

Comparaison de différentes couches MAC pour les réseaux de capteurs

Julien Vaudour, Vincent Gauthier
Rapport de Recherche INT N°06006RST
GET/INT;UMR5157 SAMOVAR

Institut National des Télécommunications, Evry, France
{julien.vaudour, vincent.gauthier}@int-evry.fr

Abstract— Les réseaux de capteurs sans fil apportent de nouvelles perspectives. Une des principales contraintes est l'économie d'énergie, la batterie devant durer pendant toute la vie du capteur. La principale source de consommation d'énergie étant le radio transmetteur du capteur, plusieurs solutions au niveau de la couche MAC ont été proposées afin d'économiser l'énergie au niveau de la radio. Le but de cet article est de comparer différentes solutions compatibles ZigBee. A partir de cette comparaison nous avons imaginé une nouvelle couche MAC.

Keywords— Réseau de capteurs, routage, IEEE 802.15.4, SMAC, TMAC, BMAC

I. LES RÉSEAUX DE CAPTEURS

Les réseaux de capteurs sans fil apportent une perspective intéressante : celle de réseaux capables de s'autoconfigurer, de se gérer sans qu'il y ait besoin d'interventions humaines. De plus, les critères de performance pour un réseau de capteurs différent de ceux des réseaux classiques et donc les solutions à apporter sont nouvelles. En effet, les capteurs sans fils ont vocation à devenir des objets "banaux" et donc doivent pouvoir s'utiliser facilement. Le réseau doit devenir transparent pour l'utilisateur. Les réseaux de capteurs trouvent des applications dans des réseaux sans fil et sans infrastructure ne nécessitant pas un débit élevé tel que la domotique ou l'agriculture intelligente. La réalisation de ce type de réseau requiert la mise en œuvre de techniques développées pour les réseaux ad-hoc; cependant, la plupart des protocoles développés pour ceux-ci ne sont pas transposables tels que ceux des réseaux de capteurs.

Les différentes problématiques autour des réseaux de capteurs concernent l'autoconfiguration d'un tel réseau, dans un contexte de mobilité et l'économie des ressources énergétiques de ses nœuds (l'objectif est qu'un capteur tienne quatre ans avec une pile AAA). Pour économiser au mieux la batterie d'un nœud, il faut que ses transmetteurs radio soient coupés la plupart du temps. Ceci pose le problème de la synchronisation des nœuds et la répartition des périodes de réveil. Il faut donc une couche MAC permettant aux nœuds d'avoir des phases de sommeil, tout en ne nuisant pas trop à la communication.

Dans ce document, nous nous intéressons uniquement aux solutions "compatibles ZigBee", cette étude se faisant dans le cadre d'un projet pour lequel Bluetooth, du fait de sa limitation en nombre de nœuds, n'est pas adapté.

II. LES COUCHES MAC POUR LES RÉSEAUX DE CAPTEURS

A. La norme 802.15.4 (LR-WPAN)

La norme 802.15.4 ([3] et [2]) est la norme retenue par ZigBee Alliance.

La norme IEEE 802.15.4 a été spécialement définie en fonction des caractéristiques des réseaux de capteurs, un faible débit et une faible consommation électrique. Elle décrit le fonctionnement de la couche physique, et de la couche MAC. Pour la couche physique deux bandes de fréquence ont été retenues la bande 2.4 GHz et la bande 868-915 MHz, c'est au total 27 canaux de communication avec 3 différents débits possibles soit 16 canaux à 250 kbps dans la bande 2.4 GHz, 10 canaux à 40 kbps dans la bande 915 MHz, et 1 canal à 20 kbps dans la bande 868 MHz. Les réseaux qui supporteront la norme 802.15.4 pourront librement choisir d'utiliser l'un des 27 canaux utilisables en fonction de sa disponibilité et du débit recherché.

Pour la couche MAC, deux topologies sont supportées par le protocole 802.15.4, la topologie en étoile, et la topologie point à point. Dans la topologie en étoile, les communications s'établissent directement entre le nœud central (coordinateur) et les capteurs, le coordinateur étant le nœud qui initie et gère les communications dans le réseau. La topologie point à point (Peer to Peer) permet à chaque nœud du réseau de communiquer avec n'importe quel autre nœud, ce qui permet de réaliser des réseaux ayant une architecture beaucoup plus complexe.

La topologie point à point nécessite l'utilisation d'un protocole de routage qui doit être géré par la couche réseau, mais cette partie n'est pas définie dans le protocole IEEE 802.15.4.

Le standard LR-WPAN permet l'utilisation de deux méthodes d'accès au canal de communication; la première utilise une méthode de type CSMA-CA similaire à celle utilisée dans les réseaux 802.11, la deuxième est une méthode de type slotted CSMA-CA (accès multiples durant un intervalle de temps). La méthode Slotted CSMA-CA utilise des supertrames; Celles-ci sont définies par le coordinateur, et ont pour principal avantage de permettre la synchronisation de l'ensemble des éléments entre eux, et ainsi d'économiser l'énergie de chacun des nœuds du réseau. Les supertrames sont tout d'abord divisées en deux parties : une partie active et une partie inactive; durant la partie

inactive le coordinateur n'interagit pas avec le réseau et se place en mode veille. Chaque supertrame commence par un beacon qui est envoyé par le coordinateur du réseau, et qui permet la synchronisation entre les nœuds et le coordinateur. Le beacon est immédiatement suivi par une zone appelée *Contention Access Period* (CAP), qui permet à chaque élément du réseau d'envoyer ou de recevoir des trames de commandes et de données.

