoit soudainement un trafic important qui correspond à une situation critique (e.g. intrusion détectée), il faut basculer le réseau dans un mode de fonctionnement qui vise à fournir la QoS, même si cela peut être au détriment de l'économie d'énergie.

Il est donc important de développer des protocoles à cycle actif pour économiser l'énergie mais qui auto-adaptent au changement de condition de trafic afin de transporter avec la QoS le trafic généré, surtout lorsque ce dernier arrive en rafale. A ce propos, les protocoles à cycle actif variable sont préférables.

Afin d'obtenir une solution permettant à la fois d'économiser l'énergie et d'offrir la QoS au trafic en rafale, dans ce papier, nous traitons deux problèmes. Le premier concerne la conception d'algorithmes et protocoles qui adaptent automatiquement leur « duty-cycle » en fonction de la variation du trafic. Le second concerne comment transporter efficacement un trafic de charge élevée, sachant que la plupart des protocoles MAC existants s'appuient sur CSMA/CA, ce qui n'est pas efficace face à la charge élevée de trafic à cause de la probabilité croissante de collisions. Ce qui est un résultat intrinsèque du CSMA/CA bien connu depuis plus de trente ans [3].

Face à ces deux problèmes nous avons développé deux solutions plus efficaces par rapport aux solutions de l'état de l'art : Queue-MAC qui propose une solution MAC en combinant CSMA/CA et TDMA, permute entre les deux modes selon la taille des queues des paquets en émission des nœuds et est implémentable sur le standard IEEE802.15.4 ; et S-CoSenS qui propose une solution inter-couche (cross-layer) MAC-routage assurant un transfert de bout en bout.

Dans la suite de ce papier, nous analysons d'abord, dans la section 2, les principaux protocoles MAC à cycle actif et la possibilité de les utiliser pour auto-adapter au changement de trafic, ainsi que les problèmes qui restent encore à résoudre. La section 3 présente le protocole Queue-MAC qui permet d'aller au-delà de l'état de l'art en terme de protocole MAC. La section 4 décrit S-CoSenS. La section 5 conclut ce papier.

## 2 Etat de l'art

Vis à vis de nos deux problèmes posés : auto-adaptation de « duty-cycle » à la variation du trafic et transmission efficace de rafale de données, nous nous focalisons uniquement sur les protocoles MAC à « duty-cycle ». Il est à noter que certains de ces protocoles MAC assurent aussi une partie des fonctions du routage car conçus pour des réseaux multi-sauts (conception cross-layer MAC-Routage). Rappelons que le problème du routage avec des nœuds fonctionnant alternativement en périodes active et inactive nécessite la synchronisation de la période active entre un émetteur et un récepteur intermédiaire (routeur), ou globalement entre des nœuds voisins. En effet, un nœud émetteur ne peut transmettre un paquet au prochain saut que lorsque ce dernier est actif.

Parmi les protocoles MAC à « duty-cycle » les plus connus et répandus, nous pouvons citer IEEE802.15.4 qui permet de définir une période active et une période inactive dans son mode avec balise (beacon-enabled mode). Un cycle est appelé « superframe » qui est balisé par une trame

« beacon » diffusée par le coordinateur afin de synchroniser les nœuds associés dans une topologie en étoile. Chaque nœud ne communique qu'avec son coordinateur qui est tout le temps actif pendant la période active. Comme les nœuds associés au coordinateur sont susceptibles de recevoir, ils sont aussi en état actif durant toute la période active. La période active est subdivisée en périodes de « beacon », CAP (Contention Access Period) basées sur CSMA/CA slotté, et optionnellement CFP (Contention Free Period) basée sur TDMA, connue aussi sous le nom de GTS (Guarenteed Time Slots) dont le nombre de slots est limité à 7. Il est à noter que le mode « beacon-enabled » de IEEE802.15.4 définit à la fois la notion de slot (16 dans la période active) et de « Backoff Period » BP qui est beaucoup plus petit et peut être considéré comme la granularité du temps discrétisé ( $aUnitBackoffPeriod = 320$  micro secondes ou équivalent à la durée de transmission de 10 octets à 250Kbit/s). Le protocole CSMA/CA slotté est alors aligné sur le début d'un BP et non d'un slot (qui contient  $n$  BP). La longueur du cycle ( $BI$  : Beacon Interval) et celle de la période active ( $SD$  : Superframe Duration) sont définies par deux paramètres  $BO$  (Beacon Order) et  $SO$  (Superframe Order) à la configuration. Avec  $BI=960x16x2^{BO}$  et  $SD=960x16x2^{SO}$  microsecondes (avec  $SO$  et  $BO$  entre 0 et 14). De ce point de vue, IEEE802.15.4 n'offre que du « duty-cycle » fixe et ne permet pas d'auto-adapter à la variation du trafic. La conception offre néanmoins la possibilité d'utilisation du TDMA (GTS) pour transmettre des paquets plus efficacement que CSMA/CA lorsque le trafic est élevé. Cette possibilité est néanmoins limitée par le nombre de slots (7 au maximum) et la complexité de la réservation de slots dont la demande de réservation passe par CSMA/CA avec une durée de réservation à déterminer au moment de la réservation, ce qui ne permet pas d'une grande adaptation à la variation du trafic. Du point de vue de l'efficacité énergétique, IEEE802.15.4 maintient tous les nœuds actifs pendant la période active, ce qui n'est pas optimal surtout quand les nœuds n'ont pas de données à transmettre ou recevoir. Un autre problème est que ce mode « beacon-enabled » est conçu pour fonctionner en topologie étoile, ne supportant pas nativement les réseaux multi-saut, bien qu'il existe des travaux sur l'ordonnancement des « beacons » [4] pour le rendre utilisable dans un réseau multi-saut organisé en arbre (cluster-tree), mais qui exhibent des difficultés de passage à l'échelle car l'ordonnancement est statique. Notons tout de même que IEEE802.15.4 est le seul standard dont l'implémentation est la plus répandue.

