

Soutenance finale

Élaboration des couches MAC et réseau pour un réseau de capteurs

Julien Vaudour

Conservatoire National des Arts et Métiers
Institut d'Informatique d'Entreprise

29 Juin 2006

Cette étude se fait dans le cadre du projet *Capteurs*

- chaîne du froid
- financé par le **R**éseau **N**ational de **R**echerche en **T**élécommunication
- dans la logique du programme initiative *Réseaux autonomes et spontanés* du GET

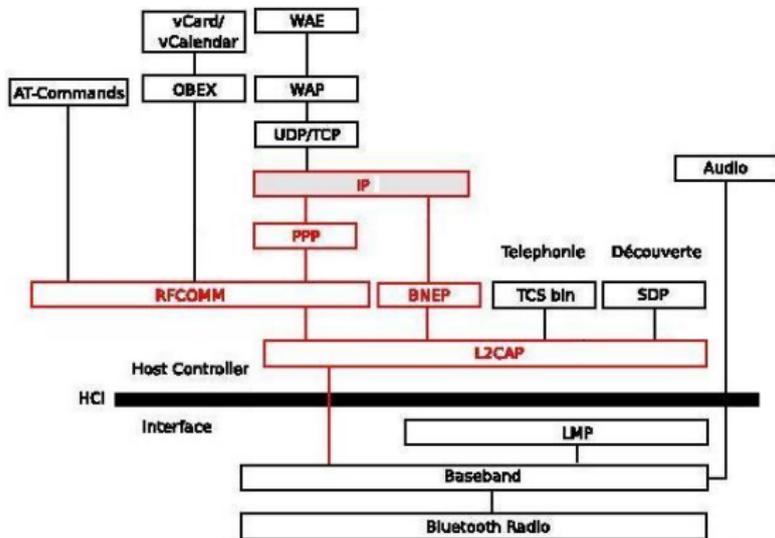
- étude de Bluetooth
une décision conséquence de cette étude de ne pas choisir Bluetooth
- étude de Zigbee.....
- résultats
- proposition

- Bluetooth
- Zigbee
- résultats
- proposition

La norme bluetooth

- 79 canaux radio
- canal Bluetooth : occupation successive de ces différents canaux
- Durée d'un *time slot* : $625 \mu\text{s}$ ($1/1600 \text{ s}$)
- à chaque *time slot*, saut de fréquence
- émission d'un paquet : 1, 3 ou 5 *time slots*
- on ne change pas de canal radio pendant l'émission d'un paquet
⇒ Reprise de la séquence là où on en serait si on n'avait émis que des paquets qui prennent un seul *time slot*

Les couches Bluetooth



La couche baseband gère les différents types de liaisons Bluetooth :

- **Liaison synchrone (SCO)** : ce type de liaisons offre un débit synchrone de 64 kb/s dans les deux directions. En théorie, Bluetooth permet 3 liaisons synchrones simultanées.
- **Liaison asynchrone (ACL)** : ce type de liaisons offre un débit maximal de 732 kb/s pour la transmission de donnée sans garantie de délai.

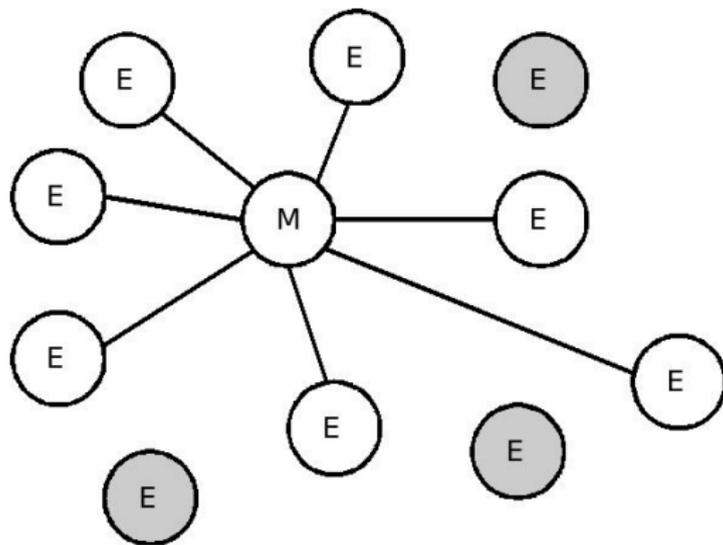
On va a priori utiliser le mode asynchrone avec LMP et/ou L2CAP.