L'accès au canal durant la période CAP suit le mécanisme CSMA-CA : l'émetteur écoute le canal pendant une durée aléatoirement tirée (algorithme de backoff) puis émet si le canal est libre. L'algorithme CSMA-CA induit une consommation électrique importante (surtout durant les périodes de fort trafic), ce qui a conduit à réduire l'intervalle valeur que peut prendre l'algorithme de backoff de 0 à 2. Ce procédé réduit la période de contention pendant laquelle l'émetteur doit écouter le canal.

Une autre zone optionnelle disponible dans la supertrame (CFP) est utilisée pour les transmissions nécessitant de la qualité de service. Le CFP est divisé en slots, dont le coordinateur gère l'allocation en fonction des besoins. Une station qui se voit attribuer un slot est assurée qu'aucune autre station ne transmettra durant cette période. L'allocation d'un slot fait suite à une négociation entre un nœud et le coordinateur. Si la négociation aboutit le coordinateur informe le nœud en plaçant une information (stream index) dans le prochain beacon (l'algorithme d'allocation n'est pas défini dans le standard).

Le mode de communication utilisant les supertrames garantit un très faible taux d'utilisation des ressources (souvent inférieur à 10%), car les nœuds formant le réseau ont un temps de veille garanti très long (les nœuds ne sont actifs que durant la période CAP).

A.1 Transfert de données

La norme IEEE 802.15.4 définit trois modèles transactionnels possibles:

- Transfert de données du coordinateur vers un nœud du réseau,
- Transfert de données d'un nœud du réseau vers le coordinateur,
- Transfert de données de deux nœuds du réseau entre eux.

Dans le cas d'une topologie en étoile, seules les deux premières transactions sont possibles car les échanges se font exclusivement avec le coordinateur du réseau (comme dans un piconet BlueTooth). Alors que, dans le cas d'une topologie point à point, les trois types de transactions sont disponibles.

A.1.a Transfert vers le coordinateur. Un nœud, désirant communiquer avec le coordinateur dans un réseau utilisant la structure des supertrames, doit d'abord se synchroniser en utilisant le beacon comme référence, puis envoyer un paquet de données durant un slot de temps et attendre un acquittement de la part du coordinateur(optionnel). Dans un réseau n'utilisant pas la structure supertrame, le nœud émet une trame de données selon le mécanisme CSMA-CA, quand il le désire.

A.1.b Transfert du coordinateur vers un nœud. Quand le coordinateur souhaite émettre une trame vers un nœud dans un réseau utilisant le mécanisme des supertrames, il doit attendre que ce nœud soit synchronisé et qu'il demande (Data request) qu'on lui envoie les données en attente.

Le même mécanisme est mis en place dans un réseau n'utilisant pas les supertrames, le nœud émet alors régulièrement une trame(Data Request), pour demander au coordinateur, s'il n'a pas de données en attente à lui envoyer.

A.1.c Transfert point à point. Dans le cas du transfert de données point à point, les nœuds doivent être constamment synchronisés et écouter constamment le canal radio, alors le mode de transfert de données ne peut être que de type CSMA-CA sans slot de temps et sans structure de supertrames. On se retrouve alors dans les conditions d'un réseau ad hoc classique.

A.2 Mediation Device

La consommation électrique des composants dans un réseau est grandement réduite, si on utilise la technologie *slotted CSMA-CA*, avec une structure en supertrame. Mais cette technologie ne peut être déployée qu'avec une topologie en étoile. La topologie en étoile n'est utilisée que dans le cas où on a de petits réseaux. Les réseaux utilisant cette topologie sont donc restreints à des applications bien particulières, alors qu'une topologie point à point(peer to peer) permet le déploiement de réseaux de grande envergure.

Dans les réseaux point à point, l'accès au canal de transmission ne peut se faire qu'avec une technologie CSMA-CA simple. Mais les nœuds étant actifs tout le temps (comme dans un réseau Wifi), cette solution est coûteuse en énergie. Pour y remédier, une technologie hybride appelée *Mediation Device* compatible avec la norme IEEE 802.15.4 peut être déployée dans les réseaux de capteurs utilisant une technologie CSMA-CA pour permettre de sauvegarder l'énergie des nœuds.

Un nœud du réseau, le coordinateur ou un autre nœud choisi aléatoirement, va jouer le rôle de médiateur entre tous les nœuds se trouvant dans son infosphère radio.

Quand un nœud X veut communiquer avec un nœud Y, il envoie une requête au médiateur. Tous les nœuds du réseau interrogent régulièrement le médiateur pour savoir si on cherche à les joindre. Donc quand le nœud Y interroge le médiateur, celui ci l'avertit que le nœud X veut entrer en contact avec lui. X et Y entament alors une communication point à point.

