



HAL
open science

Interopérabilité de protocole de communication adaptatifs basse-consommation pour des réseaux de capteurs

Elodie Morin

► **To cite this version:**

Elodie Morin. Interopérabilité de protocole de communication adaptatifs basse-consommation pour des réseaux de capteurs. Réseaux et télécommunications [cs.NI]. Université Grenoble Alpes, 2018. Français. NNT : 2018GREAM022 . tel-01903194

HAL Id: tel-01903194

<https://theses.hal.science/tel-01903194>

Submitted on 24 Oct 2018

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.

THÈSE

Pour obtenir le grade de

DOCTEUR DE LA COMMUNAUTE UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 7 août 2006

Présentée par

Elodie Morin

Thèse dirigée par **Andrzej Duda**

et coencadrée par **Mickael Maman** et **Roberto Guizzetti**

préparée au sein de

Laboratoire d'Informatique de Grenoble (LIG),

STMicroelectronics Crolles (France),

CEA-Leti Grenoble (France),

et de l'**École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique (EDMSTII).**

Interopérabilité de protocoles de communication adaptatifs basse-consommation pour des réseaux de capteurs

Thèse soutenue publiquement le **24 Avril 2018**,

devant le jury composé de :

Isabelle Guérin Lassous

Professeure, Université Claude Bernard Lyon 1, Présidente

Fabrice Valois

Professeur, INSA de Lyon, Rapporteur

Congduc Pham

Professeur, Université de Pau, Rapporteur

M. Andrzej Duda

Professeur, Université Grenoble Alpes, Directeur de thèse

M. Mickael Maman

Ingénieur, CEA Léti, Co-Encadrant de thèse

M. Roberto Guizzetti

Ingénieur, STMicroelectronics, Co-Encadrant de thèse



Table des matières

Table des matières	3
Liste des figures	5
Liste des acronymes	9
Liste des symboles	11
Avant-propos	13
Remerciements	13
Résumé en français	15
Résumé en anglais	16
Introduction	17
.1 Contexte	17
.2 Motivations	17
.3 Contributions	19
.4 Structure du document	20
I L'interopérabilité pour l'IoT	23
1 – Technologies de l'IoT	25
1.1 Solutions courte portée : les WPAN	27
1.2 Solutions LAN	41
1.3 Solutions LPWAN	44
2 – Discussion	49
2.1 Motivations pour l'interopérabilité	50
2.2 État de l'art des solutions d'interopérabilité	56
2.3 Pistes d'exploration	60
II Contributions	69
3 – Comparaison des standards pour l'IoT	71
3.1 Principe de l'analyse de consommation énergétique	73
3.2 Impact des paramètres matériels sur les résultats	78
3.3 Comparaison des durées de vie des périphériques	81
4 – Réduction du coût d'attache au réseau	93
4.1 Les protocoles de découverte de voisinage	95
4.2 Simulateur de la procédure d'attachement au réseau	102
4.3 Singer-Cyclic Difference Set	103
4.4 <i>DataJoin</i>	111
5 – Application à l'interopérabilité Google Thread/TSCH	121
5.1 Principe de fonctionnement	124
5.2 Ordonnancement des slots d'interopérabilité	128
5.3 Évaluation des performances théoriques	131
5.4 Conclusion	146

Conclusion générale	149
.1 Résumé de la contribution	149
.2 Travaux futurs	151
 Bibliographie	 153

Liste des figures

1.1	Vue d'ensemble des technologies sans-fil.	26
1.2	Format d'un paquet de données 802.15.4 (taille en bits).	28
1.3	Fonctionnement MAC d'une feuille qui souhaite envoyer un paquet à un parent en mode non- <i>beacon</i>	29
1.4	Détail d'une supertrame 802.15.4 en mode balise.	29
1.5	Opérations du 802.15.4 en mode balise.	30
1.6	Exemple d'ordonnancement TSCH.	31
1.7	État d'opération d'un nœud au sein d'un slot (Créneau temporel) TSCH.	31
1.8	Macro-machine d'état d'un nœud TSCH	32
1.9	Principe de fonctionnement du mode CSL.	33
1.10	Principe de fonctionnement du mode RIT.	34
1.11	Pile protocolaire Google Thread ©Threadgroup [1].	35
1.12	Piles Zigbee.	37
1.13	Pile protocolaire openWSN.	37
1.14	Piles BLE.	38
1.15	Présentation globale du BLE : Efficacité maximale.	40
1.16	Format minimal du paquet de données BLE (taille en bits).	40
1.17	Échange de messages BLE pour un mode de communication synchrone.	41
1.18	Opération d'un nœud dans le mode d'économie d'énergie PSM et le mode DCF du standard 802.11.	42
1.19	Structure d'un paquet 802.11 PSM (taille en bits).	42
1.20	Tableau présentant les débits accessibles (en Mb/s) avec la technologie 802.11ah, et leurs caractéristiques physiques associées.	43
1.21	Structure d'une trame de donnée 802.11ah (taille en bits).	44
1.22	Structure d'un paquet SIGFOX de voie montante (taille en bits).	44
1.23	Structure d'un paquet SIGFOX de voie descendante (taille en bits).	45
1.24	Tableau récapitulatif des débits disponibles LoRa.	46
1.25	Description du comportement d'un nœud LoRa de classe A au cours du temps.	47
1.26	Structure d'un paquet LoRa (taille en bits).	47
2.1	Taxonomie des différents mécanismes de DC utilisés au niveau MAC.	51
2.2	Les horloges imparfaites : la dérive.	53
2.3	Impact de la dérive d'horloge sur la transmission entre deux plateformes.	54
2.4	Interopérabilité de deux réseaux au niveau 3, la couche IP.	57
2.5	La mise en commun de la ressource physique.	58
2.6	Collision de deux événements MAC distincts au sein d'une unique plateforme matérielle im- pliquée dans deux communications de technologies différentes avec deux destinataires diffé- rents.	58
2.7	Pile protocolaire d'une plateforme qui réalise une optimisation conjointe multiMAC.	59
2.8	Pile protocolaire d'une plateforme qui met en place une surcouche d'optimisation MAC.	59
2.9	Interopérabilité synchrone/asynchrone : mono et multitechnologie.	61
2.10	Procédure classique d'attache au réseau.	62
2.11	Comparaison du coût de la synchronisation pour des messages applicatifs générés à des in- tervalles différents.	63
2.12	Les quatre types de transmissions pour l'interopérabilité synchrone/asynchrone.	64
3.1	Piles protocolaires des technologies comparées, et principe de fonctionnement de l'analyseur utilisé pour la comparaison.	74
3.2	Principe et variables utilisées lors de la comparaison.	75
3.3	Algorithme de calcul de la durée de vie d'un nœud.	77

3.4	Impact de la plateforme matérielle sectionnée sur la durée de vie L_t , pour une énergie initiale $E_0 = 13.5$ kJ.	79
3.5	Différentes périodes du trafic applicatif, T_a , ainsi que des tailles de données applicatives variables (s_a en octets), permettent de représenter différents débits r_a , et leurs impacts sur la durée de vie des périphériques IoT pour une énergie initiale $E_0 = 13.5$ kJ, un PER nul, et aucune dérive d'horloge.	82
3.6	Description des différents modes d'opération d'un nœud au sein d'un slot, en utilisant le modèle de consommation à 6 états pour TSCH.	83
3.7	Durées de vie estimées pour les différents débits disponibles du standard BLE 5.0, et comparaison avec un 802.15.4 théorique au débit de 2Mb/s	85
3.8	Durées de vie pour des débits applicatifs variables (r_a), un taux d'erreur de paquet moyen de 20%, une énergie initiale $E_0 = 13.5$ kJ, et des horloges parfaites.	86
3.9	Durées de vie pour différents débits applicatifs r_a et une prise en compte d'une dérive d'horloge relative maximale $\delta_{\max} = +2\Delta = 80$ ppm, pour une énergie initiale $E_0 = 13.5$ kJ, et un taux d'erreur de paquet nul (PER(<i>Taux d'erreur de paquets</i> – – ou <i>Packet Error Rate</i>) = 0).	88
3.10	Puissance consommée moyenne par T_a , PER = 0.2.	90
4.1	Taxonomie des protocoles de découverte de voisinage dans le cas de réseaux opportunistes pour l'IoT.	95
4.2	Principe de fonctionnement du système à logique floue exploitée par <i>Rapid Join</i> , l'algorithme d'ordonnancement de l'envoi des EB proposé par Duy <i>et al.</i> [2].	96
4.3	Principe de fonctionnement de l'algorithme d'ordonnancement d'envoi des EB RV/RH proposé par Vogli <i>et al.</i> [3].	97
4.4	Principe du protocole de ND SL-S.	99
4.5	Principe de fonctionnement de Nihao.	99
4.6	Modèle de découverte en temps continu PI-0M.	100
4.7	Exemple pour le 7,3,1-CDS parfait : {0,1,3}.	101
4.8	Schéma de principe du simulateur de procédure d'attache au réseau.	102
4.9	Gestion du non-alignement des slots dans le cas des CDS.	104
4.10	Principe de comparaison de la proposition S-CDS à l'existant.	106
4.11	Performances des différents protocoles de ND pour $SDC_N \approx 1,6$	108
4.12	DC radio du nœud J (DC_j) en fonction du DC d'un nœud N en termes de slots SDC_N	109
4.13	Délai d'attache pour les protocoles de ND considérés en fonction du DC d'un nœud N en termes de slots SDC_N	109
4.14	Temps passé en Rx par le nœud J des protocoles de ND considérés en fonction du DC d'un nœud N en termes de slots SDC_N	109
4.15	Légende pour les figures 4.12, 4.13, et 4.14.	109
4.16	Schéma de principe de fonctionnement de <i>DataJoin</i>	111
4.17	Répartition de l'allocation des cellules d'Enh-ACK afin de minimiser la consommation énergétique.	112
4.18	Exemple de topologie simulée pour évaluer les performances du protocole <i>DataJoin</i>	114
4.19	Exemple d'un ordonnancement TSCH avec 5 canaux logiques (CO), 5 canaux physiques (numéros dans les cellules), et une <i>multislotframe</i> de 30 slots.	114
4.20	Comparaison des performances <i>Baseline</i> et <i>DataJoin</i>	117
4.21	Comparaison de performances pour une période de paquet variable (en s).	118
5.1	Cohabitation de réseaux Thread et TSCH.	122
5.2	Présentation graphique de la problématique d'interopérabilité.	123
5.3	Détail du fonctionnement au sein d'un slot d'interopérabilité.	125
5.4	Principe de l'allocation des slots d'EB contenant les informations d'interopérabilité.	129
5.5	Comportement temporel des nœuds du réseau sur un canal fréquentiel donné.	131
5.6	Représentation temporelle du point de vue du nœud asynchrone (sur un canal physique donné).	133
5.7	Schéma présentant le calcul de l'espérance du nombre de voisins.	135
5.8	Espérance du nombre moyen de voisins.	136
5.9	Latence moyenne du trafic asynchrone (avec approximation).	137

5.10	Description détaillée du comportement temporel des différents acteurs avant et après la prise en charge d'un trafic asynchrone au sein d'un réseau synchrone.	139
5.11	Énergie consommée par le nœud asynchrone sur un intervalle T_a , pour deux valeurs différentes de T_a avec et sans la prise en charge du trafic asynchrone.	140
5.12	Énergie moyenne consommée par seconde par un nœud du réseau synchrone avec ou sans la prise en charge du trafic asynchrone à proximité.	142
5.13	Énergie moyenne consommée par seconde par un nœud du réseau synchrone TSCH, par le nœud asynchrone Thread, et la latence subie par le trafic asynchrone en fonction de la taille de la <i>slotframe</i>	143
5.14	Consommation globale dans la proposition classique, et détail de la consommation d'un nœud TSCH, du nœud Thread alimenté, et du nœud Thread contraint.	144
5.15	Consommation globale de notre proposition, et détail de la consommation d'un nœud TSCH et du nœud Thread.	145
5.16	Évaluation globale des performances.	146

Liste des acronymes

6LoWPAN	Réseau personnel (PAN) sans fil IPv6 à basse consommation — <i>IPv6 Low power Wireless PAN</i> . 35–37, 41, 42, 46, 73, 75
ACK	Acquittement. 32, 33, 47, 76, 78, 111–116, 118, 121, 126, 127, 150
ADV	Trame de synchronisation du BLE, nommée <i>ADvertisement frame</i> . 39, 40, 92
AODV	Protocole de routage ad-hoc, à la demande, et de type vecteur de distance — ou <i>Ad-hoc On-demand Distance Vector</i> . 37
AP	Point d'accès — ou <i>Access Point</i> . 41–43
ASN	Numéro de slot absolu — ou <i>Absolute Slot Number</i> . 31, 113, 115, 129, 130
BLE	Bluetooth Low Energy. 19, 26, 27, 38–41, 44, 57, 60, 63, 72, 73, 75, 78–81, 83–85, 87, 89–92, 94, 100, 149, 150
CAP	Période de communication durant laquelle l'accès au canal se fait en compétition avec d'autres utilisateurs — ou <i>Contention Access Period</i> . 29
CCA	Évaluation d'un canal libre — ou <i>Clear Channel Assessment</i> . 29
CDS	Ensemble de différences cyclique — ou <i>Cyclic Difference Set</i> . 100–105, 119
CFP	Période de communication dans laquelle l'accès au canal est déterminé <i>Contention Free Access Period</i> . 29
CoAP	Protocole d'application contraint — ou <i>Constrained Application Protocol</i> . 36, 38, 41
CSL	Ecoute coordonnée de préambule — ou <i>Coordinated Sampled Listening</i> . 28, 33, 34, 56, 124–127
CSMA/CA	Écoute d'un support à accès multiple et à évitement de collision — ou <i>Carrier Sense Multiple Access-Collision Avoidance</i> . 27–31, 35, 42, 66, 125, 127
DC	Ratio d'activité cyclique — ou <i>Duty-Cycle</i> . 30, 39, 51, 52, 56, 64, 84, 102, 105, 107, 110, 119, 121, 143
DCF	Fonction de coordination distribuée — ou <i>Distributed Coordinated Function</i> . 42, 43
DIFS	Intervalle distribué entre les trames — ou <i>Distributed InterFrame Space</i> . 79
DSSS	Étalement de spectre à séquence directe — ou <i>Direct-Sequence Spread Spectrum</i> . 28
EB	Balise améliorée — ou <i>Enhanced Beacon</i> . 12, 31, 32, 34, 61, 62, 94, 96–98, 102–107, 110–116, 121, 124, 126–134, 137, 141
ETSI	Institut européen des normes de télécommunications — ou <i>European Telecommunications Standards Institute</i> . 42, 44, 46, 83, 92, 149
HTTP	Protocole de transfert hypertexte — ou <i>HyperText Transfer Protocol</i> . 38, 41
IE	Élément d'information — ou <i>Information Element</i> . 33–35, 111–113, 126, 141, 142
IEEE	Institut des ingénieurs électriciens et électroniciens — ou <i>Institute of Electrical and Electronics Engineers</i> . 19, 41
IoT	Internet des Objets — ou <i>Internet of Things</i> . 17, 19, 20, 38, 39, 42, 44–46, 48, 50, 52, 55–58, 61, 67, 72–74, 82, 84, 85, 89, 91, 95, 149, 151, 152
IP	Internet Protocol. 35–42, 44, 57, 73, 75
KA	Maintien vivant — ou <i>Keep-Alive</i> . 32, 62, 64, 77, 87
LAN	Réseau local — ou <i>Local Area Network</i> . 26

LPWAN	Réseau personnel sans-fil à faible consommation énergétique — ou <i>Low Power Wide Area Network</i> . 26
MAC	Contrôle d'accès au support — ou <i>Medium Access Control</i> . 17, 18, 20, 21, 26–29, 33, 34, 37–39, 42–46, 48, 50–52, 54–63, 65, 67, 68, 72–76, 78, 79, 91, 122, 124–126, 146, 149, 151
MCS	Schéma de modulation et de codage — ou <i>Modulation & Coding Scheme</i> . 43, 72, 85
ND	Découverte de voisinage — ou <i>Neighbour Discovery</i> . 95, 96, 98, 100–107, 115, 119
OSI	Interconnexion de systèmes ouverts — ou <i>Open Systems Interconnection</i> . 56, 57, 149
PAN	Réseau personnel — ou <i>Personal Area Network</i> . 29, 128–130
PER	Taux d'erreur de paquets — ou <i>Packet Error Rate</i> . 6, 73, 76, 77, 81, 82, 85, 88–90
PHY	Couche physique du modèle OSI. 26–28, 36, 39, 42, 44, 48, 50, 57, 58, 65, 75, 78, 81, 84, 91, 115, 122, 151, 152
PSM	Mode d'économie d'énergie — ou <i>Power Saving Mode</i> . 26, 41, 42, 48, 72, 80, 84, 85, 89, 92, 110
QoS	Qualité de service — ou <i>Quality of Service</i> . 40
RAW	Fenêtre d'accès restreinte — ou <i>Restricted Access Window</i> . 43
RIP	Protocole de routage de l'information — ou <i>Routing Information Protocol</i> . 36
RIT	Transmission initiée par le récepteur — ou <i>Receiver Initiated Transmission</i> . 28, 34, 35, 56, 111, 124–129, 131–135, 141, 142, 146, 147
RPL	Protocole de routage pour les réseaux à faible consommation d'énergie et avec pertes — ou <i>Routing Protocol for Low-Power and Lossy Networks</i> . 36–38
RSSI	Mesure de la puissance en réception d'un signal reçu — <i>Received Signal Strength Indication</i> . 128, 130
S-CDS	Séquence de CDS générée à partir des séquences de Singer. 104–107, 110, 111, 119
SED	Noeud terminal endormi — ou <i>Sleepy End Device</i> . 34
SIFS	Court intervalle entre les trames — ou <i>Short InterFrame Space</i> . 79
slot	Créneau temporel. 5, 6, 19, 21, 29–32, 42, 48, 77, 79, 83, 97–107, 109, 110, 112–114, 118, 119, 124–132, 134, 135, 137, 138, 141, 142, 144, 145
ST	Durée d'un slot — ou <i>Slot Time</i> . 131–133, 140
TCP	Protocole de contrôle de transmissions — ou <i>Transmission Control Protocol</i> . 36, 38, 41, 42, 46
TSCH	Saut de fréquence à créneaux temporels — ou <i>Time-slotted Channel Hopping</i> . 18–21, 28, 30–32, 34, 38, 59–61, 64, 65, 68, 72, 77, 79, 80, 83, 84, 87, 89, 91, 92, 94–99, 102–106, 110–113, 115, 116, 119, 121–129, 131–133, 135, 136, 138, 141–147, 150, 152
UDP	Protocole de datagramme utilisateur — ou <i>User Datagram Protocol</i> . 36, 38, 73
WED	Noeud terminal que l'on peut réveiller — ou <i>Wakeable End Device</i> . 34
WPAN	Réseau personnel sans-fil — ou <i>Wireless Personal Area Network</i> . 26, 36
WSN	Réseaux de capteurs sans fils — ou <i>Wireless Sensor Networks</i> . 17, 18, 26, 38, 47, 48, 50, 53, 107, 149, 150

Liste des symboles

N_c	Nombres de canaux fréquentiels physiques utilisés par un réseau. 62, 64, 101–104, 110, 116, 128, 131–138, 140–145
CH	Décalage de canal physique – ou <i>CHannel</i> . 31, 112, 113
CO	Décalage de canal logique – ou <i>Channel Offset</i> . 31, 97, 112, 129, 130
DC_J	Ratio d’activité radio d’un nœud qui tente de rejoindre un réseau. 102, 103, 105, 107, 108
d_{data}	Durée d’activité pour la transmission/réception de data. 62, 138, 140
Δ	Valeur maximale de dérive d’une horloge d’un nœud. 6, 53, 54, 87, 88
δ	Dérive relative d’un nœud par rapport à un autre. 6, 53, 54, 87, 88
d_{join}	Durée en mode <i>Rx</i> lors de la phase d’attache au réseau. 61–63, 97–99, 102, 104, 107, 108, 110, 116, 118
d_{on}	Durée d’activité de la radio (<i>Rx</i> ou <i>Tx</i>). 62, 76–78, 98, 102–104, 111, 118
d_S	Durée dans l’état <i>S</i> . 27, 74, 76, 77, 138
$d_{silence}$	Durée entre deux communications. 53–55
d_{syn}	Durée d’activité pour la transmission/réception d’information de synchronisation. 62–64, 119
d_{tout}	Durée en mode <i>Rx</i> pour potentiellement recevoir une réponse à la trame de sondage envoyée précédemment. 141–143
E_{leak}	Énergie de fuite (<i>leak</i>) d’une pile, par an. 75
ϵ	Faible durée. 133
E_{Thread}	Énergie consommée par un nœud asynchrone ultra-contraint Thread pour transmettre un paquet à un nœud TSCH. 132, 138, 140, 144, 145
E_{TSCH}	Énergie consommée par un nœud TSCH pour permettre la communication avec un nœud asynchrone ultra-contraint Thread. 132, 141, 142, 144, 145
E_0	Énergie initiale disponible pour un périphérique. 6, 75, 76, 79, 80, 82, 86, 88
f_{Async}	Canal de communication d’un réseau mono-canal asynchrone. 65
$f_{Interop}$	Canal de communication d’une transmission d’interopérabilité. 65, 66
γ	Partie flottante du décalage entre le premier envoi de trame de synchronisation de la séquence d’envoi du réseau et le réveil d’un nœud. 104
GT	Intervalle de garde: temps passé en <i>Rx</i> pour prendre en compte la dérive d’horloge — ou <i>Guard Time</i> . 32, 53–55, 87, 88, 90
HS	Séquence de sauts – ou <i>Hopping Sequence</i> . 27, 28, 31, 112
κ	Entier naturel déterminant la taille de la <i>slotframe</i> . 128
L	Latence expérimentée par un trafic. 123
ℓ	Longueur du côté de l’aire de jeux. 135, 136
L_{AtoS}	Latence rencontrée un trafic en provenance d’un nœud ultra-contraint Thread et à destination d’un des nœuds du réseau TSCH. 123, 124, 132, 133, 137, 140, 141, 143, 146
L_t	Durée de vie d’un nœud fonctionnant avec une énergie initiale E_0 . 6, 76, 77, 79
L_{max}	Latence maximale expérimentée par un trafic. 123

L_{obj}	Contrainte de latence donnée par une application donnée. 51, 52, 55, 56, 128
L_{soll}	Délai avant la réception d'une trame de sollicitation. 132–135, 137, 140
N_{EB}	Nombre d'EB (Balise améliorée — ou <i>Enhanced Beacon</i>) envoyés par <i>slotframe</i> . 96, 97
N_{neigh}	Nombre de voisins à portée radio. 96, 116, 133–137, 144, 145
$N_{paquets}$	Nombres de paquets nécessaires à la transmission de la charge applicative s_a (en octets). 76, 77
N_{syn}	Nombre de période d'activité dédié à la synchronisation pure. 63
N_T	Nombre de nœuds constituant le réseau synchrone. 127, 135, 136, 143–145
N_{tr}	Nombre moyen de transmissions nécessaires pour la transmission effective d'un paquet, pour un PER donné. 76, 77
N_z	Nombre de zone d'envoi des EBs dans une <i>slotframe</i> . 113, 128, 131, 134, 137, 142
φ	n -uplet représentant une distribution des zones d'EB exploitées par le voisinage en portée radio. 134
R	Portée du signal radio, en m. 135, 136
r	Nombre de répétition d'une <i>slotframe</i> pour former une <i>mutlislotframe</i> . 97, 112
r_a	Débit de génération des données applicatives. 6, 75, 76, 80–83, 85, 86, 88
s_a	Taille des données applicatives. 6, 51, 75–77, 79, 81–86, 89, 115
SDC	Ratio de créneaux temporels actifs sur le nombre total de créneaux temporels du cycle. 105–107
seq_{length}	Longueur de la séquence de sauts utilisées dans le TSCH. 31, 113, 116
SF_{length}	Durée de la <i>slotframe</i> , en ms. 131–134, 137, 138, 140, 141, 143
SF_{size}	Taille de la <i>slotframe</i> , en nombre de slots. 112, 128, 131, 137, 140, 142
t_0	Instant de réveil d'un nœud, pour commencer ses procédures d'attache ou d'envoi de données au réseau synchrone à proximité. 32, 61, 62, 102, 133, 134
T_a	Intervalle entre deux générations de données applicatives. 6, 7, 51, 52, 54–56, 61–64, 71, 75–83, 85–90, 100, 119, 131–135, 137, 138, 140, 141, 143–146
T_{adv}	Période d'envoi de la trame de synchronisation (balise). 32, 61–64, 99, 100, 111, 112, 126
T_{ci}	Intervalle de temps entre deux réveil d'un nœud (ou <i>Check Interval</i>). 76, 87
$t_{interop}$	Instant de la transmission d'interopérabilité. 65, 66
T_{poll}	Intervalle entre deux trames de <i>poll</i> pour solliciter un émetteur potentiel. 126
T_{syn}	Période minimale d'échange de messages requise pour maintenir une connexion active. 54, 55, 62–64, 66, 76–78, 119

Avant-propos

Remerciements

Bien que la thèse soit souvent considérée comme un travail personnel et solitaire, ce parcours vers la maturité scientifique, de par sa durée et son investissement personnel, constitue une tranche de vie bien plus qu'une épreuve universitaire. De nombreuses personnes se retrouvent ainsi, de manière volontaire ou non, entre le doctorant et son doctorat. Ce sont ces personnes que je souhaite mettre en lumière à travers ces remerciements.

La première personne que je souhaite remercier est mon directeur de thèse, Andrzej Duda, qui a choisi de me faire confiance pendant ces trois années en me laissant gérer mon projet comme je l'entendais tout en étant disponible chaque fois où j'en ai eu besoin. Il a de plus su quand et comment me pousser à sortir de ma zone de confort pour produire un travail d'une plus grande qualité scientifique.

Je souhaite également remercier mes deux autres encadrants, qui ont pour moi joué un rôle tout aussi important. Ils ont en effet su m'apporter chacun une vision différente de la recherche : alors qu'Andrzej guidait mes premiers pas dans le monde de la recherche et m'apportait son expertise pour la rédaction des publications, Roberto Guizzetti m'a permis de découvrir l'aspect plus industriel, appliqué de la recherche. Enfin avec Mickael Maman, j'ai pu avoir une vision intermédiaire, plus nuancée de la recherche, la sensation de travailler avec un collègue, et d'avancer pour un projet commun (en plus d'avoir des discussions d'œnologie!).

Bien que je sois parfois têtue, j'ai pris beaucoup de plaisir à m'épanouir intellectuellement dans ma vie d'apprentie chercheuse aux côtés de ces trois encadrants.

Au delà de mes encadrants, j'ai pu compter sur la présence de nombreux collègues, de par la nature multipartite de ce contrat de thèse. Certains de ces collègues m'ont été d'une aide très précieuse, tant sur le plan de la recherche que sur le plan personnel.

Je souhaite ainsi saluer Michel Favre pour les discussions intéressantes sur les notions de temps et d'espace, mais aussi et surtout pour l'aide à la recherche de fondamentaux. Je salue également David, Jimmy, Ludovic et Benoit, du CEA, pour les discussions plus personnelles que nous avons pu avoir lors des périodes houleuses de la vie de doctorant.

L'ensemble de l'équipe Drakkar, et plus particulièrement Étienne, Franck, et Henry-Joseph, m'a par ailleurs permis de grandir en m'appuyant sur leurs compétences, partagées pendant les pauses café ; ce qui m'a permis d'avancer quand je tournais en rond.

L'aventure de ma thèse a également impactée de nombreux aspects de ma vie privée, et je voudrais continuer ces remerciements en rendant grâce aux personnes qui m'ont accompagné dans ce passage vers le monde des docteurs.

Parmi ces proches, je veux tout d'abord remercier l'ensemble de mes amis, Claire, Noémie, Jaques, Lina, David, Liviu, Gabriele, Iacob, Mickael, Christian, et Romane, avec une mention spéciale pour Grégoire, qui est venu *dormir* à la maison le soir du rendu du manuscrit, et m'a donc aidé à rédiger la conclusion du présent manuscrit jusque tôt dans la matinée.

Je souhaite de plus souligner le soutien particulier apporté par mes amis doctorants, Pierre et Timothy, également dans l'équipe Drakkar. Je les remercie pour l'amitié solide et de *qua-li-té* que nous avons construite ensemble, entre débats sur la société, tranches de fous-rires et coups de main pour terminer les publis *on time*.

Mes frères, Antoine et Louis, me permettent de me tenir à jour de la culture pop et savent toujours me faire rire. Je suis très heureuse de partager ces rigolades avec eux, même lors de la rédaction de cette *thèse qui triple pèse*, et pour cela je les remercie.

J'aimerais également remercier mes grands-parents, avec une attention particulière pour mon grand-père qui nous a quittés avant de me voir diplômée, pour leur ouverture d'esprit et l'intelligence dont ils font et ont fait preuves au quotidien, que l'on discute de ma thèse ou bien d'état d'urgence.

De plus, j'exprime ici toute ma gratitude envers mes parents, Laurence et Olivier Morin, pour leur soutien, à la fois dans la thèse et tout au long de la vie. Il est toujours plus rassurant de se lancer dans de grands projets tels le doctorat avec une famille comme celle que vous avez construit.

J'ajoute de plus un petit clin d'œil à papa, à qui j'avais reproché il y a quelques années, de s'enfermer dans sa thèse et de ne pas être disponible ... avant d'en faire de même à mon tour, et ainsi regretter ces reproches.