LMP = Link Manager Protocol

- s'occupe de l'établissement, du contrôle et de la sécurité de la liaison
- place les dispositifs dans les différents modes (park, hold, sniff et actif)

L2CAP = Logical Link Control and Adaptation Layer Protocol

- créer des liaisons asynchrones (ACL) entre les appareils
- liaisons en mode connecté ou non
- segmentation/ réassemblage



Piconet (2)

- même canal Bluetooth
- organisation maître/esclave
- 1 maître
- jusqu'à 255 esclaves
- maximum : 7 esclaves actifs, les autres étant parqués
- pas de communication maître-maître ni esclave-esclave

- **BD_ADDR** (Bluetooth Device Address) : 48 bits, identifie de manière unique un appareil Bluetooth
- **AM_ADDR** (Active Member Address) : 3 bits, identifie les esclaves actifs d'un canal Bluetooth
- **PM_ADDR** (Parked Member Address) : 8 bits, identifie les esclaves parkés.
- **AR_ADDR** (Access Request Address) : permet à un esclave parké de déterminer le *time slot* qu'il est autorisé à utiliser pour demander au maître à être déparké

L'adresse du maître (BD_ADDR) détermine la séquence des sauts de fréquence.

Les différents modes

- **Active** : esclave actif, décode les adresses pour voir si les paquets lui sont destinés
- **Sniff** : esclave toujours synchronisé avec le maître. N'écoute le réseau que pendant certains *time slots* spécifiés. Mise en veille périodique et automatique
- **Hold** : l'esclave actif d'un piconet stoppe sa radio, conserve son adresse de membre actif et reste synchronisé avec le maître et se place en mode basse consommation pour un intervalle de temps déterminé, après quoi il redevient actif
- **Park** : esclave toujours synchronisé avec le maître mais ne participe pas au trafic. Il n'a pas d'adresse de membre actif. Le maître du *piconet* lui envoie à intervalle de temps régulier des paquets de synchronisation, et à ces instants, il peut demander à redevenir actif.

Les différents modes

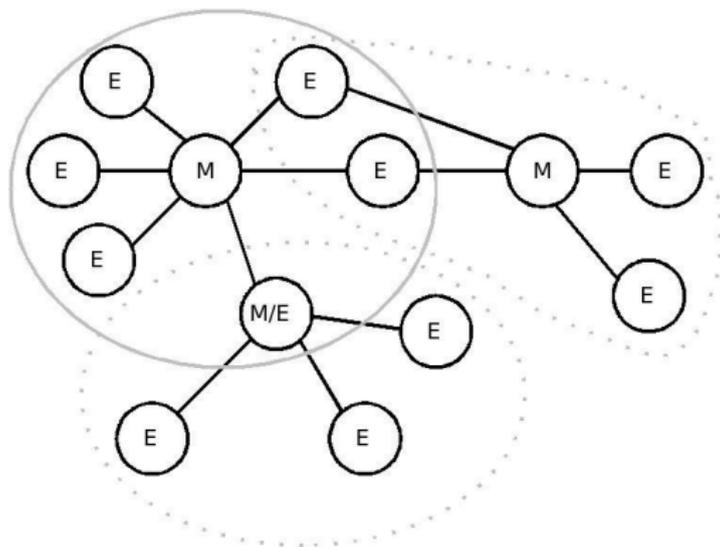
- **Active** : esclave actif, décode les adresses pour voir si les paquets lui sont destinés
- **Sniff** : esclave toujours synchronisé avec le maître. N'écoute le réseau que pendant certains *time slots* spécifiés. Mise en veille périodique et automatique
- **Hold** : l'esclave actif d'un piconet stoppe sa radio, conserve son adresse de membre actif et reste synchronisé avec le maître et se place en mode basse consommation pour un intervalle de temps déterminé, après quoi il redevient actif
- **Park** : esclave toujours synchronisé avec le maître mais ne participe pas au trafic. Il n'a pas d'adresse de membre actif. Le maître du *piconet* lui envoie à intervalle de temps régulier des paquets de synchronisation, et à ces instants, il peut demander à redevenir actif.

Les différents modes

- **Active** : esclave actif, décode les adresses pour voir si les paquets lui sont destinés
- **Sniff** : esclave toujours synchronisé avec le maître. N'écoute le réseau que pendant certains *time slots* spécifiés. Mise en veille périodique et automatique
- **Hold** : l'esclave actif d'un piconet stoppe sa radio, conserve son adresse de membre actif et reste synchronisé avec le maître et se place en mode basse consommation pour un intervalle de temps déterminé, après quoi il redevient actif
- **Park** : esclave toujours synchronisé avec le maître mais ne participe pas au trafic. Il n'a pas d'adresse de membre actif. Le maître du *piconet* lui envoie à intervalle de temps régulier des paquets de synchronisation, et à ces instants, il peut demander à redevenir actif.