B. SMAC

B.1 généralités

On peut avoir un phénomène de station cachée quand par exemple, 2 stations A et B qui ne se voient pas, voient par contre toutes les 2 une troisième station C comme sur la figure 1. Si A et B veulent toutes les deux envoyer des paquets à C, A ne voyant pas B et B ne voyant pas A, elles pensent toutes les deux que le médium pour envoyer des

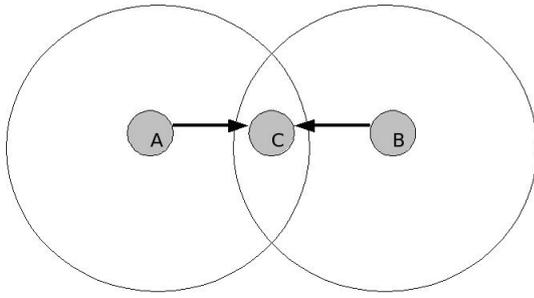


Fig. 1. Problème de stations cachées

informations à C est libre et il va donc y avoir collision.

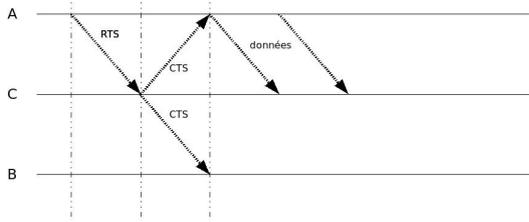


Fig. 2. Mécanisme de RTS-CTS

B.1.a . Dans un mécanisme de type RTS-CTS (*Request To Send, Clear To Send*), avant d'envoyer des données, l'émetteur envoie une trame RTS au récepteur qui lui répond par une trame CTS pour accepter ou refuser la conversation (si le CTS n'est pas reçu, on estime que le RTS a subi une collision). Les paquets RTS et CTS contiennent des informations sur la taille des données à transmettre, ainsi les voisins de l'émetteur des données (qui entendent donc le paquet RTS) et du récepteur (qui entendent donc le CTS) peuvent calculer une estimation du temps durant lequel ils ne devront pas émettre sur le canal radio. On résoud ainsi le problème des stations cachées et on assure ainsi une réservation du canal. Ainsi, en reprenant l'exemple précédent, le nœud A, comme cela est illustré sur la figure 2 peut réserver le canal et ainsi empêcher B d'envoyer des paquets au nœud C qui provoqueraient une collision.

B.2 SMAC

B.2.a Bases. SMAC (*Sensor MAC*) [6] est un protocole de couche de liaison relativement similaire au protocole 802.11 et qui utilise une méthode d'accès au canal de type CSMA-CA RTS/CTS (*Request-To-Send, Clear-To-Send*) qui permet d'éviter les collisions et le problème de station cachée. La principale innovation, apportée par ce protocole, est d'avoir développé un mécanisme de mise en veille distribué à chaque nœud du réseau dans le but de réduire la consommation énergétique des équipements réseau. Avec le protocole S-MAC chaque nœud peut périodiquement se mettre en veille (cf figure 3), la principale difficulté est alors de synchroniser les nœuds entre eux pour que la communication soit toujours possible. S-MAC définit donc une frame composée d'une période de sommeil et d'une période d'éveil. Si

un nœud est en train de transmettre ou recevoir des données, il ne suit pas son ordonnanceur de sommeil avant la fin de la transmission.

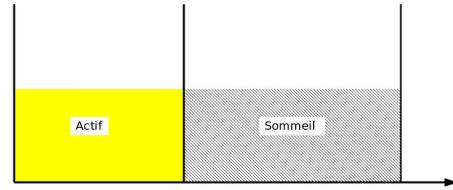


Fig. 3. Protocole S-MAC

B.2.b synchronisation. Pour synchroniser les nœuds entre eux, le protocole S-MAC permet à chacun des nœuds d'émettre des paquets SYNC qui permettent aux autres nœuds de se synchroniser. Le paquet SYNC est broadcasté à tous les voisins immédiats. La période active est divisée en deux phases, la première étant réservée à la synchronisation.

Quand un nœud se réveille, il commence par écouter. Si il n'entend rien pendant un certain temps, il choisit un ordonnancement de frame et le transmet par un paquet SYNC qui contient le temps jusqu'à son endormissement. Si au démarrage, le nœud reçoit un paquet SYNC d'un autre nœud, il suit l'ordonnancement défini dans ce paquet et transmet son propre SYNC en conséquence. Les nœuds retransmettent leur SYNC de temps en temps. Si un nœud a déjà adopté un ordonnancement et qu'il reçoit un paquet SYNC d'un autre nœud avec un autre ordonnancement, alors il doit adopter les 2 ordonnancements. Il doit aussi transmettre un paquet SYNC à l'autre nœud pour que celui-ci apprenne la présence d'un autre ordonnancement. Le fait d'adopter les deux ordonnancements signifie que le nœud va se réveiller au début des deux frames. Il faut donc éviter qu'un nœud adopte plusieurs ordonnancements.