Enfin, mais non des moindres, tous mes remerciements à Colin Chargy, mon mari depuis peu et pour mon plus grand bonheur, qui a su m'apporter à la fois un soutien technique d'ingénieur informatique et un soutien littéraire pour la relecture du manuscrit dans son intégralité. Mais surtout, il m'a offert un soutien quotidien indéfectible et une présence sans faille. Cette aventure, il l'a portée avec moi depuis le choix de faire un doctorat, jusqu'à la réalisation du pot de thèse. Je ne trouve pas les mots pour exprimer des remerciements proportionnés au soutien que j'ai reçu. Merci.

Sans toutes ces personnes, je n'aurai probablement pas réussi à terminer cette thèse avec autant de satisfaction. Merci à vous, mais également à tous celles et ceux que je n'ai pas nommé-e-s ici, qui m'ont aidée et soutenue, de près où de loin, pendant ces trois années de doctorat.

Résumé en français

L'essor rencontré par les différentes technologies dédiées aux réseaux de capteurs (WSN), a conduit au développement de plateformes capables d'opérer dans deux technologies différentes, adaptatives aux contextes de transmission. De telles plateformes ouvrent la porte à la conception de réseaux multitechnologies, que nous proposons d'exploiter dans le but de réduire la consommation d'énergie globale. Dans le but d'exploiter ces réseaux multitechnologies, nous décrivons les principales technologies de l'Internet des Objets (IoT), en les comparant sur un pied d'égalité grâce à l'analyseur que nous avons développé, puis les classifions en fonction des mécanismes MAC qu'elles exploitent. Nous analysons ensuite le lien entre le contexte applicatif (latence et fréquence de la génération de données) et le mécanisme MAC consommant le moins d'énergie pour ce contexte applicatif.

Nous remarquons alors que les technologies exploitant un mécanisme MAC synchrone sont les plus adaptées aux trafics applicatifs périodiques dont les intervalles entre les générations de données sont courts. En effet, pour ces trafics, la dérive d'horloge entraîne un coût de maintien de la synchronisation active trop élevé dans le cas de trafics périodiques rares. De plus, nous remarquons que la gestion des trafics applicatifs rares contraints en latence repose, dans les solutions existantes, sur l'utilisation d'une plateforme constamment active en mode de réception. Nous proposons alors d'exploiter les plateformes multitechnologies pour constituer un réseau synchrone dans lequel chaque nœud répartit son activité dans le temps pour globalement économiser de l'énergie pour remplacer le rôle du dispositif constamment disponible utilisé pour acheminer des trafics asynchrones contraints en latence. Nous remarquons que lors de la procédure d'attache au réseau synchrone, la situation du nœud qui tente de rejoindre un réseau synchrone dans le but d'y acheminer des données est similaire à la situation d'un nœud asynchrone qui souhaite acheminer des données au sein d'un réseau synchrone. Ainsi, nous proposons d'exploiter la phase d'attache au réseau pour acheminer des trafics émanant de nœuds asynchrones, contraints en latence, au sein d'un réseau synchrone.

Cependant, les procédures actuellement standardisées d'attache au réseau sont naïves et très coûteuses en énergie, ce qui décourage l'utilisation d'un mode de communication asynchrone, reposant sur une succession d'associations/désassociations du réseau : nous proposons deux approches pour réduire le coût de cette procédure d'attache à un réseau TSCH. La première repose sur l'exploitation de séquences mathématiques dont la propriété est d'étaler les périodes d'activités dans le temps, tout en minimisant l'impact sur la latence de la procédure, pour diminuer le coût énergétique global de la procédure d'attache. La deuxième méthode proposée exploite les trames d'acquittement (ACK) des communications TSCH pour y ajouter des éléments d'informations : la date d'envoi de la prochaine trame de synchronisation sur le même canal physique que celui utilisé pour l'envoi de la trame d'ACK. Grâce au développement d'un simulateur des performances de la phase d'attache à un réseau TSCH, nous montrons que les protocoles d'attaches proposés obtiennent de meilleures performances, soit en termes de latence, soit en termes de consommation d'énergie globale, que les protocoles d'attache classiquement utilisés dans les réseaux de capteurs.

Enfin, nous proposons d'exploiter les mécanismes de la deuxième proposition d'attache au réseau pour l'envoi de trames de sollicitation à destination d'un nœud fonctionnant avec une technologie asynchrone, permettant ainsi d'acheminer un trafic asynchrone au sein d'un réseau synchrone en une latence bornée. Nous montrons la faisabilité et prouvons l'intérêt d'une telle proposition.

Résumé en anglais

The growth of various technologies dedicated to sensor networks (WSN) has led to the development of platforms capable of operating in two different technologies, adaptive to transmission contexts. Such platforms open the door to the design of multi-technology networks, which we propose to exploit to reduce overall energy consumption. In order to exploit these multi-technology networks, we describe the main Internet of Things (IoT) technologies, comparing them on an equal footing thanks to the analyzer we developed, and classify them according to the MAC mechanisms they use. We then analyze the link between the application context (latency and frequency of data generation) and the MAC mechanism that consumes the least energy for this application context.

We note that the technologies operating with a synchronous MAC mechanism are the most suitable for periodic application traffic with short intervals between data generation. For these traffic patterns, clock drift leads to extensive traffic overhead because of the need to actively maintain synchronization for sparse periodic traffic.

Moreover, we notice that, in the existing solutions, the management of sparse application traffic management is based on the use of an always-on platform (in reception mode). We thus propose to exploit the multi-technology platforms to build a synchronous network in which each node distributes its activity over time to globally save energy by replacing the role of the always-on platform, while guaranteeing the delivery of the latency-constrained asynchronous traffic.

We notice that during the synchronous network joining phase, the situation of the node attempting to join a synchronous network is similar to the situation of an asynchronous node wanting to deliver data through a synchronous network. Thus, we propose to exploit the synchronous network joining phase to route latency-constrained traffic originating from asynchronous nodes through the synchronous network.

However, the currently standardised network attachment procedures are naïve and energy-greedy, which discourages the use of an asynchronous communication mode, based on a succession of network associations/dissociations : we thus propose two approaches to reduce the cost of the TSCH network attachment procedure.

The first is based on the use of mathematical sequences which distribute the periods of activity over time, while minimizing the impact on the latency of the procedure, in order to reduce the overall energy cost of the attachment procedure. The second proposed method exploits the acknowledgement frames (ACK) of TSCH data communications to embed the date of the next synchronization frame transmission on the same physical channel as the ACK frame. Thanks to the development of a simulator of the TSCH joining phase, we show that the proposed protocols achieve better performance, either in terms of joining latency, or in terms of overall energy consumption, than the standard joining protocols used in WSN.

Finally, we propose to exploit the mechanisms of the second proposal for sending request frames to a node operating with an asynchronous technology, thus enabling asynchronous traffic to be routed through a synchronous network in bounded latency. We demonstrate the value and feasibility of such a proposal.

Introduction

Contexte

Par la diversité de ces domaines d'applications, l'IoT (Internet des Objets — ou *Internet of Things*) fait désormais partie de nos vies. S'étendant de la surveillance des feux de forêt aux applications domotiques, l'IoT a modifié les usages, désormais tournés vers l'*intelligence* des objets : on parle en effet de villes intelligentes (*Smart Cities*), de réseaux intelligents (*Smart Grids*), ou encore de maisons intelligentes (*Smart Homes*). Cette "*intelligence*" correspond techniquement à la capacité qu'ont deux objets, la plupart du temps des capteurs/actionneurs, à s'échanger de l'information, c'est-à-dire à *communiquer*, à travers l'infrastructure réseau que l'on nomme l'Internet, dans le but de réagir à un environnement donné.

L'apparition de l'IoT, et plus particulièrement des WSN (Réseaux de capteurs sans fils — ou *Wireless Sensor Networks*) a ajouté de nouvelles contraintes de conception des solutions réseaux s'ajoutant ainsi à la liste des contraintes de robustesse, de sécurité ou en encore de partage des ressources déjà présentes dans les réseaux dits *classiques* [4].

La principale contrainte apportée par les WSN est liée à l'énergie. En effet, dans ces réseaux vastes et répandus, l'énergie devient une ressource critique : alors que le rechargement quotidien d'un *smartphone* est tolérable, une telle fréquence de remplacement des piles, ou de chargement des batteries, n'est pas acceptable pour des capteurs de l'IoT.

La contrainte d'énergie vient impacter les solutions réseau de différentes manières : en limitant la puissance de calcul, la disponibilité des nœuds, et la portée des communications radio.

En effet, pour diminuer la consommation d'énergie, on utilise des microcontrôleurs dotés de faibles ressources mémoires qui fonctionnent à de plus faibles fréquences et avec un ensemble réduit d'instructions, limitant ainsi la puissance de calcul. Ensuite, on comprend que, par souci d'économie d'énergie, les nœuds ne sont pas actifs continuellement et répartissent leurs activités dans le temps. Enfin, au niveau de la communication radio en elle-même : pour économiser de l'énergie, les nœuds utilisent des puissances d'émissions relativement faibles, ainsi que des modulations plus simples à mettre en œuvre, qui viennent impacter les débits disponibles pour une robustesse donnée.

Au-delà des contraintes énergétiques, les WSN imposent également de nouvelles contraintes en termes de mobilité, de versatilité des liens et de coexistence.

Afin de concevoir des solutions réseau capables de gérer ces différentes contraintes, de nombreuses solutions technologiques ont été proposées. En raison de la multitude et de la diversité des technologies disponibles, le marché de l'IoT s'oriente vers la commercialisation d'objets multitechnologies. De tels dispositifs appellent à la conception de solutions réseau exploitant de multiples technologies, dans le but d'optimiser les performances globales en sélectionnant la technologie la plus adaptée à un contexte applicatif donné (quantité et période des données à acheminer, contraintes de latence, distribution dans le temps de la génération des données).

Motivations

Cette thèse s'inscrit dans la continuité du projet GREENNET [5] mené par ST Microélectronique. Le but de ce projet était de créer une pile protocolaire dédiée aux plateformes à récupération d'énergie et utilisant une couche MAC (Contrôle d'accès au support — ou *Medium Access Control*) 802.15.4. À la suite du projet GREENNET, l'essor rencontré par les différentes technologies dédiées aux WSN, a conduit ST Microélectronique à développer de nouvelles plateformes multitechnologies. Dans le but d'exploiter au mieux ces plateformes émergentes, il est nécessaire d'identifier les paramètres influençant le choix d'une technologie ou d'une autre en fonction du contexte applicatif.

Nous proposons pour cela d'analyser en détail le fonctionnement de la couche MAC des différentes technologies IoT, en décrivant les modes matériels d'activités utilisés au cours du temps par des nœuds

utilisant les différentes technologies étudiées. Nous remarquons alors que les solutions WSN reposent majoritairement sur deux modes de communication MAC : les communications de nature synchrones qui répartissent l'activité des nœuds dans le temps en utilisant des mécanismes de rendez-vous, et les communications asynchrones. Les communications asynchrones exploitent différents mécanismes MAC de répartition d'activité dans le temps : les mécanismes d'échantillonnage de préambules, ceux de communications initiées par le récepteur à l'aide de trames de sollicitation, et le mécanisme à la demande de l'émetteur, qui suppose un récepteur constamment disponible.

Nous analysons que, si le mécanisme à la demande de l'émetteur consomme globalement beaucoup d'énergie à cause de l'utilisation de plateforme constamment disponible, l'avantage principal de ce mécanisme est qu'il permet d'acheminer un trafic de nature asynchrone contraint en latence, initié par un nœud asynchrone contraint en énergie.

Ainsi, nous proposons d'exploiter les plateformes multitechnologies dans le but d'acheminer un trafic de nature asynchrone contraint en latence, initié par un nœud asynchrone contraint en énergie, tout en diminuant la consommation d'énergie globale.

Nous proposons ainsi de réaliser une optimisation conjointe au niveau MAC de deux modes de communication synchrone et asynchrone contraint en latence, en limitant la consommation d'énergie globale.

Pour ce faire, nous proposons deux pistes de recherches : la première consiste à remarquer que l'hybridation des deux modes de communication synchrone et asynchrone existe déjà au sein des réseaux synchrones, lors de la phase d'attache au réseau. Nous proposons ainsi d'exploiter cette phase d'attache pour mettre en place un mode de communication asynchrone, qui consiste en une succession d'associations temporaires et des désassociations du réseau synchrone. Pour qu'une telle procédure soit intéressante, il faut cependant la comparer au coût de la phase d'attachement au réseau. Dans les technologies existantes, le coût de la phase d'attache est trop élevé pour que le processus d'associations/désassociations successives soit intéressant. Ainsi, nous proposons deux solutions peu coûteuses en énergie pour réaliser l'attachement au réseau.

La seconde piste repose sur l'exploitation des plateformes multitechnologies, capables de fonctionner simultanément à l'aide de la technologie synchrone TSCH (Saut de fréquence à créneaux temporels — ou *Time-slotted Channel Hopping*) et de la technologie asynchrone Thread. Nous présentons alors un mécanisme d'ordonnancement des activités de ces deux technologies au sein d'un même matériel pour garantir qu'une plateforme capable d'exploiter les deux technologies puisse acheminer un trafic asynchrone Thread contraint en latence. Pour ce faire, nous optimisons également le comportement de la plateforme asynchrone Thread contrainte en énergie pour garantir le bon fonctionnement global du réseau TSCH/Thread.

Lors de l'analyse des consommations globales des différentes technologies, nous remarquons par ailleurs que le maintien d'une synchronisation globale à travers le réseau représente un coût énergétique non négligeable lorsque les périodes sans envoyer de données sont longues et que l'on prend en compte les imprécisions des horloges. Ainsi, il serait également potentiellement intéressant, du point de vue énergétique, de se désynchroniser du réseau pour ne le rejoindre qu'au besoin, exploitant ainsi un mode de communication asynchrone, car désynchronisé du réseau.

La coexistence des deux modes de communication proposées ci-dessus, synchrone et asynchrone contraint en latence, ainsi que la possibilité de changer de l'un à l'autre en fonction du contexte applicatif permettrait de réduire la consommation énergétique. En effet, une telle interopérabilité permettrait à la fois en remplaçant le rôle du dispositif constamment disponible par l'exploitation de ressources judicieusement sélectionnées du réseau synchrone, mais également d'exploiter un mode de communication MAC asynchrone lorsque l'énergie dépensée dans le maintien d'une synchronisation MAC serait trop coûteuse du fait de la dérive des horloges imparfaites.

Ainsi, nous proposons, à travers ce manuscrit de présenter une optimisation conjointe de deux modes de communication MAC, synchrone et asynchrone, au sein d'un même réseau pour diminuer la consommation d'énergie globale liée à l'acheminement de trafics rares, de types asynchrones contraints en latence.

Contributions

Le travail réalisé dans le cadre de cette thèse s'organise en quatre axes de contributions selon trois axes de recherches : une contribution compare les standards de l'IoT, une double contribution se focalise sur la procédure d'attache au réseau et la dernière à la mise en œuvre de l'interopérabilité multitechnologie TSCH/Thread. Ainsi, nos principales contributions sont les suivantes :

- Nous proposons une méthode d'analyse des performances énergétiques des différentes solutions et technologies de l'IoT, ainsi que l'identification des paramètres qui impactent la consommation des différentes technologies analysées.

Notre étude montre que, pour une horloge matérielle supposée idéale, le BLE (Bluetooth Low Energy) [6] obtient les plus faibles consommations énergétiques pour l'ensemble des contextes applicatifs dans la limite de sa capacité. La technologie très longue portée LoRa [7] obtient des performances remarquables dans les contextes de très bas débits, et les solutions basées sur la spécification IEEE (Institut des ingénieurs électriciens et électroniciens — ou *Institute of Electrical and Electronics Engineers*) 802.11 [8] ne sont performantes que dans les contextes de hauts débits.

Pour des trafics d'intensités faibles et moyennes, l'ordre des technologies, en termes de consommation énergétique, est le suivant : BLE, 802.15.4 [9], TSCH [9] et les solutions 802.11 ; les technologies très longues portées ne supportant pas de tels débits.

Nous montrons également qu'au-delà du classement des technologies les unes par rapport aux autres, le maintien d'une synchronisation pour une dérive d'horloge fixée est un paramètre discriminant entre deux technologies, suivant que l'on se situe dans un contexte de données applicatives fréquentes ou non : en effet, les conclusions énoncées précédemment reposent sur l'hypothèse d'horloges matérielles parfaites, et les performances des technologies synchrones sont largement impactées lorsque la fréquence du trafic devient très faible et que l'on dimensionne des intervalles de garde pour lutter contre la dérive d'horloge.

De cette étude, nous concluons que dans des contextes applicatifs de trafics très peu fréquents, il peut être intéressant pour un nœud, du point de vue énergétique, de se désassocier du réseau pour le rejoindre à nouveau si cela est nécessaire pour l'envoi de ses données, plutôt que de maintenir sa présence dans le réseau de manière active.

Cependant, la phase d'attachement telle que standardisée actuellement représente un coût énergétique trop élevé pour que le processus d'associations/ désassociations successives soit intéressant, du point de la consommation globale d'énergie. Ainsi, nous proposerons deux solutions pour diminuer la d'énergie lors de la phase d'attache au réseau :

- La première méthode d'attachement proposée réalise un compromis entre la latence d'attachement et la consommation énergétique durant cette phase. Pour ce faire, elle exploite les propriétés de séquences mathématiques particulières dans le but de garantir une faible consommation énergétique tout en maintenant des délais d'attache au réseau acceptables.
- La seconde proposition exploite les transmissions de trafic de données, en intégrant dans les paquets d'acquittements des informations sur la date de transmission de la prochaine trame de synchronisation. Afin d'optimiser cette solution, nous proposons d'y adjoindre un algorithme intelligent d'ordonnancement des trames de synchronisation, qui s'adaptera aux trafics de données existants pour garantir des performances intéressantes en termes de latence et de consommation.
- Enfin, nous proposons d'exploiter les principes de cet ordonnanceur intelligent pour insérer des slots dédiés à des communications de nature asynchrones de type Google Thread [1] dans un ordonnancement de réseau TSCH [9], judicieusement placé pour garantir une latence maximale subie par le trafic émanant du nœud asynchrone contraint. Nous modifions également le comportement du nœud asynchrone pour lui permettre de transmettre son trafic dans un réseau synchrone à proximité, sans perturber le fonctionnement de ce dernier. De cette manière, nous proposons une interopérabilité multitechnologie, Thread et TSCH pour répondre au besoin de transmission d'un trafic asynchrone contraint en latence, tout en économisant de l'énergie en remplaçant le fonctionnement des nœuds asynchrones constamment disponibles par une activité soigneusement planifiée en temps et en fréquence des nœuds du réseau TSCH synchrone.

Structure du document

Afin de présenter ces contributions, le manuscrit est organisé en deux parties : la première partie présentera l'interopérabilité dans le contexte des technologies de l'IoT, en termes de solutions existantes ainsi que de notions nécessaires à la compréhension des travaux de contribution. Ces derniers seront présentés dans une seconde partie. Au-delà de cette organisation classique en partie; nous avons regroupé les idées présentées en chapitres, de manière à distinguer les différents aspects de notre étude.

Partie I, Chapitre 1 – Technologies de l'IoT Dans un premier temps, nous présenterons les multiples solutions technologiques pour l'IoT que nous étudierons, dans le but de donner aux lecteurs les connaissances nécessaires pour comprendre les enjeux et les intérêts de rendre ces technologies interopérables. Ainsi, à travers ce premier chapitre, nous présenterons et analyserons en détail le fonctionnement MAC des piles protocolaires standardisées et les technologies existantes pour l'IoT que nous avons sélectionnées pour participer à une potentielle interopérabilité. Nous justifierons également le choix restreint de ces technologies.

Partie I, Chapitre 2 – Discussion Dans un second chapitre, nous justifierons notre focalisation sur la gestion d'un trafic asynchrone contraint en latence à faible coût énergétique en analysant les différents mécanismes MAC existants. Nous montrerons ainsi que les mécanismes existants permettant d'acheminer un trafic asynchrone contraint en latence reposent sur l'exploitation d'un périphérique constamment disponible, coûteux en énergie. Ainsi, nous proposerons d'exploiter les plateformes contraintes en énergie capable de gérer à la fois un mode de communication synchrone et un mode de communication asynchrone contraint en latence pour diminuer la consommation globale liée à la gestion de trafic asynchrone contraint en latence. Nous présenterons ensuite les travaux réalisés en termes d'interopérabilité pour l'IoT et proposerons deux pistes d'exploration pour répondre à la problématique soulevée : l'exploitation de la phase d'attache au réseau synchrone, ainsi que la réalisation d'un réseau constitué de nœuds de technologies différentes, l'une synchrone et l'autre asynchrone.

Partie II, Chapitre 3 — Comparaison des standards pour l'IoT Au sein du troisième chapitre, nous présenterons notre première contribution [10], la réalisation d'un analyseur disponible sous forme de logiciel libre [11], pour comparer les différentes solutions existantes présentées dans le premier chapitre, dans le but d'identifier les paramètres influents sur la consommation d'énergie de chacune des solutions existantes. Les résultats de cette analyse confirmeront qu'un des facteurs déterminants de la durée de vie d'un capteur est le coût de maintien de la synchronisation au cœur du réseau. Ainsi, dans certains contextes applicatifs, il pourrait être énergétiquement intéressant de permettre à un nœud de se désynchroniser du réseau, puis de rejoindre par la suite, pour lui donner un mode de communication proche de celui des technologies asynchrones.

Cependant, si une telle opération est d'ores et déjà réalisable, le coût de la procédure d'attachement au réseau ne peut être négligé. Les solutions techniques permettant actuellement de rejoindre un réseau TSCH, seule solution existante multicanal et multisaut qui économise de l'énergie à chacun des nœuds du réseau, sont actuellement trop consommatrice en énergie pour qu'il y ait réellement un intérêt à exploiter un mode de communication non synchronisé.

Partie II, Chapitre 4 — Réduction du coût d'attache au réseau C'est pourquoi nous proposerons, au sein d'un quatrième chapitre, deux solutions pour diminuer le coût de la phase d'attache au réseau. L'une repose sur l'utilisation de propriétés mathématiques de séquences spécifiques pour l'envoi ou l'écoute de trames de synchronisation [12]. L'autre exploite les communications de données au sein du réseau pour transmettre des informations de synchronisation, et optimiser l'envoi des trames de synchronisation grâce à un ordonnanceur intelligent. La réalisation d'un simulateur qui évalue les performances en termes de coût énergétique global et de temps passé pour réaliser la phase de découverte d'un réseau TSCH permettra de valider les performances de ces deux contributions : par rapport aux solutions existantes, les deux protocoles d'attache proposés obtiennent de meilleures performances, soit en termes de délai d'attache, soit en termes de consommation énergétique lors de la phase d'attache.

Partie II, Chapitre 5 — Application à l’interopérabilité Google Thread/TSCH Au cours du cinquième chapitre, qui constituera l’ultime contribution de ce rapport, nous proposerons de réaliser une interopérabilité entre une couche MAC de communication synchrone (TSCH) avec une couche MAC asynchrone (Thread), dans le but de garantir, à l’aide d’un cœur de réseau disponible par intermittence et donc peu consommant en énergie, l’acheminement d’un trafic asynchrone contraint en latence. Nous proposerons ainsi un nouveau mode de communication, hybride, s’appuyant sur l’allocation de slots dédiés à la communication asynchrone au sein du réseau synchrone. Nous exploiterons l’ordonnanceur intelligent du chapitre 4 pour l’acheminement du trafic asynchrone dans une latence bornée. Nous modifierons également le comportement du nœud asynchrone pour qu’il intègre ses communications asynchrones contraintes en latence dans le réseau synchrone, sans pour autant perturber ce dernier. Nous évaluerons théoriquement les performances de la solution proposée, et montrerons qu’une telle interopérabilité permet de réduire la consommation globale d’énergie si l’on tolère une faible latence sur le trafic asynchrone.

Enfin, nous clôturerons ce manuscrit par une conclusion générale, qui fera le bilan des principales conclusions de différents travaux menés dans ce rapport, avant de prendre du recul sur ces travaux dans le but d’identifier les directions principales qu’il nous semble important de donner aux recherches futures dans le domaine de l’interopérabilité, en particulier dans le domaine des réseaux de capteurs à faible consommation d’énergie.

Première partie

L'interopérabilité pour l'IoT

Technologies de l'IoT

Sommaire

Remerciements	13
Résumé en français	15
Résumé en anglais	16
.1 Contexte	17
.2 Motivations	17
.3 Contributions	19
.4 Structure du document	20

Introduction

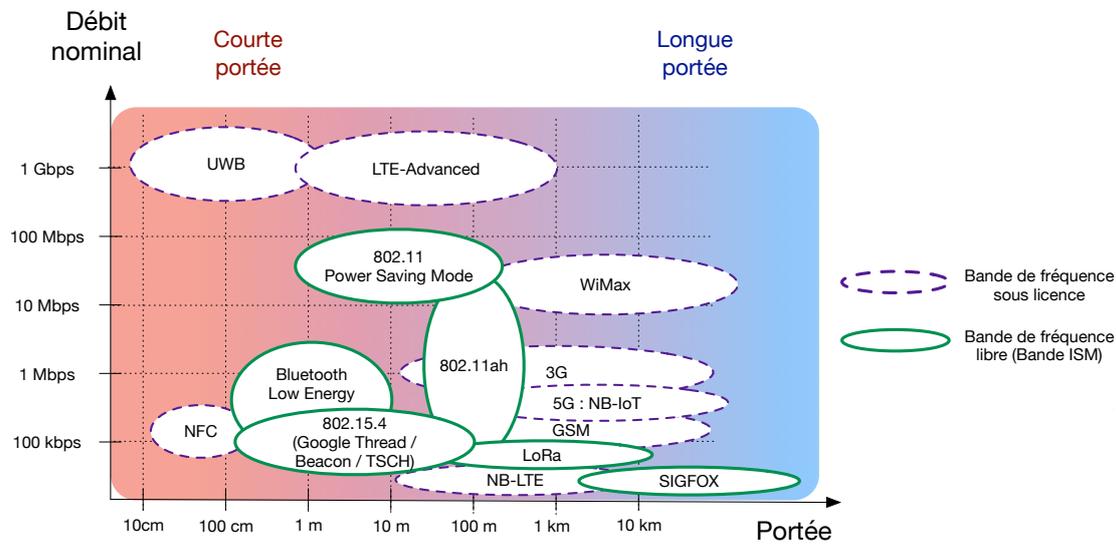


FIGURE 1.1 – Vue d'ensemble des technologies sans-fil.

La figure 1.1 nous donne une représentation classique de la "carte" des technologies sans-fil existantes. Cette représentation positionne chacune des technologies dans un plan portée/débit sur l'air, et permet rapidement d'identifier la zone d'application de chacune des solutions techniques.

Sur la figure, on peut remarquer qu'une partie des technologies utilise des bandes de fréquences sous licence, c'est-à-dire que l'autorité d'un état réserve des bandes de fréquences pour ces technologies, comme c'est le cas avec l'ARCEP¹, en France : c'est cette autorité qui attribue les bandes de fréquences aux militaires, ou encore aux opérateurs téléphoniques. Alors que l'obtention d'une licence sur une bande donnée se monnaie auprès de l'ARCEP, le spectre non licencié offre aux innovateurs un outil précieux : l'accès à certaines bandes de fréquence à faible coût. Dans le contexte des WSN, le déploiement d'infrastructure de capteurs pour tous et tous les usages, nécessite des solutions à bas coût. Dans ce chapitre, nous présentons les principales technologies sans-fil utilisant les bandes de fréquences non licenciées, ouvertes à tous les utilisateurs à faible coût.

Nous organisons ce chapitre en trois sections, en présentant les technologies par ordre de portée croissante, comme montré sur la figure 1.1 :

- La première section présente les technologies PHY (Couche physique du modèle OSI) à courte portée, dont le cœur de cible sont les réseaux de type WPAN (Réseau personnel sans-fil — ou *Wireless Personal Area Network*), le 802.15.4 et ses différents modes, ainsi que le BLE (Bluetooth Low Energy).
- La seconde présente les technologies dédiées aux réseaux LAN (Réseau local — ou *Local Area Network*), développées dans le but d'une faible consommation énergétique : 802.11b PSM (Mode d'économie d'énergie — ou *Power Saving Mode*) et 802.11ah.
- Enfin, la troisième section décrit le fonctionnement des nouvelles technologies pour les réseaux à longue portée, les LPWAN (Réseau personnel sans-fil à faible consommation énergétique — ou *Low Power Wide Area Network*) : LoRa et SIGFOX.

De manière à analyser le fonctionnement des différentes couches MAC au cours du temps, nous présentons les différents modes matériels dans lesquels une plateforme va se trouver lorsque le nœud souhaite remonter un maximum de données à destination d'un parent, en liaison point à point.

Nous supposons qu'un nœud de communication peut être modélisé par ses deux composants les plus consommateurs en énergie : le microcontrôleur et la radio. Ainsi, nous définissons quatre modes

1. Autorité de Régulation des Communications Electronique et des Postes

de consommations matériels (*Rx*, *Tx*, *Idle* et *Sleep*) qui correspondent aux combinaisons possibles des modes de chaque composant, comme illustré dans le tableau 1.1.

	Radio Éteinte	Radio en réception (Rx)	Radio en transmission (Tx)
CPU Actif	<i>Idle</i>	<i>Rx</i>	<i>Tx</i>
CPU Éteint	<i>Sleep</i>	\emptyset	\emptyset

TABLEAU 1.1 – États de consommation des composants matériels.