S-MAC [5] est l'un des premiers protocoles MAC à « duty-cycle » conçu pour des réseaux multi-sauts. Chaque nœud commute périodiquement entre une période active et une période inactive pour économiser l'énergie. Chaque nœud est libre de choisir sa propre « duty-cycle ». Afin qu'un nœud émetteur puisse trouver un nœud routeur voisin pendant sa période active, S-MAC adopte la synchronisation suivante. Chaque nœud diffuse périodiquement vers ses nœuds voisins son ordonnancement de périodes actives et inactives. Ainsi chaque nœud mémorise les ordonnancements de tous ses voisins afin de savoir quand un voisin particulier entrera dans sa période active pour lui transmettre sa donnée. Si plusieurs nœuds doivent transmettre leurs données à un même nœud routeur ou récepteur, ils utilisent le CSMA/CA de IEEE802.11, y compris RTS/CTS. Normalement dans un petit réseau où tout le monde peut entendre tout le monde, c'est le premier nœud qui diffuse son ordonnancement qui impose son rythme car tous ses voisins se synchronisent avec lui. En général, il se peut que plusieurs nœuds diffusent leurs ordonnancements différents dans des zones de diffusions distinctes. Un nœud voisin ayant reçu des ordonnancements différents adopte alors ces différents rythmes, se réveillant ainsi à chaque période active différente. Dans S-MAC, le nœud qui diffuse en premier son ordonnancement est un nœud synchroniseur et les nœuds voisins sont des nœuds suiveurs. Une première remarque que nous pouvons formuler est qu'il s'agit d'un protocole à « duty-cycle » fixe et qui ne s'adapte pas automatiquement à la variation du trafic, sauf un nœud le modifie explicitement en devenant un nouveau synchroniseur (mécanisme non prévu dans S-MAC). Une autre remarque est que le délai de bout en bout dépend de la période de chaque nœud du chemin, ce qui peut être très long. Une troisième remarque est que son trafic de contrôle (overheads) représente une part importante du trafic total car non seulement il

exige l'utilisation de RTS/CTS, mais aussi la diffusion des messages de synchronisation ou resynchronisation. Enfin pour tolérer la dérive des horloges, la période active doit être suffisamment longue, ce qui l'empêche d'être utilisé avec un « duty-cycle » ultra faible.

B-MAC [6] adopte la fameuse technique LPL (Low Power Listening). Les nœuds commutent périodiquement entre état actif et inactif (radio éteinte), avec l'intervalle d'échantillonnage. L'état actif est en général de très courte durée, juste ce qu'il faut pour permettre au nœud d'échantillonner le canal. Quand un nœud se réveille, il allume sa radio et vérifie l'état du canal (CCA : Clear Channel Assessment). S'il ne détecte aucune activité, il se rendort. Dans le cas contraire, il reste actif pour recevoir le paquet. Après la réception, le nœud repasse en mode inactif, sauf qu'il doit continuer à relayer le paquet. Du côté de l'émetteur, chaque transmission d'un paquet est précédée par la transmission d'un long préambule. La taille du préambule doit être plus longue que l'intervalle d'échantillonnage afin d'être sûr de pouvoir être détecté par un récepteur voisin (prochain saut). De cette façon, le récepteur est averti, donc réveillé, pour recevoir le paquet de données. B-MAC offre une bonne efficacité énergétique et la période active de chaque nœud récepteur peut être prolongée ou raccourcie en fonction de la charge de l'émetteur. Il est donc à « duty-cycle » dynamique, auto-adaptant à la variation du trafic. B-MAC propose aussi une interface de haut niveau, permettant de reconfigurer l'intervalle de réveil (échantillonnage) pour trouver un bon compromis entre l'énergie et le débit du réseau. Puisque B-MAC utilise CSMA/CA pour l'accès au médium, il souffre du problème de débit lors de la charge élevée à cause des collisions. Un autre problème est la sur-écoute du préambule par des nœuds voisins car même si le paquet n'est destiné qu'à un nœud particulier (prochain saut), tous les autres nœuds voisins doivent quand même écouter le préambule jusqu'à la fin. Notons toutefois que cette sur-écoute n'a pas que des effets néfastes car elle peut aussi servir pour choisir des routeurs multiples, comme discuté plus loin dans ce papier.

X-MAC [7] est une amélioration de B-MAC pour résoudre essentiellement le problème de sur-écoute. Au lieu de transmettre un grand préambule, X-MAC le divise en un ensemble de petits paquets de préambules, chacun contenant l'adresse du destinataire du paquet à transmettre, et le transmet tout en insérant un intervalle de temps entre eux. Ces intervalles permettent au nœud destinataire d'envoyer un acquittement lorsqu'il reçoit l'un de ces paquets préambules. Une fois l'émetteur reçoit l'acquittement, il sait que le nœud prochain saut est réveillé et interrompt l'envoi du reste des paquets de préambules, puis envoie immédiatement le paquet vers le destinataire. Comme B-MAC, X-MAC propose aussi l'auto-adaptation de la durée de sommeil en fonction de la variation du trafic. Par rapport à B-MAC, X-MAC permet d'améliorer l'efficacité énergétique et réduire le délai grâce au préambule raccourci. Néanmoins comme expliqué précédemment, X-MAC ne peut choisir qu'un seul routeur pour faire avancer le paquet vers sa destination dans un réseau multi-saut, même s'il existe de chemins multiples dont l'exploitation aurait pu rendre la transmission de bout en bout plus robuste [8]. Le problème du faible débit lorsque la charge est élevée reste toujours non résolu.

Dans la section suivante, nous présentons Queue-MAC qui combine CSMA et TDMA afin de préserver l'efficacité de traiter le trafic faible et la capacité de passage à l'échelle de CSMA d'une part, et d'obtenir l'efficacité de traiter le trafic élevé par TDMA d'autre part. En effet, selon le résultat bien connu de Bux [9], CSMA est mieux adapté au cas du trafic faible, tandis que TDMA est mieux adapté au trafic élevé.

### 3 Le protocole Queue-MAC

Queue-MAC [10] est un protocole MAC hybride CSMA/TDMA qui s'adapte dynamiquement à la variation du trafic. Deux points clés sont considérés dans notre conception. Premièrement nous devons disposer d'une jauge indicatrice reflétant la variation du trafic. Deuxièmement nous devons

pouvoir utiliser à la fois CSMA/CA et TDMA pour traiter efficacement le cas du trafic faible et le cas du trafic élevé. L'indicateur du trafic devrait nous servir à commuter entre les deux modes, atteignant ainsi l'objectif de l'auto-adaptation. D'autres considérations secondaires nous ont conduit à choisir une solution proche du standard IEEE802.15.4 et à conserver une période minimale CSMA/CA afin de minimiser la configuration statique et préserver la bonne capacité du passage à l'échelle.

La longueur de la queue d'émission des nœuds est utilisée comme l'indicateur du trafic courant, d'où vient le nom Queue-MAC. Quand le trafic augmente, la période active CSMA/CA est étendue par l'ajout des slots TDMA, permettant ainsi de gérer efficacement l'augmentation du trafic en rafale ayant des contraintes de la QoS. Lorsque le trafic est faible, seul CSMA/CA est présent dans la période active.