B.2.c Overhearing Avoidance. Chaque nœud maintient un vecteur d'allocation du réseau (*network allocation vector NAV*) dans lequel il indique l'activité de ses voisins. Chaque paquet de données contient un champ indiquant le temps restant. Lorsqu'un nœud reçoit un paquet de données qui ne lui est pas destiné, il met à jour son NAV. A chaque intervalle de temps, la valeur du NAV est décrétementée (si elle est non nulle). Ainsi une valeur non nulle du NAV signifie qu'il y a une transmission. Le nœud peut donc s'endormir et ne se réveiller ensuite que lorsque le NAV devient nul.

C. TMAC

C.1 Bases

Dans T-MAC (*Timeout MAC*) [5], chaque nœud se réveille périodiquement pour communiquer avec ses voisins. On a donc, comme dans S-MAC des périodes actives et inactives. Les nœuds communiquent entre eux en utilisant

un mécanisme de RTS/CTS (*Request-To-Send, Clear-To-Send*) qui permet d'éviter les collisions et le problème de station cachée. Un nœud écoute le canal radio et peut potentiellement transmettre tant qu'il est dans la période active. Une période active se termine quand aucun événement d'activation ne se produit pendant un temps TA . Un événement d'activation est :

- le déclenchement d'un temporisateur de frame
- la réception de données sur la radio
- la détection (grâce à un indicateur de puissance du signal sur la radio) d'une communication sur le canal radio
- l'acquittement de ses propres paquets
- la connaissance grâce à l'écoute qui précède l'émission de RTS et CTS que la transmission de données d'un voisin s'est terminée.

TA est donc le temps minimum d'écoute à vide par frame.

C.2 Synchronisation

Quand un nœud se réveille, il commence par écouter. Si il n'entend rien pendant un certain temps, il choisit un ordonnancement de frame et le transmet par un paquet SYNC qui contient le temps jusqu'au début de la frame suivante. Si au démarrage, le nœud reçoit un paquet SYNC d'un autre nœud, il suit l'ordonnancement défini dans ce paquet et transmet son propre SYNC en conséquence. Les nœuds retransmettent leur SYNC de temps en temps. Si un nœud a déjà adopté un ordonnancement et qu'il reçoit un paquet SYNC d'un autre nœud avec un autre ordonnancement, alors il doit adopter les deux ordonnancements. Il doit aussi transmettre un paquet SYNC à l'autre nœud pour que celui-ci apprenne la présence d'un autre ordonnancement. Le fait d'adopter les deux ordonnancements signifie que le nœud va avoir un événement d'activation au début de chacune des deux frames. Les nœuds ne doivent débiter une transmission de données qu'au début de leur propre période active. A ce moment, à la fois les voisins ayant le même ordonnancement et ceux l'ayant adopté en plus sont réveillés.

TA doit être suffisamment long pour prendre en compte l'attente avant l'émission du paquet SYNC et le temps de la réception éventuelle d'un paquet RTS.

C.3 Overhearing Avoidance

Le mécanisme utilisé dans S-MAC peut-être utilisé dans T-MAC en option. C'est juste une option. Cela peut entraîner dans certaines conditions qu'un nœud manque des RTS et des CTS.

C.4 FRTS

Avec T-MAC, il peut y avoir un problème quand le trafic est essentiellement unidirectionnel (ce qui est le cas quand les capteurs doivent surtout remonter les informations vers une station collectante). Par exemple (voir figure 4) supposons qu'un nœud A doit envoyer des données à un nœud D en passant par les nœuds B et C, A ayant pour voisin B, B ayant pour voisins A et C, C ayant pour voisins B et D et D ayant pour voisin C. Considérons le nœud C avec des paquets en attente pour le nœud. Le nœud C peut perdre

la contention à cause de B (en recevant un paquet RTS de B) ou à cause de A (en recevant un paquet CTS de B en réponse à un paquet RTS que B a reçu de A). Considérons le deuxième cas. C ayant reçu le CTS de B, il doit rester silencieux et donc D ne reçoit rien. Par conséquent il retourne en sommeil. Quand A a fini de transmettre les données, B émet un paquet d'acquittement (ACK) que tous ses voisins reçoivent (et donc C). C peut donc espérer obtenir la contention pour transmettre ses paquets à D mais D étant en sommeil, il ne peut pas recevoir le paquet RTS de C.

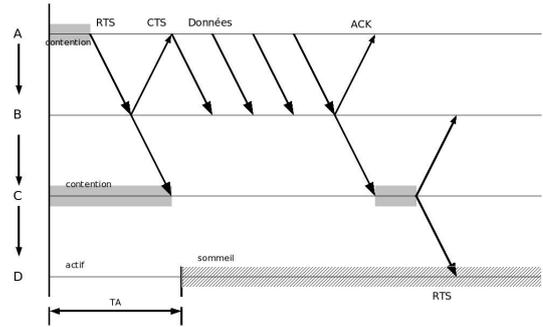


Fig. 4. Problème de l'endormissement précoce

Une des solutions est l'introduction du *future request-to-send* (FRTS). Dans l'exemple précédent (voir figure 5), quand C reçoit le CTS de B, C envoie un paquet FRTS à D (B entend ce paquet) qui contient la longueur de la communication de données bloquante (cette information est contenue dans le CTS que C a reçu). Ainsi le nœud peut déterminer qu'il sera ensuite le destinataire d'un paquet RTS et ne pas être endormi quand ce paquet arrivera. (Il faut tenir compte du temps de réception d'un éventuel paquet FRTS pour le calcul de TA). L'ajout du mécanisme de FRTS entraîne qu'il faut rajouter un paquet *Data-Send* (DS) avant l'émission de paquets de données. En effet puisque C émet le paquet FRTS après avoir reçu le paquet CTS de B, A envoie un paquet DS (de même taille qu'un paquet FRTS) après avoir reçu le paquet CTS. Ainsi il ne risque pas de perdre pas la contention et les données qu'il doit envoyer à B n'entrent pas en collision avec le paquet FRTS que B reçoit.