Ainsi, la description du fonctionnement MAC via ces modes de consommation matériels nous permettra d'utiliser par la suite un modèle de consommation énergétique dérivé des travaux de Vilajosana *et al.* [13] pour exprimer la consommation d'énergie au cours de l'intervalle t de la manière suivante :

$$E(t) = \sum_S P_S \times d_S, \quad S \in \{Tx, Rx, Idle, Sleep\} \tag{1.1}$$

où P_S est la puissance consommée dans l'état S et d_S la durée dans l'état S au cours de l'intervalle t .

1.1 Solutions courte portée : les WPAN

Les solutions à courte portée dans les bandes de fréquences libres s'articulent majoritairement autour de deux standards :

- Le standard IEEE 802.15.4
- La spécification BLE

Alors que le 802.15.4 est normalisé par l'IEEE², le BLE est quant à lui standardisé par la Bluetooth SIG³.

1.1.1 Le standard IEEE 802.15.4

1.1.1.1 Couche PHY

Les canaux fréquentiels utilisés par le standard 802.15.4 sont :

- 16 canaux physiques dans la bande des 2.4 GHz (majoritairement utilisée)
- 10 canaux dans la bande des 900 MHz
- 1 canal dans la bande des 868 MHz

Le 802.15.4 travaille donc majoritairement dans la même bande ISM que celle du 802.11 et du BLE. Afin de réduire le risque d'interférence, les largeurs de bandes sont beaucoup plus faibles dans le cas du 802.15.4 que dans le cas du 802.11 : alors que les canaux 802.11 mesurent 22 MHz, ceux du 802.15.4 n'utilisent qu'une largeur de 2 MHz, et l'on considère des canaux de 5 MHz pour limiter encore les interférences.

Ainsi, on peut utiliser un canal 802.15.4 entre deux canaux actifs 802.11. C'est la couche PHY qui s'occupe de sélectionner le meilleur canal de communication en fonction de l'environnement dans le cas des technologies monocanal. Dans les cas des technologies multicanales, la couche PHY se charge de sélectionner un sous-ensemble des fréquences PHY pour former la séquence de sauts de fréquence (*HS*) utilisée par le réseau.

Le rôle de la couche PHY est d'assurer la transmission au niveau physique, c'est-à-dire qu'elle s'occupe de la procédure de détection d'énergie afin de détecter la présence d'une communication de la technologie 802.15.4, dans le voisinage, ou encore de s'assurer que le canal est libre avant transmission par le mécanisme de CSMA/CA (Écoute d'un support à accès multiple et à évitement de collision— ou *Carrier Sense*

2. Institut des ingénieurs électriciens et électroniciens — ou *Institute of Electrical and Electronics Engineers*

3. Organisation Bluetooth— ou *Special Interest Group*

Multiple Access-Collision Avoidance)⁴.

Elle s'occupe également de gérer le changement de canal de communication si besoin dans le cas des technologies monocanales, ou de retirer une fréquence de la séquence *HS* si les conditions sur ce canal sont dégradées. Le principe général de fonctionnement d'envoi des données de la couche PHY est le suivant : à partir des données binaires que l'on souhaite envoyer au niveau MAC, on réalise d'abord une correspondance entre une séquence de bits et des symboles. Ces symboles sont alors multipliés par un code d'étalement, qui va élargir le spectre fréquentiel du message, afin d'être plus robuste aux brouillages rencontrés lors de la transmission : c'est de l'étalement de spectre DSSS (Etalement de spectre à séquence directe — ou *Direct-Sequence Spread Spectrum*).

Les codes obtenus sont ensuite modulés afin d'obtenir le signal à transmettre sur la radio. Lorsque l'on travaille en 802.15.4 dans la bande des 2.4GHz, la modulation utilisée est une modulation O-QPSK⁵. Cependant, si l'on travaille dans les fréquences plus basses, on module le signal en *Binary Phase Shift-keying* (BPSK), une modulation plus simple avec seulement deux points dans la constellation.

1.1.1.2 Couche MAC

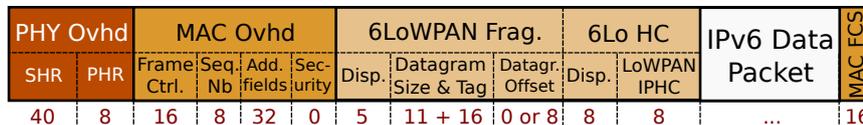


FIGURE 1.2 – Format d'un paquet de données 802.15.4 (taille en bits).

Format de trames

Il existe 6 types de trames MAC dans le standard 802.15.4 :

- les trames balises, ou *beacons*, qui servent à l'organisation en *supertrame* dans le mode correspondant,
- les trames de données,
- les trames d'acquittements,
- les trames à but générique (ou *Multipurpose*),
- et les trames de commandes, pour lesquelles on remplace le champ des données par un identifiant de commande, suivi des données liées à la commande.
- les trames de fragmentation, utilisées lorsqu'une fragmentation est nécessaire à cause de trames étalées en fréquence (en plus de la modulation DSSS classique) pour les modes à très faible consommation d'énergie.

Les différents modes MAC

La couche MAC du standard 802.15.4 s'organise en plusieurs modes de communications. Nous présentons ici uniquement les modes que nous utilisons dans la suite de nos travaux : le mode non-*beacon*, le mode *beacon*, le mode TSCH (Saut de fréquence à créneaux temporels — ou *Time-slotted Channel Hopping*), le mode CSL (Ecoute coordonnée de préambule — ou *Coordinated Sampled Listening*) et le mode RIT (Transmission initiée par le récepteur — ou *Receiver Initiated Transmission*).

4. Le principe du CSMA/CA est le suivant : Avant d'émettre, une station écoute le canal. Si le canal est libre, la station peut passer à la phase suivante de la transmission. Si le canal est occupé, le nœud augmente le compteur du nombre d'essais de transmission de la trame correspondante et attend une période aléatoire (*backoff*) avant de détecter le canal une deuxième fois.

Le *backoff* est choisi aléatoirement et uniformément parmi un intervalle particulier : la fenêtre de contention (*CW*). Ce *CW* sera doublé à chaque fois qu'un canal est détecté occupé (ou lorsqu'une collision est détectée) jusqu'à une taille maximale. De plus, une fois le canal libéré et la transmission effectuée, le *CW* est réduit au minimum.

Ce mécanisme évite la collision : si deux stations souhaitent émettre deux paquets en même temps, il peut arriver qu'elles détectent un canal libre en même temps, mais la probabilité qu'elles choisissent le même *backoff* est faible et diminue alors que la taille du *CW* augmente.

5. La modulation O-QPSK est une modulation à quadrature de phases ayant subi un *offset* entre ce qui est envoyé sur la voie I et la voie Q de l'émetteur (on évite ainsi des sauts de phases de π)

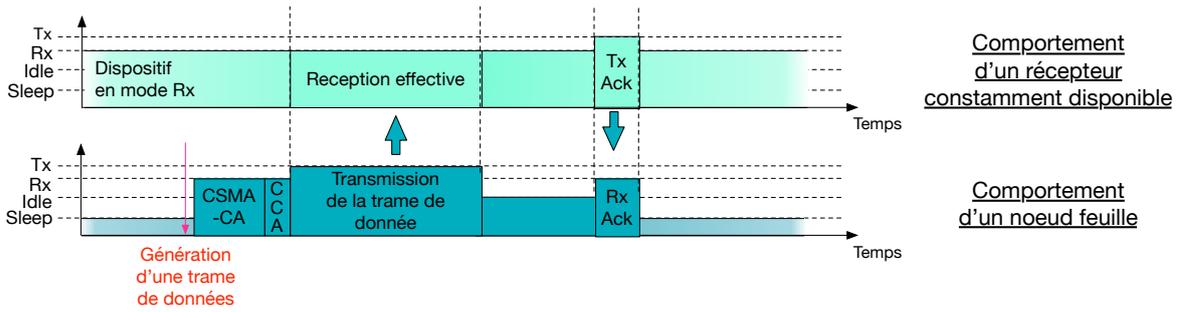


FIGURE 1.3 – Fonctionnement MAC d'une feuille qui souhaite envoyer un paquet à un parent en mode non-*beacon*.

Le mode non-*beacon*

Dans la version sans trames de synchronisation du standard 802.15.4, les transmissions sont organisées, au sein de la couche MAC de manière asynchrone : les nœuds ne partagent pas d'information temporelle globale.

Lorsqu'il souhaite transmettre un paquet (donnée ou contrôle), un nœud feuille, c'est-à-dire un nœud en périphérie du réseau, active sa radio dans un premier temps en mode de réception (*Rx*) pour réaliser l'algorithme de CSMA/CA du standard 802.15.4. Il réalise ensuite un CCA (Évaluation d'un canal libre — ou *Clear Channel Assessment*) et, si le canal est détecté comme libre, envoie son paquet. Le canal fréquentiel utilisé pour les transmissions est le même pour toutes les communications du réseau et correspond au canal fréquentiel sélectionné par le coordinateur *PAN*, le responsable du réseau, lors de la phase d'initialisation du réseau.

Ce mécanisme de transmission suppose qu'un récepteur est constamment à l'écoute sur le canal choisi ; c'est pourquoi les parents de tels nœuds sont considérés comme des périphériques constamment disponibles en mode *Rx*.

Si ce mode de communication est coûteux en énergie du point de vue du dispositif alimenté, il permet néanmoins d'économiser le maximum d'énergie du point de vue des terminaisons ou feuilles du réseau.

Le mode *beacon* : utilisation de trames balises

Dans le mode de communication utilisant les trames balises, une structure en *supertrame* organise l'envoi des trames dans le temps. La figure 1.4 présente cette *supertrame* : elle commence par l'envoi d'une trame de *beacon*, qui annonce la durée de la prochaine *supertrame* et permet de synchroniser les périphériques avec le coordinateur. Ainsi, chacun des périphériques pourra soit transmettre, soit s'endormir après ce *beacon*, pour la durée annoncée par le coordinateur.

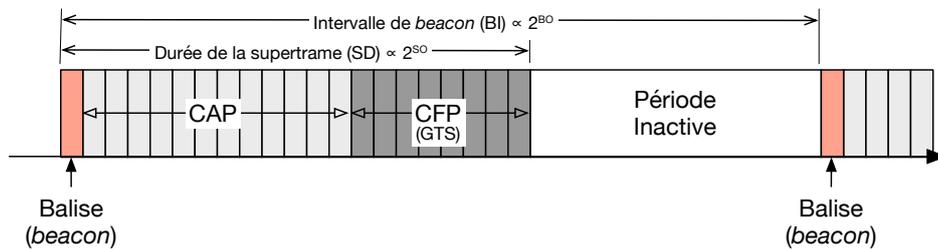


FIGURE 1.4 – Détail d'une supertrame 802.15.4 en mode balise.

À la suite du *beacon*, on peut noter une première période nommée CAP (Période de communication durant laquelle l'accès au canal se fait en compétition avec d'autres utilisateurs — ou *Contention Access Period*), durant laquelle les données des périphériques sont envoyées suivant la méthode d'accès CSMA/CA (cf. note de pied de page de la page 28), suivie d'une période de CFP (Période de communication dans laquelle l'accès au canal est déterminé *Contention Free Access Period*), constituée de différents *Guaranteed Time Slots*, des slot (Créneau temporel)s de temps réservés pour un émetteur particulier. Ces slots sont attribués par le coordinateur responsable de la *supertrame*. Un périphérique utilisateur peut demander l'allocation de plusieurs slots d'affilée au coordinateur *PAN* (Réseau personnel — ou *Personnal Area Network*), qui

décide de l'allocation des GTS en fonction des disponibilités et du remplissage de la *superframe*. Enfin, on remarque une période d'inactivité durant laquelle tous les périphériques s'endorment afin d'économiser de l'énergie.

Chacun des nœuds du réseau qui n'est pas une feuille gère sa propre *superframe* pour éviter les collisions avec les *superframes* de son parent. Les paramètres qui influent sur les cycles d'activités des nœuds sont les suivants :

- *SD*, la durée de la *superframe* — ou *Superframe Duration*,
- *BI*, l'intervalle entre l'envoi de trame de *beacon* — ou *Beacon Interval*,

Comme le montre la figure 1.4, la durée de *superframe* *SD* est fonction du paramètre *SO*, l'ordre de *superframe* — ou *Superframe Order*. De la même manière *BI*, l'intervalle de *beacon*, dépend de l'ordre de *beacon* *BO* — ou *Beacon Order* utilisé par le coordinateur. En utilisant les paramètres par défaut du standard pour le paramètre, on trouve que pour des valeurs de *BO* et *SO* comprises entre 0 et 14 (valeurs minimales et maximales du standard), le DC (Ratio d'activité cyclique — ou *Duty-Cycle*) le plus petit qu'un coordinateur puisse atteindre est de l'ordre de 6%, pour des valeurs de *BI* ~ 4.2 min ($BO = 14$), et *SD* $\sim 15,36$ ms ($SO = 0$).

De plus, les nœuds fonctionnant avec le mode *balise activée* peuvent ainsi atteindre de très faible niveau de consommation d'énergie : un nœud peut en effet éteindre sa radio en toute sécurité une fois qu'il a réalisé ses opérations de la *superframe* courante et ne se réveiller qu'à la balise suivante, réduisant ainsi sa période d'activité de *SD* à la durée d'une trame de *beacon* uniquement.

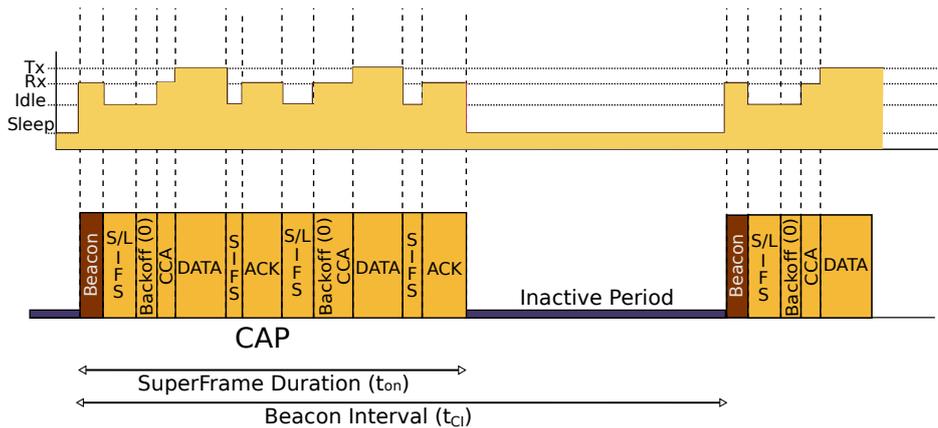


FIGURE 1.5 – Opérations du 802.15.4 en mode balise.

La figure 1.5 présente le cas d'un nœud qui souhaite envoyer un maximum de données pendant la période active d'une *superframe*, ainsi que ces différents modes d'opérations. Sur cette figure, on suppose que le nœud qui veut envoyer des données ne réalise pas de CSMA/CA (*backoff* = 0), mais que cela ne l'empêche pas d'avoir accès au canal (on peut ainsi supposer que ce nœud est un fils unique).

Le mode TSCH

Le mode TSCH fait partie de l'amendement de 2012 du standard 802.15.4 [14], et a été récemment intégré dans la version 802.15.4-2015 [9]. Cet amendement traite de la nécessité d'un accès déterministe pour les applications industrielles : le saut de fréquence permet au réseau d'être plus robuste aux évanouissements de canal et aux interférences liées aux trajets multiples; et un ordonnanceur évite les collisions en allouant périodiquement des cellules de transmission dédiées dans un ordonnancement, définies par un canal fréquentiel et un créneau temporel particulier.

Ainsi, TSCH cible principalement les réseaux industriels qui requièrent un niveau accru de robustesse, fiabilité, disponibilité et sécurité.

Organisation en *slotframe*. La figure 1.6 montre un exemple d'ordonnancement TSCH, représenté par une *slotframe*, un ensemble de cellules affectées à une paire de nœuds. Une *cellule* de communication TSCH est définie par un couple (*slot*, *canal logique*) et peut être de type :

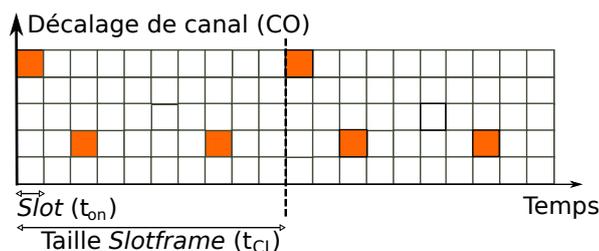


FIGURE 1.6 – Exemple d’ordonnancement TSCH.

- **dédiée** : un couple de nœuds émetteur/récepteur peut échanger de l’information sans collisions. Les nœuds non concernés peuvent rester éteints.
- **partagée** : les nœuds sont en compétition pour l’accès au canal, en utilisant la méthode d’accès CSMA/CA, minimisant ainsi le risque de collision.
- **libre** : aucun nœud n’est affecté à une cellule libre : tous les nœuds peuvent s’endormir et c’est une ressource disponible de l’ordonnancement.
- **advertisement** : les cellules dédiées à l’envoi de trames de synchronisation, les *Enhanced Beacons*, ou EBs. Les transmissions dans ces cellules sont de types *broadcast*, afin que des nœuds qui tentent de rejoindre le réseau les entendent.

L’ASN (Numéro de slot absolu — ou *Absolute Slot Number*) représente le nombre total de slots écoulés depuis le début du réseau. La norme ne définit pas comment établir l’ordonnancement pour la transmission des trames de données, mais certaines solutions ont déjà été proposées [15, 16, 17].

Saut de fréquence. Les nœuds utilisent le saut de canal en changeant périodiquement de canal fréquentiel, suivant une *séquence de sauts*, une liste ordonnée de canaux physiques. Il s’agit le plus souvent d’une liste pseudo-aléatoire des canaux fréquentiels disponibles, connue par tous les nœuds du réseau TSCH.

La relation suivante donne le canal physique (*CH*) pour un slot donné :

$$CH = HS[(ASN + CO) \bmod seq_{length}],$$

où *HS* est une séquence de sauts, *CO* est l’identifiant du canal logique, et seq_{length} , la longueur de la séquence de sauts *HS*.

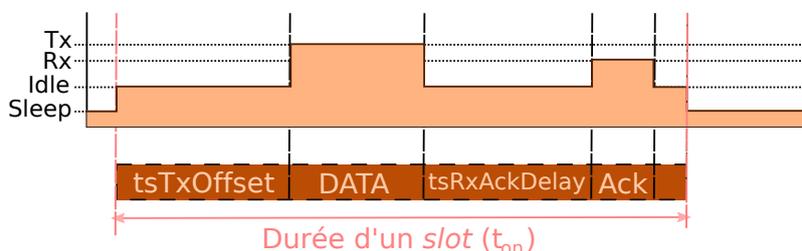


FIGURE 1.7 – État d’opération d’un nœud au sein d’un slot TSCH.

Organisation d’un slot. Chaque nœud infère son état pour une cellule particulière à partir de l’ordonnancement : il peut être en état d’émission, de réception ou de veille, comme le montre la figure 1.7.

La figure 1.7 présente le détail de fonctionnement d’un nœud au sein d’un slot d’envoi, avec les différents modes d’opérations associés.

Le format des trames de données TSCH est le même que celui du 802.15.4 classique (voir Fig. 1.2), la seule différence étant que le champ *Numéro de séquence* est omis.

Attache au réseau. Afin qu'un nœud rejoigne le réseau TSCH déployé, le processus est le suivant : les nœuds qui ont déjà rejoint le réseau envoient des trames de EB avec une période T_{adv} , un nœud qui tente de rejoindre le réseau, et se réveille à l'instant t_0 , surveille en mode Rx le ou les canaux à la recherche d'une balise pour se synchroniser au réseau. Le standard 802.15.4 ne spécifie pas non plus la manière d'ordonnancer l'envoi des trames d'EB.

Mécanisme de synchronisation. Une fois qu'un nœud a rejoint le réseau, il n'est plus nécessaire d'utiliser les balises pour la synchronisation. Lorsqu'un émetteur veut envoyer un paquet, il allume sa radio $tsTxTxOffset$ après le début d'un slot (voir Fig.1.7).

A la réception d'une trame, le récepteur calcule la différence entre l'heure d'arrivée de la trame et l'heure d'arrivée prévue $tsTxTxOffset$ pour ajuster son horloge par rapport à l'émetteur.

De plus, les trames d'ACK (Acquittement) peuvent également contenir un horodatage de manière à réajuster la valeur de l'horloge de l'émetteur du paquet de données.

Toutefois, s'il n'y a pas de trafic de données, les nœuds doivent utiliser des messages de KA (Maintien vivant — ou *Keep-Alive*) pour la synchronisation. L'intervalle maximum entre les messages de KA (trame de données vide) est défini de manière à ce que la dérive d'horloge ne dépasse jamais la durée de GT (Intervalle de garde: temps passé en Rx pour prendre en compte la dérive d'horloge — ou *Guard Time*). Vis-à-vis des notations du standard TSCH, GT se définit de la manière suivante :

$$GT = tsTxOffset - tsRxOffset \tag{1.2}$$

Puisque nous supposons une dérive symétrique de l'horloge au récepteur et à l'émetteur, on peut déduire :

$$KA_{période} \leq \frac{GT}{2 * \text{precision d'horloge}} \tag{1.3}$$

Ainsi, en supposant une précision d'horloge de 40 ppm, et les valeurs par défaut du standard [14] (Tableau 52e), nous trouvons une valeur de $GT = 1000 \mu s$, dont on déduit

$$KA_{période} \approx 12.5s. \tag{1.4}$$

Machine d'état. Ainsi, nous pouvons représenter les différents états de synchronisation d'un nœud au sein d'un réseau TSCH par la machine d'état de la figure 1.8.

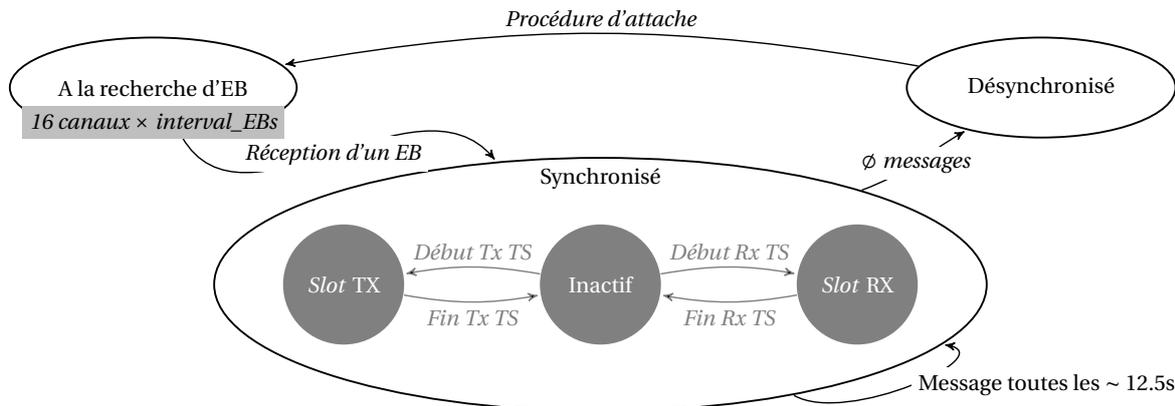


FIGURE 1.8 – Macro-machine d'état d'un nœud TSCH – TS signifie créneau temporel, ou *timeslot*.

Le mode *Coordinated Sampled Listening* (CSL)

Les protocoles à échantillonnage de préambule. Le CSL s'inscrit dans la famille des protocoles asynchrones à échantillonnage de préambule, fondée autour du concept d'écoute à faible consommation d'énergie (LPL, ou *Low Power Listening*). Cette famille regroupe notamment les protocoles B-MAC [18], MFP [19] PS-ALOHA [20], DPS-MAC [21], CONTIKIMAC [22] et le mode CSL du standard 802.15.4.

Les évolutions des protocoles MAC permettent aux récepteurs d'émettre des trames d'acquiescement (ACK) après réception d'un des paquets de préambule, dans le but de raccourcir la durée d'envoi du préambule de la part de l'émetteur, puisqu'il transmet alors ses données dès qu'un récepteur est disponible. Ce mécanisme réduit non seulement la consommation d'énergie, mais aussi le délai de transmission. X-MAC [23], ENB-MAC [24], CSMA-MPS [25], TICER [26] et C-MAC [27] sont des exemples d'utilisation de ce mécanisme d'échantillonnage de préambule avec des trames d'acquiescement. Le détail de chacun des protocoles MAC peut être trouvé dans l'état de l'art de l'évolution des protocoles MAC réalisé par Huang *et al.* [28].

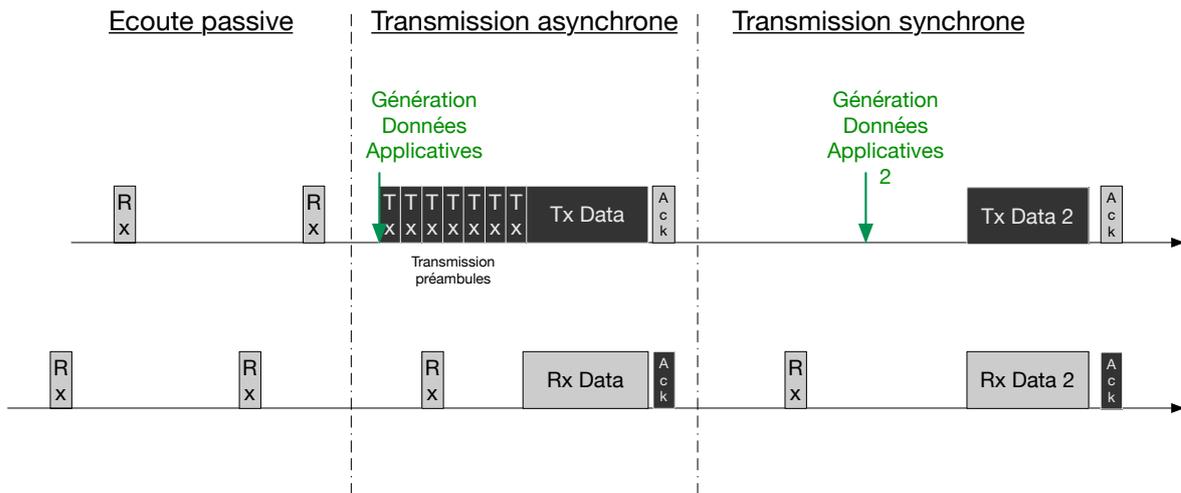


FIGURE 1.9 – Principe de fonctionnement du mode CSL.

Échantillonnage passif. La figure 1.9 présente le fonctionnement global du mode de communication CSL de la version 2015 du standard 802.15.4 [9]. Dans ce mode, lorsqu'il n'y a pas de communication, tous les nœuds du réseau fonctionnent de manière identique : ils se réveillent périodiquement pour échantillonner le canal physique de communication, en quête d'un potentiel préambule. Cette phase est représentée sur la partie gauche de la figure 1.9, avant la génération de la donnée applicative.

Le rôle du transmetteur. Lorsque l'un des nœuds décide de transmettre une donnée à la demande d'une application, il entre dans le rôle d'émetteur. Dans ce rôle, il commence par signaler la présence d'un trafic à envoyer par l'envoi consécutif de trames de préambules, contenant la date d'envoi de la trame de donnée. Ces préambules sont des trames de types 802.15.4 *Multipurpose* contenant uniquement un IE (Elément d'information — ou *Information Element*) de date de rendez-vous, lui-même constitué de deux champs : la date de l'envoi de la trame, ainsi qu'un intervalle d'écoute du nœud, dans le but d'anticiper de potentielles communications futures. Ainsi, pour envoyer des données, seul le champ contenant la date d'envoi de la donnée est utilisé, l'intervalle de réveil étant annoncé comme nul.

La réception d'une trame. De cette manière, lorsqu'un potentiel récepteur se réveille pour échantillonner le canal, et après avoir vérifié qu'il est le destinataire du préambule, il reçoit la date de la transmission de donnée et peut alors éteindre son récepteur jusqu'à la date annoncée, comme on peut le voir sur la partie centrale de la figure 1.9.

Une fois la transmission réalisée, les deux nœuds retournent dans le mode d'écoute passive, et échantillonnent périodiquement le canal à la recherche de potentiels préambules.

Mode de communication synchronisé. Le mode CSL du standard permet également un mode de communication synchrone en point à point via l'exploitation du champ "Intervalle de réveil" de l'IE de rendez-vous : en effet, grâce à ce champ, un nœud peut informer un de ses voisins de l'intervalle entre ses réveils, afin que les nœuds qui souhaitent communiquer avec lui envoient leurs trames dans ses intervalles, comme le montre la partie droite de la figure 1.9 (on suppose que l'intervalle de réveil a été précisé plus tôt via l'envoi de préambules). Cependant, un tel mode de communication synchrone point à point ne peut fonctionner à l'échelle d'un grand réseau, et c'est pourquoi, même si le groupe de travail de Thread prévoit d'ajouter ce mode à leur spécification, le mode CSL serait exploité uniquement entre un *Master* et un nœud périphérique de type WED (Noeud terminal que l'on peut réveiller — ou *Wakeable End Device*) : un nœud SED (Noeud terminal endormi — ou *Sleepy End Device*) qui fonctionnerait en mode CSL.

Étendre ce mode de fonctionnement synchronisé à la globalité du réseau revient à poser les mêmes problématiques de partage temporel et de synchronisation globale du réseau que lors de la création d'un réseau en mode TSCH.

Par ailleurs, le mode CSL peut être activé sur plusieurs canaux fréquentiels en répétant le processus d'activité sur un canal présenté sur la figure 1.9 sur chacun des canaux exploités.