Afin d'assurer son implémentabilité sur la majorité des composants existants, Queue-MAC est volontairement inspiré du standard IEEE802.15.4 et peut fonctionner dans des réseaux multi-sauts adoptant la structure de superframe (par exemple Zigbee « cluster-tree »). Comme IEEE802.15.4, Queue-MAC ne gère pas nativement le routage multi-saut mais laisse gérer le problème de synchronisation des coordonneurs par ailleurs, en suivant par exemple l'ordonnancement proposé dans [4]. Les nœuds parents tels que routeurs (ou têtes de clusters) diffusent périodiquement des « beacons » pour diviser le temps en superframes répétitives contenant chacune une période active et une période inactive. Sans modifier l'en-tête MAC standard, nous ajoutons un champ spécifique dans la partie données de la trame IEEE802.15.4, appelé « indicateur de la longueur de la queue », pour décrire la charge d'un nœud en terme du nombre de paquets dans sa file d'attente d'émission. La nouvelle structure de la superframe compte une période TDMA de taille variable dont les slots sont ajoutés ou supprimés en fonction de la charge des nœuds. Un coordonneur (parent) peut connaître la valeur de chaque indicateur de ses nœuds enfants car l'indicateur est inséré dans des paquets qui lui sont transmis précédemment durant la période CSMA/CA (technique de « piggybacking »). Quand le trafic augmente, le coordonneur annonce des slots TDMA dans son « beacon » pour allonger la période active en ajoutant plus de slots, augmentant ainsi la bande passante.

Nous avons implémenté ce protocole sur des cartes SOC STM32W108 [11]. La comparaison avec le protocole IEEE802.15.4 optimisé (c'est à dire en mode « beacon-enabled » et en supposant qu'on peut utiliser les GTS sans passer par la phase de réservation) a montré une grande amélioration de performances.

### 3.1 Structure des trames

Queue-MAC nécessite deux modifications du standard IEEE802.15.4. La structure de la « superframe » est redéfinie comme illustrée dans FIG.1.

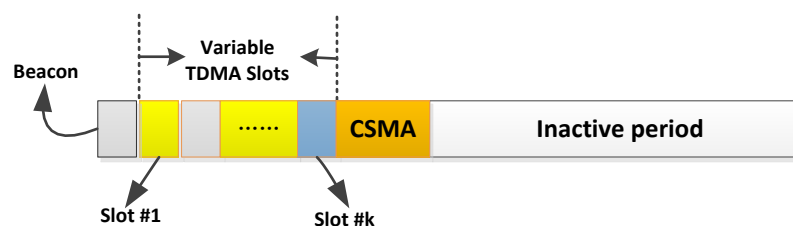


FIG. 1 - Structure de la « superframe » de Queue-MAC

La période variable TDMA est ajoutée dont les slots sont alloués aux nœuds selon leur nombre de paquets en attente de transmission. La plupart du trafic est transporté par TDMA. Si nous avions que la période TDMA, il va falloir donner au moins un slot par nœud, sinon les nœuds ne pourront jamais obtenir des slots TDMA supplémentaires et leur charge ne pourra pas être apprise par le

coordinateur. Pour résoudre ce problème, une période CSMA/CA de taille fixe suit la période TDMA. Durant la période CSMA/CA, chaque nœud aura sa chance de transmettre un ou plusieurs paquets selon CSMA/CA dans lesquels il insère le nombre de paquets restants dans sa queue d'émission. Comme la plupart du trafic a été transporté durant la période TDMA, le trafic restant dans la période CSMA/CA est généralement faible, ce qui réduit considérablement la probabilité de collisions et donc les retransmissions. Il est donc important de mettre la période TDMA avant la période CSMA/CA. Il est aussi à noter que la taille de la période CSMA/CA doit être suffisamment grande pour permettre à chaque nœud d'avoir une probabilité raisonnable de réussir sa transmission. La minimale correspond à la durée de transmission d'au moins un paquet et un accusé de réception, noté  $T_{slot}$ . La taille conseillée est  $mT_{slot}$  où  $m$  est le nombre de nœuds enfants du coordinateur. Quant au problème de la détermination de la taille optimale, nous le laissons au choix de l'utilisateur en fonction de la qualité de service désirée.

L'allocation des slots TDMA est assurée par le coordinateur selon la charge de chaque nœud enfant. Pour cela la trame « beacon » du standard IEEE802.15.4 est aussi modifiée comme illustrée dans FIG. 2.

Header	Slot #	ID list	Slot allocationg list
--------	--------	---------	-----------------------

FIG. 2 - Structure de la trame « beacon »

Une liste d'identificateurs (ID list) et une liste de slots alloués (slot allocating list) ont été ajoutées. La liste d'ID maintient la liste des nœuds à qui des slots ont été alloués. La liste des slots alloués maintient le nombre de slots alloués à chaque nœud de la liste d'ID. Une fois un paquet est reçu, ces deux listes sont mises à jour par le coordinateur. En recevant un paquet d'un nœud  $i$ , le coordinateur cherche dans la liste d'ID si le nœud  $i$  a déjà des slots alloués. Sinon, le coordinateur regarde la valeur de l'indicateur de queue. Si la valeur est zéro, le coordinateur reçoit le paquet normalement. Si la valeur est non nulle, le coordinateur ajoute  $ID i$  dans la liste d'ID. Selon une certaine stratégie d'allocation (voir sous-section suivante), le coordinateur ajoute dans la liste des slots alloués le nombre de slots qu'il faut pour la superframe suivante. Si  $ID i$  a déjà des slots alloués précédemment mais la valeur de l'indicateur est zéro,  $ID i$  est supprimé de la liste d'ID. Si la valeur de l'indicateur est non nulle, le coordinateur met à jour le nombre de slots alloués à nœud  $i$ .

Une stratégie possible d'allocation de slots TDMA est présentée dans la sous-section suivante.

### 3.2 Allocation de slots TDMA

Supposons que les paquets sont de taille fixe dont la durée de transmission plus la confirmation de l'acquittement correspond à un slot. Le nombre de slots alloués à chaque nœud est proportionnel à la valeur de l'indicateur de la longueur de sa file d'attente. S'il y a  $n$  nœuds qui ont besoin d'être alloués de slots, le nombre total de slots à allouer,  $C$ , est donné par l'équation (1).

$$C = \sum_{i=1}^n N_i \quad (1)$$

où  $N_i$  est le nombre de slots alloués à nœud  $i$ .

Comme la longueur de la superframe est fixe, il y a une borne supérieure de  $C$  qui est donnée par l'équation (2).

$$M = \frac{T_S - T_B - T_C - T_R}{T_{slot}} \quad (2)$$

où  $T_S$  est la durée de la superframe,  $T_B$  et  $T_C$  sont respectivement la durée de la « beacon » et de la période CSMA/CA.  $T_R$  est le temps maximum que le coordinateur a besoin pour retransmettre les

paquets collectés vers le niveau supérieur (d'un arbre cluster-tree).  $T_{slot}$  est la durée d'un slot.  $M$  représente donc le nombre maximal de slots qui peuvent être alloués durant une superframe.

Prenons  $K$  comme le nombre de slots réellement alloués, on a alors :

$$K = \min(C, M) \quad (3)$$

Dans le cas où  $C$  est supérieur à  $M$ , le coordinateur alloue les  $M$  slots proportionnellement à la valeur des indicateurs.

Par rapport au schéma du « duty-cycle » fixe du standard IEEE802.15.4, Queue-MAC, avec sa stratégie d'allocation de slots TDMA, fournit visiblement une meilleure performance. La sous-section suivante donne une comparaison expérimentale.

