Réseaux sans fil

# Contenu du cours

12h de CM avec Guitton :

* Réseaux de capteurs sans fil
* MAC de 802.15.4
* Routage WSN
* Programmation WSN

12h de CM avec Chalhoub

* Médium sans fil
* MAC de 802.11
* Routage MANET

Evaluation :

* Ecrit #1 🡪 partie Chalhoub (33%)
* Ecrit #2 🡪 2 parties : Chalhoub (33%) et Guitton (67%)

# Introduction

Types de réseaux filaires :

* LAN : Local Area Network
* WAN : Wide Area Network
* MAN : Métropole Area Network

Types de réseaux sans fil :

* BAN : Body Area Network (2m)
  + Niveau perso
  + Pacemaker
  + Bracelet sportif
* WPAN : Wireless Personnal Area Network (10m)
  + Niveau personne mais autour de la personne
  + Pièces
* WLAN : Wireless LAN (100m – 1km)
  + Bâtiment
* WAN : Wide Area Network
  + Comme WAN filaire mais sans fil
* LP-WAN : Low Power WAN
  + Basse conso
* LR-WPAN : Low Rate WPAN

Réseaux de capteurs sans fil (Wireless Sensor Networks) :

* Nombreux équipements pas chers si possible
* Capable de faire des traitements
* Dispose d’une mémoire
* Communication sans fil
* Source d’énergie
* Capteurs / actionneurs

Le but des réseaux de capteurs sans fil est de faire de la surveillance (en tout genre).

* Réseaux statiques avec infrastructure (station de base)

Autres réseaux :

* WSN : statique
* VANET : Vehiculor (qui peut bouger) et Ad Hoc (sans infrastructure)
* MANET : Mobile Ad-Hoc (comme les VANET mais bouge plus lentement)

Caractéristiques importantes :

* Portée
* Robustesse / bruitage
  + Si le signal est bon ou pas
  + Compréhensible même avec des pertes
* Energie
  + A quelle fréquence on doit recharger les batteries
* Débit
  + A quelle vitesse jeux parler
  + Vitesse de la communication

# IEEE 802.15.4

Catégorie LR-WPAN (Low Rate Wireless Personnal Area Network) : débit faible mais du coup dépense en énergie faible.

Le 802.15.4 définit la couche physique et la couche MAC (les couches en dessous sont définis par d’autres protocoles).

Fréquence de couche physique :

* 868 MHz – EU : 1 canal
* 9,5 MHz – US : 30 canaux
* 2450 MHz : 16 canaux séparés

Pour un débit de 250 kbps

Plus le débit est faible, plus la portée peut être grande.

Plus la fréquence est basse plus la qualité du signal peut être améliorée.

Taille maximum d’une trame est de 127 octets.

## Couche MAC

### Topologie :

* FFD : Full Function Device
  + Routage
* RFD : Reduced Function Device

Un RFD ne peut parler qu’à un FFD.

Dans une topologie étoile le nœuds centrale sera forcément un FFD.

Dans une topologie paire à pair (réseaux maillé) :

* Les RFD sont relié qu’à un seul FFD
* Les FFD peuvent être relié à d’autres FFD et à un RFD

### Adressage

* Adresse longue (64 bits)
* Adresse courte (16 bits)
  + Topologique

### Accès au médium

* Beacon-enabled mode
  + Mode économie d’énergie
  + On transmet e façon régulière un beacon
  + Entre 2 beacons on a un intervalle **BI**
  + Juste après le beacon on a une période où on est actif (là où on est le plus précis) : **SD** (Superframe Duration)
  + Après cette période on retombe dans une période d’inactivité (jusqu’au prochain beacon)
  + Duty-cycle = 1% (la proportion d’activité : SD/BI)
  + BO : Beacon Order

SO : Superframe Order

0 <= SO <= BO <= 14

SD = 15,36ms \* 2^SO

BI = 15,36ms \* 2^BO

* + On divise la supertrame en 16 slots de taille équivalente : les 16 slots sont répartis en 2 groupes : le CAP et le CFP

CAP : Contention Access Period (tout le monde peut parler en même temps on aura des collisions si plusieurs personnes veulent parler en même temps)

CSMA/CA : Carrier Sense Multiple Access with collision Avoidance 🡪 efficace à bas débit quand on parle beaucoup

CFP : Collision Free Period (la prise de parole se fait à tour de rôle, on n’a donc pas de collision)

TDMA : Time Division Multiple Access

Taille appartient à [0,7] slots

* Non beacon enabled mode
  + Pas de beacon périodique
  + Minislot mais pas de synchronisation
  + Même algo sauf que CW = 1
  + Pas de sommeil => pas d’économie d’energie

### CAP

Algo CSMA / CA

* NB : nombre de tentative (le nombre de fois qu’on essaie d’accéder au médium)
* BE : Backoff Exponent (délai entre 2 tentatives)
* CW : Congestion Window (durée d’écoute du canal avant TX)