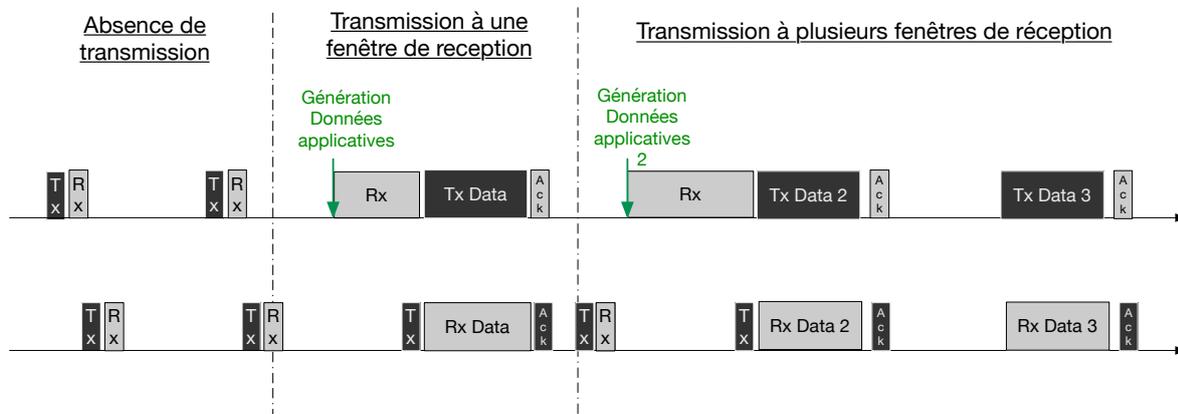


FIGURE 1.10 – Principe de fonctionnement du mode RIT.

Le mode *Receiver Initiated Transmission* (RIT)

Si c'est le récepteur qui se rend initialement disponible, c'est ce dernier qui se signale sur le canal de communication, et l'émetteur de la transmission répondra à ses sollicitations lorsqu'il sera à son tour disponible pour la transmission : on parle alors de transmissions initiées par le récepteur.

Dans ces modes de communication, c'est le nœud récepteur qui choisit l'instant de transmission en interrogeant l'émetteur de données grâce à des trames de sollicitation (ou *poll*). L'exploitation de trames de *polling* est notamment utilisée par les propositions A-MAC[29], RICER [26], RI-MAC [30], et par le mode RIT du standard 802.15.4, présenté en détail dans la section suivante.

Fonctionnement hors transmission de données. Alors que, dans le mode CSL, les nœuds échantillonnent le canal périodiquement en mode *Rx* lorsqu'aucune transmission de données n'a lieu, les nœuds qui fonctionnent en mode RIT envoient périodiquement des trames de sollicitation, les *RIT Data Request*, qui viennent périodiquement demander aux voisins de ce nœud s'ils ont une trame à leur destination, comme le montre la partie gauche de la figure 1.10.

Les trames de *RIT Data Request* sont des trames de commande MAC contenant un IE de type RIT, qui peut également être inclus dans des trames MAC de types EB ou trames de données.

L'IE RIT peut contenir jusqu'à trois types d'informations : la date de la prochaine écoute du nœud émetteur de la trame RIT, l'intervalle entre les écoutes successives, et le nombre d'écoutes successives avant de repartir dans le mode passif.

Ainsi, deux cas peuvent se présenter : selon que le nœud émetteur du RIT souhaite allouer une seule fenêtre de transmission ou qu'il préfère donner plusieurs créneaux de transmissions possibles. Dans le premier cas, seule l'information de la date de la prochaine écoute est transmise dans l'IE RIT. S'il souhaite allouer plusieurs fenêtres de transmissions, il est possible d'exploiter les deux autres champs du RIT IE dans

le but d'ouvrir plusieurs fenêtres de communication successives. La partie droite de la figure 1.10 présente la situation d'écoutes répétées alors que la partie centrale représente l'utilisation d'une fenêtre d'écoute unique.

A la suite de la transmission d'une requête RIT, les nœuds à l'origine des trames de sollicitation doivent se positionner en mode *Rx* pendant un court laps de temps (à partir de la date annoncée dans l'IE RIT), pour donner la possibilité à l'un de leurs voisins de leur transmettre une trame de données. La longueur de cet intervalle correspond à la durée minimale passée en mode *Rx* qui garantit la transmission correcte d'une trame de donnée depuis un voisin.

Rôle de transmetteur. Lorsque l'un des nœuds génère une donnée applicative à envoyer à un voisin, il se place dans le mode *Rx* de manière continue jusqu'à réception d'une trame de sollicitation, c'est-à-dire d'une trame contenant l'IE RIT. A réception de cette dernière, le nœud décode l'information sur la date du prochain instant d'écoute, et transmet sa trame de données à cette date-là, en respectant une procédure de CSMA/CA classique.

Si la trame RIT contient des informations sur plusieurs fenêtres d'écoute successives, le nœud peut alors transmettre plusieurs paquets en exploitant ces différentes fenêtres.

1.1.1.3 Exemple de piles protocolaires basées sur le standard 802.15.4

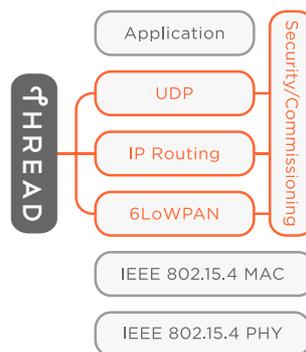


FIGURE 1.11 – Pile protocolaire Google Thread ©Threadgroup [1].

Google Thread

Présentation générale. Google Thread est une pile protocolaire ayant pour visée les *smart homes* ou maisons intelligentes, qui prévoit le déploiement de différentes solutions pour améliorer l'habitat : gestion de l'éclairage, des accès à la maison, surveillance et sécurité et gestion d'énergie pour le bâtiment. La proposition est basée sur des standards éprouvés et ouverts tels que le 6LoWPAN (Réseau personnel (PAN) sans fil IPv6 à basse consommation — *IPv6 Low power Wireless PAN*) [31], l'IP (Internet Protocol)v6 et le 802.15.4 [9]. La figure 1.11 présente la pile protocolaire utilisée par Thread pour proposer des réseaux maillés, capables d'autoguérison, ne comprenant pas de points uniques de défaillance.

Un réseau Thread consiste en un *Leader* qui choisit, pour tout le réseau, le canal fréquentiel utilisé ainsi que le *PAN ID* et les informations de sécurité.

Autour de ce *Leader*, des *Routers* forment une dorsale de nœuds Thread toujours actifs, qui échangent des informations pour maintenir un cœur de réseau disponible à la réception du trafic de feuilles, des nœuds périphériques du réseau, majoritairement endormies.

Spécificité MAC. Dans un réseau Thread, les feuilles sont nommées des *Sleepy End Devices* (SED) et passent la plupart du temps endormies. C'est dans le but d'économiser de l'énergie pour ces nœuds que les nœuds de la dorsale sont constamment disponibles : ainsi, les SED peuvent obtenir des durées de vie de l'ordre de dizaines d'années alors qu'ils ne sont simplement alimentés que par de petites batteries [1].

Afin de gérer ces nœuds endormis dans le réseau, à la fois pour la gestion du réseau ainsi que pour le trafic descendant dans ces feuilles, Thread propose un système de *proxy*⁶ mis en place par la notion de "parent" Thread : après une phase de négociation à l'initialisation, un *Router* Thread peut agir comme le parent d'un SED, et ainsi gérer les messages à la destination de ce nœud SED qui circulent dans le réseau, ainsi que le trafic venant de ce périphérique pour l'intérieur du réseau.

Une fois que le nœud SED est réveillé, il transmet ses données à son parent. À ce moment, le nœud parent sait qu'il peut transmettre les données stockées à destination du SED en réponse à la sollicitation du SED.

Les couches hautes. Thread utilise la couche 6LoWPAN, qui permet de faire l'interface avec la couche IP de niveau 3, en morcelant et rassemblant les paquets IP trop grands pour pouvoir être encapsulés dans des paquets de PHY de 127 octets, la taille maximale imposée par le standard 802.15.4.

Le protocole de routage utilisé par la spécification Thread est RIP (Protocole de routage de l'information — ou *Routing Information Protocol*), l'un des plus élémentaires protocoles de routage, tel que présenté dans la RFC 2453 [32]. C'est un protocole de type vecteur de distance, qui suppose que les nœuds du réseau qui prennent part au routage RIP envoient périodiquement les informations de leur table de routage à l'ensemble de leurs voisins, ce qui permet de continuellement propager l'information de l'état des routes à tous les voisins. La transmission des routes à tous les voisins représente un coût énergétique non négligeable, que la solution Thread n'a pas à prendre en compte puisque les seuls nœuds qui prennent part au routage sont les nœuds alimentés, non contraints en énergie.

Le protocole de transport utilisé est l'UDP (Protocole de datagramme utilisateur — ou *User Datagram Protocol*), comme dans le cas de la pile CoAP (Protocole d'application contraint — ou *Constrained Application Protocol*) utilisée par Palattella *et al.* [33] pour proposer une pile standardisée pour l'*Internet des objets importants*.

La couche applicative repose quant à elle sur l'utilisation de services applicatifs CoAP, par exemple le *framework* HomeKit d'Apple [34], le *framework* libre IoTivity [35] de la fondation OpenConnectivity [36], ou encore le *framework* applicatif Zigbee, utilisé par la pile Zigbee 3.0.

La pile Zigbee

La Zigbee Alliance est une association d'entreprises à but non lucratif qui s'est fondée en 2002 et compte désormais plus de 400 membres (dont Huawei, Schneider Electric, Philips, Samsung, Siemens, Texas Instrument (TI)). Le but de cette association est de fournir des standards permettant de réaliser des installations Zigbee simplement interopérables.

L'élaboration des protocoles Zigbee permet de répondre à l'augmentation des demandes dans le domaine de l'automatisation de gestion de bâtiments, ainsi que des réseaux dits intelligents. Ces protocoles doivent donc, en plus de prendre en compte cet aspect d'automatisation, remplir les caractéristiques classiques des WPAN, c'est-à-dire une sécurité accrue, être fiables et autoréparateurs, être flexibles et extensibles, et enfin être peu chers et simples à mettre en place. De plus, il faut respecter la contrainte de faible consommation liée aux réseaux des capteurs.

Les familles de protocoles Zigbee. La Zigbee Alliance génère les différents standards Zigbee. Il existe à l'heure actuelle 3 spécifications Zigbee : Zigbee RF4CE, Zigbee 3.0 et Zigbee IP. Le standard RF4CE, bien que très prometteur pour le contrôle à distance, ne satisfaisait pas d'ambition de portée applicative évoluée, importante pour la plupart des communications. Le standard Zigbee 3.0 est actuellement le plus répandu, car historiquement plus présent, même si les implémentations tendent à évoluer vers la pile Zigbee IP, plus interopérable au niveau applicatif.

En effet, la figure 1.12 permet de comprendre rapidement que la pile protocolaire Zigbee 3.0 repose sur de nouveaux protocoles de communications – notamment l'APS ou la ZCL –, alors que la pile Zigbee IP utilise des protocoles d'ores et déjà connus et éprouvés – IPv6, TCP (Protocole de contrôle de transmissions — ou *Transmission Control Protocol*), RPL (Protocole de routage pour les réseaux à faible consommation d'énergie et avec pertes — ou *Routing Protocol for Low-Power and Lossy Networks*).

6. Traduction littérale : *mandataire*. Le *proxy* est un composant qui joue le rôle d'intermédiaire en se plaçant entre deux entités du réseau pour faciliter la communication entre ces deux équipements. Parfois, le *proxy* peut être utilisé dans le but de sécuriser la transmission, mais ce n'est pas le but ici.

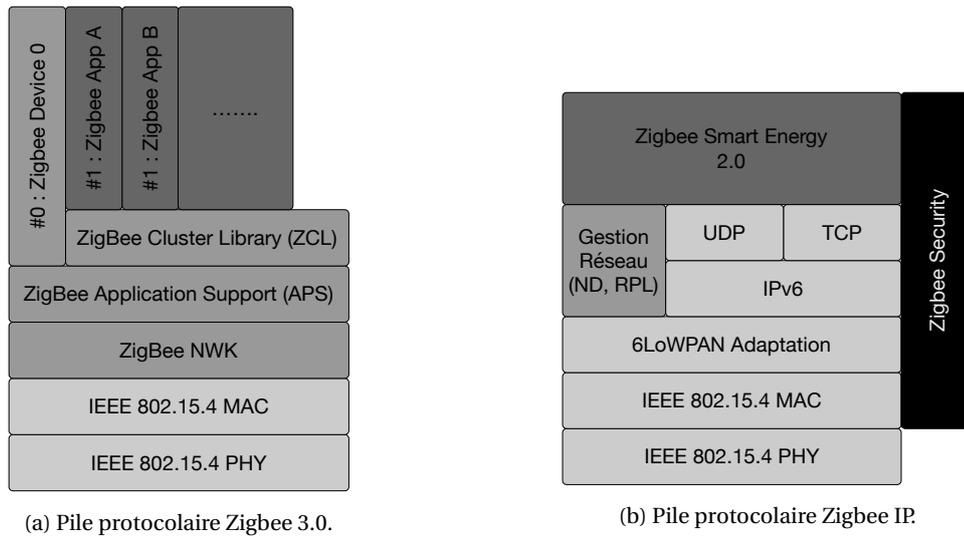


FIGURE 1.12 – Piles Zigbee.

Au niveau MAC, les piles protocolaires Zigbee utilisent les deux modes de communication les plus courants du standard 802.15.4 historique : le mode synchronisé avec balises et le mode non-*beacon*.

Couches hautes. La couche réseau est différente suivant la pile Zigbee sélectionnée, comme le montre la figure 1.12. La pile Zigbee IP pile fonctionne grâce au protocole IPv6, accompagné de l'interface 6LoWPAN, et le protocole de routage utilisé dans cette pile est le protocole RPL, qui repose principalement sur la maintenance d'un arbre de collecte IPv6 pour l'extraction de trafic hors du réseau.

La pile Zigbee 3.0 quant à elle, fonctionne grâce à la couche Zigbee PRO NWK. Le routage utilisé par cette couche est l'AODV (Protocole de routage ad-hoc, à la demande, et de type vecteur de distance — ou *Ad-hoc On-demand Distance Vector*). L'idée clé de ce routage est qu'il fonctionne à la demande, c'est-à-dire que contrairement à RPL, il n'établit pas de table de routage permanente.

Pour ce qui est des couches applicatives, les deux piles Zigbee IP et Zigbee 3.0 reposent sur l'utilisation de profils applicatifs définis par la Zigbee Alliance, qui décrivent les objets, leurs attributs et les fonctionnalités associées à travers un framework applicatif, à la manière d'IoTivity [35].

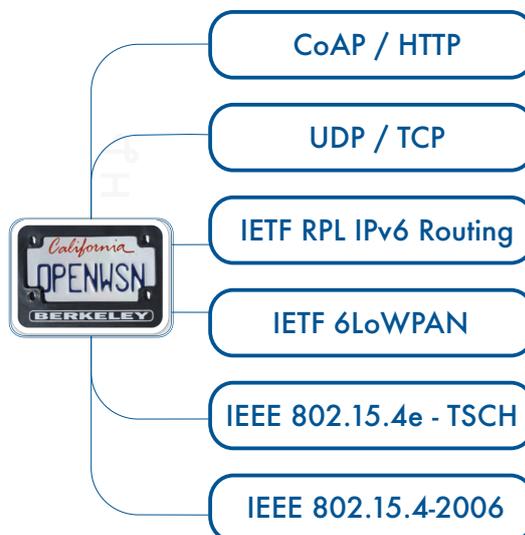


FIGURE 1.13 – Pile protocolaire openWSN.

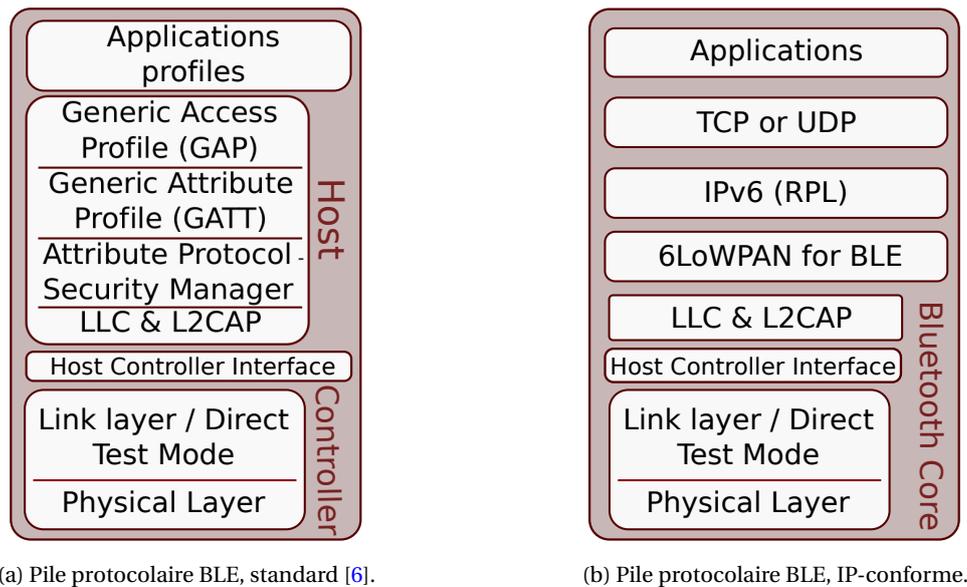


FIGURE 1.14 – Piles BLE.

OpenWSN

La figure 1.13 présente une implémentation de la pile protocolaire proposée par le groupe de travail 6TiSCH [37] : le projet openWSN. OpenWSN vise à construire une implémentation standard ouverte d'une pile complète de protocoles réseau contraints pour les réseaux de capteurs sans fil et l'IoT. Cette pile repose sur le concept l'utilisation du mode MAC TSCH. Au-dessus de la couche MAC, la pile basse consommation pour les réseaux à pertes (*lossy*) exploite les normes IETF⁷, y compris la couche de gestion et d'adaptation 6TiSCH de l'IETF (un profil de configuration minimal [38], le protocole 6top [39] et différentes fonctions pour réaliser l'ordonnancement).

Cette pile est également complétée par une implémentation de 6LoWPAN, le protocole RPL en mode *non-storing*, l'UDP, le TCP et les protocoles applicatifs CoAP et HTTP (Protocole de transfert hypertexte — ou *HyperText Transfert Protocol*), permettant un accès simple et facilement interopérable avec d'autres utilisateurs de ces standards ouverts.

1.1.2 Bluetooth Low Energy

BLE est la version basse consommation de la norme Bluetooth, qui vise à réduire la consommation d'énergie et permet ainsi aux utilisateurs de faire de l'IoT tout en utilisant un périphérique Bluetooth classique. Pour ce faire un *contrôleur* Bluetooth spécifique aux WSN a été proposé : c'est le contrôleur BLE. Il est principalement utilisé pour les applications de surveillance, telles que les moniteurs de fréquence cardiaque ou le contrôle à distance de la température d'une pièce. Un avantage important de la technologie BLE est sa popularité —BLE est déjà déployé dans la majorité des ordiphones et il est largement utilisé pour des applications mobiles.

Cette technologie définit une relation maître/esclave entre deux nœuds dans une topologie en étoile. BLE ne supporte pas encore la topologie maillée, mais la topologie *scatternet* (une collection d'arbres) est déjà disponible dans la version 5.0 puisqu'un nœud peut à la fois être maître et esclave, dans deux communications différentes.

La pile classique Bluetooth est présentée sur la partie gauche de la figure 1.14, tandis qu'une pile hybride IPv6/BLE est présentée sur la droite. Cette pile IP-conforme est proposée par le groupe de travail 6lo de l'IETF [40]. Bien que la pile protocolaire BLE standard soit la plus fréquemment implémentée, nous avons choisi ici de présenter la pile IP-conforme, car elle est plus proche des autres piles protocolaires présentées dans ce rapport, et nous permettra ainsi de comparer les protocoles sur un pied d'égalité au sein du

7. L'*Internet Engineering Task Force*, abrégée IETF, littéralement traduit de l'anglais en « Groupe de travail d'ingénierie d'Internet » est un groupe informel, international, ouvert à tous, qui participe à l'élaboration des standards Internet.

troisième chapitre (Comparaison des standards pour l'IoT).

Dans cette section, nous détaillerons donc essentiellement le fonctionnement des couches PHY et MAC de la pile IP-conforme, les couches hautes étant communes à la plupart des piles protocolaires de l'IoT.

1.1.2.1 Couche PHY

La couche PHY de BLE divise la bande de fréquences ISM 2.4 GHz en 40 canaux. La largeur de chaque canal est de 2 MHz avec un espacement entre les canaux de 2 MHz. Trois canaux sont dédiés à la signalisation, alors que les 37 autres canaux sont réservés au transfert de données. La modulation utilisée par le BLE est GFSK⁸ pour obtenir un débit binaire maximal de 1 Mb/s (2 Mb/s dans la version 5.0). Afin d'être robuste aux interférences et aux évanouissements de canal, Bluetooth réalise un étalement de spectre par saut de fréquences. Les nœuds BLE peuvent changer de canal fréquentiel de manière périodique ou à un instant logique donné, en fonction du mode de saut de canal fréquentiel utilisé.

La récente version 5.0 permet de s'adapter à différentes contraintes en termes de portée et de robustesse grâce à quatre schémas de codage conduisant à plusieurs débits physiques : 125 kb/s, 500 kb/s, 1 Mb/s et 2 Mb/s.

1.1.2.2 Couche MAC

Les nœuds BLE peuvent fonctionner dans quatre modes différents : *advertising*, *scanning*, *slave* ou *master*. Les deux premiers modes sont utilisés dans la phase d'attache au réseau alors que les deux derniers sont atteints une fois qu'une liaison est établie entre deux nœuds, c'est-à-dire qu'un nœud peut communiquer par des *événements de connexion*. Dans ce cas, le maître et l'esclave se réveillent de manière synchrone pour échanger des trames. Les événements de connexion se répètent périodiquement dans le temps, à chaque *intervalle de connexion*. Les connexions sont maintenues jusqu'à ce que l'un des nœuds veuille fermer la liaison, ou que le nombre maximum d'intervalles de connexion sans trafic (*latence de l'esclave*) soit atteint. Par conséquent, les paramètres *intervalle de connexion* et *latence de l'esclave* définissent le DC minimal d'un esclave.

Établissement de la connexion Le mécanisme de jonction à un réseau BLE est initié par le nœud entrant, qui est en mode *advertising* : il signale sa présence à l'aide de paquets ADV (Trame de synchronisation du BLE, nommée *ADvertisement frame*) sur un ou plusieurs des trois canaux dédiés à la signalisation. Les nœuds en mode *scanning* balayent quant à eux ces canaux afin de découvrir les paquets ADV des potentiels futurs esclaves.

Lorsqu'un maître entend un paquet ADV, il répond par une *requête de connexion*, contenant les informations pour les événements de connexion qui se produiront entre ces deux nœuds : intervalle de connexion, latence de l'esclave, date du début du premier événement de connexion, ainsi que le canal fréquentiel de données qu'ils utiliseront pour le premier événement de connexion.

Ainsi, les deux nœuds basculent sur le canal de données et changent de mode : le nœud scanneur devient maître tandis que le nœud qui se signalait devient esclave (cf. figure 1.17).

Maintient de la connexion active. Au début de chaque intervalle de connexion, l'esclave se réveille et attend un paquet de sondage (*poll paquet*) de son maître avant d'envoyer son paquet, dans le but d'éviter une possible collision avec le paquet d'un autre esclave à destination du même maître.

Le mécanisme de réveil n'est pas obligatoire à chaque intervalle si l'esclave n'a pas de donnée à transmettre durant cet intervalle de connexion, sauf si le nombre maximum d'intervalles de connexion sans trafic de données (*latence de l'esclave*) a déjà été atteint par l'esclave.

La figure 1.15 présente le fonctionnement du BLE dans le cas d'un débit maximal : l'esclave envoie autant de données que possible dans l'intervalle de connexion. De plus, la figure 1.16 présente le format

8. *Gaussian Frequency Shift Keying*, un mode de modulation de fréquence numérique dans lequel le signal modulé varie entre des fréquences prédéterminées. Contrairement à la FSK, plutôt que de moduler directement la fréquence avec les symboles de données numériques en changeant "instantanément" la fréquence au début de chaque période de symboles, le GFSK filtre les impulsions de données avec un filtre gaussien pour rendre les transitions plus fluides.

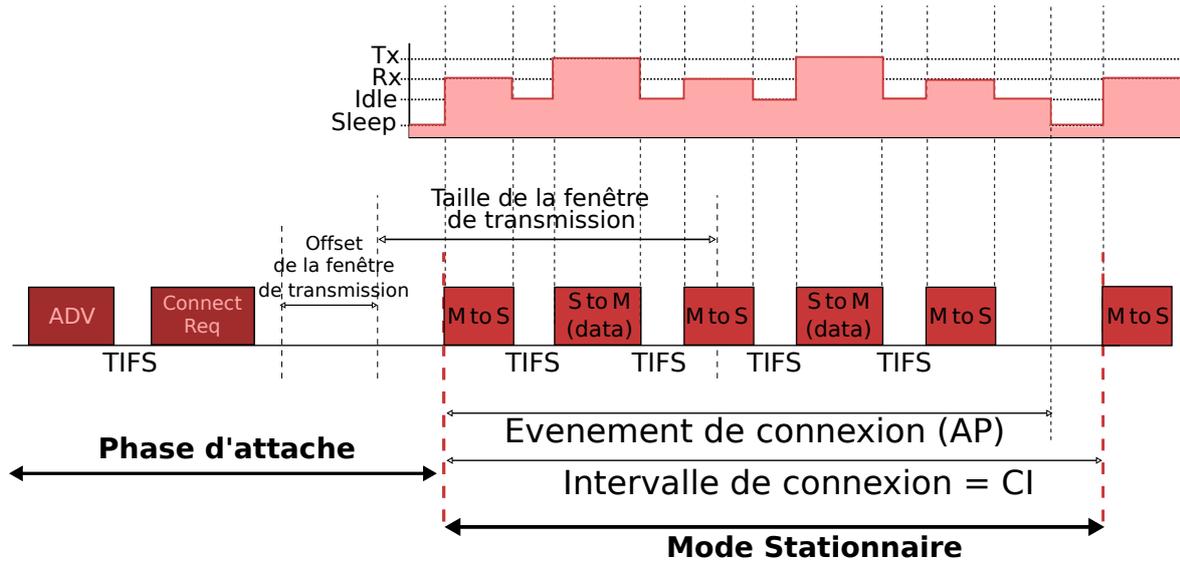


FIGURE 1.15 – Présentation globale du BLE : Efficacité maximale.

Paquets BLE non-codés
(1Mb/s and 2Mb/s)

PHY Ovhd		MAC Ovhd						LL Ovhd		6LoWPAN HC		IPv6 Data Packet	PHY CRC	
Preamble	Access address	LLID	Next SN	Seq. Nb	MD	Length	Length	Channel ID	Dispatch	LoWPAN IPHC	...	PHY CRC		
8 or 16	32	2	1	1	1	3	5	3	16	16	8	8	...	24

Paquets BLE codés
(125kb/s and 500kb/s)

PHY Ovhd		PHY TERM1				MAC Ovhd						LL Ovhd		6LoWPAN HC		IPv6 Data Packet	PHY CRC	PHY TERM2
Preamble	Access address	Code Ind.	TERMI	LLID	Next SN	Seq. Nb	MD	Length	Length	Channel ID	Dispatch	LoWPAN IPHC	...	PHY CRC	PHY TERM2			
80 μ s	256 μ s	16 μ s	24 μ s	2	1	1	1	3	5	3	16	16	8	8	...	24	3	

x2 ou x8 (μ s)

FIGURE 1.16 – Format minimal du paquet de données BLE (taille en bits).

d'une trame de données telle qu'envoyée par un esclave à son maître, dans le cas d'une communication IPv6.

Le mode de communication asynchrone. BLE est connu pour son mode connecté, purement synchrone, mais il existe aussi un mode de communication asynchrone. BLE peut en effet également transmettre 31 octets dans des trames d'ADV (et 31 octets de plus dans les paquets SCAN REQ). Ces paquets sont envoyés périodiquement jusqu'à ce qu'un paquet soit capté et acquitté par un scanner, à la manière de la procédure d'attachement au réseau.

Schéma temporel du BLE. La figure 1.17 résume le fonctionnement global du BLE, depuis la procédure d'attachement au réseau jusqu'à l'envoi de trames de données périodiquement entre deux nœuds.

1.1.2.3 Les couches hautes

LLC & L2CAP. Les couches de contrôle (LLC — ou *Link Layer Control*) et d'adaptation des liens logiques (L2CAP — ou *Logical Link Control & Adaptation Protocol*) permettent de supporter le multiplexage, la segmentation et le réassemblage des paquets des couches plus hautes, ainsi que la QoS (Qualité de service — ou *Quality of Service*). Elles permettent ainsi aux protocoles de haut niveau d'échanger des paquets

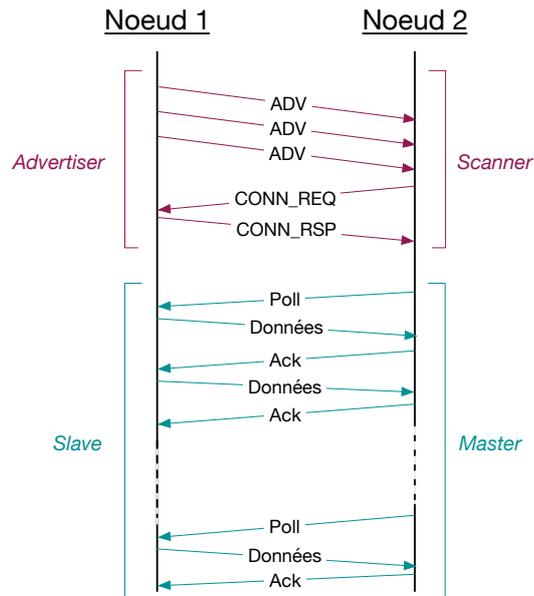


FIGURE 1.17 – Échange de messages BLE pour un mode de communication synchrone.

de taille maximale de 64 ko, et autorisent un contrôle de flux par canal logique de communication.