Le mécanisme de FRTS ne doit être utilisé que sous certaines conditions de trafic car sinon cela rajoute une consommation d'énergie qui peut être évitée.

D. BMAC

BMAC (*Berkeley MAC*) [4] a été développé par l'Université de Berkeley et est actuellement utilisé dans TinyOS avec la couche physique de la norme IEEE 802.15.4 pour les capteurs compatibles ZigBee.

BMAC est basé principalement sur deux principes : l'analyse du bruit sur le canal radio et sur l'écoute basse consommation.

Quand un nœud veut envoyer un paquet, il détermine si le canal radio est utilisé par un autre nœud ou pas en écoutant

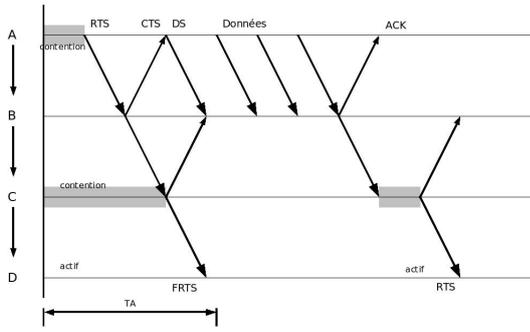


Fig. 5. Mécanisme de FRTS

le "bruit" en se basant sur un indicateur de puissance du signal. Si il n'y a pas de bruit, le canal est libre et il peut donc émettre. Avant d'envoyer des données il doit émettre un préambule.

Les nœuds sont en sommeil la plupart du temps et se réveillent à intervalles réguliers comme illustré sur la figure 6. A leur réveil ils écoutent le bruit sur le canal radio. Si il n'y a pas de bruit sur le canal radio, le nœud retourne en sommeil. Si il y a du bruit, cela signifie que des données vont arriver (à cause du préambule). Le préambule doit être suffisamment long, pour que tous les nœuds (et donc le destinataires des données) puissent l'entendre. Ensuite quand les données commencent à arriver, les nœuds qui ne sont pas destinataires de ces données retournent en sommeil. Un des avantages de BMAC est qu'il ne nécessite pas de synchronisation entre les nœuds.

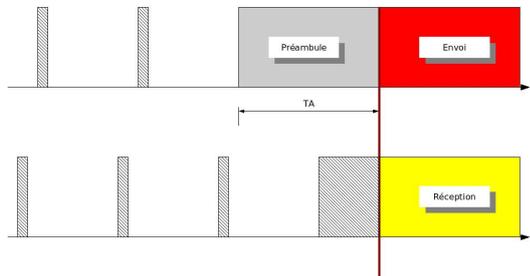


Fig. 6. BMAC: les nœuds se réveillent à intervalles réguliers

III. COMPARAISON PAR SIMULATION

Nous cherchons à comparer SMAC, TMAC et BMAC avec comme critère de performances la consommation d'énergie de la radio.

A. le simulateur

MACSimulator[1] est un simulateur basé sur OMNeT++ et écrit à l'origine par les auteurs de [5]. Il permet de simuler différentes couches MAC en utilisant plusieurs modèles de trafic. Les résultats de simulation obtenus avec ce simulateur pour chaque nœud :

- pour la couche application : le nombre de paquets envoyés et reçus

- pour la couche réseau : le nombre de paquets envoyés, reçus et encore en queue ainsi que le nombre de paquets dropés
- pour la couche mac : le nombre de paquets envoyés et reçus ainsi que le temps passé à envoyer et recevoir des paquets, le temps d'*overhearing* et le temps d'*overhead*
- pour la couche radio : le temps passé avec la radio coupée, le temps passé en réception et en envoi en mode normal et en mode basse consommation.

Ce mode basse consommation n'a rien à voir avec le concept d'écoute basse consommation introduit dans BMAC. Il s'agit ici juste d'un état actif de la radio avec une consommation plus faible que peuvent adopter certains modèles de radio quand ils ne sont pas en train de recevoir ou de transmettre des paquets.

MACSimulator est donc bien adapté à l'étude de la consommation d'énergie des capteurs. De plus un certain nombre de couches MAC sont déjà implémentées dans le simulateur dont SMAC, TMAC et CSMA. De plus un mécanisme d'écoute basse consommation est prévu ce qui facilite la mise en place de BMAC. Dans les résultats de simulations, on utilise la moyenne de la consommation d'énergie sur l'ensemble du réseau.