Les différents modes

- **Active** : esclave actif, décode les adresses pour voir si les paquets lui sont destinés
- **Sniff** : esclave toujours synchronisé avec le maître. N'écoute le réseau que pendant certains *time slots* spécifiés. Mise en veille périodique et automatique
- **Hold** : l'esclave actif d'un piconet stoppe sa radio, conserve son adresse de membre actif et reste synchronisé avec le maître et se place en mode basse consommation pour un intervalle de temps déterminé, après quoi il redevient actif
- **Park** : esclave toujours synchronisé avec le maître mais ne participe pas au trafic. Il n'a pas d'adresse de membre actif. Le maître du *piconet* lui envoie à intervalle de temps régulier des paquets de synchronisation, et à ces instants, il peut demander à redevenir actif.



Un *scatternet* est le résultat de l'interconnexion de plusieurs *piconets*.

- Un appareil Bluetooth ne peut être maître que dans un seul *piconet*.
- Un pont interconnectant 2 *piconets* peut être soit esclave dans les 2 *piconets*, soit maître dans l'un et esclave dans l'autre.

Plusieurs solutions ont été proposées :

- formation immédiate sans esclaves parqués
- formation immédiate avec esclaves parqués
- formation à la demande

Les limites de Bluetooth

- gestion du scatternet
- interférence entre les piconets
 - ⇒ limite à 8 piconets
 - ⇒ moins de 64 nœuds actifs

Les limites de Bluetooth

- gestion du scatternet
- interférence entre les piconets
 - ⇒ limite à 8 piconets
 - ⇒ moins de 64 nœuds actifs

Les limites de Bluetooth

- gestion du scatternet
- interférence entre les piconets
 - \Rightarrow limite à 8 piconets
 - \Rightarrow moins de 64 nœuds actifs

Les limites de Bluetooth

- gestion du scatternet
- interférence entre les piconets
 - \Rightarrow limite à 8 piconets
 - \Rightarrow moins de 64 nœuds actifs

- Bluetooth
- Zigbee
- résultats
- proposition

- faible consommation électrique
- opère sur la bande 2.4 GHz et la bande 868-915 MHz
- 27 canaux de communication
 - 16 canaux à 250 kbps dans la bande 2.4 GHz
 - 10 canaux à 40 kbps dans la bande 915 MHz
 - 1 canal à 20 kbps dans la bande 868 MHz

2 topologies :

- étoile :
 - un coordinateur
 - communication entre le coordinateur et les autres nœuds
 - synchronisation sur le coordinateur avec beacon
 - *Data Request*
 - contrôle de la mise en veille au niveau du nœud
 - supertrame
- point à point :
 - communication directe entre les nœuds
 - écoute permanente
 - synchronisation
 - pas de supertrame

2 topologies :

- étoile :

- un coordinateur
- communication entre le coordinateur et les autres nœuds
- synchronisation sur le coordinateur avec beacon
- *Data Request*
- contrôle de la mise en veille au niveau du nœud
- supertrame

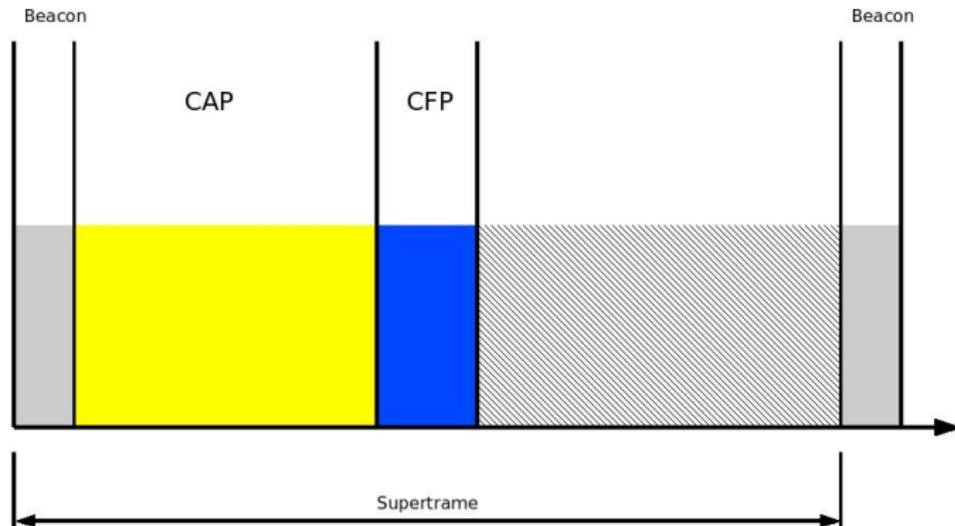
- point à point :

- communication directe entre les nœuds
- écoute permanente
- synchronisation
- pas de supertrame

2 topologies :

- étoile :
 - un coordinateur
 - communication entre le coordinateur et les autres nœuds
 - synchronisation sur le coordinateur avec beacon
 - *Data Request*
 - contrôle de la mise en veille au niveau du nœud
 - supertrame
- point à point :
 - communication directe entre les nœuds
 - écoute permanente
 - synchronisation
 - pas de supertrame

IEEE 802.15.4 : superframe



2 parties : active et inactive.