6LoWPAN for BLE. Le groupe 6lo [40] a proposé une adaptation de la couche 6LoWPAN des réseaux 802.15.4 (RFC 4919 [31] et RFC 4944 [41]) pour le BLE, de manière à décrire comment les paquets IPv6 sont transportés par le BLE, en utilisant les techniques 6LoWPAN.

Les principales responsabilités de 6LoWPAN sont la fragmentation et le réassemblage des paquets de haut niveau, ainsi que la compression de l'entête IPv6. Comme la couche L2CAP, elle sert d'adaptation à la couche IPv6.

IPv6, UDP/TCP et les couches applicatives. Ces couches sont communes à la pile classique TCP/IP, et ne seront pas présentées ici, car trop éloignées du point de vue de notre étude.

Nous précisons ici simplement que la couche IPv6 est la couche qui permet à tous les objets connectés de communiquer avec l'Internet classique. En effet, ce sont ces adresses qui permettent d'identifier les objets de manière unique sur le réseau classique, comme nous l'avons vu dans l'introduction de ce manuscrit.

Enfin, nous mentionnons rapidement l'utilisation possible de la couche applicative CoAP, qui est une version optimisée pour les réseaux de capteurs du protocole HTTP, utilisée au-dessus du protocole IPv6.

1.2 Solutions LAN

1.2.1 802.11b Power Saving Mode (PSM)

Le PSM (Mode d'économie d'énergie — ou *Power Saving Mode*) est un mode d'optimisation énergétique du standard 802.11 spécifié dès la version 802.11b du standard de l'IEEE, c'est-à-dire en 1999 [42]. Ce standard a ensuite été utilisé par l'Alliance Wi-Fi lors de la réalisation de la certification de compatibilité Wi-Fi. Il cible l'efficacité énergétique en mettant en place des cycles de fonctionnement alternant des périodes d'activité et de veille pour les nœuds associés à l'AP (Point d'accès — ou *Access Point*).

Dans le 802.11b, les nœuds peuvent transmettre sur l'un des treize canaux de largeur 22 MHz séparés par 5 MHz.

La révision 802.11b du standard spécifie une modulation qui repose sur l'utilisation de clés de code complémentaire⁹ (CCK) de 8 bits/symbole, ce qui correspond à un débit binaire de 11 Mb/s.

Le standard 802.11 propose un mode dédié à l'économie d'énergie, appelé PSM, et permet à un nœud d'opérer dans deux modes différents : état actif et inactif (le mode de gestion de l'énergie du nœud est activé

9. Complementary Code Keying

ou non via un champ de contrôle de trame de l'en-tête MAC pour chaque paquet). La technologie clé de ce mode est que, de temps en temps, la radio peut être désactivée si elle n'est pas utilisée.

La norme 802.11b PSM utilise le mode DCF (Fonction de coordination distribuée — ou *Distributed Coordinated Function*) pour organiser le partage du support entre les différentes stations. Ce mode s'appuie sur l'algorithme CSMA/CA, présenté au pied de page de la page 28.

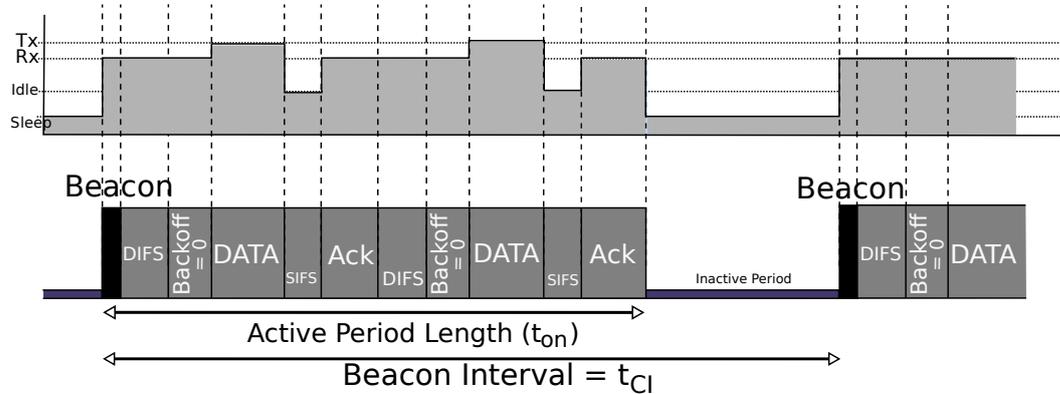


FIGURE 1.18 – Opération d'un nœud dans le mode d'économie d'énergie PSM et le mode DCF du standard 802.11.

La figure 1.18 présente les principes du mode DCF utilisé dans 802.11b PSM lorsqu'une station envoie deux paquets d'affilée à destination d'un AP. Pour cette figure, par souci de simplicité, nous avons supposé que la station est la seule associée avec l'AP.

Une valeur habituelle de l'intervalle des *beacons* est de 100 ms dans les réseaux classiques 802.11. Cependant, le champ "intervalle d'écoute", qui décrit la valeur de cet intervalle, est exprimé en nombre de slots, et est encodé sur deux octets, ce qui correspond à une valeur maximale de ~ 18.6 heures.

PHY Ovhd		MAC Ovhd				IPv6 Ovhd						IPv6 Data Packet	PHY CRC		
Preamble	PLCP Header	Ctrl	Dur-ation	Seq. Ctrl	FCS	Adresses	Vers.	Traffic Class	Flow label	PL Size	Nxt hdr	Hop Lim.	@	...	PHY CRC
72	32	16	16	16	32	48 x 4	4	8	20	16	8	8	128 x 2	...	16

FIGURE 1.19 – Structure d'un paquet 802.11 PSM (taille en bits).

La figure 1.19 présente la structure d'une trame de données envoyées avec le standard 802.11b PSM, dans sa forme minimale (pas de prise en compte de la sécurité).

Couches hautes Le standard 802.11b PSM ne définit que les couches PHY et MAC, sur lesquelles le protocole IP est prévu pour fonctionner, sans nécessité de couche d'adaptation à la manière de 6LoWPAN. Ainsi, la pile classique d'utilisation du standard 802.11b PSM est très proche de la pile TCP/IP classique.

1.2.2 802.11ah

La dernière version du projet de norme 802.11ah [43] est une extension de la norme IEEE 802.11 et répond spécifiquement aux principales exigences de l'IoT : une portée et une fiabilité accrue ainsi qu'une faible consommation d'énergie.

1.2.2.1 Couche PHY

La différence avec les couches physiques du 802.11b PSM présentées ci-dessus réside dans l'exploitation de la bande de fréquence ISM 868 MHz. Elle peut être ainsi être vu comme une couche PHY optimisée 802.11ac [44], qui exploite les bandes de fréquences inférieures à 1 GHz en mode PSM.

Le choix de la bande sous-1 GHz vient de l'objectif de limiter les émissions dans la bande 2,4 GHz. En outre, la limitation européenne du ratio d'activité dans cette sous-bande, préconisée par l'ETSI (Institut européen des normes de télécommunications — ou *European Telecommunications Standards Institute*), n'est pas applicable pour 802.11ah puisque le schéma de transmission repose sur les concepts de *Listen Before*

Talk¹⁰ et de *Adaptive Frequency Agility*¹¹ [45]. Comme en 802.11ac, il y a 11 MCS (Schéma de modulation et de codage — ou *Modulation & Coding Scheme*) disponibles, fonctionnant dans des sous-bandes de différentes largeurs.

Caractéristiques physiques		MCS Correspondant	Largeur de bande				
Modulation	Débit de codage		1MHz	2MHz	4MHz	8MHz	16MHz
BPSK	1/2 & 2x répétition	MCS0	0,30	0,65	1,35	2,93	5,85
BPSK	1/2	MCS1	0,60	1,30	2,70	5,85	11,70
QPSK	1/2	MCS2	0,90	1,95	4,05	8,78	17,55
QPSK	3/4	MCS3	1,20	2,60	5,40	11,70	23,40
16-QAM	1/2	MCS4	1,80	3,90	8,10	17,55	35,10
16-QAM	3/4	MCS5	2,40	5,20	10,80	23,40	46,80
64-QAM	2/3	MCS6	2,70	5,85	12,15	26,33	52,65
64-QAM	3/4	MCS7	3,00	6,50	13,50	29,25	58,50
64-QAM	5/6	MCS8	3,60	7,80	16,20	35,10	70,20
256-QAM	3/4	MCS9	4,00	-	18,00	39,00	78,00
256-QAM	5/6	MCS10	0,15	-	-	-	-

FIGURE 1.20 – Tableau présentant les débits accessibles (en Mb/s) avec la technologie 802.11ah, et leurs caractéristiques physiques associées. Tableau inspiré de la présentation du brouillon 802.11ah réalisé par Khorov *et al.* [46].

Le tableau 1.20 résume les caractéristiques physiques de ces différents MCS en termes de modulation et de codage utilisés ainsi que les débits qu'ils impliquent.

Le découpage de la sous-bande ISM sous-1 Ghz en différents canaux physiques aux États-Unis et en Europe sont disponibles dans le brouillon de spécification 802.11ah [43].

Ainsi, le débit de données minimum est donné pour le MCS10 et une largeur de bande de 1 MHz, en utilisant une modulation BPSK¹², avec un codage à 2 répétitions : le débit disponible est alors de 150 kb/s. De même, le débit de données maximum en Europe est de 7,8 Mb/s, et est obtenu, par l'utilisation du MCS8 et d'une largeur de bande de 2 MHz. Aux États-Unis, avec 16 MHz de largeur de bande, et en utilisant le MCS9, une modulation 256-QAM¹³, on peut atteindre un débit de 78 Mb/s.

1.2.2.2 Couche MAC

De la même manière que le PSM du 802.11, 802.11ah utilise le mode de communication DCF (voir Fig. 1.18).

Pour réduire la probabilité de collision dans les réseaux avec des milliers de stations et ainsi améliorer l'efficacité énergétique, le groupe de travail 802.11ah a développé le schéma d'accès RAW (Fenêtre d'accès restreinte — ou *Restricted Access Window*).

L'idée clé du RAW est de limiter l'ensemble des stations qui accèdent au canal aux mêmes instants, et de répartir leurs tentatives d'accès sur une période plus longue. Pour ce faire, RAW divise les stations en groupes, et le canal est découpé en zones temporelles. On affecte ensuite chaque zone temporelle à un groupe de stations. En diffusant les éléments d'information des paramètres RAW au sein des trames de *beacon*, l'AP attribue un ou plusieurs intervalles d'accès limité au médium, appelés fenêtres restreintes, ou RAW. Au sein de ces fenêtres restreintes, seul un groupe de stations peut accéder au canal [47].

Les travaux de [48] ont par ailleurs montré que DCF et RAW obtiennent des performances similaires dans les petits réseaux, en nombre de stations.

La figure 1.21 montre la structure d'une trame de données 802.11ah. Elle permet la transmission d'une trame de donnée de manière plus efficace que le 802.11 classique, notamment en réduisant la longueur du préambule physique, ainsi que la taille totale de l'entête MAC, par rapport au standard 802.11 classique.

10. Écoute du canal avant de transmettre.

11. Capacité à changer de canal fréquentiel de transmission si la qualité de celui utilisé se dégrade.

12. Modulation à changement de phase avec deux états — ou *Binary phase-shift keying*

13. Modulation d'amplitude à quatre états — ou *Quadrature Amplitude Modulation*

PHY Ovhd	MAC Ovhd				IPv6 Ovhd							IPv6 Data Packet	PHY CRC	
560 μ s si < 1Mhz	Frame Ctrl.	Seq. Ctrl.	FCS	Adresses	Vers.	Traffic Class	Flow label	PL Size	Nxt. hdr	Hop Lim.	@	128 x 2	...	16
240 μ s sinon	16	0	32	32	4	8	20	16	8	8				

FIGURE 1.21 – Structure d'une trame de donnée 802.11ah (taille en bits).

1.3 Solutions LPWAN

1.3.1 SIGFOX

SIGFOX se présente comme une "solution de connectivité cellulaire globale pour l'IoT". Le principe de la solution SIGFOX réside bien dans l'aspect cellulaire de la technologie, ce qui fait de la société SIGFOX un fournisseur de connectivité au même titre qu'un fournisseur de réseau mobile : SIGFOX ne fournit pas seulement des périphériques à ses clients, ils fournissent également une infrastructure à laquelle ces derniers connectent leurs appareils, dans le but de récupérer les données des périphériques via l'Internet classique.

SIGFOX étant une technologie propriétaire, les détails des couches PHY/ MAC ne sont pas accessibles au public. On peut néanmoins trouver des informations sur cette technologie via les documents publics disponibles sur le site de SIGFOX. Les sections suivantes présenteront globalement notre compréhension de la technologie via les documents accessibles.

1.3.1.1 Couche PHY

SIGFOX est une technologie à bande étroite qui fonctionne dans la bande de fréquences de 868 MHz en Europe et de 903 MHz aux États-Unis. Les débits PHY disponibles sont de 100 b/s pour une bande passante de 100 Hz et 1000 b/s pour une largeur de canal de 1 kHz en Europe (et 600 b/s aux États-Unis), obtenus grâce aux modulations DBPSK¹⁴ et GFSK¹⁵.

Alors que les débits disponibles semblent faibles par rapport aux technologies IoT présentées précédemment, avec une sensibilité de -140 dBm, la portée théorique en espace libre est de l'ordre de quelques dizaines de kilomètres [49].

Les appareils SIGFOX peuvent transmettre jusqu'à 140 messages par jour à la station de base, avec une charge utile de paquet maximale de 12 octets. Vu la petite taille de trame, SIGFOX encapsule les données utilisateurs directement dans des trames MAC, sans utiliser de couches protocolaires supplémentaires. Ainsi, SIGFOX n'utilise pas le protocole IP pour s'interconnecter à l'Internet, et diffère ainsi des solutions IoT classiques.

Bien que les débits SIGFOX disponibles soient faibles, la longue portée de la technologie et la facilité de déploiement sont des caractéristiques importantes pour de nombreuses applications IoT.

En outre, la réglementation de l'ETSI impose une limitation de capacité : SIGFOX fonctionne dans la sous-bande de 1 GHz, ce qui implique un cycle d'utilisation maximal de 0,1% ou 1% dans la bande 863 – 870 MHz (selon la sous- bande sélectionnée) [45]. Il est à noter que 140 messages par jour correspondent à la limite de 1% de rapport cyclique avec une implémentation SIGFOX à 100 b/s.

1.3.1.2 Couche MAC

PHY Hdr		MAC Hdr		User Data Packet	HM/CRC
Preamble	Frame Sync	End Device ID			
32	16	32		0 to 96	0 16

FIGURE 1.22 – Structure d'un paquet SIGFOX de voie montante (taille en bits).

14. *Differential Binary Phase Shift Keying*. Une modulation où les bits sont représentés par un changement d'état BPSK. Aucun changement est traduit par un 0, un changement se traduit par un 1.

15. cf. Modulation employée dans le BLE.

Voie montante. La technologie SIGFOX se concentre sur les appareils à très faible débit : la charge utile ne peut être encapsulée que dans 5 conteneurs de tailles 1, 4, 8 ou 12 octets. Cela permet de transmettre un maximum de données de 1680 octets par jour. Cette limitation, bien que faible, couvre la plupart des besoins pour les appareils qui transmettent des données telles que la localisation d'un équipement, l'indice de consommation d'énergie, l'alarme ou tout autre type d'information de base d'un capteur, c'est-à-dire la plupart des demandes de l'IoT.

La couche MAC de SIGFOX est simple puisqu'il ne s'agit pas d'un réseau synchronisé : un périphérique qui veut transmettre des données les encapsule dans un paquet SIGFOX (voir la figure 1.22), qu'il émet trois fois, sur des canaux fréquentiels physiques différents sélectionnés aléatoirement, avec un codage de canal différent à chaque émission.

Ce mécanisme de transmission à trois reprises était à l'origine un besoin réel, car aucun trafic descendant n'était utilisé, de sorte que la transmission ne pouvait pas être acquittée. Ainsi, pour s'assurer que la station de base reçoive un paquet, les terminaux se protègent des mauvaises conditions de canal en transmettant trois copies du même paquet.

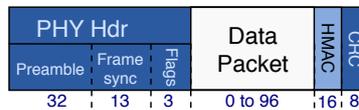


FIGURE 1.23 – Structure d'un paquet SIGFOX de voie descendante (taille en bits).

Voie descendante. Récemment, SIGFOX a ajouté à son réseau la fonctionnalité de transmettre un maximum de quatre messages de huit octets de charge utile, par appareil et par jour. Ces messages de huit octets permettent d'envoyer des données de configuration aux périphériques terminaux si nécessaire, mais on peut optimiser l'autonomie de la batterie des périphériques en n'exploitant uniquement la communication montante si aucune communication bidirectionnelle n'est nécessaire.

Comme les appareils de SIGFOX ne se synchronisent pas avec le réseau, c'est à chacun des terminaux d'interroger la station de base pour obtenir des données sur la liaison descendante.

Pour gérer le trafic de voie descendante, SIGFOX réalise du partage de ressource fréquentielle de la manière suivante : le spectre disponible est découpé en une sous-bande dédiée à la voie montante, et une sous-bande dédiée à la descendante. La station de base transmet ainsi le paquet de liaison descendante sur le même décalage fréquentiel que le paquet de requête, mais dans la sous-bande utilisée pour la liaison descendante. En supposant que le paquet de requête soit envoyé sur une fréquence de -10kHz par rapport à la fréquence centrale de la sous-bande de liaison montante, le paquet de liaison descendante sera alors envoyé sur le canal fréquentiel écarté de -10kHz par rapport à la fréquence centrale de la sous-bande dédiée à la voie descendante. Par conséquent, aucune communication préalable n'est nécessaire pour sélectionner les canaux de liaison descendante. Le format d'un paquet de voie descendante est présenté sur la figure 1.23.

Mode d'opération d'un périphérique au cours du temps. SIGFOX est une technologie asynchrone : les nœuds n'ont pas besoin de se réveiller pour maintenir une synchronisation avec la station de base.

Le modèle de consommation d'énergie de SIGFOX est donc plus simple que celui des autres technologies : les nœuds sont dans l'état *Tx* pendant la transmission des données et dans l'état *Sleep* entre les transmissions.

Notons que SIGFOX définit un temps d'inter transmission pendant lequel le nœud est en mode *Idle*. Cependant, ce temps est négligeable par rapport à la durée de la transmission d'une trame de données.

Grâce à ce mode de fonctionnement asynchrone, bien que la consommation instantanée lors d'une transmission soit élevée et du fait d'une forte puissance d'émission, la durée de vie atteignable par les périphériques SIGFOX reste encore très acceptable : les articles disponibles mettent en avant le fait que certains capteurs peuvent atteindre une durée de vie d'une vingtaine d'années pour trois messages transmis par jour.

1.3.2 LoRa

Pour les applications IoT longue portée, la LoRa Alliance propose, à la manière de SIGFOX, l'exploitation d'une topologie cellulaire avec des stations de base et un système de passerelles, qui reçoivent les paquets des périphériques IoT et relaient les données vers un serveur LoRa *via* une connexion TCP.

Actility (un partenaire de la LoRa) et d'autres partenaires ont permis à 6LoWPAN de s'ajouter à la pile protocolaire LoRa, mais comme la technologie LoRa est plus proche de SIGFOX en termes de profil applicatif visé, pour une comparaison équitable entre les deux, nous présenterons ici la pile LoRa classique qui réalise des transmissions de données utilisateurs directement au-dessus de la couche MAC.

1.3.2.1 Couche PHY

LoRa fonctionne dans la même bande fréquentielle que SIGFOX et 802.11ah, c'est-à-dire la bande sous-1 GHz, et utilise une technologie propriétaire, qui repose sur l'étalement de spectre grâce à des séquences orthogonales (OSCC¹⁶). La modulation utilisée par les systèmes LoRa est le CSS¹⁷, qui améliore considérablement la sensibilité du récepteur et, comme pour la plupart des modulations à étalement de spectre, utilise toute la largeur de bande du canal pour diffuser un signal.

En modulation CSS, l'étalement est obtenu en générant un signal *chirp* qui varie continuellement en fréquence. L'avantage de cette méthode est que les décalages de synchronisation et de fréquence entre l'émetteur et le récepteur sont équivalents, ce qui réduit considérablement la complexité de la conception du récepteur. Cet étalement rend la technologie LoRa robuste au bruit de canal, et insensible aux décalages en fréquence provoqués par l'utilisation d'oscillateur à faible coût.

De plus, la modulation LoRa intègre un mécanisme de correction d'erreur directe (FEC¹⁸), qui ajoute de la redondance dans le but de détecter et corriger des erreurs liées à la propagation du signal radio.

Le facteur d'étalement (ou *Spreading Factor*) permet de décrire l'ampleur de l'augmentation d'occupation de la bande spectrale ainsi que le gain de codage des systèmes à étalement spectre. Dans la technologie LoRa, la notation SF représente le logarithme de base deux du facteur d'étalement, tel qu'il est défini dans la plupart des systèmes à étalement. Le facteur d'étalement effectif correspond donc à une valeur de 2^{SF} . Pour une largeur de bande et un SF donnés, la modulation LoRa conduit aux débits physiques disponibles présentés dans la figure 1.24.

Facteur d'étalement	Débits binaires (kb/s)		Débits binaires (kb/s)		Débits binaires (kb/s)	
SF = 6	Largeur de bande 125 kHz, débit de codage 4/6	7.81	Largeur de bande 250 kHz, débit de codage 4/6	15.63	Largeur de bande 500 kHz, débit de codage 4/6	31.25
SF = 7		4.55		9.12		18.22
SF = 8		2.60		5.21		10.41
SF = 9		1.46		2.93		5.86
SF = 10		0.81		1.63		3.26
SF = 11		0.45		0.90		1.79
SF = 12		0.24		0.49		0.98

FIGURE 1.24 – Tableau récapitulatif des débits disponibles LoRa : débits calculés à partir des informations de Semtech [50].

Ainsi, la modulation LoRa offre des débits de données de 0,25 à 11 kb/s dans la bande de fréquence européenne. 250 b/s correspond à un facteur d'étalement de 12 pour une bande passante de 125 kHz, alors que 11 kb/s résulte d'un facteur d'étalement de 7 et 250 kHz de bande passante.

Les périphériques LoRa peuvent également utiliser la modulation FSK¹⁹, qui leur permet d'atteindre un débit de données plus élevé de 50 kb/s.

Par ailleurs, l'utilisation de la bande fréquentielle sous-1 GHz par LoRa doit, de la même manière que SIGFOX, être en conformité avec la réglementation ETSI : le rapport cyclique d'activité maximal doit être

16. *Orthogonal Sequence Spread Spectrum* : Séquence à étalement de spectre orthogonale

17. Étalement de Spectre à base de *Chirp* — ou *Chirp Spread Spectrum*

18. *Forward Error Correction* : Correction des erreurs de transmission

19. Modulation à changement de fréquence — ou *Frequency Shift Keying*

compris entre 0,1 et 10 %, en fonction de la sous-bande utilisée [45].

1.3.2.2 Couche MAC LoRAWAN

Les appareils terminaux peuvent transmettre sur n'importe quel canal disponible, à tout moment, en utilisant n'importe quel débit de données disponible, tant que les réglementations locales concernant l'utilisation du spectre sont respectées (réglementation de l'ETSI en Europe) et que le terminal change de canal de façon pseudo-aléatoire pour chaque transmission.

LoRa définit trois classes de dispositifs en fonction de leurs capacités énergétiques :

- Les périphériques de classe A permettent des communications bidirectionnelles, la transmission par liaison montante de chaque périphérique étant suivie de deux courtes fenêtres de réception pour la liaison descendante. Le créneau de transmission en voie montante, programmé par le périphérique, est basé sur ses propres besoins de communication (protocole de type ALOHA). La classe A comprend les appareils les plus contraints en énergie, et correspond aux applications qui n'exigent une communication en liaison descendante du serveur que peu de temps après l'envoi d'une transmission en liaison montante (par exemple les trames d'acquiescement). Les besoins de communications en liaison descendante survenant à tout autre moment devront attendre la prochaine liaison montante pour y répondre, à la manière de SIGFOX.
- Les périphériques de classe B permettent l'utilisation d'un plus grand nombre d'emplacements de réception. En plus des fenêtres de réception des terminaux de classe A, les dispositifs de classe B ouvrent des fenêtres de réception supplémentaires de manière périodique. Afin que le terminal puisse ouvrir sa fenêtre de réception aux instants périodiques programmés, il doit maintenir une synchronisation avec la passerelle ou la station de base à laquelle il est rattaché. Pour cela, une balise est envoyée depuis la passerelle de manière périodique, à la manière de la technologie 802.15.4 à balises. Ceci permet au serveur de savoir quand le terminal est à l'écoute, et ainsi autoriser des transmissions descendantes de manière plus fréquente.
- Les périphériques de classe C sont actifs en réception quasiment continuellement : ces créneaux de réception sont uniquement fermés lors de l'émission. Les terminaux de classe C consommeront évidemment plus d'énergie que les terminaux de classe A ou B, mais ils offrent une latence plus faible pour les communications de nature descendantes.

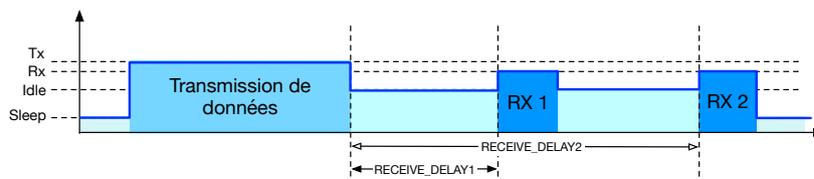


FIGURE 1.25 – Description du comportement d'un nœud LoRa de classe A au cours du temps.

Dans le contexte de transmission de données d'un nœud contraint en énergie, comme le sont la plupart des transmissions au sein des WSN, nous décrivons le fonctionnement d'un périphérique LoRa de classe A au cours du temps à la manière présentée sur la figure 1.25 : un périphérique qui souhaite envoyer des données allume sa radio en mode *Tx* pour la durée de la transmission (dépendante de la longueur de la trame et de la modulation utilisées), et reste ensuite dans un état *Idle* entre la transmission et une première fenêtre de réception (dédié à la réception d'un ACK), puis se positionne dans l'état *Rx* pour la réception d'une trame d'ACK pendant deux créneaux temporels successifs réservés aux communications de voies descendantes, , comme le montre la figure 1.25.

PHY Ovhd		MAC Ovhd	Frame format					User Data Packet	MIC	PLCRC
Preamble	PHDR + CRC	MHDR	Dev@	FCtrl.	FCnt.	FOpts (options)	FPort	...	32	16
8 symb.	32	8	32	8	16	0	1			

FIGURE 1.26 – Structure d'un paquet LoRa (taille en bits).

La figure 1.26 montre la structure du paquet de données LoRa.

La taille maximale des données utilisateur dépend du débit de données. Pour le minimum et le maximum des débits de données disponibles en Europe, la taille maximale des données est respectivement de 59 et 250 octets.

Conclusion

Au sein de ce chapitre, nous avons présenté les acteurs majeurs des technologies de l'IoT, en détaillant le fonctionnement PHY et MAC de chacune des piles protocolaires grâce à l'analyse du temps passé dans les différents états matériels consommants. Nous avons également présenté les choix des différentes technologies pour transporter des données applicatives dans le contexte contraint des WSN.

La diversité des solutions proposées, tant en termes de portée que de débit, rend difficile la comparaison entre toutes ces solutions.

Technologie	Largeur de bande	Topologie principale	Débits PHY (Mb/s)	Portée annoncée	Modulation	Latence à un saut
2.4 GHz :						
BLE 5.0	2 Mhz	Scatternet	0.125, 0.500, 1 ou 2	10 m	GFSK	3 ms
802.15.4	5 MHz	Maillée	0.250	50 m	O-QPSK	20 ms
TSCH	5 MHz	Maillée	0.250	50 m	O-QPSK	10 ms (1 slot)
802.11b PSM 11	22 MHz	En étoile	11	100 m	CCK/DSSS	1.5 ms
<1 GHz :						
802.11ah						
- MCS10	1 MHz	En étoile	0.150	< 1 km	BPSK * 2	N/A
- MCS8	2 MHz	En étoile	7.8	200 m	256-QAM	N/A
- MCS9	16 MHz	En étoile	78	100 m	256-QAM	N/A
EU SIGFOX	100 Hz	En étoile	100 b/s ou 1000 b/s	~ 40 km	DBPSK	2 s
EU LoRa	max 250 kHz	En étoile	0.250 kb/s jusqu'à 11 kb/s	jusqu'à 10 km	LoRa	500 ms

TABLEAU 1.2 – Tableau récapitulatif des technologies présentées.

Le tableau 1.2 permet en effet de résumer la diversité des solutions existantes, mais ne permet en aucun cas de définir un critère de comparaison de ces technologies. Par exemple, le fait d'obtenir des longues portées se paye au prix d'un débit plus faible et/ou d'une plus forte consommation énergétique instantanée lors de la transmission, ce qui rend la comparaison complexe.