B. Le réseau simulé

Le réseau considéré est composé de cent nœuds répartis sur une grille de dix par dix. Pour calculer les consommations d'énergie, on a utilisé les caractéristiques de la puce radio RFM1000. Il n'y a pas de mode basse consommation et donc l'émission et la réception en mode basse consommation dans la simulation seront considérées comme si elles étaient en mode normal pour effectuer les calculs de consommation d'énergie. La consommation de la puce est la suivante :

- 0,02 mA en sommeil
- 4,0 mA en réception
- 10,0 mA en émission

On ne prend en considération que les résultats pour lesquels au moins quatre-vingt-dix pour cent des paquets émis ont été reçus.

C. Modèle Nodes-to-Sink

Dans le modèle de trafic Node-To-Sink, les nœuds envoient des messages à un nœud unique que nous avons choisi dans un coin de notre grille, en l'occurrence, le nœud 0. Les messages sont routés en utilisant un algorithme non déterministe du plus court chemin. Aucune aggrégation de données n'est réalisée. Ce modèle est particulièrement intéressant car cela peut représenter le trafic vers une station collectante ou le trafic vers un *clusterhead* si on veut utiliser de la clusterisation.

Pour TMac, on utilise une période de cycle de six cent dix millisecondes et une longueur de quinze millisecondes pour le *timeout*. On utilise TMAC avec l'*overhearing avoidance* et le mécanisme de FRTS. Pour SMac, on utilise une période de cycle de une seconde et on fait varier la longueur de la période active.

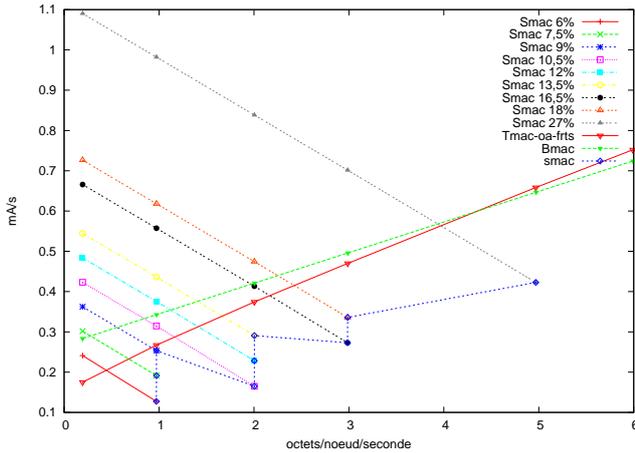


Fig. 7. réseau à un saut avec une taille de paquet de 20 octets

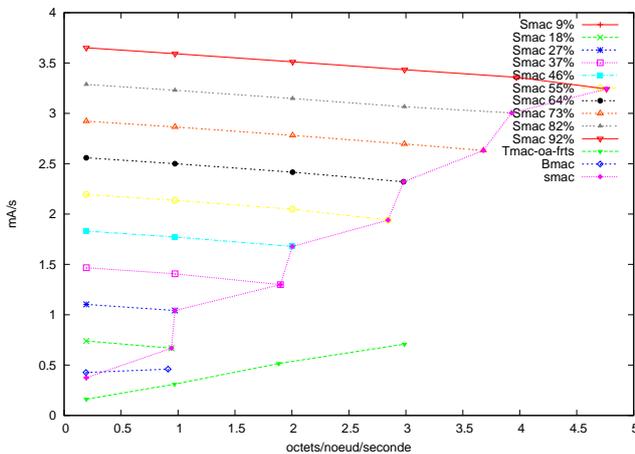


Fig. 8. réseau multi-saut avec une taille de paquet de 20 octets

Sur la figure 7, tous les nœuds sont dans le même voisinage radio. Dans le cas d'un réseau multisaut (figure 8), chaque nœud n'a à sa portée radio que les huit nœuds qui l'entourent (sauf dans le cas des nœuds du bord de la grille).

Sur les figures 7 et 8, on trace la courbe de SMAC pour différentes valeurs de la période active (exprimée ici en pourcentage de la période de cycle). Dans les deux cas, pour SMAC, on constate que, à longueur de période active fixée, la consommation d'énergie moyenne sur le réseau décroît quand la charge de trafic croît, pour atteindre pour un certain trafic une consommation minimum. Au delà de cette charge de trafic, il y a saturation avec des pertes de paquets dues à des débordements de buffer. La baisse de la consommation d'énergie quand le trafic croît est due au mécanisme d'*overhearing avoidance*. En effet, plus un nœud émet, plus il "oblige" ses voisins à dormir pendant leur période active,

ce qui entraîne qu'ils consomment moins d'énergie. La saturation est due au fait que si un nœud contraint ses voisins trop souvent à dormir pendant leur période active, ceux-ci n'auront plus le temps d'envoyer leur paquets, d'où un débordement de buffer.