### 3.3 Evaluation expérimentale de performances

Pour une comparaison équitable, IEEE802.15.4 est aussi implémenté sur les même cartes SOC STM32W108 [11] mais en l'optimisant pour l'allocation des GTS. En effet, nous avons aussi utilisé la technique de « piggybacking » au lieu des paquets de réservation supplémentaires. La différence est que le nombre de GTS est toujours limité à 7.

Le scénario de test est composé d'un seul cluster avec une tête de cluster (coordinateur) et un nombre de nœuds enfants, variable allant jusqu'à 30 (variant ainsi la charge du réseau). Chaque nœud génère en moyen 2 paquets par seconde, soit environ un paquet par superframe, selon le processus de Poisson. La taille maximale des files d'attente d'émission des nœuds enfants est de 50 paquets. Pour les deux protocoles MAC nous avons choisi la taille d'un slot à 3,84ms, correspondant approximativement à la durée de transmission d'un paquet de 95 octets, soit 3,04ms, suivi de l'espace inter-trame et d'un acquittement. Cela correspond à  $SO=2$  et  $BO=5$  dans IEEE802.15.4. Ce qui donne une superframe de 491,52ms et période active de 61,44ms, soit 16 slots. Pour Queue-MAC, la durée de la période CSMA/CA est 40ms. Chaque expérimentation dure 40 secondes et est reconduite 10 fois afin d'obtenir des moyennes relativement stables du point de vue statistique.

La figure 3 (gauche) montre le délai maximal mesuré pour Queue-MAC (noté Q-MAC pour la concision) et IEEE802.15.4 par rapport à la variation de la charge. On voit que le délai de Queue-MAC est stable, tandis que celui de IEEE802.15.4 augmente rapidement une fois la charge dépasse la capacité de transmission d'une période active (maximum 15 paquets). Le délai maximal de IEEE802.15.4 est borné vers 25 secondes car cela correspond à la transmission de 50 paquets si chaque superframe permet de transmettre un paquet en cas de saturation.

La figure 3 (droite) donne le détail en terme du nombre de paquets reçus par chaque protocole (débit ou « throughput »). Il est clair que Queue-MAC arrive à transmettre tous les paquets générés, alors que IEEE802.15.4 n'en transmet qu'une partie. La différence correspond au rejet des paquets à cause de la saturation des files d'attente.

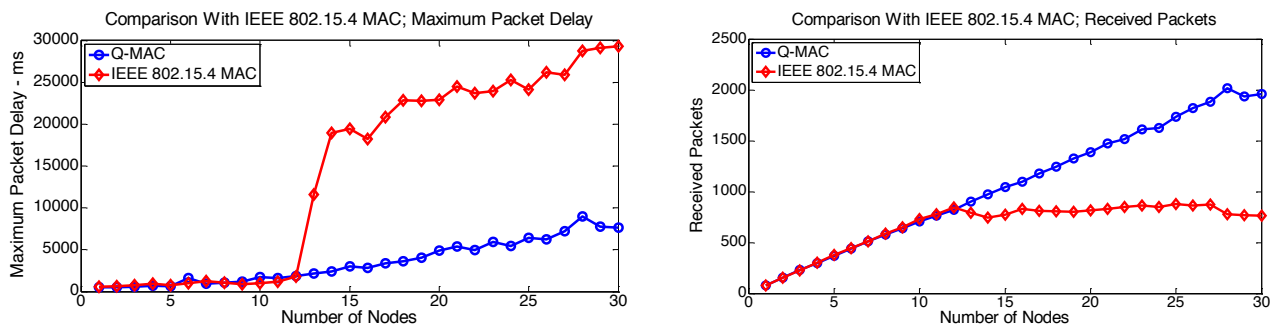


FIG. 3 - Délai maximal (gauche) et Paquets reçus (droite) par Queue-MAC et IEEE802.15.4



La figure 4 (gauche) illustre la variation de la taille de la file d'attente d'émission en fonction de la variation de la charge. Avec Queue-MAC, nous arrivons à maintenir cette taille à un niveau très faible, montrant ainsi l'efficacité indéniable de la partie TDMA dynamique.

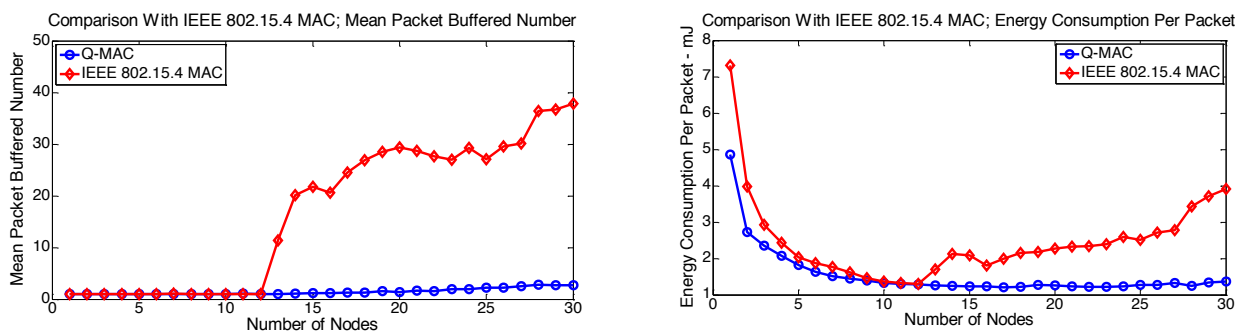
L'efficacité énergétique est toujours l'un des critères les plus importants des réseaux de capteurs sans fil. Nous avons mesuré cette efficacité selon l'équation (4) qui mesure l'énergie consommée par paquet transmis avec succès et divisée ensuite par le taux du succès.

$$\frac{E_{Total}}{N_{success}} / \frac{N_{success}}{N_{Total}} \quad (4)$$

où  $E_{Total}$  est l'énergie totale consommée par la tête du cluster (coordinateur). Nous supposons que la part de l'énergie consommée par des nœuds enfants est une constante pour les deux protocoles.  $N_{success}$  est le nombre de paquets reçus avec succès,  $N_{Total}$  le nombre total de paquets générés.

La raison pour laquelle nous divisons l'énergie consommée par paquet réussi par le taux de succès est pour mesurer équitablement l'efficacité. En effet, selon l'efficacité des protocole MAC, pour un nombre donné de paquets générés, un protocole moins efficace peut rejeter une proportion importante de paquets et ne transmet avec succès qu'une petite partie. Il se peut que ce mauvais protocole consomme moins d'énergie par paquet car peu de paquets ont été transmis. Alors en la divisant par le taux de succès, nous donnons une mesure plus juste.