La partie active :

- *beacon* : synchronisation
- CAP (*Contention Access Period*) : CSMA-CA
- CFP (*Contention Free Period*) :
 - optionnelle
 - slots réservés
 - géré par le coordinateur
 - allocation des slots non défini dans le standart

2 parties : active et inactive.

La partie active :

- *beacon* : synchronisation
- CAP (*Contention Access Period*) : CSMA-CA
- CFP (*Contention Free Period*) :
 - optionnelle
 - slots réservés
 - géré par le coordinateur
 - allocation des slots non défini dans le standart

2 parties : active et inactive.

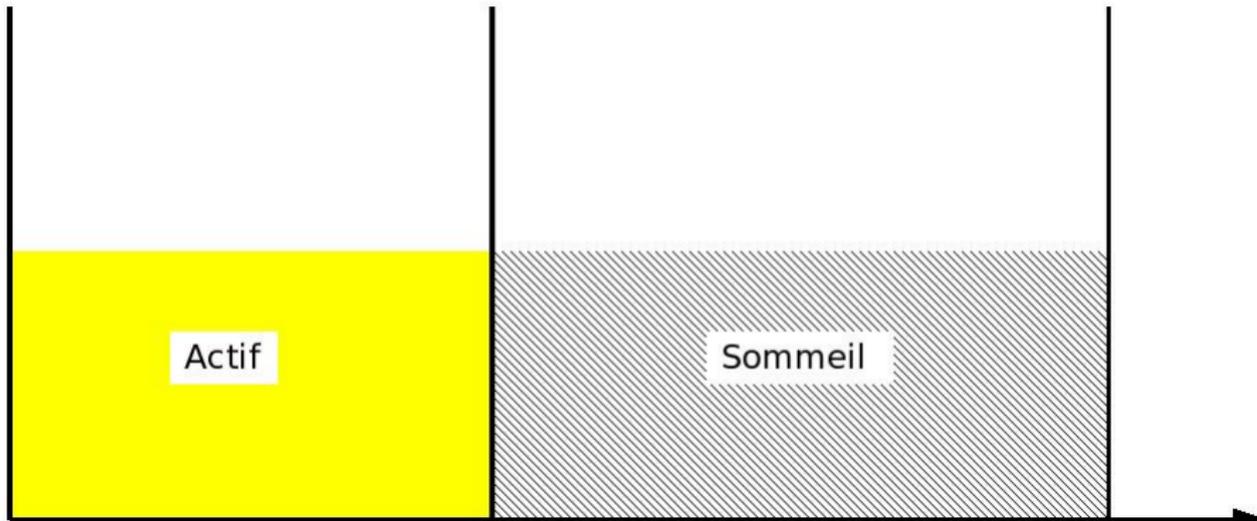
La partie active :

- *beacon* : synchronisation
- CAP (*Contention Access Period*) : CSMA-CA
- CFP (*Contention Free Period*) :
 - optionnelle
 - slots réservés
 - géré par le coordinateur
 - allocation des slots non défini dans le standart

2 parties : active et inactive.

La partie active :