La durée du CAP est divisée en mini-slots 🡪 la durée en 2 mini-slots est de 320 micro secondes

Déroulement de l’algorithme :

* Attente frontière minislot
* Init CW = 2, BE = 3, NB = 1
* Backoff = random(0, 2^BE – 1)
* Attente de 1 minislot et backoff –
* Tant que le backoff != 0 on continue sinon on passe à la suite
* CCA : Clean Channel Assesment (on check si le canal est dispo ou pas) 🡪 écoute pendant environ 320 micro secondes
* Si le canal est IDLE (dispo)
  + CW –
  + Si CW > 0, on retourne au CCA sinon on continue
  + Si CW = 0 🡪 **transmission**
* Si le canal est BUSY (occupé)
  + NB = NB + 1
  + BE = min(BE + 1, 8)
  + CW = 2
  + On retourne à l’étape du calcul du backoff

Cet algorithme ne garantie pas qu’il n’y aura pas de collision, il essaie de les éviter.

Après la réception d’une trame => on attend 1 minislot de 320 micro secondes et on envoie un ACK. SI pas d’ACK on retransmet (avec CSMA / CD)

### CFP

La CFP est divisée en GTS (Guaranteed Time Slot) alloué à un nœud spécifique (sur demande)

### Transmission entre coordinateur et terminaux

Si on prends un réseaux avec des coordinateurs et des terminaux (End Device)

Les transmissions ED 🡪 coordinateur sont décidés par les ED

Transmission coordinateurs 🡪 ED

Soit après une communication ED 🡪 coordinateur

Soit c’est indiqué dans le beaocn (frame pending)

# Adressage

Adresse longue (64 bits)

Adresse courtes (16 bits)

SAAM = Stochastic Address Allocation Mechanism

DAAM = Distributed Address Allocation Mechanism

Cluster tree :

FFD

coordinateur

RFD

Rm = #max de fils routeur (ici = 2)

Cm = #max de Children/ fils (ici = 4)

Lm = Length max (ici = 3)

Le coordinateur à une adresse et s’occupe de toutes les autres. Il attribue donc les adresses des FFD puis divise en 2 les adresses restantes et les donnes aux RFD. Les RFD prennent une adresse et assigne des adresses aux FFD. Il sépare ensuite en 2 les adresses restantes et les donnent aux routeurs suivants et ainsi de suite.

Le nombre d’adresse que l’on saute entre 2 end devices :

Cskip(d) = 1 + Cm \* (Lm – d – 1) si Rm = 1 (avec d la profondeur)

Sinon

(Cm \* Rm^(Lm-d-1) + Rm – Cm – 1) / (Rm – 1)

Adresse du xème routeurs (x >= 1) :

Aparent + (x -1) \* Cskip(d) + 1

Adresse yème feuille (y >= 1) :

Aparent + Rm \* Cskip(d) + y

# Programmation de motes en NesC

## Définition

* Motes : Nœud d’un réseau de capteurs (arduino, TelosB, …)
* NesC : langage basé sur le C
* TinyOS : OS des motes

## NesC

* Evenement :
  + Ne peut pas être interrompu
  + Court
  + Déclencher par un stimulus
  + Peut provoquer d’autres événements
* Taches
  + Plus long
  + Peut être interrompus que par un évènement
* Composants
  + Implémente des commandes
  + Utilisent des évènements
  + Commande implémente des fonctions
  + Les commandes ne peuvent pas lancer d’évènement
* Ordonnanceur
  + Ordonne en FIFO
  + Les taches peuvent être interrompus par des évènements on reprend la tache après l’évènement
  + Les évènements peuvent s’arrêter entre eux
* Mot clefs
  + Poster une tache : post
  + Appeler une commande : call
  + Lancer un évènement : signal

# Routage

## Dans Zigbee

* Standardiser par un consensus d’entreprise.
* AODV : si adresses SAAM
* Routage hiérarchique : si DAAM

### Routage hiérarchique

* Si Adresse destination = Adresse nœud alors transmission couche supérieur
* Si nœud = RFD alors envoie au père
* SI Adresse nœud = 0 ou Adresse destination appartient à [Anoeud + 1 ; Anoeud + Rm\*Cskip(D-1) + Cm – Rm] alors

Si Adest est une feuille directe alors envoie feuille

Sinon calculer x <- 1 + |(Adest – Anoeud – 1) / Cskip(d)| alors envoie au xème FFD

Sinon envoie au père

### AODV

Ad-Hoc On Deman Distance Vector

* 2 types de routages
  + Routage pro-actif : on garde les routes même si elles ne sont pas utilisées
  + Routage actif : si une route n’est plus utilisée on la supprime, il faut donc recalculer les routes souvent.
* AODV est un routage actif
* Etape établissement de route
  + RREQ = Route REQest
    - On envoie ce message à tous les nœuds qui renvoient à tous les nœuds
    - Quand la destination reçoit le message elle répond par un RREP au premier nœud qui lui a envoyé le RREQ (on considère que se sera le plus rapide)
    - On remonte ainsi jusqu’à la source
  + RREP = Route REPly