La figure 4 (droite) compare l'efficacité énergétique des deux protocoles. Lorsque la charge est faible, l'efficacité est moindre car la période active consomme autant d'énergie sans rien transmettre que de transmettre des paquets. Cette consommation par paquet diminue lorsque le trafic augmente jusqu'au point où la période active de IEEE802.15.4 est saturée. Au-delà, l'efficacité de IEEE802.15.4 se dégrade à cause du taux de succès de plus en plus faible. Tandis que Queue-MAC arrive à maintenir son efficacité jusqu'à une charge élevée. Ce qui démontre encore une fois que Queue-MAC a la capacité d'assurer à la fois une bonne performance lorsque la charge est élevée et une efficacité énergétique.



**FIG. 4** - Taille moyenne de la file d'attente (gauche) et efficacité énergétique (droite) de Queue-MAC et IEEE802.15.4

Comme expliqué précédemment Queue-MAC présente deux défauts. Premièrement, comme le routage entre voisins n'est pas nativement inclus dans sa conception, son utilisation dans des réseaux de grande taille peut s'avérer compliquée car la synchronisation entre des têtes de clusters n'est pas toujours aisée. Deuxièmement, comme ce qu'on peut constater de la figure 4 (droite), Queue-MAC souffre du problème de l'énergie lorsque la charge est très faible à cause de la période CSMA/CA fixe. Réduire cette période au minimum (équivalent d'un slot) peut réduire la consommation d'énergie mais risque de former un goulot d'étranglement lors du trafic élevé car il y

a peu de probabilité qu'un nœud puisse gagner l'accès au médium par CSMA/CA lorsque le nombre de nœuds chargés est important.

En général, Queue-MAC fait mieux par rapport aux protocoles de l'état de l'art en terme de l'efficacité de transmission des rafales de trafic élevé, tout en garantissant une faible consommation lors du trafic faible (conditionné par la taille de la période CSMA/CA). Ce défaut peut ne pas être gênant lorsqu'il y a un minimum de trafic dans le réseau (typiquement la transmission périodique avec de longues périodes) car il suffit de caler la durée de la période CSMA/CA de Queue-MAC à la quantité de paquets périodiques. Mais cette consommation lors du faible trafic est probablement plus élevée par rapport à la conception de B-MAC ou X-MAC.

Dans la section suivante, nous présentons S-CoSenS qui permet de remédier partiellement à ce problème de manque d'efficacité lors du trafic faible.

## 4 S-CoSenS

S-CoSenS (Sleep, Collect, then Send as a burst) [12] est une autre solution que nous avons développée pour résoudre nos deux problèmes posés précédemment : auto-adaptation à la variation du trafic et transmission plus efficace que CSMA/CA lors de la charge élevée du trafic.

S-CoSenS décompose un cycle en trois périodes : période de sommeil (SP), période de collecte (WP) et période de transmission en rafale ou en mode « burst » (TP). L'auto-adaptation à la variation du trafic est réalisée grâce à l'estimation en ligne par le routeur du trafic entrant pendant la période de collecte, permettant ainsi de redimensionner la durée de la période de collecte du prochain cycle.

Pour traiter efficacement le problème de la charge élevée, à la différence de Queue-MAC qui s'appuie sur TDMA, S-CoSenS utilise la technique de CSMA/CA mais avec la transmission en rafale (mode « burst ») entre routeurs. Si un routeur a plusieurs paquets à transmettre, au lieu de passer par la phase d'attente de la fenêtre de contention aléatoire et le CCA pour la transmission de chaque paquet, il n'y a que le premier paquet qui suit cette procédure CSMA/CA. Une fois ce premier paquet transmis (et acquitté), les paquets suivants sont transmis l'un après l'autre, comme un train de paquets (mode « burst »).

Par rapport à Queue-MAC, il intègre en plus nativement dans sa conception la considération du routage entre les nœuds voisins (inspirée de B-MAC). Précédant la transmission en rafale, le routeur émetteur doit commencer par transmettre un long préambule à la façon du LPL de B-MAC.

Par ailleurs, S-CoSenS est conçu en supposant que le réseau est structuré en deux niveaux hiérarchisés : niveau nœuds simples et niveau routeurs. Ce qui correspond à une organisation en arbre « cluster-tree », topologie logique la plus fréquente des réseaux de capteurs sans fil. Afin que la transmission des paquets entre routeurs ne soit pas perturbée par celle des nœuds simples, nous avons défini deux priorités qui sont réalisés grâce aux différentes valeurs d'attente de la fenêtre de contention (paramètre *macMinBE* du standard IEEE802.15.4).

Enfin, comme dans la conception de Queue-MAC, la considération d'implémentabilité nous a conduit à choisir IEEE802.15.4 comme protocole de base. S-CoSenS est donc directement implémentable au dessus du standard.

La sous-section suivante détaille le fonctionnement de S-CoSenS.

### 4.1 Principe de S-CoSenS

#### Topologie en arbre « cluster-tree » à deux niveaux

Selon le standard IEEE802.15.4 et la pratique industrielle définie par l'Alliance Zigbee [13], deux types de nœuds peuvent exister : FFD (Full Functional Device) et RFD (Reduced Functional Device). Seul les nœuds FFD assurent le routage. L'ensemble forme un réseau à deux niveaux selon l'architecture en arbre du type « cluster-tree ».

Dans la conception de S-CoSenS nous adoptons cette organisation du réseau qui correspond à la plupart des besoins d'applications réelles. En plus, par rapport à une organisation plate où chaque nœud est aussi un routeur, le problème de routage est grandement simplifié car un grand réseau complexe est devenu ainsi petit du point de vue du routage.

Les nœuds simples RFD sont associés aux nœuds routeurs FFD selon la procédure d'association de Zigbee, formant ainsi une relation enfants-parent. Notons dès à présent que dans S-CoSenS, le protocole de routage d'arbre hiérarchique de Zigbee est modifié selon nos travaux précédents [14] afin d'avoir la possibilité de choisir un chemin qui n'est pas forcément le parent. Ceci permet d'améliorer la fiabilité du routage et réduire le délai de bout-en-bout quand le « duty-cycle » des nœuds routeurs est pris en compte.

### Protocole MAC de S-CoSenS

Un nœud simple est tout le temps endormi sauf quand il a un paquet à transmettre. Comme les nœuds simples représentent la majorité des nœuds d'un réseau, cette approche permet d'économiser globalement l'énergie de façon très efficace.

Un nœud routeur adopte un fonctionnement en cycle, illustré par la figure 5.