Dans un réseau à un saut (figure 7), SMAC présente de meilleures performances que TMAC et BMAC. On peut également constater que TMAC est plus performant que BMAC sur les faibles débits tandis que pour des débits un peu plus élevés c'est l'inverse car la consommation d'énergie pour BMAC croît moins vite avec le débit que pour TMAC. Dans un réseau multisaut (figure 8), BMAC et TMAC sont meilleurs que SMAC. La consommation d'énergie par rapport au débit croît beaucoup plus vite pour SMAC que pour TMAC et BMAC, en considérant à chaque fois pour SMAC la consommation minimale pour le débit considéré. Cependant BMAC sature très tôt, et on ne peut pas aller au delà de 1 octet par seconde; ceci est notamment dû à un problème de collisions car BMAC n'a aucun mécanisme pour éviter le problème de stations cachées. Cependant la consommation d'énergie pour BMAC semble croître moins avec le débit que pour TMAC. On peut donc imaginer, au vu de l'allure des courbes, que si on n'avait ce problème de saturation dans BMAC, que cette couche MAC serait plus performante que TMAC à partir de débits relativement peu élevés. SMAC peut tenir des débits plus importants que TMAC.

D. Modèle Unicast local

Dans le modèle unicast utilisé ici, l'émetteur choisi aléatoirement parmi ses voisins un nœud qui sera le destinataire du message, chaque nœud ayant pour voisin les 8 nœuds qui l'entourent (sauf dans le cas d'un nœud au bord de la grille).

Pour TMac, on utilise une période de cycle de six cent dix millisecondes et une longueur de quinze millisecondes pour le *timeout*. On utilise TMAC avec l'*overhearing avoidance*, ce qui donne les meilleures performances. On n'a pas gardé les autres configurations d'option de TMAC pour assurer une meilleure lisibilité de la courbe. Pour SMac, on utilise une période de cycle de une seconde et on fait varier la longueur de la période active. Pour tracer la consommation de SMAC sur la figure 9, nous avons utilisé la même manière que pour les figures 7 et 8. La consommation d'énergie pour SMAC est supérieure à celle de TMAC et BMAC. TMAC est plus performant que BMAC pour les faibles débits, mais l'évolution de la consommation d'énergie par rapport au débit étant plus faible pour BMAC que pour TMAC, BMAC est meilleur sur le débit plus élevé (environ pour les débits supérieurs à 20 octets/seconde). Contrairement à ce qu'on observe pour le modèle Node-to-Sink, TMAC et BMAC saturent pour des débits semblables. SMAC peut tenir des trafics légèrement plus élevés que ces derniers.

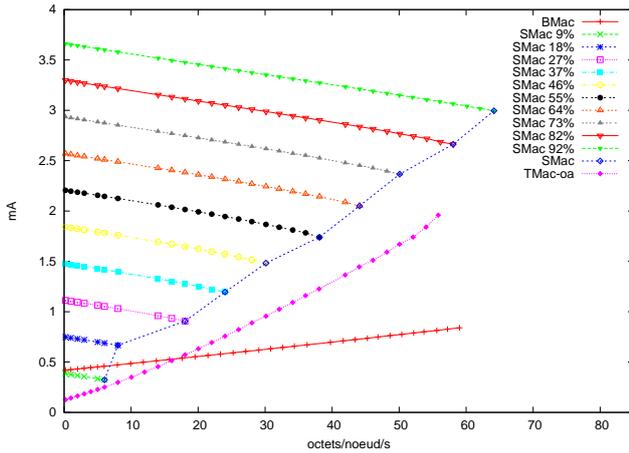


Fig. 9. Local Unicast avec une taille de paquet de 20 octets

IV. NOTRE PROPOSITION : ÉVOLUTION DE BMAC

A. Limitations de BMAC

L'envoi de paquets dans BMAC est précédé d'un long préambule pour que les nœuds puissent avoir connaissance que des paquets vont être envoyés. Dans le cas où un nœud doit émettre un flux continu important de paquets de données, émettre des préambules "longs" pour chaque paquet représente une consommation d'énergie pas forcément négligeable et qui peut être évitée (on appelle par la suite préambule long un préambule qui est suffisamment long pour couvrir au moins un cycle endormissement-éveil dans un protocole utilisant un principe d'écoute basse consommation comme dans BMAC). De plus, pendant l'émission de ce flux, les nœuds voisins de l'émetteur autre que le destinataire des paquets vont continuer à se réveiller périodiquement inutilement. Cela augmente la consommation d'énergie, surtout que ces nœuds, chaque fois qu'ils se réveillent, sont obligés d'attendre de recevoir l'entête d'un paquet pour se rendre compte que le bruit sur le canal radio ne correspond pas à des paquets qui leur sont destinés. Par ailleurs BMAC, avec son système d'évaluation du bruit sur le canal pour déterminer s'il est libre ou non, ne permet pas de se protéger du problème de stations cachées.

Cependant BMAC, par sa simplicité, est très intéressant. Par son absence de besoin de synchronisation, il permet que l'ajout de nouveaux nœuds soit aisé et sans trop de risque de dégradation de performances. Dans SMAC et TMAC, lorsque l'on ajoute un nouveau nœud, il y a toujours le risque que ce nouveau nœud adopte un autre cycle, obligeant ainsi ces voisins à se synchroniser sur ce cycle en plus de celui qu'ils suivaient déjà, dégradant ainsi leurs performances en terme de consommation d'énergie. De plus l'augmentation de la consommation d'énergie avec l'augmentation de débit dans BMAC est plus faible que pour TMAC.