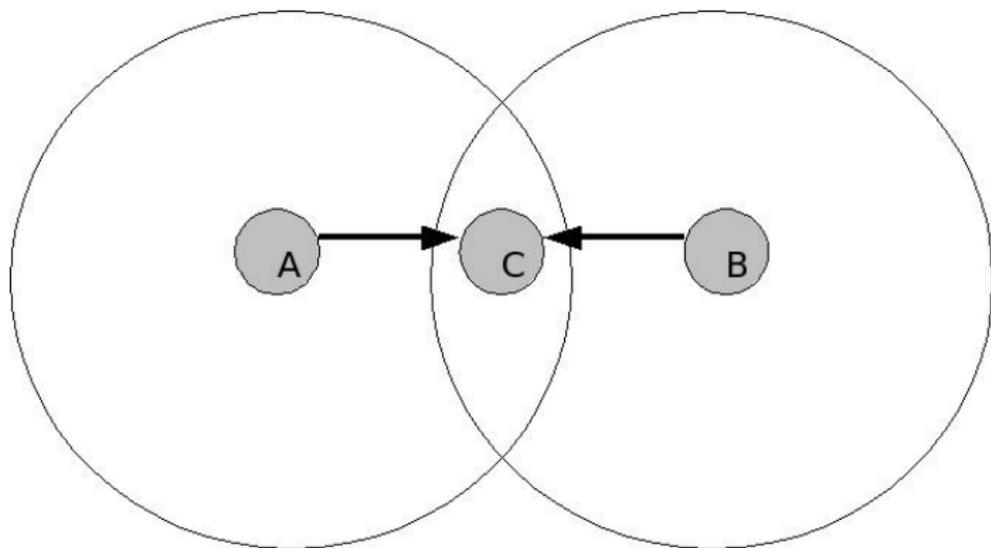
- *beacon* : synchronisation
- CAP (*Contention Access Period*) : CSMA-CA
- CFP (*Contention Free Period*) :
 - optionnelle
 - slots réservés
 - géré par le coordinateur
 - allocation des slots non défini dans le standart



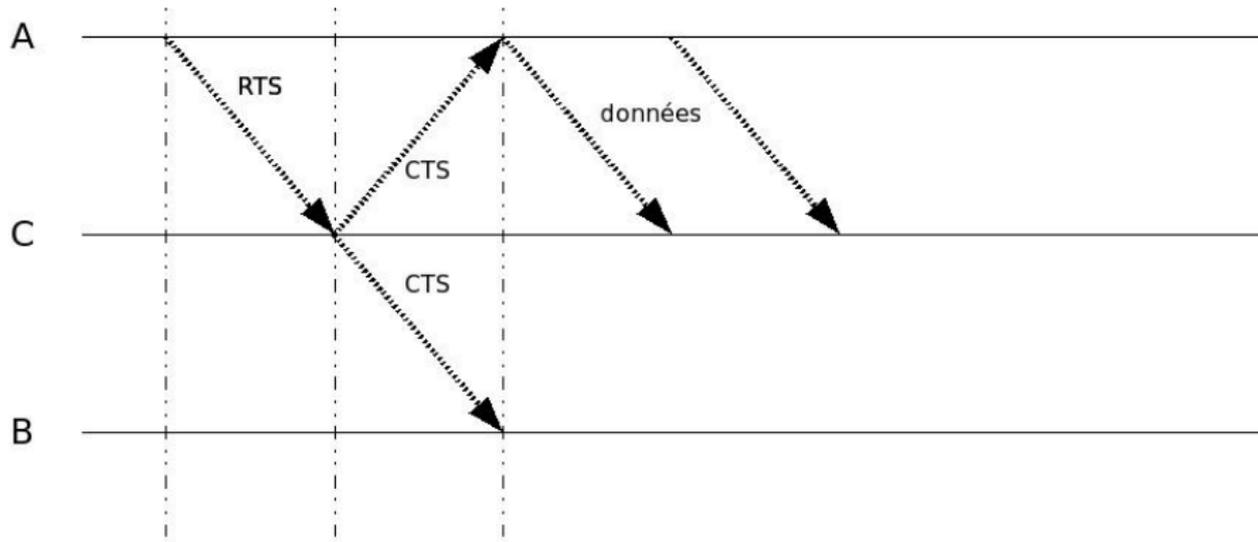
SMAC (2)

- alternance période active/inactive
- synchronisation :
 - non critique
 - trame SYNC
- CSMA-CA RTS/CTS
 - problème des stations cachées
 - source : émet une trame RTS
 - destination : répond par une trame CTS
- *Overhearing Avoidance*