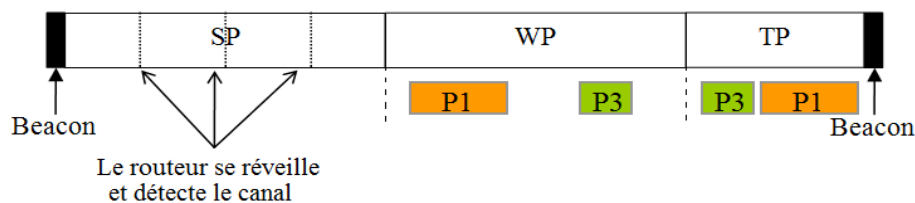


FIG. 5 - Réseau en arbre « cluster-tree » à deux niveaux

Le cycle est balisé par des « beacons » diffusés par le routeur. Chaque cycle est composé de trois périodes. La période SP (Sleep Period) correspond à la période du sommeil du routeur durant laquelle le routeur se réveille périodiquement, avec un intervalle de réveil constant, pendant une très courte durée pour détecter d'éventuelles transmissions provenant des routeurs voisins qui raccourciraient la période du sommeil. La période WP (Waiting Period) permet au routeur de collecter des paquets transmis par leur enfants (nœuds simples, mais aussi d'autres routeurs situés à un niveau inférieur de l'arbre). Dans S-CoSenS, nous fixons l'intervalle SP+WP à une constante, mais la frontière entre les deux périodes bouge dynamiquement en fonction de la charge du trafic réseau. Si le trafic est faible, la période SP s'allonge tandis que la période WP se raccourcit. Si le trafic est fort, la période SP diminue, voire disparaît, au profit de l'augmentation de la période WP. La période TP (Transmission Period) correspond au temps qu'il faut pour transmettre tous les paquets collectés par le routeur dont le premier en CSMA/CA avec un préambule à la façon B-MAC de la durée de transmission au moins égale à l'intervalle de réveil, suivi du train de paquets sans CSMA/CA (mode burst). Notons que ce mode « burst » peut être perturbé, soit suite à une erreur de transmission (ou collision), soit par un paquet émis par un nœud simple (même si la probabilité est faible). Dans ce cas particulier, le train peut être interrompu et repris par la même procédure de CSMA/CA pour le premier paquet interrompu, suivi du reste du train des paquets restants.

Les nœuds simples transmettent, pendant WP, selon CSMA/CA non-slotté de IEEE802.15.4 avec  $macMinBE=3$  (valeur par défaut pour le calcul de la fenêtre de contention). Les routeurs transmettent leur premier paquet de leur file d'attente des paquets collectés, durant TP, selon CSMA/CA non-slotté mais avec  $macMinBE=2$ , ce qui leur donne une priorité probabiliste d'accès au médium par rapport aux nœuds simples. Notons que CSMA/CA slotté aurait pu être adopté mais dans la version actuelle de S-CcSenS, nous avons préféré la version non-slotté afin de préserver la simplicité maximale lors de l'implémentation.

Les nœuds simples se synchronisent grâce au « beacon » dans lequel le routeur indique la durée de la période SP du cycle qui commence. Un nœud simple, qui a un paquet à transmettre, doit donc se réveiller, attendre l'arrivée d'un « beacon » de son parent, attendre l'écoulement de la période SP, puis transmettre en utilisant CSMA/CA son paquet. Il est à noter que contrairement à la plupart des protocoles MAC synchrones, un nœud simple dans S-CoSenS n'a besoin de se synchroniser que lorsqu'il a un paquet à transmettre, minimisant ainsi sa consommation d'énergie.

L'auto-adaptation à la variation du trafic se fait par un algorithme simple, basé sur l'estimation du trafic et est exécuté par chaque routeur. L'estimation du trafic est calculée selon la moyenne exponentielle pondérée glissante du nombre de paquets reçus, noté par  $N_{avg}$ , pendant chaque WP jusqu'à la (k-1)ème WP. Cet algorithme est similaire à l'estimation du RTT (Round Trip Time) dans une session TCP. Notons par  $n_{k-1}$  le nombre de paquet reçus pendant la (k-1)ème WP, nous obtenons l'estimation de la valeur de  $N_{avg}$  à la kème WP par l'équation (5).

$$N_{avg} = (1 - \alpha)N_{avg} + \alpha n_{k-1} \quad (5)$$

où  $\alpha$  est le poids pour suivre de plus ou moins près la variation du trafic à court (ou long) terme dont la valeur peut être fixée par l'utilisateur selon la nature du trafic. Dans [12] nous avons utilisé deux seuils différents  $\alpha_1=0,008$  et  $\alpha_2=0,01$  dont les résultats seront présentés dans la sous-section suivante. Nous invitons le lecteur à consulter [12] pour les détails sur l'estimation de la valeur de WP de chaque cycle (qui est proportionnelle à  $N_{avg}$ ). Pour une raison pratique, la valeur de WP est bornée. Dans nos évaluations qui suivent, elle est bornée par  $WP_{min}=0,01$  secondes et  $WP_{max}=0,07$  (ou  $WP_{max}=0,7$ ) secondes.

### **Routage entre routeurs voisins**

Bien que la conception de S-CoSenS se focalise sur les mécanismes MAC, elle intègre aussi la considération du routage entre les routeurs voisins. Le principe est inspiré de B-MAC et fonctionne de la manière suivante. Chaque routeur, une fois entré dans sa période TP, doit s'assurer que le routeur prochain saut se réveille. Pour cela, il diffuse un préambule à la façon B-MAC dont la durée de transmission est au moins égale à l'intervalle de réveil. Tout routeur, même dans sa période SP, doit se réveiller avec cet intervalle pour détecter le médium. Le routeur émetteur s'assure alors qu'il y aura un prochain saut prêt à l'écouter. Bien évidemment, si un routeur voisin est dans sa période WP, il reçoit le paquet transmis ainsi que le train du reste des paquets s'ils lui sont destinés. Si le routeur voisin est en sa période TP, le routeur émetteur va détecter que le médium est occupé et tentera de nouveau selon CSMA/CA.

Ce mode de se synchroniser au rythme de l'intervalle de réveil des routeurs voisins représente un intérêt de ne pas avoir besoin de la synchronisation explicite, évitant ainsi la complexité NP-dure de l'ordonnancement des cycles entre des routeurs voisins. C'est aussi une façon d'assurer le passage à l'échelle de notre solution.

Il est à noter que nous avons choisi de transmettre un long préambule à la façon de B-MAC au lieu de préambule « échantillonné » à la façon de X-MAC pour une raison de robustesse du routage. En effet, X-MAC intègre l'adresse du prochain saut dans son préambule pour éviter de réveiller inutilement les autres nœuds voisins non-concernés (problème de sur-écoute). Le routeur prochain saut interrompt la transmission d'autres « échantillons » de préambule par l'envoi d'un acquittement. De ce fait, X-MAC ne permet que l'unicast et le routage uni-chemin. Dans S-CoSenS, comme les paquets collectés dans un routeur ne sont pas nécessairement pour la même destination, ce routage uni-chemin ne convient pas. Par ailleurs, attendre que le prochain saut concerné se réveille peut aussi prendre du temps. Dans S-CoSenS, nous pratiquons le routage opportuniste en exploitant tous les prochains sauts possibles au lieu d'un seul. Le prix à payer est une efficacité moindre, un peu similaire à ce que B-MAC peut perdre par rapport à X-MAC. Mais nous verrons

dans la sous-section suivante que ce petit sacrifice est largement compensé par le fait de transmettre en rafale des paquets.