Le troisième chapitre, qui constituera la première contribution de ce manuscrit, s'attachera à mettre en place un système de comparaison des technologies, en les mettant sur un pied d'égalité, de manière à mesurer les performances d'une solution donnée vis-à-vis d'une autre.

Discussion

Sommaire

1.1 Solutions courte portée : les WPAN	27
1.1.1 Le standard IEEE 802.15.4	27
1.1.1.1 Couche PHY	27
1.1.1.2 Couche MAC	28
1.1.1.3 Exemple de piles protocolaires basées sur le standard 802.15.4	35
1.1.2 Bluetooth Low Energy	38
1.1.2.1 Couche PHY	39
1.1.2.2 Couche MAC	39
1.1.2.3 Les couches hautes	40
1.2 Solutions LAN	41
1.2.1 802.11b Power Saving Mode (PSM)	41
1.2.2 802.11ah	42
1.2.2.1 Couche PHY	42
1.2.2.2 Couche MAC	43
1.3 Solutions LPWAN	44
1.3.1 SIGFOX	44
1.3.1.1 Couche PHY	44
1.3.1.2 Couche MAC	44
1.3.2 LoRa	46
1.3.2.1 Couche PHY	46
1.3.2.2 Couche MAC LoRAWAN	47

Introduction

L'interopérabilité correspond à la capacité de matériels, de logiciels ou de protocoles différents à fonctionner ensemble et à partager des informations²⁰.

L'apparition de nouvelles plateformes capables de gérer simultanément de multiples technologies [51, 52, 53, 54] permet de repenser les réseaux WSN, en intégrant l'usage de multiples technologies. Ce type d'équipement motive ainsi des travaux de recherche dans le but d'améliorer les performances réseau grâce à la mise en place d'une interopérabilité.

Ainsi, dans ce chapitre, nous déterminons dans une première section la problématique que nous cherchons à résoudre via l'exploitation de ces plateformes multitechnologies. Pour ce faire, nous catégorisons les différents mécanismes MAC dédiés à l'économie d'énergie, et analysons l'impact de ces différents mécanismes sur la consommation d'énergie, en fonction du trafic applicatif. De cette analyse, nous déduisons que la prise en charge de trafics asynchrones contraints en latence requiert actuellement la présence d'un nœud constamment disponible en mode *Rx*, ce qui constitue une dépense énergétique importante.

Nous proposons alors d'exploiter les plateformes multitechnologies pour diminuer la consommation globale liée à la prise en charge de trafics asynchrones rares et contraints en latence, tout en garantissant une faible consommation de chacun des nœuds.

Nous présentons de plus les propositions de travaux d'interopérabilité de la littérature, et donnons également les pistes d'explorations pour mettre en place une telle interopérabilité : l'exploitation des deux modes de communications synchrone et asynchrone au sein d'un réseau de même technologie en utilisant à la phase d'attache du réseau, et la réalisation d'un réseau multitechnologie, une synchrone et l'autre asynchrone.

Ce chapitre répond ainsi aux trois questions suivantes :

- Quelles sont les motivations scientifiques pour l'interopérabilité?
- Quel est l'état de l'art des solutions d'interopérabilité?
- Quelles sont les pistes d'exploration quant à l'interopérabilité?

2.1 Motivations pour l'interopérabilité

2.1.1 La diversité des applications

Comme vu dans le chapitre précédent, les technologies pour l'IoT, et plus particulièrement pour les WSN, sont diverses. La diversité des technologies peut tout d'abord s'exprimer en termes de caractéristiques intrinsèques aux différentes solutions.

Les caractéristiques principales à prendre en compte lors du choix d'une technologie pour répondre aux besoins de l'utilisateur sont :

- La **portée** : Alors que certaines technologies permettent de réaliser des transmissions longue portée, de l'ordre de la dizaine de kilomètres, certaines solutions sont prévues pour opérer dans le cadre des réseaux personnels, c'est-à-dire à faible portée.
- Le **débit** : Un haut débit permet de transporter une grande quantité d'information par intervalle de temps, ce qui peut être nécessaire dans le cas de transmission d'image, par exemple. En revanche, cela implique généralement l'utilisation d'une plus large bande de fréquence, coûteuse par ailleurs. L'IoT, et les WSN en particulier se focalisent plutôt sur l'acheminement de faibles à très faibles débits.
- La **robustesse** : Les différents mécanismes mis en place pour garantir des performances en termes de robustesse aux interférences, ou aux mauvaises conditions de canal, peuvent influencer le choix de la technologie à utiliser. La robustesse d'une technologie peut provenir de la modulation qu'elle utilise, mais aussi du codage exploité. Ces deux paramètres dépendent directement de la couche PHY sélectionnée, caractérisée par les modulations et les codages disponibles. Des mécanismes de couches MAC permettent également d'augmenter la fiabilité des transmissions (acquittements, retransmissions et évitements de collisions par exemple).

20. Définition de larousse.fr

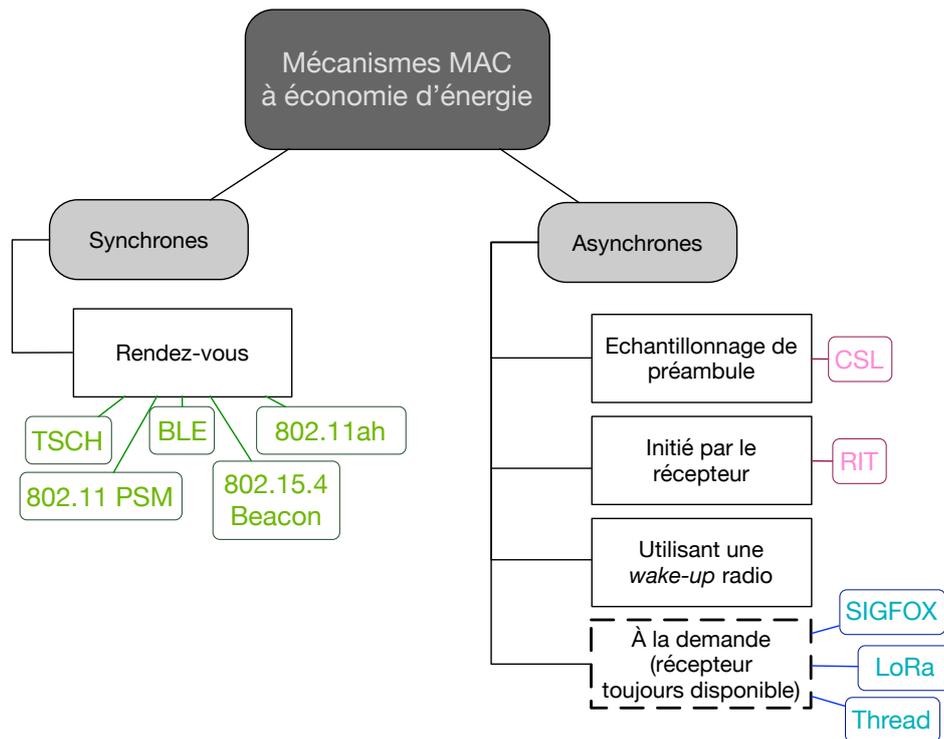


FIGURE 2.1 – Taxonomie des différents mécanismes de DC utilisés au niveau MAC. Figure inspirée des travaux de Carrano *et al.* [55], dans laquelle nous avons rajouté une catégorie en pointillés, ainsi que les technologies présentées au chapitre précédent, en couleurs.

- La **bande de fréquence** : La répartition des ressources fréquentielles est une problématique politique, et dépend donc de la législation en vigueur. Ainsi, suivant la zone de déploiement (Europe ou Etats-Unis par exemple) d'une solution, il reviendra plus ou moins intéressant de déployer l'une ou l'autre des technologies, en fonction du spectre disponible. D'autre part, les canaux fréquentiels utilisés sont en lien direct avec la portée d'une technologie, et doivent donc être pris en compte lors du déploiement.
- Enfin, la **latence** induite par la nature des technologies lors des communications, qui peut être bornée par une latence maximale, garantie ou non, suivant les technologies.

Ces caractéristiques doivent être exploitées pour répondre aux besoins utilisateurs liés au contexte applicatif, défini par les contraintes suivantes :

- la **période** de génération des données applicatives, T_a , inversement proportionnelle à la fréquence de génération de ces données,
- la **quantité** de données à envoyer à chaque génération de données (s_a),
- la **distribution** dans le temps de la génération de données à envoyer,
- et enfin les contraintes en termes de **latence** applicative L_{obj} .

La diversité et la multitude de critères permettant de trier les technologies rendent difficile une comparaison équitable entre toutes les technologies. Cependant, afin de déterminer la pertinence d'une interopérabilité entre deux technologies, il nous paraît essentiel de pouvoir comparer les technologies sur un pied d'égalité, ce que nous ferons à travers une première contribution (Chapitre 3 – Comparaison des standards pour l'IoT).

Cette comparaison nous permettra, entre autres, de confirmer l'hypothèse avancée dans les travaux de Carrano *et al.* [55] : les contraintes applicatives des trafics périodiques, déterminées par le triplet (T_a, s_a, L_{obj}) auront un impact sur la nature du mécanisme MAC utilisé par les nœuds du réseau dans le but d'économiser de l'énergie.

En effet, pour économiser de l'énergie chacun des nœuds va répartir son activité dans le temps en respectant un certain ratio d'activité, le DC (Ratio d'activité cyclique — ou *Duty-Cycle*). Cette répartition dans

le temps est gérée par différents mécanismes MAC, par exemple l'échantillonnage de préambule ou l'utilisation de trames balises, mentionnés dans le chapitre précédent.

Dans l'étude de Carrano *et al.* [55], les auteurs présentent une classification des différents mécanismes de DC exploités au niveau MAC : la figure 2.1 présente la taxonomie proposée. Sur cette figure, nous avons reporté le nom des solutions MAC présentées au chapitre précédent et configurées dans le mode de communication le plus répandu, en les classifiant dans la catégorie adéquate.

Les travaux de Carrano *et al.* se concentrent sur les mécanismes de DC MAC utilisés par l'ensemble des nœuds du réseau, dans le cas de nœuds contraints en énergie.

Cependant, nous avons ajouté en pointillés sur la figure le mécanisme de DC mis en place par les nœuds asynchrones contraints en énergie, qui ne peuvent pas supporter une activité périodique fréquente : ces nœuds ne se réveillent qu'à la demande de l'application, donc leur ratio d'activité dans le temps est imposé par les paramètres applicatifs.

Ce fonctionnement suppose que le réseau, constituant une dorsale de nœuds actifs continuellement, soit constamment disponible en mode *Rx* de manière à ce que les nœuds asynchrones contraints à la périphérie de cette dorsale puissent transmettre sans se soucier de la disponibilité d'un récepteur.

On remarque que les mécanismes de répartition du temps d'activité utilisés par les technologies de IoT présentées au chapitre précédent se concentrent principalement sur le mécanisme "Synchrone avec prise de rendez-vous", avec cinq technologies exploitant ce mode, et le mécanisme "Asynchrone avec l'un des nœuds toujours disponible", utilisé par trois des technologies présentées.

Au-delà de la classification des mécanismes MAC, les travaux de Carrano *et al.* [55] présentent les liens entre les mécanismes de DC mis en place au niveau de la couche MAC et le couple de paramètres applicatifs (T_a, L_{obj}) pour lesquels ces mécanismes sont optimaux en termes de consommation d'énergie globale.

En s'appuyant sur les travaux de Carrano *et al.* [55], nous présentons dans la section suivante les différents types de trafics applicatifs pour lesquels les technologies IoT permettent le meilleur support du point de vue de l'énergie.

2.1.2 Les profils applicatifs des technologies de l'IoT

Comme mentionné dans l'introduction, les trafics applicatifs sont décrits par l'intervalle entre les instants de génération de données applicatives T_a , la répartition de ces instants dans le temps, et la latence applicative ciblée L_{obj} . Dans cette section, nous proposons de catégoriser les trafics applicatifs de la manière suivante :

- Les trafics périodiques de fréquence élevée, c'est-à-dire pour lesquelles on a de faibles valeurs de T_a , contraints ou non en latence.
- Les trafics périodiques de faible fréquence – grandes valeurs de T_a –, ou apériodiques, contraints ou non en latence.

Nous montrerons dans le chapitre de comparaison qu'il existe un intervalle de génération des données seuil $T_{a|seuil}$, pour lequel les trafics applicatifs de période $T_a \leq T_{a|seuil}$ sont ceux pour lesquels les technologies MAC synchrones avec des rendez-vous sont meilleures du point de vue de la consommation énergétique globale.

Pour les trafics apériodiques, et pour les trafics périodiques dont la période applicative est plus élevée que cette valeur $T_{a|seuil}$, nous nous appuyons sur les travaux de Carrano *et al.* [55] pour avancer que le mécanisme MAC le plus économe en énergie correspond à l'utilisation du mode de communication MAC asynchrone, du fait d'un coût trop élevé du maintien d'une association MAC active dans les solutions synchrones utilisant des rendez-vous.

Pour comprendre les limites d'un mécanisme de synchronisation à base de trames échangées à des instants de rendez-vous, la prochaine section présente en détail l'impact de la dérive d'horloge sur le coût de maintien d'une association au niveau MAC.

2.1.2.1 Impact de la dérive d'horloge

Pour tous les mécanismes MAC à prise de rendez-vous, il faut nécessairement que chacun des nœuds soit actif au moment de la date de rendez-vous. Pour ce faire, les horloges de chacun des nœuds d'une com-

munication maintiennent une valeur horaire commune aux deux protagonistes de cette communication : cela représente leur horloge partagée.

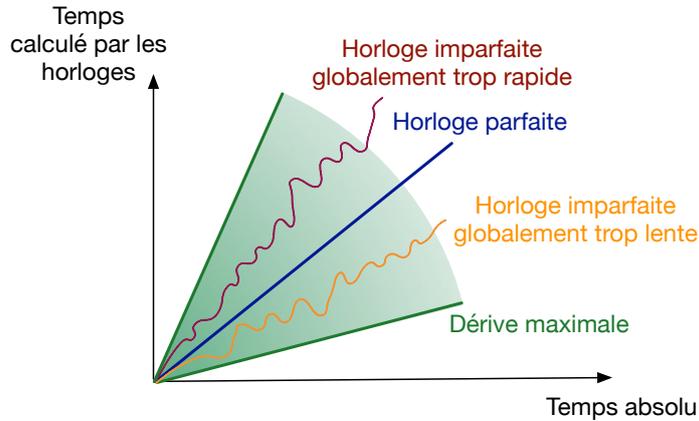


FIGURE 2.2 – Les horloges imparfaites : la dérive.

Imperfection des horloges. Comme les horloges matérielles sont imparfaites, elles ne peuvent calculer précisément l’heure de rendez-vous : en effet, les horloges dérivent au cours du temps, comme le montre l’exemple de la figure 2.2. Cette remarque est d’autant plus vraie dans le contexte des WSN, où, par souci de coût financier, les horloges déployées sur les plateformes matérielles sont moins précises. Sur la figure 2.2, nous représentons le temps absolu réel en abscisse, et le temps calculé par différentes horloges en ordonnée. On peut voir que les horloges peuvent dériver, soit en allant globalement plus vite, soit en allant globalement moins vite que l’horloge parfaite de référence.

Nous supposons donc que les périphériques ont des horloges imparfaites $C(t)$, dont la dérive est bornée par Δ , calculées de la manière suivante :

$$C(t) = \left| \frac{dC(t)}{dt} - 1 \right| \leq \Delta \quad (2.5)$$

La précision d’horloge Δ est indiquée par les fabricants. Une valeur communément utilisée pour les horloges de plateformes WSN est de $\Delta = 40$ ppm. Cela signifie que le fabricant garantit que la dérive d’horloge soit cantonnée dans un cône (représenté sur la figure) qui ne s’éloigne pas de l’horloge parfaite de plus de 40 parties par million, soit une dérive maximale de $40 \mu s$ toutes les secondes.

Dimensionnement d’un intervalle de garde. Cette dérive d’horloge impacte les rendez-vous entre les nœuds. En effet, si un nœud se réveille trop tôt, ou trop tard, par rapport à la date de rendez-vous fixée pour échanger le message de synchronisation (ou le message de donnée), la transmission ne peut avoir lieu.

Comme à la fois l’horloge de l’émetteur et du récepteur peuvent dévier de l’horloge parfaite, la différence maximale entre les horloges des deux périphériques ($\delta_{\max} = \Delta_1 - \Delta_2$) est limitée par 2Δ (en supposant des horloges identiques $\Delta_1 = \Delta_2 = \Delta$) comme le montre la figure 2.3. C’est pourquoi, pour compenser la dérive que peuvent éprouver les deux horloges durant un intervalle sans communications, d_{silence} , il faut dimensionner un intervalle de garde GT tel que :

$$GT \geq 2 \Delta d_{\text{silence}}. \quad (2.6)$$

Ainsi, pour un intervalle d_{silence} tel que $d_{\text{silence}} = 100$ s et une dérive d’horloge $\Delta = 40$ ppm, la durée nécessaire pour compenser la dérive d’horloge est de $GT = 8$ ms.

La figure 2.3 présente les relations de temps entre un émetteur et un récepteur, qui doivent être actifs simultanément pour échanger une trame de données. Le premier cas (cas idéal) suppose des horloges parfaites, de telle sorte que l’émetteur active sa radio en mode Tx exactement quand le récepteur allume la sienne dans le mode Rx . Les trois autres cas illustrent la même opération, en prenant en compte un intervalle de garde GT , tolérant une dérive d’horloge relative maximale de $\delta_{\max} = 2\Delta$, ce qui permet de déterminer la durée de l’intervalle de garde GT (voir l’équation 2.6).

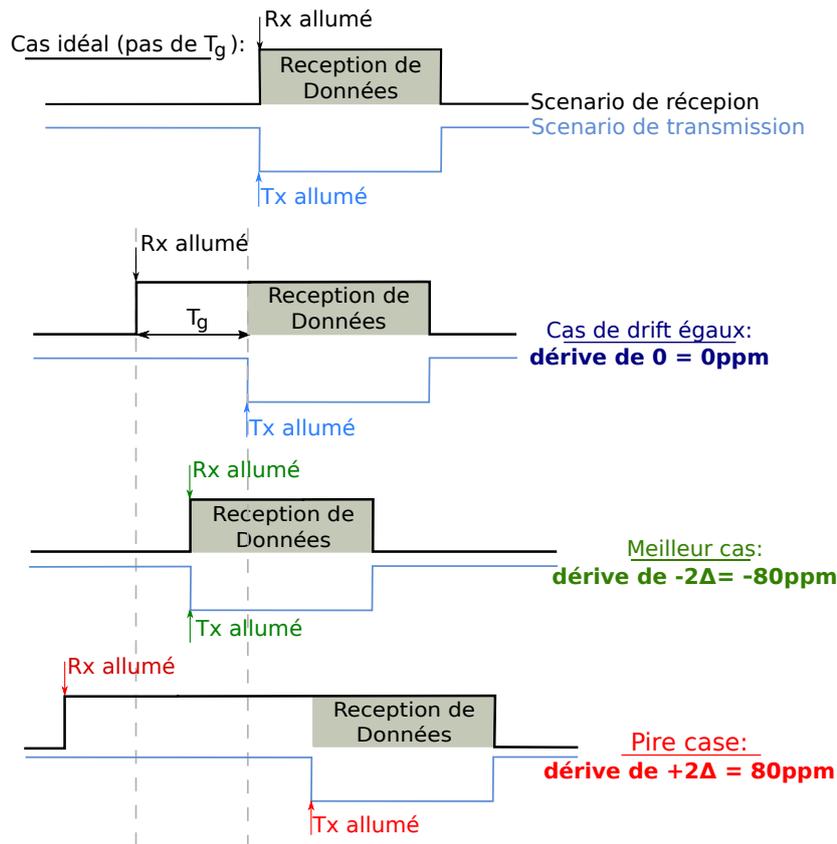


FIGURE 2.3 – Impact de la dérive d’horloge sur la transmission entre deux plateformes.

Les cas présentés correspondent à différentes valeurs de la dérive relative δ des horloges l’une par rapport à l’autre : nous présentons les cas extrêmes où $\delta = -2\Delta$, 0 et $+2\Delta$. On peut observer que le cas $\delta = -2\Delta$ correspond, du point de vue de la consommation d’énergie, à la même situation que le cas idéal.

Ainsi, les technologies synchrones qui fonctionnent avec le mécanisme de rendez-vous, doivent, pour maintenir la synchronisation, dimensionner des intervalles de garde directement proportionnels à la durée passée sans échanges de données d_{silence} .

Nous utilisons un modèle de consommation énergétique basé sur le temps passé dans les états consommateurs, décrit par l’équation 1.1 (page 27). Nous supposons également que le mode *Rx* consomme plus que le mode *Sleep* étant donnée la définition de ces modes (cf. tableau 1.1, page 27), ce qui implique que la durée de l’intervalle de garde GT qui n’est pas utilisé pour la réception d’un paquet représente une surconsommation énergétique.

L’échange périodique de messages de synchronisation. Du fait de la prise en compte de la dérive d’horloge, on comprend que, si la période des transmissions est trop faible, les horloges vont subir des dérives trop importantes, nécessitant la mise en place de très grands intervalles de garde, les transmissions effectives de données pouvant avoir lieu pendant toute la durée de l’intervalle. L’incertitude sur les instants de transmission liée à la prise en charge des dérives d’horloges altère le déterminisme des solutions proposées, rendant complexe la garantie d’une certaine latence pour un trafic à plusieurs sauts, par exemple. Ainsi, pour éviter que l’incertitude sur la date de transmission soit trop élevée, les mécanismes MAC synchrones utilisant des rendez-vous imposent un échange périodique d’information de synchronisation, de période T_{syn} , de manière à réduire la durée d_{silence} , et ainsi diminuer la longueur de l’intervalle GT .

L’envoi de ces informations de synchronisation peut être couplé avec l’envoi de données, réduisant ainsi la proportion d’énergie dépensée pour la synchronisation (on met en commun les surcoûts protocolaires). Ainsi, dans le cas de trafic périodique, on a $d_{\text{silence}} = \min(T_a, T_{\text{syn}})$.

Détermination de la valeur $T_{a|\text{seuil}}$. Si la période du trafic applicatif T_a est au plus égale à celle des messages de synchronisation T_{syn} , il est possible de réaliser cette opération à chaque fois qu’il est néces-

saire d'envoyer des informations de synchronisation, ce qui diminue fortement le surcoût énergétique lié à l'envoi de trames dédiées au maintien de la synchronisation.

C'est pourquoi le maintien d'une association MAC devient intéressant lorsque la période du trafic est plus faible ou égale à celle requise pour les messages de synchronisation ($T_a \leq T_{\text{syn}}$) : en effet, dans ce cas, la fréquence des transmissions applicatives vient réduire la durée d_{silence} , et donc celle de l'intervalle de garde GT , réduisant ainsi le niveau d'énergie perdue pour l'écoute passive durant l'intervalle de garde.

Notons que dans le cas de la technologie 802.15.4 en mode *beacon*, l'intervalle entre les messages de synchronisation T_{syn} avoisine les 4 min, ce qui, dans le cas d'un trafic applicatif de faible fréquence, correspond à un intervalle de garde de 192 ms. Cette durée passée en mode *Rx* correspond à de l'énergie dépensée sans pour autant servir directement la réception d'un paquet, si l'on considère le pire cas de dérive des horloges (cf. dernière ligne de la figure 2.3).

Ainsi, comme le confirmeront également les résultats de notre étude comparative relatifs à la dérive d'horloge, on comprend que le mode de communication synchrone avec prise de rendez-vous est approprié au trafic dont la période du trafic T_a est relativement faible (fréquence élevée), de manière à garder un coût énergétique lié au maintien de la synchronisation relativement faible.

Les travaux de Carrano *et al.* considèrent un intervalle seuil entre les occurrences de trafic applicatif pour laquelle les modes de communication synchrones sont intéressants du point de vue de la consommation globale, $T_{a|\text{seuil}}$, pour une période applicative T_a inférieure à la seconde, c'est-à-dire $T_a|\text{seuil} < 1$ s.

Nous nuancerons cette valeur en expliquant que $T_{a|\text{seuil}}$ n'est pas seulement influencée par le coût de maintien de synchronisation, mais également par le coût d'une réalisation de cette même communication de manière asynchrone.

2.1.2.2 Trafics périodiques tels que $T_a \leq T_{a|\text{seuil}}$

Trafics peu ou non-contraints en latence : grandes valeurs de L_{obj} . Pour les trafics applicatifs non contraints en latence et de périodes applicatives plus faibles que la valeur $T_{a|\text{seuil}}$, l'utilisation d'un mode de communication synchrone utilisant des rendez-vous est recommandée. En effet, en l'absence de contrainte de latence, seuls les mécanismes mentionnés ci-dessus (dimensionnement d'un intervalle de garde et période du trafic applicatif) viennent impacter la consommation énergétique globale.

Contrainte de latence : L_{obj} faible. Pour les trafics applicatifs de périodes plus faibles que $T_{a|\text{seuil}}$ contraints en latence, le mode de synchronisation exploitant des rendez-vous peut également représenter le minimum de consommation énergétique, si le concepteur du réseau paramètre correctement les instants de réveil des nœuds.

En effet, dans le but de réduire la latence d'une communication entre deux nœuds, il faut que l'émetteur de la transmission soit actif dès que la donnée est générée, et que le récepteur soit en activité radio au même instant. Ainsi, si la génération de données applicatives est périodique, il est possible d'organiser les périodes d'activité radio des nœuds à la suite des périodes d'activité des nœuds pour la génération de données. Cette logique peut également être suivie en ordonnant intelligemment les périodes d'activités des nœuds lors de communications multisauts : les travaux de Juc *et al.* [56] et de Keshavarzian *et al.* [57] proposent une organisation des périodes d'activité des nœuds dans le temps pour répondre aux contraintes de latence d'un trafic applicatif à destination du coordinateur du réseau, ce qui correspond à la majorité du trafic de l'IoT.

Ainsi, le mode de communication synchrone qui utilise un système de rendez-vous est approprié pour les trafics applicatifs périodiques dont le paramètre applicatif T_a est faible par rapport à $T_{a|\text{seuil}}$, quelle que soit la valeur du paramètre L_{obj} .

2.1.2.3 Trafics a périodiques ou périodiques tels que $T_a \geq T_{a|\text{seuil}}$

Comme nous l'avons vu au paragraphe précédent, le maintien d'une association MAC active requiert l'envoi périodique d'information de synchronisation.

Dans le cas des trafics a périodiques, ou dans le cas des trafics applicatifs tels que $T_a \geq T_{a|\text{seuil}}$, l'envoi de ces informations de synchronisation ne peut pas être couplé avec l'envoi des données, car ces dernières sont trop peu fréquentes vis-à-vis de la période requise pour le maintien de la synchronisation T_{syn} .

De ce fait, le surcoût énergétique représenté par le maintien de la synchronisation est trop élevé, et on comprend alors le choix de se tourner vers des modes de communication asynchrones.

Non contraints en latence. Si les trafics applicatifs ne possèdent pas de contrainte de latence particulière, c'est-à-dire que le but est de transmettre l'information, mais qu'il n'est pas nécessaire de garantir la durée d'exécution de cette transmission. Les mécanismes MAC de transmission asynchrone reposant sur les modes d'échantillonnage de préambule, ainsi que le mode initié par le récepteur sont appropriés pour acheminer de tels trafics asynchrones : ils permettent de ne pas dépenser d'énergie pour le maintien d'une association MAC et présente l'avantage de gérer les trafics de nature asynchrones sans pour autant reposer sur l'exploitation d'un nœud constamment actif en mode *Rx*.

En effet, pour ces deux modes, en l'absence de transmission, chacun des nœuds du réseau se réveille de manière périodique et indépendante d'un nœud à l'autre, ce qui permet d'économiser globalement de l'énergie.

Lors de la génération d'une donnée, le nœud qui vient de générer cette donnée se rend disponible à un instant donné, et peut soit attendre le signal de l'autre nœud, pour garantir une superposition d'activité, soit s'il connaît les instants d'activité du destinataire envoyer son paquet à cet instant.

Les mécanismes dédiés à la faible consommation d'énergie du standard 802.15.4, CSL et RIT présentés au chapitre précédent, sont des exemples de tels mécanismes MAC, comme le montre la figure 2.1.

Contraints en latence. Pour les trafics possédant une contrainte de latence forte, c'est à dire une valeur faible de L_{obj} , et un grand intervalle entre les générations de données applicatives T_a , ou de nature apériodique, les modes de communications asynchrones présentés précédemment ne conviennent pas.

En effet, soit ces derniers vont consommer trop d'énergie du point de vue du nœud émetteur des données pour maintenir une synchronisation (dans le cas des technologies synchrones), soit une latence va s'appliquer du fait de la nature asynchrone des périodes d'activité des nœuds (dans le cas des mécanismes de transmission reposant sur l'échantillonnage de préambule ou de transmission initiée par le récepteur).

Ainsi, les contraintes de latence sur des trafics de type asynchrone ont été résolues en mutualisant la consommation d'énergie en un point du réseau qui servira de centralisateur pour récupérer les données des périphériques lui étant rattaché, à la manière des technologies SIGFOX et LoRa. Une telle disposition suppose un récepteur constamment actif et disponible en mode *Rx*, dans le but d'obtenir une latence la plus faible possible.

Ce mode de communication est également exploité au sein des réseaux Thread, où le réseau est constitué d'une dorsale constamment disponible, alors que les feuilles périphériques sont asynchrones et transmettent leurs données dès que possible.

