

Applications Mobiles et Internet des Objets

Suite du cours sur l'IoT

Thibault CHOLEZ - thibault.cholez@loria.fr

TELECOM Nancy - Université de Lorraine

LORIA - INRIA Nancy Grand-Est

CC BY-NC-SA 3.0

25/01/2016

Plan

- 1 802.15.4
- 2 6LoWPAN
- 3 RPL
- 4 COAP
- 5 Contiki OS

Plan

- 1 802.15.4
- 2 6LoWPAN
- 3 RPL
- 4 COAP
- 5 Contiki OS

MAC IEEE 802.15.4

La couche MAC IEEE 802.15.4

d'après "Réseaux de Capteurs Sans Fils" de Yacine CHALLAL

- Utilise deux modes d'adressage IEEE 64-bit et 16-bit
- Plusieurs fréquences : 868Mhz (20k/s), 900Mhz (40k/s), 2400Mhz (250kb/s)
- Utilise une structure de trame simple
- Permet d'utiliser le mécanisme de beaconing ; réveil périodique, vérification de l'arrivée d'un beacon
- Économise l'énergie à travers la mise en veille entre deux beacons, et les noeuds ne devant pas router ou recevoir les données aléatoirement peuvent se mettre en veille.
- Assure une transmission fiable de données
- Offre une sécurité AES-128

Deux types d'implantation

Full Function Device (FFD)

- Implante l'ensemble de la norme
- Compatible avec toutes les topologies
- Peut être coordinateur PAN (1 max par réseau)

Reduced Function Device (RFD)

- Implante une sous-partie de la norme
- Implantation minimum
- Ne peut pas être PAN
- Limité au rôle de feuille dans le réseau

MAC IEEE 802.15.4

Champs principaux de la trame

- Préambule pour synchroniser les noeuds
- adresse IEEE 64-bit (globalement unique) ou 16-bit "short" (unique dans PAN)
- Type de la trame : Beacon, Command, Data, Ack
- max. frame size 127 octets ; max. frame header 25 octets

Network Beacon

- Identifie et organise le réseau
- Décrit la structure de la super-trame
- Indique la présence de données
- Présent uniquement lorsque le réseau est actif
- Optionnel, géré par le PAN

Mécanisme d'accès au canal

Network Beacon

IEEE 802.15.4 utilise CSMA/CA décliné en deux versions selon la configuration du réseau :

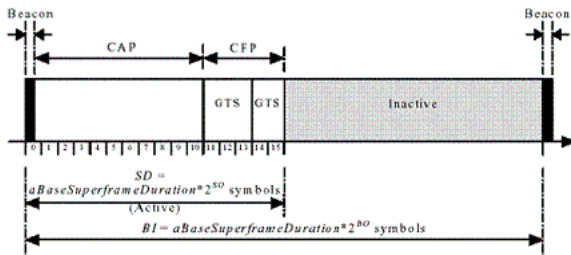
- Si le "beaconing" n'est pas utilisé, IEEE 802.15.4 utilise CSMA/CA sans slots (classique). Adapté pour capteurs émettant peu (ex : interrupteur), coordinateur alimenté.
- Si le "beaconing" est utilisé, IEEE 802.15.4 utilise CSMA/CA avec slots et la structure super-trame. Adapté aux topologies ad-hoc, synchronise les noeuds.

Mécanisme d'accès au canal

Super-frame

La super-frame est composée de deux parties :

- Inactive-Period : toutes les stations dorment ZZZzzz
- Active : Période active composée de
 - Contention access period (CAP) : channel partagé en CSMA/CA
 - Contention free period (CFP) : temps de parole garanti par noeud (GTS)



Mécanisme d'accès au canal

Super-trame

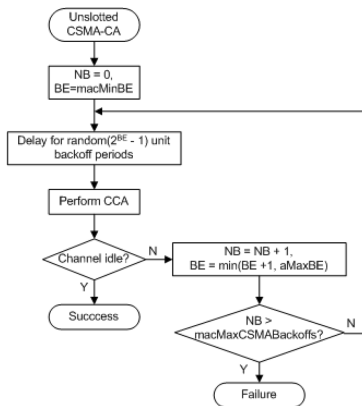
Deux paramètres déterminent la longueur de la super-trame :

- Superframe Order (SO) : détermine la longueur de la période active
- Beacon Order (BO) : détermine la longueur d'une période de beacon.

Dans CFP, un GTS (Guaranteed Time Slots) peut être constitué de plusieurs slots, attribués à un seul noeud, pour transmission (t-GTS) ou réception (r-GTS).