Une autre conséquence est que S-CoSenS autorise un routeur de choisir non plus systématiquement son nœud parent, mais le premier routeur voisin qui se réveille et qui peut faire progresser le ou les paquets vers la ou les destinations. De ce point de vue, le routage ne suit pas nécessairement l'arbre établi lors de l'association des nœuds parent-enfants, mais est capable d'explorer tous chemins possibles, rendant le routage plus robuste et réduisant potentiellement le délai de bout-en-bout. Notons qu'une approche similaire est adoptée dans [8] pour améliorer X-MAC.

## 4.2 Evaluation de performances

Le scénario simulé est un réseau multi-saut composé de 25 routeurs et 125 nœuds simples, déployé sur une zone rectangulaire de  $1000 \times 1000 \text{m}^2$ . Cette surface est subdivisée en 25 zones. Chacune d'elles est couverte par un routeur et 5 nœuds simples déployés aléatoirement. Le récepteur (sink) est localisé au milieu.

Nous avons implémenté sous OPNET S-CoSenS, S-MAC et X-MAC afin de comparer leurs performances. Le modèle de consommation d'énergie correspond à celui des nœuds MicaZ.

Les figures 6 et 7 montrent respectivement l'efficacité énergétique, le délai de bout-en-bout et le débit (en bit/s) des trois protocoles simulés en fonction de la variation de la charge (paquets de taille de 500 bits, générés par des nœuds simples vers le récepteur selon un processus de Poisson).

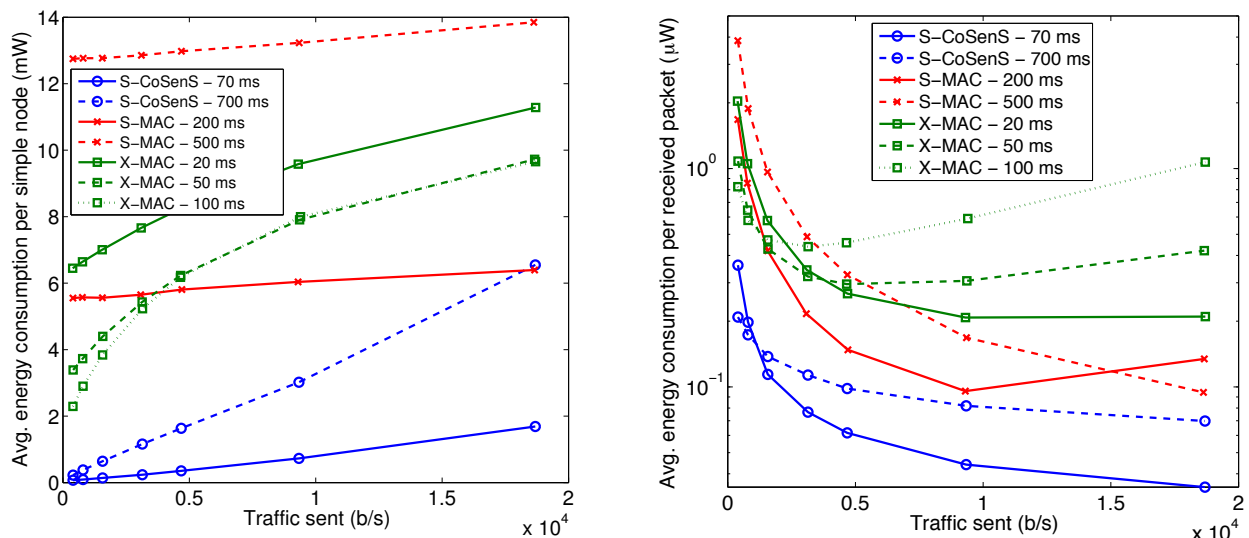


FIG. 6 – Energie consommée par nœuds simples et efficacité énergétique selon l'équation 4 (droite)

S-CoSenS – 70ms et 700ms correspondent à la valeur de SP+WP, et avec l'intervalle de réveil de 10ms. S-MAC - 200ms et 500ms correspondent aux périodes actives de 200ms et 500ms sur un cycle d'une seconde. X-MAC – 20, 50 et 100ms correspondent aux périodes de réveil, toutes avec un intervalle de 7ms qui sépare les échantillons du préambule.

La figure 6 montre que S-CoSenS exhibe la meilleure efficacité énergétique grâce à l'endormissement des nœuds simples, qui est proche de l'optimale. Au fait, quand on regarde séparément la consommation des nœuds simples (Fig.6 gauche) et des routeurs (courbes non illustrées ici à cause de l'espace limité), S-CoSenS consomme le moins pour les nœuds simples et plus que X-MAC (pour le cas de 700ms). La figure 6 (droite) montre l'énergie consommée par nœud en moyenne, nœuds simples et routeurs confondus. On voit que quand le trafic est faible, tous les protocoles exhibent une efficacité moindre à cause des périodes actives non totalement utilisées pour transporter des données. L'efficacité augmente (la valeur diminue) jusqu'à une certaine

charge, puis diminue de nouveau avec l'augmentation de la charge pour les protocoles autres que S-CoSenS. Ceci est dû à la diminution du taux de succès que l'on peut voir à la figure 7.

La figure 7 (gauche) montre que le délai de bout-en-bout de S-CoSenS pour une durée SP+WP faible (ici 70ms) est comparable aux autres. Le grand délai pour le cas de 700ms est dû au fait qu'un nœud simple doit attendre la prochaine période WP, qui est au pire 700ms à chaque saut.

La figure 7 (droite) montre que le débit (throughput) de S-CoSenS reste à son meilleur niveau avec l'augmentation de la charge. Nous remarquons que S-MAC, avec une longue période active de 500ms (duty-cycle=50%), se comporte aussi très bien, dû à sa bande passante importante (la moitié du débit nominal). Mais ceci est au détriment de la consommation d'énergie (qui est la plus élevée de tous en cas de faible charge). Par ailleurs, nos simulations révèlent aussi que X-MAC souffre plus de l'augmentation de la charge que les autres protocoles. En effet, nous observons que les collisions entre les échantillons de préambules augmentent rapidement, qui est la cause principale de la dégradation du débit.