SMAC (3) : Problème de station cachée



SMAC (4) : Mécanisme de RTS/CTS



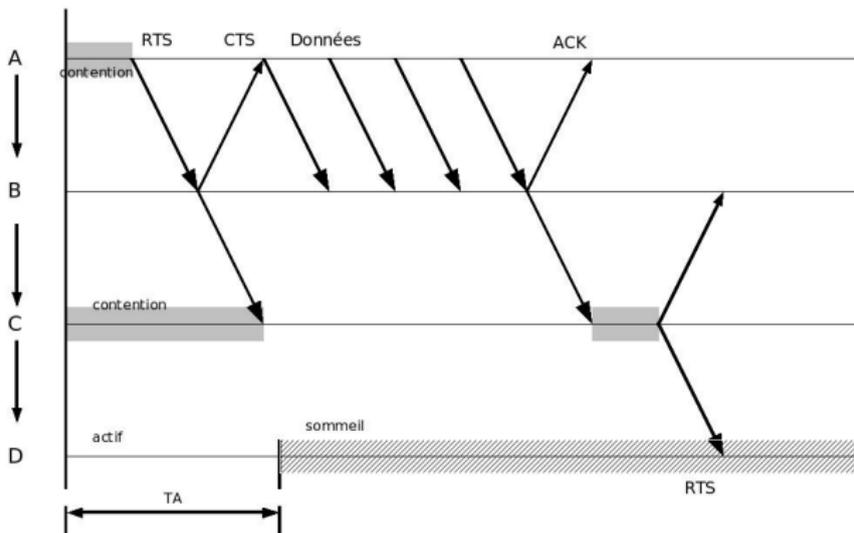
TMAC = *Timeout MAC*

Amélioration de SMAC

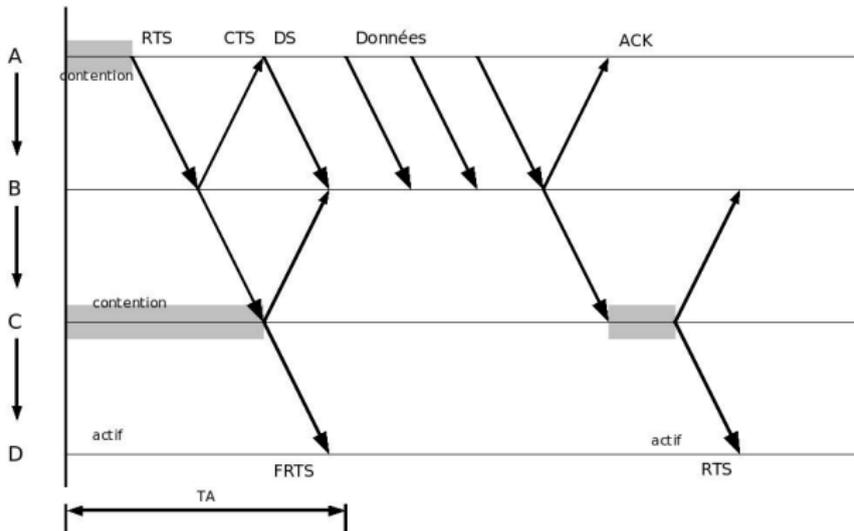
Fenêtre temporelle avec *timeout*

⇒ Retour en sommeil des nœuds si aucune données à recevoir

TMAC (2) : Problème d'endormissement précoce



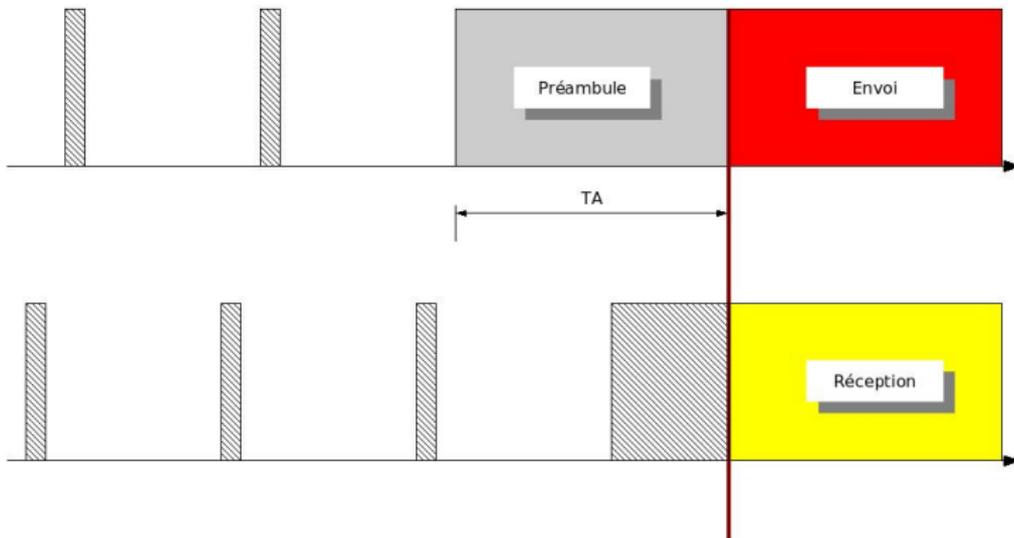
TMAC (3) : Mécanisme de FRTS



BMAC = *Berkeley MAC*

- utilisé dans TinyOS pour les capteurs compatibles ZigBee
- évaluation de la liberté du canal basé sur le bruit
- écoute à basse consommation du canal
- préambule avant l'émission d'un paquet
- contrôle du canal à intervalles réguliers