2.1.3 Nos objectifs d'interopérabilité

Nous remarquons que la contrainte de latence pour les trafics asynchrones apériodiques (ou de périodes très larges) implique, parmi les solutions proposées dans l'IoT, soit la disponibilité continue d'un récepteur (à la manière de LoRa ou SIGFOX), soit une infrastructure synchronisée, globalement coûteuse pour le nœud asynchrone contraint en énergie.

Dans les travaux de cette thèse, nous souhaitons exploiter l'interopérabilité pour proposer une solution qui permettrait de gérer des trafics asynchrones contraints en latence tout en utilisant les mécanismes de ratio d'activité DC utilisés dans les technologies synchrones pour remplacer le fonctionnement d'un dispositif constamment disponible.

Ainsi, notre problématique peut se définir de la manière suivante : comment exploiter l'interopérabilité pour garantir la réception d'un trafic asynchrone avec une latence faible tout en maintenant une consommation d'énergie globale faible.

2.2 État de l'art des solutions d'interopérabilité

Des travaux ont déjà été menés quant à la problématique globale de l'interopérabilité, et se situent à différents niveaux du modèle OSI (Interconnexion de systèmes ouverts — ou *Open Systems Interconnec-*

tion). Dans cette partie, nous présentons les travaux existants ainsi que leur positionnement, dans le but de préciser notre cadre de travail vis-à-vis de cet état de l'art.

2.2.1 Interopérabilité en exploitant un protocole commun

Le modèle OSI décrit un système de communication en couches d'abstraction, présentant chacune un des aspects à mettre en place pour permettre une interconnexion : ainsi deux objets exploitant le même protocole pour répondre à la problématique de niveau X peuvent, par construction du modèle, s'interconnecter au niveau X .

Cela signifie que la plupart des objets de l'IoT, s'ils ne sont pas directement compatibles, peuvent communiquer jusqu'à une passerelle de niveau X , qui se chargera de traduire dans une autre technologie de plus bas niveau le message à transférer au destinataire.

Dans l'Internet classique, de telles passerelles sont nommées des *routeurs* et s'interconnectent au niveau de la troisième couche, c'est à dire la couche *réseau*, grâce à l'utilisation du protocole IP [58], qui facilite l'interconnexion de réseaux de technologies de couches basses différentes (802.11 et Ethernet par exemple).

De tels routeurs peuvent se schématiser de la manière présentée sur la figure 2.4 : deux piles basses (MAC/PHY) distinctes pour une seule couche IP commune au sein d'un même équipement matériel.

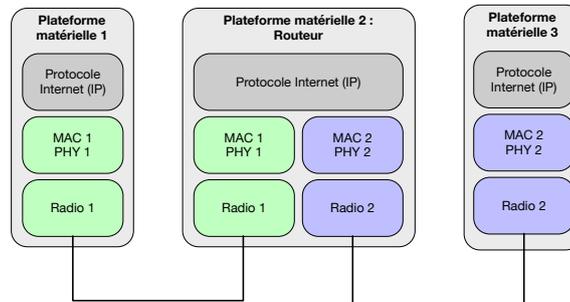


FIGURE 2.4 – Interopérabilité de deux réseaux au niveau 3, la couche IP.

Des exemples de ces passerelles sont désormais courants sur le marché de l'IoT [52, 53, 54], car elles permettent de réaliser une interconnexion de niveau 3 sans aucune modification des piles utilisant le protocole IP existantes.

Une telle manière de réaliser interopérabilité constitue une première solution d'interopérabilité, mais nécessite cependant le déploiement physique d'un routeur capable d'exécuter deux piles basses (PHY/MAC) en parallèle, consommatrices en énergie. Dans l'Internet classique, ces routeurs sont alimentés par le réseau électrique.

Dans le contexte des réseaux de capteurs, l'alimentation de tels routeurs représente un coût élevé de déploiement de l'infrastructure (raccordement des routeurs au réseau électrique), et représente également une forte contrainte en termes de déploiement topologique : il faut positionner le routeur à un endroit stratégique du réseau, de manière à ce qu'un maximum de périphériques puissent être à sa portée.

2.2.2 Partage de la ressource physique

La cohabitation de deux couches MAC distinctes

Afin de réduire la consommation des plateformes permettant l'interopérabilité, de nombreuses propositions mutualisent la ressource physique de la radio en utilisant des radios capables d'exécuter différentes couches PHY, reconfigurables à la demande.

La figure 2.5 présente la schématisation du fonctionnement de ces plateformes au regard des couches basses du modèle OSI. L'idée d'une telle solution est de simplifier l'architecture matérielle – une seule radio – tout en réalisant une interopérabilité multiprotocole MAC.

Comme les modulations exploitées par le BLE et le 802.15.4 sont proches, les plateformes IoT exploitant la même radio matérielle et capables de réaliser à la fois les technologies BLE, Zigbee et/ou Thread, sont courantes [59, 60, 61].

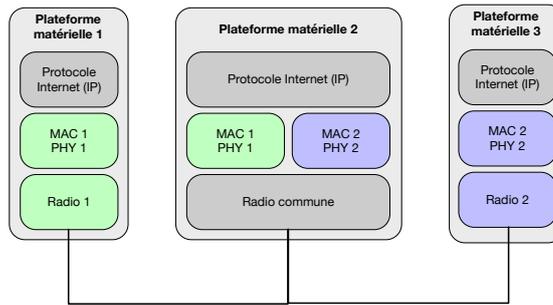


FIGURE 2.5 – La mise en commun de la ressource physique.

De plus, l'essor de la SDR²¹ a permis le développement de telles infrastructures, puisque la reconfiguration logicielle de l'interface radio à la demande, rend possible la mutualisation d'une ressource radio physique, ainsi que son exploitation tout en faisant coexister des couches PHY et MAC distinctes, pour ainsi créer des plateformes multistandards.

Une telle approche pose tout de même les contraintes de partage de la ressource matérielle que constitue la radio, au sein d'une même plateforme matérielle, ce qui peut impliquer un irrespect des relations temporelles des événements des deux couches MAC concurrentes au sein du même nœud.

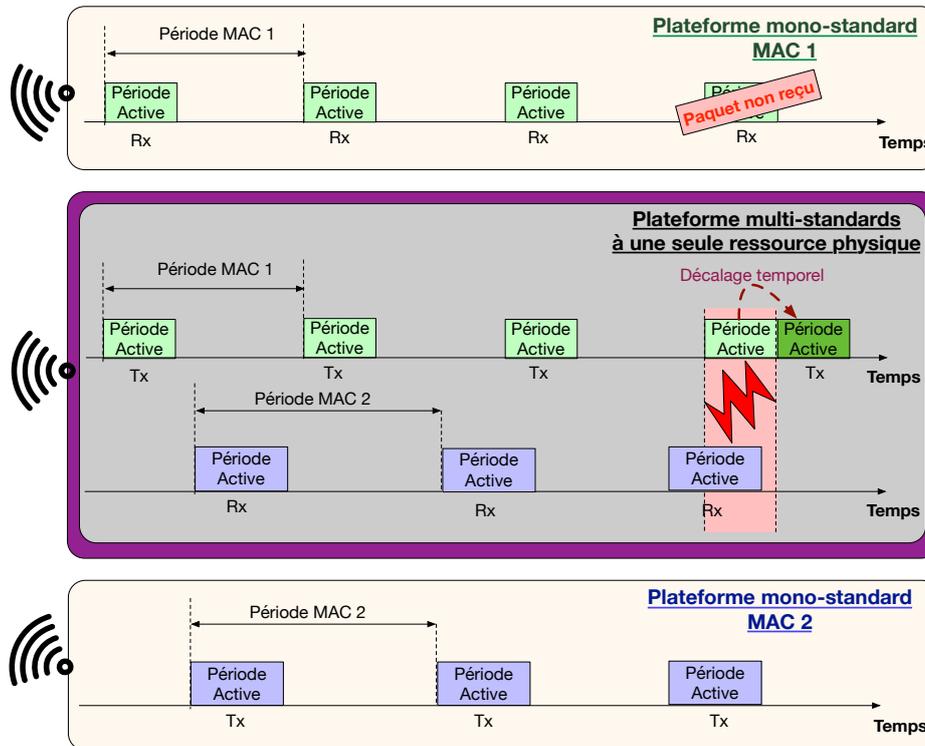


FIGURE 2.6 – Collision de deux événements MAC distincts au sein d'une unique plateforme matérielle impliquée dans deux communications de technologies différentes avec deux destinataires différents.

La figure 2.6 donne un exemple de ces collisions d'obligations temporelles internes, dans le cas de deux couches MAC de nature périodiques, de périodes différentes, s'exécutant en parallèle au sein d'un unique nœud multistandard (le nœud du milieu sur la figure 2.6).

Optimisation conjointe multiMAC

Dans le but de gérer ces contraintes temporelles entre les protocoles dans le cas d'une ressource matérielle commune, la réalisation d'une optimisation conjointe au niveau de la couche MAC, entre deux technologies de l'IoT apparaît nécessaire.

21. *Software Define Radio* ou Radio Logicielle, c'est-à-dire un émetteur/récepteur radio réalisé principalement par une architecture logicielle

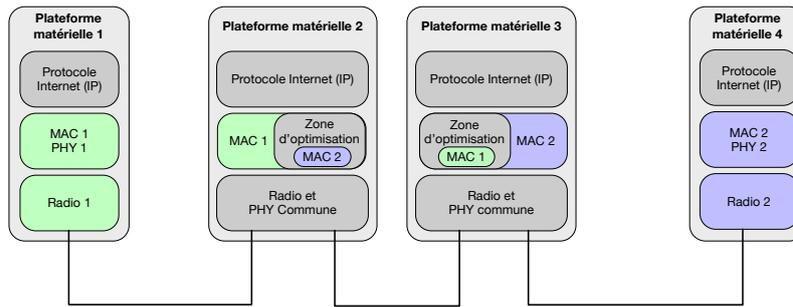


FIGURE 2.7 – Pile protocolaire d'une plateforme qui réalise une optimisation conjointe multiMAC.

Ainsi, une interopérabilité de niveau MAC implique de définir le comportement et les ajustements liés à la zone d'interopérabilité au niveau de la couche MAC, c'est-à-dire les zones d'optimisation grisées de la figure 2.7 : le but n'est plus de seulement faire cohabiter les deux protocoles MAC de technologies différentes, mais de les faire fonctionner conjointement, en exploitant les temps d'inactivité garantis de l'un des protocoles pour y insérer des créneaux d'activité du second.

Nous proposons donc de réaliser une optimisation conjointe des performances et de la consommation énergétique de deux protocoles MAC de natures différentes, synchrone (de type TSCH) et asynchrone (de type Thread), pour répondre à la problématique établie dans la section précédente : acheminer un trafic asynchrone contraint en latence grâce à un réseau de nature synchrone qui répartit son activité dans le temps de manière périodique. Une telle interopérabilité imposera, comme suggéré par la figure 2.7, une optimisation au niveau de chacun des protocoles existants : TSCH et Thread.

2.2.3 Propositions d'interopérabilité MAC existantes

L'ensemble des travaux existants en termes d'optimisation MAC se présente de la façon suivante : des propositions de structures agissant en tant que surcouche MAC, qui permettent de décrire, définir et faire cohabiter différentes couches MAC, comme le montre la figure 2.8. Les couches MAC utilisées par les surcouches proposées ne sont pas spécifiées, car ces travaux se veulent génériques.

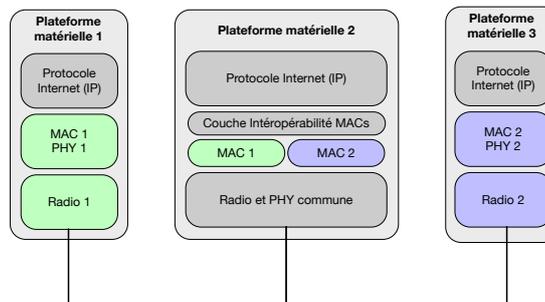


FIGURE 2.8 – Pile protocolaire d'une plateforme qui met en place une surcouche d'optimisation MAC.

MetaMAC [62], et plus tard Multi-MAC [63], proposent la réalisation d'une surcouche MAC qui pilote l'utilisation de différentes couches MAC sous-jacentes en fonction du contexte (qualité de lien radio, densité du trafic, ratio de trames en erreur par exemple), comme présenté sur la figure 2.8. Les travaux de Gul [64] présentent différentes optimisations pour équipements multistandards SDR en s'appuyant sur les travaux de Farago *et al.* [62].

Ces travaux supposent cependant uniquement un trafic dont la probabilité d'apparition suit une loi de Poisson, et ne prennent pas en compte l'existence d'un trafic périodique et déterministe, par exemple le suivi de la température d'une pièce au cours du temps.

A l'inverse, la proposition de Kim *et al.* [65], se focalise uniquement sur la gestion de trafics applicatifs de nature périodiques, et ne traite pas des trafics de nature aperiodiques, par exemple un signal d'alarme.

PluralisMAC, proposé par De Mil *et al.* [66], permet de gérer des besoins de trafics applicatifs différents, en découpant le temps en périodes dédiées à différents types de trafic périodiques : une première zone

temporelle gère les trafics contraints en latence, une seconde ceux contraints en consommation d'énergie par exemple. Les travaux de Zhang *et al.* [67] et Ansari *et al.* [68] reposent sur le même principe, même si ces solutions sont plus flexibles, car le changement de mode de communication n'est pas défini à l'avance et peut s'organiser à la volée.

Guo *et al.* [69] ont proposé une prise de décision verticale et multicritère de reconfiguration MAC du réseau, dans le but de minimiser l'impact de la collision de timings, tels que présentés sur la figure 2.6.

De telles architectures reconfigurables sont certes plus efficaces que de rester dans un seul mode de communication MAC, mais elles permettent uniquement de reconfigurer le fonctionnement d'une couche MAC, et l'on retombe ainsi dans le cas d'une interopérabilité avec deux couches MAC cloisonnées, comme le montre la figure 2.5, avec les problématiques mentionnées sur la figure 2.6.

Ainsi, les solutions existantes ne permettent pas d'optimiser conjointement deux couches MAC de natures différentes : soit l'on suppose un trafic périodique, soit un trafic Poissonien, soit l'on n'optimise pas conjointement, mais on fait simplement cohabiter ces deux types de trafics.

Contrairement à ces travaux, nous proposons de réaliser une interopérabilité en optimisant conjointement deux couches MAC à la manière des zones MAC grisées sur la figure 2.7.

Ainsi, nous proposons de prendre en charge deux types de trafics : périodique et aperiodique, tout deux potentiellement contraints en latence, en travaillant avec une unique plateforme matérielle multitechnologie, et en optimisant conjointement l'exécution de deux couches MAC différentes dans le but d'économiser globalement l'énergie consommée par le réseau.

2.3 Pistes d'exploration

Nous avons défini notre problématique comme l'exploitation de l'interopérabilité pour acheminer un trafic applicatif envoyé en mode de communication asynchrone en respectant une faible latence, et une faible consommation d'énergie globale.

Nous remarquons que, lors de la phase d'attache au réseau, le nœud qui tente de rejoindre un réseau synchrone déployé se trouve dans une situation similaire à celle d'un nœud asynchrone, vis-à-vis du réseau synchrone déployé : bien que capable de maintenir une synchronisation par la suite, l'état du nœud au réveil ne lui permet pas de connaître les paramètres temporels du réseau (période de l'ordonnancement, ressources allouées).

On peut alors faire le parallèle entre un nœud d'une technologie asynchrone qui souhaite envoyer un ou des paquets de données à un cœur de réseau synchrone, et un nœud d'une technologie synchrone non encore synchronisé, qui souhaite envoyer un paquet de demande d'allocation de ressources, comme le montre la figure 2.9.

Les concepteurs du protocole BLE ont d'ailleurs remarqué la similarité des processus, puisque le mode de communication non connecté du BLE, c'est à dire le mode de communication asynchrone de BLE, exploite exactement le mécanisme utilisé par rejoindre le réseau pour envoyer des trames de données de manière asynchrone.

Pendant, dans le cas d'un réseau BLE, le *Master* de la communication est considéré comme disponible pour la communication asynchrone, de la même manière que les routeurs Thread sont continuellement en mode *Rx* pour recevoir les trafics provenant des périphériques feuilles asynchrones aux alentours. Une telle disponibilité continue représente un coût énergétique que nous aimerions diminuer en exploitant la technologie TSCH pour économiser de l'énergie pour l'ensemble des nœuds du réseau.

Ainsi, comme les nœuds d'un réseau TSCH permettent d'obtenir un réseau multisaut pour une faible consommation à chacun des nœuds, nous proposons deux solutions adaptées à TSCH pour répondre à la problématique d'un trafic asynchrone traité avec une faible latence et une faible consommation énergétique globale :

- Dans un premier temps, nous proposons de réaliser une interopérabilité multimode (synchrone et asynchrone), en exploitant la phase d'attache à un réseau TSCH pour transmettre des informations asynchrones avec une faible latence et un faible coût énergétique, à la manière de BLE.
- Dans un second temps, nous proposons de réaliser une interopérabilité multitechnologie Thread et TSCH, c'est-à-dire une interopérabilité de mode de communication MAC synchrone et asynchrone contraint en latence, en exploitant les mécanismes mis en place pour l'interopérabilité multimode.

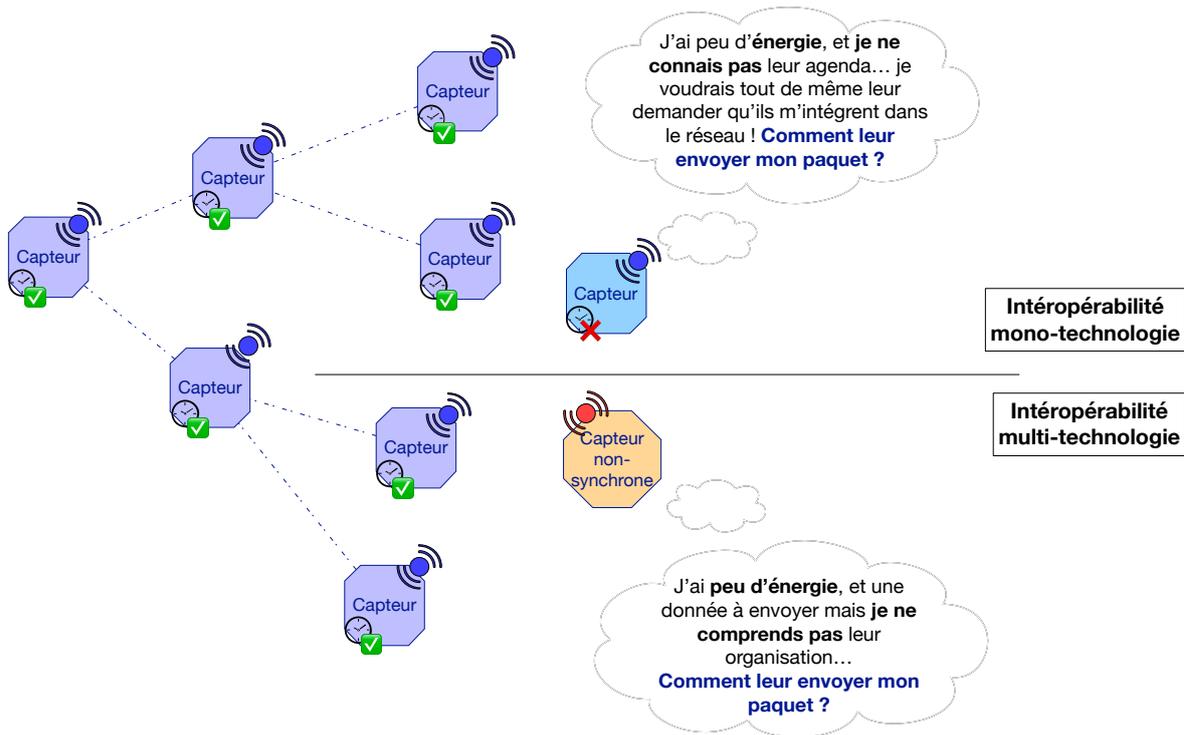


FIGURE 2.9 – Interopérabilité synchrone/asynchrone : mono et multitechnologie.

2.3.1 Exploitation de la phase d'attache

Dans cette partie, nous proposons d'exploiter un mode de communication reposant sur l'utilisation successive de périodes d'association et de désassociation au réseau, dans le but d'apporter un mode de communication asynchrone avec une faible latence dans les réseaux TSCH.

Cependant, les procédures d'attachement proposées dans le standard 802.15.4-2015 font partie de la phase la plus consommatrice de la vie du capteur. Ainsi, l'intérêt d'une telle procédure est à mettre en relation avec le coût énergétique de maintien de la synchronisation : en effet, si le coût de la procédure d'attache est trop élevé, il est peut-être plus intéressant du point de vue de la consommation d'énergie globale de ne pas se désassocier.

Comme nous l'avons mentionné à la page 27 dans l'équation 1.1, nous considérons un modèle de consommation d'énergie pour lequel l'énergie consommée par une plateforme IoT est directement proportionnelle au temps passé dans les états consommateurs, résumés dans le tableau 1.1 de la page 27.

Ainsi, pour mesurer le coût énergétique du maintien d'une association MAC active, ainsi que le coût d'attache au réseau, nous utiliserons comme métrique le temps passé dans l'état Rx pendant ces deux phases opératoires, représentatif de l'énergie consommée à un facteur multiplicatif près, la puissance instantanée consommée par la plateforme dans l'état Rx .

Nous considérons ici le cas de trafics périodiques de périodes variables, T_a , et évaluons l'intérêt de réaliser cette procédure de désynchronisation du réseau pour se resynchroniser au besoin.

Coût de la procédure d'attache au réseau. La figure 2.10 rappelle le schéma standard de la procédure d'attache au réseau pour un nœud du réseau 802.15.4 en mode TSCH [9] : un nœud qui souhaite rejoindre le réseau se réveille et balaye en continu un des canaux radio jusqu'à recevoir une trame de balise : les trames d'EB (Balise améliorée — ou *Enhanced Beacon*). La durée entre l'instant de réveil du nœud et la réception effective d'une trame de synchronisation, directement proportionnelle à l'énergie dépensée durant la phase d'attache au réseau, est notée d_{join} . D'après la figure 2.10, d_{join} vaut :

$$d_{\text{join}} = T_{\text{adv}} - t_0. \tag{2.7}$$

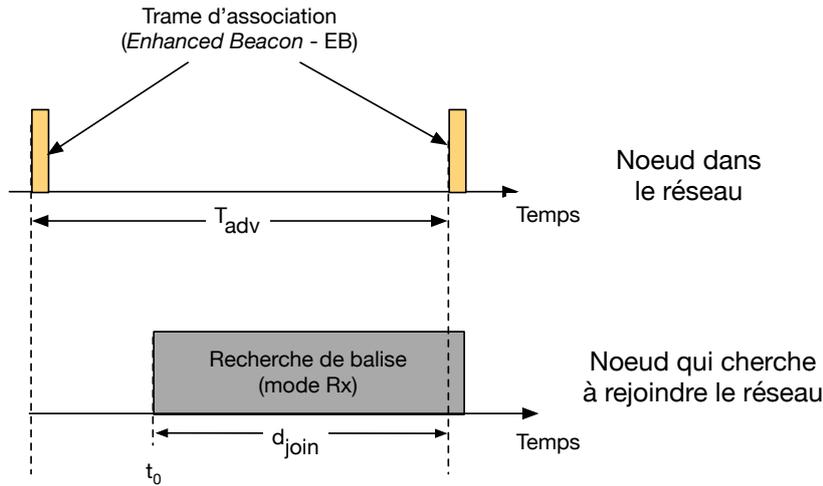


FIGURE 2.10 – Procédure classique d’attache au réseau.

Ainsi, dans le pire cas, c’est-à-dire quand t_0 se produit juste après le début de la première trame d’EB et que t_0 devient négligeable devant T_{adv} , on obtient :

$$d_{\text{join|max}} = T_{adv} \quad (2.8)$$

De plus, en supposant que le décalage entre la date d’envoi de la trame de synchronisation et le réveil du nœud est tiré aléatoirement et est uniformément réparti, car le nœud qui se réveille n’a pas de connaissance *a priori* de l’ordonnancement du réseau, on obtient un délai moyen d’attache au réseau de :

$$d_{\text{join|moy}} = \frac{T_{adv}}{2} \quad (2.9)$$

Il est à noter que par souci de simplicité, nous avons représenté les comportements des nœuds sur un seul canal fréquentiel, le standard ne préconisant pas actuellement de mode spécifique à l’utilisation de multiples canaux. Ainsi, si l’on suppose que N_c canaux sont utilisés par le réseau, le comportement peut être extrapolé dans la dimension fréquentielle, et le temps d’attache au réseau par la procédure classique vaut :

$$d_{\text{join|multicanal}} = d_{\text{join|monocanal}} \times N_c \quad (2.10)$$

Coût de maintien de la synchronisation. Le coût pour maintenir une synchronisation dépend quant à lui de la fréquence d’envoi des données. Nous avons vu que les technologies synchrones requièrent un échange périodique de messages dans le but de maintenir une association active, et ainsi pouvoir considérer cette association comme perdue en absence de réception de ces paquets. Ces messages de synchronisation sont des messages de KA (Maintien vivant — ou *Keep-Alive*), et l’intervalle entre deux KA sera noté T_{syn} .

La figure 2.11 présente deux cas de trafics périodiques, l’un dans le cas de trafic applicatif de période élevée, et l’autre dans le cas de trafic applicatif relativement rare.

Sur cette figure, nous avons noté d_{on} la durée d’activité d’un nœud, c’est-à-dire la durée passée dans un état matériel différent de *Sleep*.

Nous avons de plus rajouté les deux variables suivantes : le temps passé en activité pour la transmission de données d_{data} et celui passé en activité pour l’échange d’information de synchronisation d_{syn} . Ainsi, sur la figure 2.11, nous retrouvons deux cas vis-à-vis du surcoût énergétique induit par les messages de synchronisation.

Lorsque $T_a < T_{\text{syn}}$, c’est-à-dire quand la période des messages de données est plus faible que l’intervalle nécessaire de KA pour maintenir une association MAC active (cf. figure 2.11a), le surcoût en temps d’activité de la radio engendré par l’information de synchronisation est négligeable devant le temps passé avec radio active liée à l’envoi de données applicatives, soit

$$d_{\text{syn}} \ll d_{\text{data}} \quad (2.11)$$

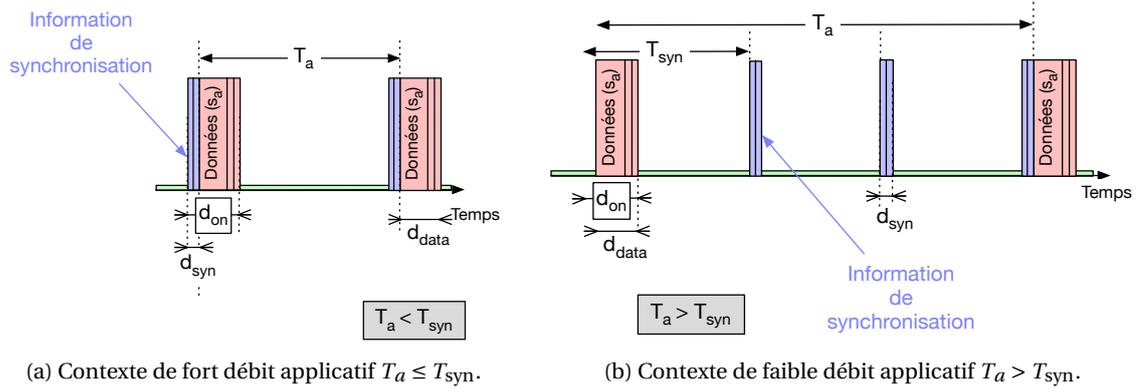


FIGURE 2.11 – Comparaison du coût de la synchronisation pour des messages applicatifs générés à des intervalles différents.

Dans ce cas, on peut dire que le maintien d'une association active n'implique qu'un faible surcoût, que l'on peut négliger.

Lorsque $T_a > T_{syn}$, et c'est le cas des trafics que nous souhaitons rendre interopérables avec le réseau synchrone, il est nécessaire d'ajouter des périodes d'activité de la radio uniquement dans le but de maintenir une association MAC active, comme le montre la figure 2.11b.

En supposant que, quelque soit l'instant de génération des données applicatives, on fait correspondre l'envoi de ces dernières avec la transmission d'informations de synchronisation en "décalant" l'envoi de ces données, on peut déduire le nombre de périodes d'activité dédiées à la synchronisation pure N_{syn} de la manière suivante :

$$N_{syn} = \lfloor \frac{T_a}{T_{syn}} \rfloor - 1 \quad (2.12)$$

On notera qu'on enlève une période au ratio, car il est possible de négliger le surcoût d_{syn} dans le cas de transmission de données (cf. Équation 2.11).

Ainsi, le surcoût, en temps d'activité, lié au maintien de la synchronisation $d_{synCost}$ sera :

$$d_{synCost} = N_{syn} \times d_{syn} \quad (2.13)$$

De plus, la prise en compte des dérives d'horloge vient augmenter le temps d_{syn} , ce qui augmentera encore ce surcoût. L'équation 2.13 vient confirmer ces propos, puisque le surcoût augmente lorsque N_{syn} augmente, mais aussi lorsque d_{syn} augmente.