Dans CAP, le concept de slots n'est pas utilisé. CAP est divisé en de plus petits slots de contention. Chaque slot de contention a une longueur de 20 symboles. C'est la plus petite unité de contention backoff. Dans ce cas, les noeuds entrent en contention suivant CSMA/CA avec slots.

802.15.4 CSMA/CA algorithm



From An Enhanced Reservation-Based MAC Protocol for IEEE 802.15.4 Networks, José A. Afonso et al.

L'algorithme CSMA/CA

Principe

Chaque noeud doit maintenir trois variables pour chaque tentative de transmission

- NB : nombre de fois l'algorithme CSMA/CA fait backoff durant la tentative de transmission en cours
- BE (Backoff Exponent) : détermine le nombre de périodes backoff qu'un noeud doit attendre avant de tenter d'accéder au canal.
- CW (Contention Window) : Longueur de la fenêtre de contention ; le nombre de slots backoff sans aucune activité de canal avant de commencer la transmission. CW est Initialisé à 2 et remis à 2 si le canal est détecté occupé. Une station doit détecter 2 CCA (Clear Channel Activity) avant d'entrer en contention.

Plan

- 1 802.15.4
- 2 6LoWPAN**
- 3 RPL
- 4 COAP
- 5 Contiki OS

IPv6 Low power Wireless Personal Area Networks

Objectif et problématique

- Objectif : assurer une compatibilité IPv6 au dessus de IEEE 802.15.4
- PSDU 802.15.4 de 127 octets...
 - 802.15.4 MAC header = 25 octets (+21 si AES-CCM-128)
 - 127-25-40(IPv6)-8(UDP) = 54 octets utiles (sans chiffrement)
 - 127-46-40-8 = 33 octets utiles (AES-CCM-128)
- Difficultés : taille importante des en-têtes IPv6, MTU IPv6 de 1280 octets, contraintes liées à 802.15.4
- Besoin de fragmentation + ré-assemblage, compression, simplification du plan de contrôle (NDP)

Protocole clé

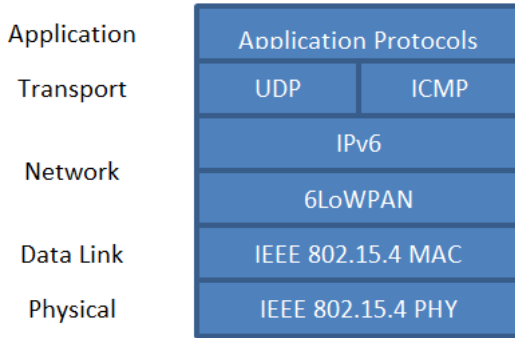
- Standardisé par l'IETF : RFC 4944, spécifications ouvertes
- Largement supporté : Contiki, TinyOS, ZigBee, Android, etc.

Principe de 6LoWPAN

Compression Principles (RFC 4944)

- Omit any header fields that can be calculated from the context, send the remaining fields unmodified
- Nodes do not have to maintain compression state (stateless compression)
- Support (almost) arbitrary combinations of compressed / uncompressed header fields

Pile protocolaire 6LoWPAN



En-têtes d'encapsulation

Différentes en-têtes 6LoWPAN

- Suivent directement le payload 802.15.4, "Dispatch code" sur 1 octet
- Chaque en-tête peut être suivie par 0 ou plusieurs autres
- Ordre à respecter : Mesh Addressing Header > Fragmentation Header > HC1 Header

A LoWPAN encapsulated IPv6 datagram:

```
+-----+-----+-----+
| IPv6 Dispatch | IPv6 Header | Payload |
+-----+-----+-----+
```

A LoWPAN encapsulated LOWPAN_HC1 compressed IPv6 datagram:

```
+-----+-----+-----+
| HC1 Dispatch | HC1 Header | Payload |
+-----+-----+-----+
```

A LoWPAN encapsulated LOWPAN_HC1 compressed IPv6 datagram with mesh addressing:

```
+-----+-----+-----+-----+-----+
| Mesh Type | Mesh Header | HC1 Dispatch | HC1 Header | Payload |
+-----+-----+-----+-----+-----+
```

A LoWPAN encapsulated LOWPAN_HC1 compressed IPv6 datagram with fragmentation:

```
+-----+-----+-----+-----+-----+
| Frag Type | Frag Header | HC1 Dispatch | HC1 Header | Payload |
+-----+-----+-----+-----+-----+
```


Communication Mesh

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|1 0|V|F|HopsLft| originator address, final address
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