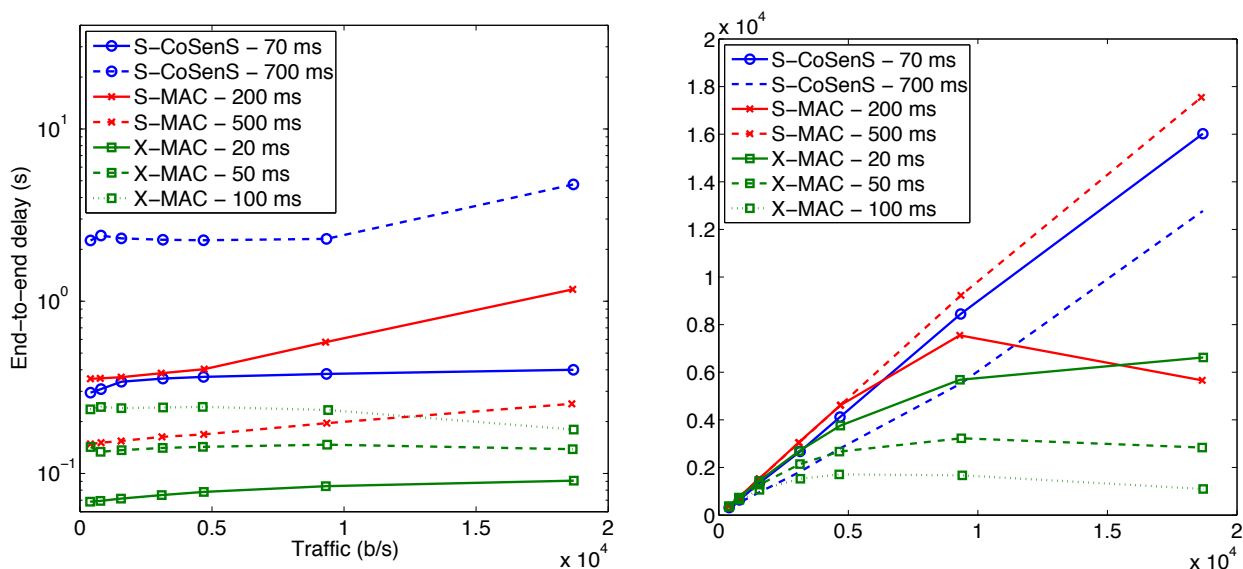


FIG. 7 - Délai de bout-en-bout (gauche) et débit (droite)

## 5 Conclusion et perspectives

Dans ce papier nous avons proposé trois contributions. Tout d'abord nous avons mené une analyse des principaux protocoles à « duty-cycle » et soulevé les problèmes de l'auto-adaptation à la variation du trafic et de la transmission inefficace du trafic élevé. Ensuite Queue-MAC a été présenté qui auto-adapte à la variation du trafic grâce à l'indicateur de la file d'attente et au transport plus efficace du trafic élevé avec la combinaison CSMA/TDMA. Enfin le principe de S-CoSenS est décrit qui utilise le mode « burst » pour résoudre le problème du manque d'efficacité de la transmission du trafic élevé avec CSMA/CA, l'estimation en ligne du trafic pour auto-adapter à la variation du trafic, et le principe du LPL (low power listening) pour synchroniser les routeurs voisins lors du routage.

Les protocoles MAC étant relativement bien étudiés, Il nous paraît intéressant d'étendre nos résultats selon deux directions futures : l'intégration des nœuds mobiles dans la conception des protocoles MAC à cycle actif, et la conception des protocoles de routage à QoS sur des protocoles MAC à cycle actif. [15] est un exemple de la première direction qui a étendu X-MAC afin de pouvoir choisir le premier nœud voisin qui se réveille comme le prochain saut, au lieu du nœud fixe qui peut ne pas être dans le voisinage du nœud émetteur à cause de sa mobilité depuis la dernière mise à jour de sa table de voisins. [8] est un exemple de la deuxième direction de recherche qui

propose un routage opportuniste permettant d'exploiter un sous-ensemble de voisins qui se réveillent en premier et font progresser le paquet vers sa destination.

## 6 Remerciements

Ce travail a été partiellement soutenu par le projet ANR blanc international QUASIMODO entre l'Université de Lorraine et l'Université de Zhejiang (Chine), sous les contrats N°ANR2010INTB020601 et N°NSFC61061130563. L'auteur remercie Shuguo Zhuo (Université de Zhejiang) et Bilel Nefzi (Université de Lorraine), les deux contributeurs respectifs de Queue-MAC et S-CoSenS.

## 7 Références

- [1] Prayati A., Antonopoulos C., Stoyanova T., Koulamas C., et Papadopoulos G. (2010), *A modeling approach on the TelosB WSN platform power consumption*, Journal of Systems and Software, vol. 83, No. 8, pp. 1355–1363, 2010.
- [2] IEEE 802.15.4 Standard (2006), *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE: Piscataway, NJ, USA, 2006.
- [3] Kleinrock, L. et Tobagi, F. (1975), *Packet Switching in Radio Channels: Part I-- Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics*, IEEE Trans. on Communications, vol.23, no.12, pp. 1400- 1416, Dec. 1975.
- [4] Koubaa A., Alves M., Attia M., et Nieuwenhuys A.V. (2007), *Collision-free beacon scheduling mechanisms for IEEE 802.15.4/ ZigBee cluster-tree wireless sensor networks*, Proceedings of the 7th International Workshop on Applications and Services in Wireless Networks, May 2007.
- [5] Ye W., Heidemann J., et Estrin D. (2002), *An energy-efficient MAC protocol for wireless sensor networks*, Proceedings of the IEEE Infocom, pages 1567–1576, 2002.
- [6] Polastre J., Hill J., et Culler D. (2004), *Versatile low power media access for wireless sensor networks*, Proceedings of SenSys'04, pages 95–107, New York, NY, USA, 2004.
- [7] Buettner M., Yee G., Anderson E., et Han R. (2006), *X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks*. Proceedings of SenSys'06, pp. 307-320, Nov. 2006.
- [8] Landsiedel O., Ghadimi E., Duquennoy S., et Johansson M. (2012), *Low Power, Low Delay: Opportunistic Routing meets Duty Cycling*, Proceedings of IPSN2012, April 16-19, Beijing, China.
- [9] Bux W. (1981), *Local-area sub networks: A performance comparison*, IEEE Trans. on Communications, Vol. 29, No. 10, pp. 1465–1473, 1981.
- [10] Zhuo S.G., Song Y.Q., Wang Z., et Wang Z.B. (2012), *Queue-MAC: A queue-length aware hybrid CSMA/TDMA MAC protocol for providing dynamic adaptation to traffic and duty-cycle variation in wireless sensor networks*, Proceedings of the IEEE WFCSS2012, May 21-24, 2012, Lemgo/Detmold, Germany.
- [11] STM32W108 datasheet: <http://www.datasheetarchive.com/STM32W108-datasheet.html>
- [12] Nefzi B. (2011), *Mécanismes auto-adaptatifs pour la gestion de la qualité de service dans les réseaux de capteurs sans fil*, Thèse de doctorat INPL Nancy, 21/09/2011.
- [13] Zigbee (2008), *Specification Document 053474r17*, <http://www.zigbee.org>, January 2008.
- [14] Nefzi B., et Song Y.Q. (2007), *Performance Analysis and improvement of ZigBee routing protocol*, Proceedings of the 7th IFAC FeT'2007, Toulouse, France, 2007.
- [15] Ba P.D, Gueye B., Ning I. et Noel T. (2011), *MoX-MAC: a low power and efficient access delay for mobile wireless sensor networks*, Proceedings of the IFIP WMNC'2011, Toulouse, France, Oct. 2011.