Les auteurs de BMAC [4] proposent d'implémenter un mé-

canisme de RTS-CTS dans une couche au dessus de BMAC. Cependant cette solution, bien que résolvant le problème de terminal caché, laissent certaines des autres limitations décrites ci-dessus sans réponse.

B. la nouvelle couche MAC

Notre solution consiste à implémenter un mécanisme de RTS-CTS directement dans BMAC et à partir de cela en tirer des optimisations. Cette couche MAC, bien que reposant sur des principes tirés de BMAC et de SMAC, constitue une nouvelle couche MAC car elle en diffère de manière significative. En voici le principe de fonctionnement.

B.1 Fonctionnement

Comme dans BMAC, les nœuds se réveillent à intervalles réguliers et chaque phase de réveil commence par une phase d'écoute du canal. Si il n'y a pas de bruit sur le canal, le canal est libre. Dans le cas contraire, le canal est occupé et le nœud reste éveillé jusqu'au début de la réception du paquet. Lorsqu'un nœud veut émettre un ou des paquets de données, au début de sa phase d'éveil il écoute le canal pendant une durée aléatoire dans un intervalle prédéterminé. Si il n'y a pas de bruit sur le canal, il émet un paquet RTS précédé (comme les paquets dans BMAC) d'un préambule suffisamment long pour que tous ses voisins puissent savoir qu'un paquet va arriver. Si il peut accepter ce transfert de données, le destinataire du RTS répond par un paquet CTS avec lui aussi un préambule long et reste éveillé. L'émission du CTS est précédée d'une période de contention, comme pour le RTS. Le CTS contenant des informations sur la taille des données qui vont être transmises, les voisins du destinataire des données vont pouvoir estimer la durée de cet échange et se mettre totalement en sommeil durant cette période, c'est-à-dire que durant cette période, ils abandonnent les réveils réguliers. Après la réception du CTS, la source des données émet un paquet DS (Data Send) précédé d'une période de contention et d'un préambule long, ce paquet contenant des informations sur la taille des données qui vont être émises avant que ses voisins puissent passer en sommeil total durant la période d'émission des données. Les paquets de données sont ensuite émis sans préambule long.

B.2 Comparaison par rapport à BMAC et TMAC

L'avantage de cette solution par rapport à BMAC est valable si les nœuds ont un nombre suffisamment grand de données à transmettre à la fois. En effet, si les nœuds ont relativement peu de données à transmettre (comme par exemple un ou deux paquets à la fois), BMAC est bien meilleur et notre solution est plus lourde et entraîne une plus grande consommation d'énergie sauf peut-être si la répartition du trafic et la topologie du réseau font qu'il y a un grand risque de collision à cause du problème de terminaux cachés. L'avantage de cette solution, par rapport à l'implémentation d'un mécanisme de RTS-CTS dans une couche au dessus de BMAC, est que l'on peut tirer tous les

avantages de ce mécanisme puisqu'il est directement implémenté dans la couche MAC. Ainsi on n'émet pas des préambules longs pour les paquets de données et les nœuds ne pouvant pas participer au trafic (un voisin de l'émetteur ou du récepteur ne doit pas émettre pendant la transmission sous peine de provoquer une collision) sont mis en sommeil total.

L'avantage par rapport à SMAC et TMAC est que cette solution ne nécessite pas de synchronisation.

Il faut cependant voir si l'introduction d'un mécanisme de RTS/CTS ne risque pas de dégrader la faible évolution de la consommation d'énergie par rapport au débit qu'on a pu constater dans BMAC.

V. CONCLUSION

Dans cette article nous avons effectué une comparaison entre les trois couches MAC (SMAC, TMAC, BMAC) pour réseaux de capteurs en terme d'énergie. Les simulations effectuées nous ont permis de montrer que TMAC et BMAC étaient meilleurs que SMAC en terme de consommation énergétique. Dans les simulations effectuées nous avons aussi montré que BMAC consommait moins d'énergie que TMAC et SMAC dans le cas de débit élevé, alors que dans le cas de faibles débits, TMAC est meilleur que BMAC. Cependant il est à noter que BMAC sature à des débits relativement faibles, cela est dû au phénomène de stations cachées. C'est pourquoi il est intéressant d'intégrer un mécanisme de RTS/CTS dans BMAC par pallier à ce problème. L'introduction de ce mécanisme s'accompagne de quelques optimisations permettant d'économiser un peu d'énergie lors de l'émission de paquets de données.

REFERENCES

- [1] the consensus project site. <http://www.consensus.tudelft.nl>.
- [2] A. D. de Mazieux, V. Gauthier, M. Marot, and M. Becker. État de l'art sur les réseaux de capteurs. *INT 05001RST*, avril 2005.
- [3] IEEE. 802.15.4-2003 standard for information technology - telecommunication and information exchange between systems - lan/wan - part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for wireless personal area networks (lr-wpan)., 2003.
- [4] J. Polastre, J. Hill, and D. E. Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems, SenSys 2004*, pages 95–107, Baltimore, MD, USA, November 2004. ACM.
- [5] T. van Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys 2003*, pages 171–180, Los Angeles, California, USA, November 2003. ACM.
- [6] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks. In *Proceedings of the IEEE Infocom*, pages 1567–1576, New York, NY, USA, June 2002. IEEE.