- Deux types d'adresses possibles : adresse 64-bit (EUI-64) si $V/F = 0$ ou adresse 16-bit si $V/F = 1$
- V : "Very first" : adresse originelle, F "Final Destination"

Deux modes de routage

- Mesh-under (couche adaptation 6LoWPAN), latence réduite
- Route-over (couche réseau IPv6), meilleure fiabilité

Fragmentation et ré-assemblage

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 1 0 0 0 |   datagram_size   |   datagram_tag   |
+-----+-----+-----+-----+-----+-----+-----+-----+

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 1 1 0 0 |   datagram_size   |   datagram_tag   |
+-----+-----+-----+-----+-----+-----+-----+-----+
| datagram_offset |
+-----+-----+

```

Les paquets IPv6 sont fragmentés avant d'être transmis sur le WSN. Temps pour ré-assembler : 60 sec. Pas de récupération possible. En-tête de fragmentation :

- 5 bits "dispatch" : 1er fragment = 11000, les suivants 11100 ;
- 11 bits "datagram_size" : taille du paquet IP original ;
- 16 bits "datagram_tag" : identifie les fragments issus d'un même paquet IP ;
- 8 bits "datagram_offset" : numéro du fragment si différent du premier.

Compression d'en-tête

Principes

- Compression basée sur les informations connues par tout le WSN (pas d'état de compression entre noeuds)
- Supprimer toute information qui peut être recalculée
- Différentes combinaisons d'en-têtes compressées / non compressées
- Meilleure compression obtenue pour les adresses locales
- Une compression HC1 peut être suivie d'une compression HC2
- En-têtes non-compressibles placées après les tags HC1 ou HC1/HC2 (compression partielle)

```

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| HC1 encoding |      Non-Compressed fields follow...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Construction de l'en-tête HC1

Encodage des adresses

- PI : Préfixe complet suivant l'en-tête
- PC : Préfixe compressé (préfixe de lien-local sous-entendu)
- II : Identifiant d'interface suivant l'en-tête
- IC : Identifiant d'interface compressé (déductible depuis l'adresse MAC)

IPv6 source address (bits 0 and 1) | IPv6 destination address (bits 2 and 3):

00: PI, II

00: PI, II

01: PI, IC

01: PI, IC

10: PC, II

10: PC, II

11: PC, IC

11: PC, IC

Construction de l'en-tête HC1

| Traffic Class and Flow Label (bit 4): | | Next Header (bits 5 and 6): |
|---|--|--------------------------------------|
| 0: not compressed (8 bits for Traffic Class + 20 bits for Flow Label) | | 00: not compressed (8 bits are sent) |
| | | 01: UDP |
| | | 10: ICMP |
| | | 11: TCP |
| 1: Traffic Class and Flow Label are zero | | |

Exemple : meilleure compression possible

- Adresses IPv6 locales : déduites depuis adresses MAC
- Taille du paquet déduite depuis taille de la frame ou taille du fragment
- Traffic Class et Flow Label à zero, Hop Limit gardé à 8 bits
- Résultat : 2 octets à la place de 40 octets

Compression d'autres en-têtes : HC2

HC2 encoding(bit 7):

0: No more header compression bits

1: HC1 encoding immediately followed
by another header compression (UDP, ICMP, or TCP)

Exemple : compression supplémentaire pour UDP

- UDP ports source et destination compressés sur 4 bits :
 $P + \text{shortportvalue}(P = 61616)$
- Longueur du segment omis : calculée à partir de l'en-tête IPv6
 $\text{UDP length} = \text{IPv6 header(Payload Length)} - \text{extension headers}$

Plan

- 1 802.15.4
- 2 6LoWPAN
- 3 RPL**
- 4 COAP
- 5 Contiki OS

RPL

On reprend ce support S167-S214 :

`http://thibault.cholez.free.fr/teaching/AMIO/AMIO_Wireless_Sensor_Network_Lahmadi.pdf`

Plan

- 1 802.15.4
- 2 6LoWPAN
- 3 RPL
- 4 COAP**
- 5 Contiki OS

Constrained Application Procol

On suit ce support S24-S44 :

<http://fr.slideshare.net/carlosralli/curso-coap-v3slideshare>

Plan

- 1 802.15.4
- 2 6LoWPAN
- 3 RPL
- 4 COAP
- 5 **Contiki OS**

Contiki OS

On reprend ce support S93-S110 :

`http://thibault.cholez.free.fr/teaching/AMIO/AMIO_Wireless_Sensor_Network_Lahmadi.pdf`