BMAC (2)



- Bluetooth
- Zigbee
- **résultats**
- proposition

- OMNeT++
 - simulateur à évènements discrets
 - modules en C++
 - langage pour les scénarios : NED
- *framework* MACSimulator
 - permet de simuler les couches MAC
 - adapté à l'étude de la consommation d'énergie par la radio

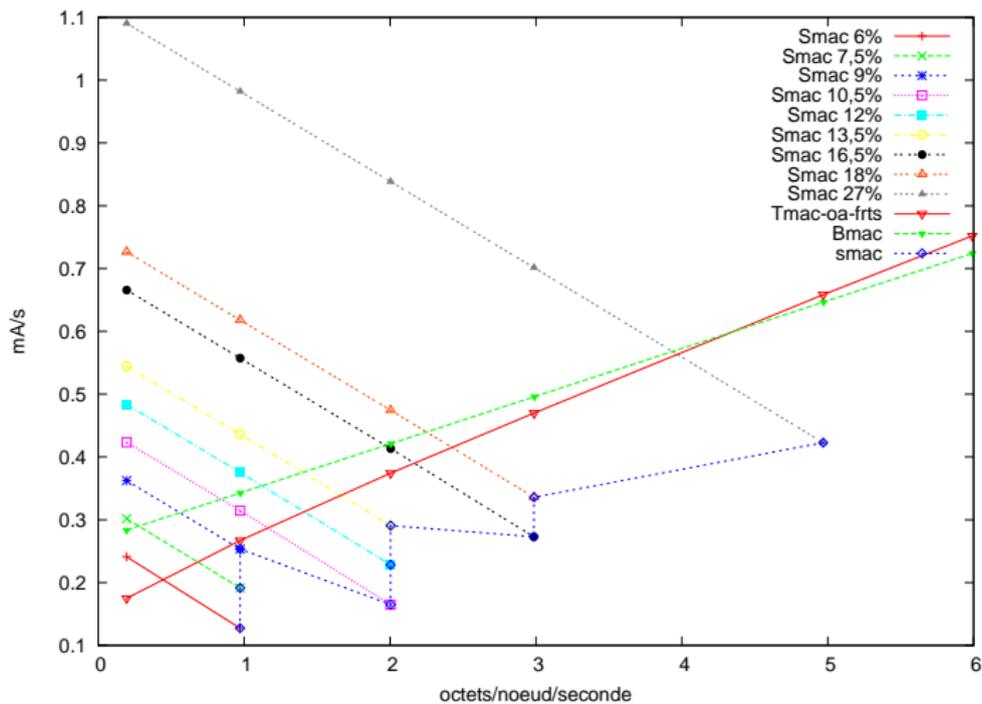
- station collectante
- les nœuds émettent vers cette station
- routage : algorithme non déterministe du plus court chemin

- station collectante
- les nœuds émettent vers cette station
- routage : algorithme non déterministe du plus court chemin

- station collectante
- les nœuds émettent vers cette station
- routage : algorithme non déterministe du plus court chemin

- 100 nœuds
- grille 10x10
- chipset radio RFM1000 :
 - 0,02 mA en sommeil
 - 4,0 mA en réception
 - 10,0 mA en émission
- seuls les résultats avec au moins 90% des paquets arrivés sont considérés