## DSDV

Destination Sequence Dictance Vector

* Similaire à RIP : split horizon, poison update
* Routage pro-actif

X

Nœud actuel

dest

Z

Y

inter

Si Y + Z < X alors

Passer par le nœud intermédiaire pour dest

* Split horizon
  + On ne dit pas a un nœuds qu’il existe des routes qui passe par ce nœuds (on ne raconte pas une blague à la personne qui nous l’a raconté
* Poison update
  + Si on perd une de nos routes, on informe ses voisins que l’on a perdu la route
  + Les autres nœuds font l’update de leur route si ils passent par la route qui a était perdu

## OLSR

Optimized Link State Routing

* MPR : Multi point relay
  + Ensemble (minimum) de nœuds a 1 saut qui peut couvrir tous les nœuds à exactement 2 sauts
  + Réduire les répétitions
  + On ne dit pas à tous les nœuds a 1 saut de prévenir un autre nœud a 2 sauts, on dit à 1 seul nœud à 1 saut qui connait le nœuds à 2 saut
* MS(x) = ensemble des nœuds qui ont x pour MPR
* Des messages Hello : découverte des voisins à 1 – 2 sauts, MPR
* Des messages TC (Topologie Control) -> envoyé seulement pour des MPR (un nœud envoie un TC à destination d’un MS)
* On m’informe que des liens (N, MS(N))

# Révision

0

2

1

25

25

25

76

26

1

49

48

1

7

6

Cm = 6

Rm = 3

Lm = 3

Cskip(0) = (1 – Cm + Rm + Cm \* Rm^(Lm – d – 1)) / Rm – 1

= (-1 – 6 + 3 + 6 \* 3^(3-0-1)) / 2

= 6 + 54 / 2 = 25

Cskip(1) = (-7 + 3 + 6 \* 3) / 2 = 7

Cskip(2) = (-7 + 3 + 6) / 2 = 1

## Examen 2017-2018 session 1

### Exo 2.1 :

Schéma du CAP decoupe en plein de partie.

Si backoffMax + CCA + tempsTx + turnaround + ACK > temps restant

* ((2^BE -1)\*320µs) + (2\*320µs) + temps TX + (1 \* 320µs) + ACK

Alors reporter transmission

Les transmissions en fin de CAP et les trames entres les CAP sont reporté dans le CAP suivant donc beaucoup de transmission au début d’un CAP.

On a beaucoup de collisions en début de CAP car on a beaucoup de trames mais aussi le BE retrouve sa valeur initiale de 3 (donc le backoff diminue), cette valeur est donc très (trop) optimiste.

En 2006 on stocke la valeur de BE si une trame est reportée d’un CAP à un autre.

BE conserve sa valeur en début de CAP, car la compétition est au moins égal à celle en fin de CAP précédent.

## Examen 2018-2019 session 1

### Exo 2.1

Si trame urgente alors

Demander un GTS : espace dédié à nous (trame de requête pendant un CAP)

Attendre le prochain Beacon (info des GTS dans les beacons)

Attendre le GTS

La demande se fait dans une période de sommeil et sera envoyé au prochain CAP. Il faudra ensuite attendre le prochain beacon pour avoir notre GTS attribué et donc attendre le GTS dans CFP (en espérant e avoir 1 de libre). Donc le temps en entre la création et le prochain beacon + un jour + entre le beacon et le GTS

Non on n’est pas sûr d’avoir le temps d’envoyer la requête pour la demande de GTS, si il y a trop de demande on ne pourra faire cette demande.

Le nombre de GTS est limité, on n’est donc pas sur d’en avoir un de disponible tout de suite.

Quand on a un GTS, on est sûr de pouvoir transmettre.

L’approche peut être longue pour obtenir un GTS, on n’est pas sûr d’avoir un GTS mais quand on a un GTS on est sûr de pouvoir transmettre.

Une solution serait de pré-allouer les GTS à chaque nœuds

### Exo 2.3

LoRa ( = long range) 🡪 couche physique

LoRaWAN = couche MAC 🡪 si on veut parler, on parle donc beaucoup de collision.

Trame de 20 octets = 0,45s avec SF = 10

= 1,8s avec SF = 12

Pour SF = 10 🡪 0,45s

SF = 11 🡪 0,45 \* 2 = 0,9s

SF = 12 🡪 0,9 \* 2 = 1,8s

Il faut ajouter un délai aléatoire pour désynchroniser les nœuds qui veulent transmettre en même temps.

Délais random entre 1 et 3s

Le délai est mauvais et généré des collisions quand le SF est grands. C’est encore pire si la taille de la trame augmente.

SF = 10 🡪 1/2s

SF = 9 🡪 1/4s

SF = 8 🡪 1/8s

SF = 7 🡪 1/16s env 80ms

Il faut donc adapter le SF à la taille de la trame.