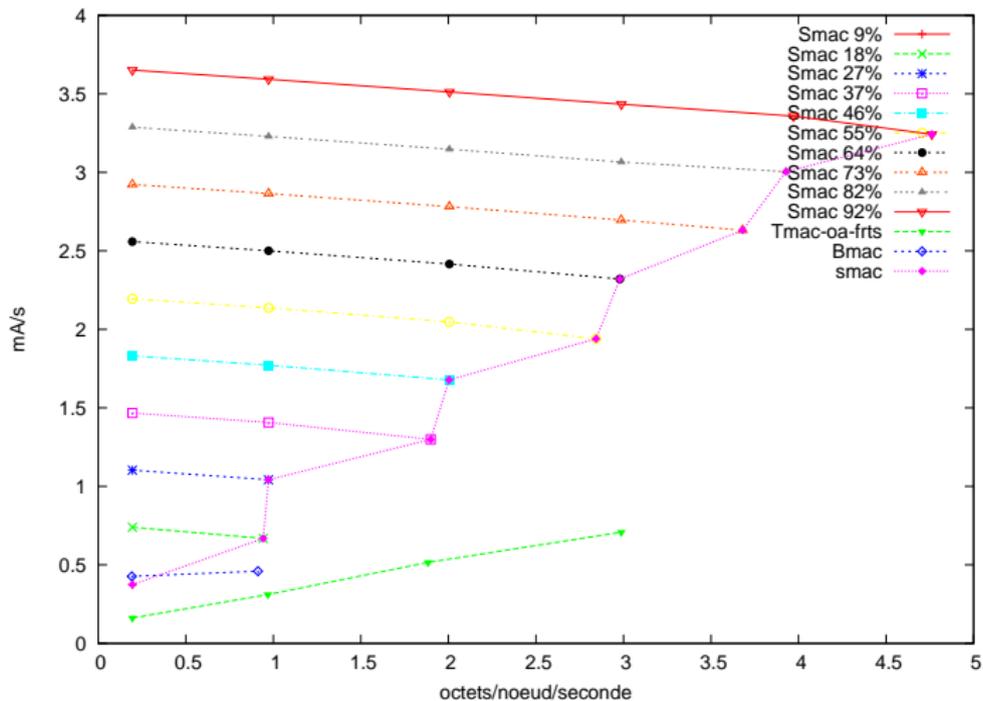
Node-To-Sink : réseau à un saut



Node-To-Sink : réseau à un saut (2)

- SMAC meilleur que TMAC et BMAC
- TMAC meilleur que BMAC sur les petits débits
- il faut paramétrer SMAC en fonction du débit attendu

Node-To-Sink : réseau multi-sauts



Node-To-Sink : réseau multi-sauts (2)

- TMAC meilleur que SMAC et BMAC
- saturation de BMAC à un débit très faible
- collisions avec BMAC
- SMAC supporte des débit plus élevés que TMAC
- il faut paramétrer SMAC en fonction du débit attendu

- Bluetooth
- Zigbee
- résultats
- **proposition**

- BMAC

- préambule "long" sur tous les paquets
- problème de stations cachées
- limitation de débit dans un réseau multisaut
- réveil inutile des nœuds quand les voisins sont en communication

- TMAC

- mécanisme de synchronisation
- limitation en débit

- BMAC
 - préambule "long" sur tous les paquets
 - problème de stations cachées
 - limitation de débit dans un réseau multisaut
 - réveil inutile des nœuds quand les voisins sont en communication
- TMAC
 - mécanisme de synchronisation
 - limitation en débit

- BMAC

- préambule "long" sur tous les paquets
- problème de stations cachées
- limitation de débit dans un réseau multisaut
- réveil inutile des nœuds quand les voisins sont en communication

- TMAC

- mécanisme de synchronisation
- limitation en débit

- BMAC

- préambule "long" sur tous les paquets
- problème de stations cachées
- limitation de débit dans un réseau multisaut
- réveil inutile des nœuds quand les voisins sont en communication

- TMAC

- mécanisme de synchronisation
- limitation en débit

- BMAC

- préambule "long" sur tous les paquets
- problème de stations cachées
- limitation de débit dans un réseau multisaut
- réveil inutile des nœuds quand les voisins sont en communication

- TMAC

- mécanisme de synchronisation
- limitation en débit

- BMAC
 - préambule "long" sur tous les paquets
 - problème de stations cachées
 - limitation de débit dans un réseau multisaut
 - réveil inutile des nœuds quand les voisins sont en communication
- TMAC
 - mécanisme de synchronisation
 - limitation en débit

- BMAC
 - préambule "long" sur tous les paquets
 - problème de stations cachées
 - limitation de débit dans un réseau multisaut
 - réveil inutile des nœuds quand les voisins sont en communication
- TMAC
 - mécanisme de synchronisation
 - limitation en débit

Les limitations de BMAC et TMAC

- BMAC
 - préambule "long" sur tous les paquets
 - problème de stations cachées
 - limitation de débit dans un réseau multisaut
 - réveil inutile des nœuds quand les voisins sont en communication
- TMAC
 - mécanisme de synchronisation
 - limitation en débit

Proposition

- mécanisme d'écoute basse consommation
- mécanisme de RTS/CTS
- *overhearing avoidance*
- optimisation de la taille des préambules



Omnet++ community site.

<http://www.omnetpp.org>.



the consensus project site.

<http://www.consensus.tudelft.nl>.



Alexandre Delye de Mazieux, Vincent Gauthier, Michel Marot, Julien Vaudour, Monique Becker, Rahim Kacimi, Riadh Dhaou, and André-Luc Beylot.

État de l'art, réseaux de capteurs sans fil.

Délivrable, projet Capteurs, mai 2006.



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.



Joseph Polastre, Jason Hill, and David 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.



Tijs van Dam and Koen 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.



Wei Ye, John Heidemann, and Deborah 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.