Dans le but de diminuer ce surcoût de consommation lié au maintien de la synchronisation, nous proposons d'exploiter un mode de communication non synchronisé, comme le propose BLE : les nœuds qui ont très peu de trafic à envoyer se désynchroniseraient du réseau pour ne se resynchroniser qu'à la demande. L'intérêt d'une telle méthode de communication ne sera effectif que lorsque le coût énergétique de rejoindre le réseau sera suffisamment faible par rapport au coût de maintien d'une synchronisation active, qui se traduit par $E_{join|moy} \leq E_{syn|moy}$. En utilisant le modèle de consommation d'énergie présenté par l'équation 1.1, le seuil à partir duquel il est plus intéressant, du point de vue de la consommation d'énergie de se désassocier se traduit de la manière suivante :

$$d_{join|moy} \times P_{Rx} \leq d_{synCost} \times P_{Rx}.$$

En simplifiant les valeurs de P_{Rx} et en remplaçant par les valeurs obtenues plus haut, on peut également l'écrire de la façon suivante :

$$\frac{T_{adv}}{2} \leq d_{syn} \times (\lfloor \frac{T_a}{T_{syn}} \rfloor - 1) \quad (2.14)$$

Ainsi, nous pouvons déduire la période à partir de laquelle il peut être intéressant, du point de vue énergétique, de transmettre des données au réseau de manière asynchrone plutôt que de payer le coût de maintien de la synchronisation, dans un contexte de réseau monocanal :

$$T_a \geq \frac{T_{syn}}{2} \times (\frac{T_{adv}}{d_{syn}} + 1) \quad (2.15)$$

En considérant une valeur de dérive d’horloges de 40 ppm, et en utilisant les valeurs par défaut du standard 802.15.4 [9], nous avons montré que $T_{syn} = 12.5$ s et $T_{adv} = 30$ s (cf. équation 1.4). De plus, nous supposons que le temps d’activité nécessaire à l’envoi d’un message de KA, en prenant en compte la dérive d’horloge maximale, est de $d_{syn} \approx 1,5$ ms.

Avec ces valeurs, il devient intéressant dans un contexte monocanal d’utiliser un mode de communication asynchrone, lorsque la période du trafic applicatif serait plus faible, en moyenne que

$$T_a \geq \frac{12.5}{2} \times \left(\frac{30}{1,5 \times 10^{-3}} + 1 \right)$$

c’est-à-dire $T_a > 125000$ s (environ une journée et demie). Cette valeur très élevée montre que l’exploitation d’un mode de communication asynchrone n’est, dans la situation existante, pas intéressante du point de vue de la consommation, car le coût énergétique de la phase d’attache au réseau est trop élevé. Comme nous l’avons vu plus tôt, l’extension à un modèle multicanal viendra encore multiplier par le nombre de canaux N_c cette période T_a minimale à partir de laquelle un nœud aurait intérêt d’exploiter ce mode de communication asynchrone.

Ainsi, dans le but de créer une interopérabilité multimode synchrone/asynchrone, il est nécessaire de travailler à la réduction du coût de la procédure d’attache au réseau, dans le but de rendre cette interopérabilité réaliste. C’est pourquoi, dans le second chapitre des contributions, nous proposerons deux méthodes de réductions des coûts d’attache à un réseau TSCH.

2.3.2 Réalisation d’une interopérabilité multitechnologie

La seconde manière de répondre à la problématique de recevoir un trafic applicatif asynchrone, avec une faible latence et une faible consommation d’énergie globale, consiste en la réalisation d’une interopérabilité entre deux technologies, l’une synchrone pour l’aspect d’économie d’énergie globale, et l’autre asynchrone, pour l’aspect faible latence.

Parmi les technologies étudiées (une synchrone et l’autre asynchrone faible latence), nous identifions trois types de périphériques :

- Les périphériques asynchrones contraints en énergie, qui envoient leurs données de manière asynchrone et qui sont également la source du trafic que nous souhaitons acheminer à faible latence et avec peu d’énergie.
- Les nœuds contraints en énergie du réseau synchrone, qui suivent une activité périodique dans le temps pour économiser de l’énergie en respectant un ratio d’activité DC.
- Les périphériques toujours disponibles pour recevoir une communication, utilisés par la technologie asynchrone pour recevoir les trafics applicatifs avec une faible latence.

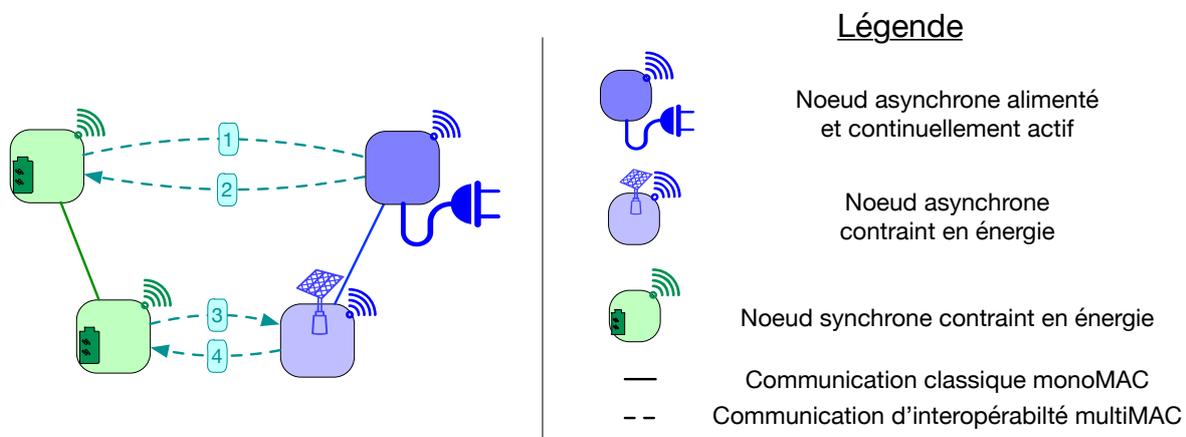


FIGURE 2.12 – Les quatre types de transmissions pour l’interopérabilité synchrone/asynchrone.

La figure 2.12 présente les différents types de communications pouvant avoir lieu entre ces trois types de périphériques.

Échange de données entre deux nœuds de technologies différentes.

Comme n'importe quelle transmission classique, une transmission d'interopérabilité entre un nœud synchrone et un nœud asynchrone correspond à la superposition au même instant t_{Interop} de l'activation de la radio d'un nœud émetteur en mode transmission (Tx) et celle du destinataire en mode réception (Rx). Ces deux radios doivent nécessairement être actives sur le même canal fréquentiel f_{Interop} dans le but de communiquer, et doivent également exploiter un même protocole de niveau PHY.

Ainsi, pour permettre une communication entre un nœud d'une technologie synchrone, et un nœud d'une technologie asynchrone, il est nécessaire de définir, pour cette communication le triplet (t_{Interop} , f_{Interop} , protocole PHY) auprès de chacun des nœuds impliqués dans la transmission d'interopérabilité.

Détermination de la technologie PHY. Comme nous l'avons précisé sur la figure 2.7, nos travaux se focalisent sur la réalisation d'une interopérabilité au niveau MAC, et supposent l'utilisation d'une couche physique commune. Ainsi, l'étape de sélection de la technologie PHY n'est pas nécessaire dans notre contexte, puisqu'elle est commune aux deux piles avec lesquelles nous travaillons. Pour une comparaison des couches PHY en termes de débit et de surcoût protocolaire, voir les travaux du chapitre suivant.

Détermination du canal de communication f_{Interop} . Les technologies de réseaux asynchrones, par besoin de disponibilité de nœuds continuellement actifs vis-à-vis des nœuds contraints, sont monocanales et utilisent un unique canal dédié f_{Async} pour communiquer, à la manière des technologies LoRa et Thread.

Les nœuds des réseaux synchrones, quant à eux, peuvent opérer sur un ou plusieurs canaux parmi les canaux disponibles de la technologie, suivant si la technologie qu'ils exploitent est de nature monocanal ou multicanal. La technologie avec laquelle nous avons choisi de travailler, le TSCH, exploite des mécanismes de sauts de fréquence : les nœuds du réseau TSCH peuvent changer de canal au cours du temps.

Pour déterminer le canal de communication sur lequel les nœuds de différents réseaux vont pouvoir échanger des données, f_{Interop} , il faut prendre en compte le fait que les nœuds du réseau asynchrone échangent des données sur le canal f_{Async} , et uniquement ce dernier.

Ainsi, il appartient aux nœuds du réseau synchrone, capables de changer de canal de s'adapter pour activer leurs radios sur le canal $f_{\text{Interop}} = f_{\text{Async}}$ dans le but d'échanger des données avec les nœuds du réseau asynchrone à proximité.

Cependant, une telle proposition revient à rendre l'interopérabilité monocanale, puisque les communications liées à cette interopérabilité auraient lieu uniquement sur le canal f_{Interop} .

Ainsi, nous proposons que les nœuds du réseau TSCH réalisent le processus d'interopérabilité sur chaque canal fréquentiel disponible, à la fois dans le but d'être disponible pour n'importe quel réseau asynchrone voisin, mais également parce qu'on pourrait imaginer exploiter la robustesse apportée par la nature multicanal du TSCH pour transporter des informations du réseau asynchrone lorsque les conditions de ce dernier sont dégradées.

Par exemple, si le canal f_{Async} utilisé par le réseau asynchrone venait à être brouillé alors qu'un nœud asynchrone contraint est à portée d'un nœud du réseau synchrone, ce nœud asynchrone contraint pourrait exploiter la proximité du réseau synchrone pour transmettre ses données sur un canal fréquentiel de meilleure qualité, tiré aléatoirement parmi les canaux fréquentiels utilisés par la technologie synchrone.

Instant de communication t_{Interop} . La mise en place d'une interopérabilité entre une technologie synchrone et une technologie asynchrone signifie, après avoir déterminé le canal fréquentiel de communication f_{Interop} , qu'il faut déterminer les instants t_{Interop} des quatre types de transmissions présentées sur la figure 2.12 :

1. Un nœud synchrone contraint souhaitant communiquer avec un nœud asynchrone constamment actif.
2. Un nœud asynchrone constamment actif souhaitant communiquer avec un nœud synchrone contraint.
3. Un nœud synchrone contraint souhaitant envoyer un paquet à un nœud asynchrone contraint.
4. Un nœud asynchrone contraint souhaitant communiquer avec un nœud synchrone contraint.

Les prochains paragraphes présentent des solutions pour déterminer les instants de chacune de ces communications, sans perturber le fonctionnement des réseaux synchrones et asynchrones déjà en place.

2.3.2.1 Nœud synchrone contraint → Nœud asynchrone constamment actif

Les technologies synchrones répartissent l'activité des nœuds grâce à des cycles d'activité périodiques, de période minimale T_{syn} .

Pour permettre la transmission de trafics d'un nœud synchrone vers un nœud asynchrone continuellement actif, comme le nœud asynchrone est toujours disponible pour la réception, il est facile pour le nœud synchrone d'insérer au sein de l'une de ses périodes d'activité une transmission dédiée au trafic à destination du périphérique continuellement actif, sur le canal physique f_{Interop} .

La transmission suivrait une procédure identique à celle d'une transmission classique de la technologie synchrone, afin que les voisins du nœud synchrone à l'origine de la transmission pour le nœud asynchrone puissent considérer cette transmission de manière transparente.

Si les mécanismes d'évitement de collisions ne sont pas présents dans la technologie synchrone, pour ne pas collisionner avec d'autres transmissions, le nœud synchrone devra réaliser une procédure d'évitement des collisions de type CSMA/CA ou d'écoute avant de transmettre.

Le nœud asynchrone constamment en mode Rx pourrait être occupé à une autre communication (par exemple dans une configuration de *nœuds cachés*)²².

Cependant, cette problématique est la même qu'au sein du réseau asynchrone uniquement, et de nombreuses solutions existent afin de réduire l'impact de tels scénarios sur le bon fonctionnement du réseau, par exemple le mécanisme de vérification de disponibilité du récepteur basé sur l'échange des trames RTS/CTS²³.

2.3.2.2 Nœud asynchrone constamment actif → Nœud synchrone contraint

Afin de permettre des communications émanant d'un nœud asynchrone constamment actif à destination d'un nœud synchrone contraint en énergie, il faut que le nœud asynchrone alimenté signale son trafic et réserve une ressource pour ses trafics auprès du nœud synchrone contraint, à la manière d'une procédure classique du réseau synchrone. Bien évidemment, cette procédure dépend de la technologie synchrone utilisée. Cependant, quelle que soit cette technologie, il est possible pour le nœud asynchrone, non contraint en énergie, de réaliser une telle procédure.

Ainsi, la communication du nœud asynchrone peut être initiée à un instant t_{Interop} et sur un canal fréquentiel f_{Interop} qui correspondent à une des ressources allouées par le nœud synchrone.

2.3.2.3 Nœud synchrone → Nœud asynchrone contraint

Déterminer l'instant de disponibilité du nœud asynchrone contraint pour la réalisation d'une transmission dont il est le destinataire n'est pas une problématique propre à l'interopérabilité, puisque la même problématique existe lors d'une transmission asynchrone classique initiée par un nœud asynchrone constamment actif.

Comme nous l'avons vu, par exemple dans le cas de la technologie Thread, la solution classiquement proposée pour la transmission à destination d'un nœud rarement disponible repose sur l'exploitation d'un parent, jouant le rôle de *proxy*, qui stocke les trafics à destination du nœud dormant jusqu'à la réception

22. Problème des nœuds cachés : situation dans laquelle deux nœuds A et B sont à portée d'un troisième nœud C alors que les nœuds A et B ne sont pas à portée l'un de l'autre.

23. *Ready To Send* (RTS) / *Clear To Send* (CTS) : Mécanisme optionnel du standard 802.11 qui permet de résoudre la problématique des nœuds cachés : un nœud transmet d'abord un message appelé RTS (signifiant "Demande pour émettre") contenant des informations sur le volume des données qu'il souhaite émettre et la vitesse de transmission. Le récepteur répond un CTS (qui signifie "Le champ est libre pour émettre"), avant que le nœud émetteur transmette effectivement sa donnée. Ainsi, un nœud dans une situation de nœud caché pourra entendre le message de CTS et ainsi différer sa propre transmission.

d'une sollicitation de la part de ce dernier. Dans la communication d'interopérabilité, deux entités peuvent jouer le rôle de *proxy*.

Un des nœuds constamment actifs du réseau asynchrone peut être, comme dans le cas d'une transmission asynchrone classique, le *proxy* du nœud asynchrone contraint. Dans ce cas de figure, la communication d'un nœud synchrone contraint à destination d'un nœud asynchrone contraint se transforme alors en une communication d'un nœud synchrone contraint à destination d'un nœud alimenté, déjà analysée au paragraphe 2.3.2.1.

Une autre solution serait de déporter le rôle de *proxy* sur un des nœuds du réseau synchrone : c'est-à-dire qu'un des nœuds synchrones stockerait tout le trafic à destination du nœud asynchrone contraint.

La technologie Thread par exemple, permet également une fonctionnalité de multiples parents, ce qui rendrait possible la distinction entre le parent dans le réseau asynchrone, du *proxy* d'interopérabilité, si nécessaire.

Alors que la première proposition, suppose qu'un nœud constamment actif du réseau asynchrone soit à la fois dans la couverture radio d'un nœud du réseau synchrone et d'un nœud asynchrone contraint, la deuxième proposition est moins contrainte en termes de topologie physique puisque la seule nécessité est que le nœud asynchrone contraint soit à portée d'un nœud synchrone parent d'interopérabilité.

En revanche, cette deuxième proposition exploite une transmission d'un nœud asynchrone contraint à destination d'un nœud synchrone également contraint en énergie, que nous présentons dans le paragraphe suivant.

2.3.2.4 Nœud asynchrone contraint → Nœud synchrone

Alors que les autres types de transmissions permettraient de retomber dans des problématiques déjà étudiées, nous n'avons pas trouvé d'études relatives à la problématique de transmission d'un nœud asynchrone contraint en énergie, et donc incapable de maintenir une synchronisation, à destination d'un nœud synchrone disponible par intermittence.

Récemment, les travaux d'Harrisson *et. al.* [70] ont d'ailleurs montrés que les solutions proposées dans les réseaux de capteurs ont tendance à minimiser la détection d'événement rare (c'est à dire de nature asynchrone) dans un réseau de capteurs à économie d'énergie. Nous proposons de se focaliser sur cette problématique.

Si l'intérêt d'une interopérabilité synchrone/asynchrone permet principalement d'apporter un relai du trafic asynchrone par des nœuds synchrones, qui économisent de l'énergie contrairement au nœud asynchrone continuellement actif, une telle interopérabilité pourrait, comme nous l'avons déjà mentionné, permettre aux nœuds dormants de bénéficier de la capacité du réseau synchrone à réaliser des transmissions sur des canaux différents.

Dans ce cas de figure, le réseau synchrone devrait donc être capable d'acheminer un trafic asynchrone, c'est-à-dire transmis à n'importe quel instant, et sur n'importe lequel des canaux fréquentiels autorisés par la spécification de sa technologie, avec une latence donnée. C'est précisément le besoin que nous avons déterminé au début de ce chapitre, et c'est pour cette transmission d'interopérabilité que nous allons proposer une solution au cours de la dernière contribution de ce rapport.

Conclusion

Dans ce deuxième chapitre, nous avons dans un premier temps analysé les différents mécanismes MAC dédiés à l'économie d'énergie, dans le but d'identifier les catégories de mécanismes les mieux adaptés à différents trafics applicatifs. Nous avons ainsi montré que le mécanisme MAC existant pour la prise en charge de trafics asynchrones contraints en latence repose sur l'exploitation d'un nœud asynchrone alimenté constamment en mode *Rx*, consommateur en énergie.

Nous avons alors proposé de réaliser une interopérabilité qui exploite les mécanismes MAC des nœuds synchrones voisins pour remplacer le rôle des périphériques asynchrones alimentés, tout en garantissant l'acheminement du trafic asynchrone avec une faible latence.

Nous avons ensuite présenté les différentes solutions existantes d'interopérabilité dans l'IoT, et également précisé la nature de l'interopérabilité que nous proposons : une optimisation conjointe de niveau

MAC telle que présentée sur la figure 2.7.

Enfin, nous avons exposé les pistes d'explorations pour la réalisation d'une optimisation conjointe au niveau MAC permettant la gestion d'un trafic asynchrone contraint en latence au sein d'un réseau synchrone.

Nous avons ensuite remarqué qu'une telle interopérabilité de mode de communication (synchrone et asynchrone) est déjà existante dans les technologies présentées, durant la phase d'attache au réseau où un nœud non synchronisé tente de rejoindre un réseau synchrone à proximité.

Nous proposons ainsi deux solutions différentes pour réaliser une interopérabilité entre deux modes de communication MAC, le mode synchrone et le mode asynchrone contraint en latence :

- l'exploitation de la phase d'attache au réseau comme mode de communication asynchrone au sein d'un réseau synchrone,
- et la réalisation d'une interopérabilité multitechnologie, l'une synchrone et l'autre asynchrone, dans le but de répondre à la problématique posée.

Nous avons également montré que l'interopérabilité multimode synchrone/asynchrone au sein d'une même technologie via l'utilisation de la phase d'attache sera trop coûteuse en énergie en utilisant les mécanismes actuels d'attache au réseau. C'est pourquoi, dans nos contributions, nous proposerons deux solutions de réduction du coût énergétique d'attache à un réseau synchrone.

Enfin, nous avons montré que la réalisation d'une interopérabilité de technologies différentes (l'une synchrone et l'autre asynchrone utilisant un récepteur constamment actif) peut se résumer à l'étude de la communication initiée par le nœud asynchrone contraint en énergie et à destination d'un nœud synchrone également contraint, les autres types de communications nécessaires à l'interopérabilité ne présentant pas de problématiques particulières.

Ainsi, dans la dernière contribution de ce rapport, nous proposerons un mécanisme pour réaliser l'interopérabilité entre deux modes de communication du standard 802.15.4 : le mode non-*beacon* asynchrone utilisé par la technologie Thread et le mode synchrone TSCH.

Deuxième partie

Contributions

Comparaison des standards pour l'IoT

Sommaire

2.1 Motivations pour l'interopérabilité	50
2.1.1 La diversité des applications	50
2.1.2 Les profils applicatifs des technologies de l'IoT	52
2.1.2.1 Impact de la dérive d'horloge	52
2.1.2.2 Trafics périodiques tels que $T_a \leq T_{a \text{seuil}}$	55
2.1.2.3 Trafics aperiodiques ou périodiques tels que $T_a \geq T_{a \text{seuil}}$	55
2.1.3 Nos objectifs d'interopérabilité	56
2.2 État de l'art des solutions d'interopérabilité	56
2.2.1 Interopérabilité en exploitant un protocole commun	57
2.2.2 Partage de la ressource physique	57
2.2.3 Propositions d'interopérabilité MAC existantes	59
2.3 Pistes d'exploration	60
2.3.1 Exploitation de la phase d'attache	61
2.3.2 Réalisation d'une interopérabilité multitechnologie	64
2.3.2.1 Nœud synchrone contraint → Nœud asynchrone constamment actif	66
2.3.2.2 Nœud asynchrone constamment actif → Nœud synchrone contraint	66
2.3.2.3 Nœud synchrone → Nœud asynchrone contraint	66
2.3.2.4 Nœud asynchrone contraint → Nœud synchrone	67

Introduction

Une faible consommation d'énergie est importante pour la plupart des dispositifs IoT, car ils fonctionnent principalement sur piles, et doivent néanmoins atteindre des durées de vie de l'ordre de plusieurs années sans remplacement de la batterie, puisque le remplacement de la batterie d'un nœud n'est pas acceptable au vu de l'ampleur des déploiements ciblés.

Jusqu'à récemment, les appareils ultra-basse consommation reposaient principalement sur le standard 802.15.4 [71] qui a été spécialement conçu pour une faible consommation d'énergie. Cependant, les travaux de Tozlu [72] ont montré que les appareils 802.11 peuvent également bénéficier d'une bonne efficacité énergétique en utilisant le mode 802.11b PSM (Mode d'économie d'énergie — ou *Power Saving Mode*) [42], et parfois même surpasser la technologie 802.15.4 en termes de consommation énergétique. Le BLE (Bluetooth Low Energy) a également ajouté un autre concurrent aux normes de faibles puissances, et, au-delà des solutions de faibles portées, les technologies émergentes à longues portées telles que 802.11ah [43], LoRa [7] et SIGFOX [73] visent également une haute efficacité énergétique.

Le premier chapitre, présentant ces technologies, a fourni une compréhension approfondie de la consommation d'énergie des différents protocoles, notamment en décrivant le mode opératoire des nœuds lors de la transmission de paquet à destination d'un puits à l'aide des différentes modes matériels *Rx*, *Tx*, *Idle* et *Sleep*.

Dans ce chapitre, nous présentons une comparaison des durées de vie estimées des dispositifs IoT fonctionnant dans les réseaux sans fil que nous nommons, par souci de simplicité, de la manière suivante :

- La version 5.0 de Bluetooth Low Energy : **BLE**
- IEEE 802.15.4 : **802.15.4**
- IEEE 802.15.4 TSCH (Saut de fréquence à créneaux temporels — ou *Time-slotted Channel Hopping*) : **TSCH**
- IEEE 802.11b PSM à 11 Mb/s : **PSM 11**,
- IEEE 802.11ah au débit minimal : **10_1** (MCS 10, 1 MHz de largeur de bande)
- IEEE 802.11ah avec sa valeur de débit maximal en Europe : **8_2** (MCS 8, 2 MHz de largeur de bande)
- IEEE 802.11ah avec sa valeur de débit maximal aux États-Unis : **9_16** (MCS 9, 16 MHz de largeur de bande)
- LoRa pour sa plus faible valeur de débit en Europe : **LoRa EUMin**
- LoRa pour sa valeur de débit maximale en Europe : **LoRa EUMax**
- SIGFOX pour sa plus faible valeur de débit en Europe : **SIGFOX EUMin**
- SIGFOX pour sa plus forte valeur de débit en Europe : **SIGFOX EUMax**

Cette étude est motivée par l'observation que de nombreuses technologies sans fil différentes peuvent être utilisées pour IoT et qu'il n'y a pas, à notre connaissance, de comparaison exhaustive dans la littérature qui aiderait à prendre la bonne décision quant au déploiement d'un réseau donné. Bien que certaines caractéristiques des technologies considérées, comme la portée ou la largeur de bande utilisée, ne soient pas directement comparables, il est nécessaire de procéder à une comparaison approfondie de l'efficacité énergétique afin d'identifier leurs principaux attributs et limites.

Il existe de nombreuses présentations des différents réseaux sans fil [74, 75, 76, 49, 77, 78], mais à notre connaissance, aucun travail antérieur ne compare autant de technologies sans fil pour l'IoT, du point de vue de l'efficacité énergétique, en les plaçant toutes sur un pied d'égalité. Certains articles se contentent d'étudier deux technologies [79, 80], mais il manque alors une comparaison avec les autres solutions disponibles.

Une telle comparaison avec toutes les technologies réunies sur un pied d'égalité permet de mettre en lumière les avantages et les inconvénients respectifs des différentes technologies, afin d'orienter les développements futurs vers l'amélioration des performances et de la consommation d'énergie.

Par ailleurs, ce chapitre a également pour but d'étendre les modèles de consommation d'énergie qui reflètent le fonctionnement des couches MAC à ratio d'activité cyclique, ce qui est souvent négligé dans les études sur la consommation d'énergie.

Hormis les articles récents qui donnent les principes du modèle énergétique que nous utilisons [13, 81, 5], les modèles précédents de consommation d'énergie ne tiennent compte que des principaux aspects

suivants : puissance de transmission, distance entre deux nœuds, taille des paquets et perte de chemin [82, 83, 84, 85, 86].

Cette approche a permis de modéliser uniquement le comportement de la couche physique et elle ne reflétait pas de manière réaliste les dispositifs de l'IoT qui fonctionnent par alternance de cycles d'activité et d'endormissement. Notre objectif est également d'étendre les récents modèles réalistes de consommation d'énergie à toutes les technologies considérées et de les rendre disponibles par le biais d'un outil disponible sous forme de logiciel libre [11], permettant à quiconque de faire varier les paramètres du réseau et d'estimer rapidement l'impact sur les durées de vie des périphériques du réseau.

Enfin, nous voulons évaluer l'impact des protocoles de niveau IP sur des périphériques IoT contraints, en tenant compte non seulement des performances matérielles et du fonctionnement MAC intermittent, mais aussi du surcoût protocolaire (en-têtes et fragmentation), ainsi que du coût de maintien d'une association active, de la synchronisation temporelle entre deux nœuds, de la dérive d'horloge et des conditions du canal de transmission.

Nous supposons l'utilisation des protocoles IPv6/6LoWPAN chaque fois que cela est possible, dans le but d'établir une comparaison sur un pied d'égalité, comme cela est présenté sur la figure. 3.1.

De cette façon, nous évaluons l'impact de toutes les couches sur la consommation et la performance énergétique des périphériques IoT. L'objectif général de ce chapitre est d'aider tous les utilisateurs des technologies de l'IoT à sélectionner la technologie qui répond le mieux à leurs besoins, du point de vue de l'efficacité énergétique.

3.1 Principe de l'analyse de consommation énergétique

3.1.1 Hypothèses et méthodologie

Dans ce chapitre, nous considérons le cas d'un nœud contraint en énergie (fonctionnant sur batteries ou à récupération d'énergie) qui génère et envoie des données à destination d'un puits (trafic en liaison montante). Nous adoptons le point de vue d'un dispositif IoT qui est synchronisé au réseau et souhaite rester actif dans ce réseau le plus longtemps possible.

Nous limitons la portée de ce chapitre à l'étude d'un lien unique, car il s'agit du cas le plus courant, une sorte de référence pour les comparaisons (il est à noter que beaucoup d'articles limitent la portée à un cas de lien unique, par exemple le modèle de consommation d'énergie analytique de B-MAC proposé par Polastre *et al.* [87]).

De plus, la topologie de tous les réseaux considérés, sauf le 802.15.4, est une topologie en étoile dans laquelle un nœud se synchronise ou s'associe à un nœud suffisamment alimenté en énergie pour obtenir une connectivité à Internet. Dans le cadre des travaux futurs, nous prévoyons d'étendre l'analyse aux réseaux multisaits 802.15.4, ainsi qu'aux réseaux BLE maillés.

Notre méthodologie consiste à analyser la consommation d'énergie de chaque technologie en considérant un modèle énergétique réel dans lequel les éléments les plus consommateurs d'énergie d'un nœud (la radio et le microcontrôleur) alternent entre les états actifs et inactifs (*Tx*, *Rx*, *Idle* et *Sleep*) en suivant les schémas présentés dans le chapitre 2 de consommation d'énergie, et les contraintes de temps définies par les couches MAC.

Nous dérivons notre modèle des propositions précédentes pour les technologies 802.15.4 [13, 81, 5], et les étendons à d'autres technologies MAC. Ensuite, nous l'utilisons pour analyser chacune des technologies en fonction de différents paramètres.

Comme le présente la figure 3.1, nous supposons que le trafic applicatif est composé de paquets UDP de tailles variables au-dessus des protocoles IPv6/6LoWPAN lorsque cela est possible, ou directement au-dessus de la couche MAC si IPv6 n'est pas supporté. L'analyse tient compte de l'énergie dépensée dans les retransmissions de trames de données en raison de la corruption des trames et des collisions.

Nous considérons la consommation d'énergie pour différentes valeurs de paramètres de plateformes matérielles (c'est-à-dire la consommation d'énergie par état) pour trouver les paramètres conduisant à une consommation d'énergie minimale. Nous adoptons ensuite les plateformes matérielles à consommation d'énergie minimale pour calculer la durée de vie des technologies considérées pour un trafic applicatif d'intensité variable, un PER (Taux d'erreur de paquets — ou *Packet Error Rate*) et une précision d'horloge

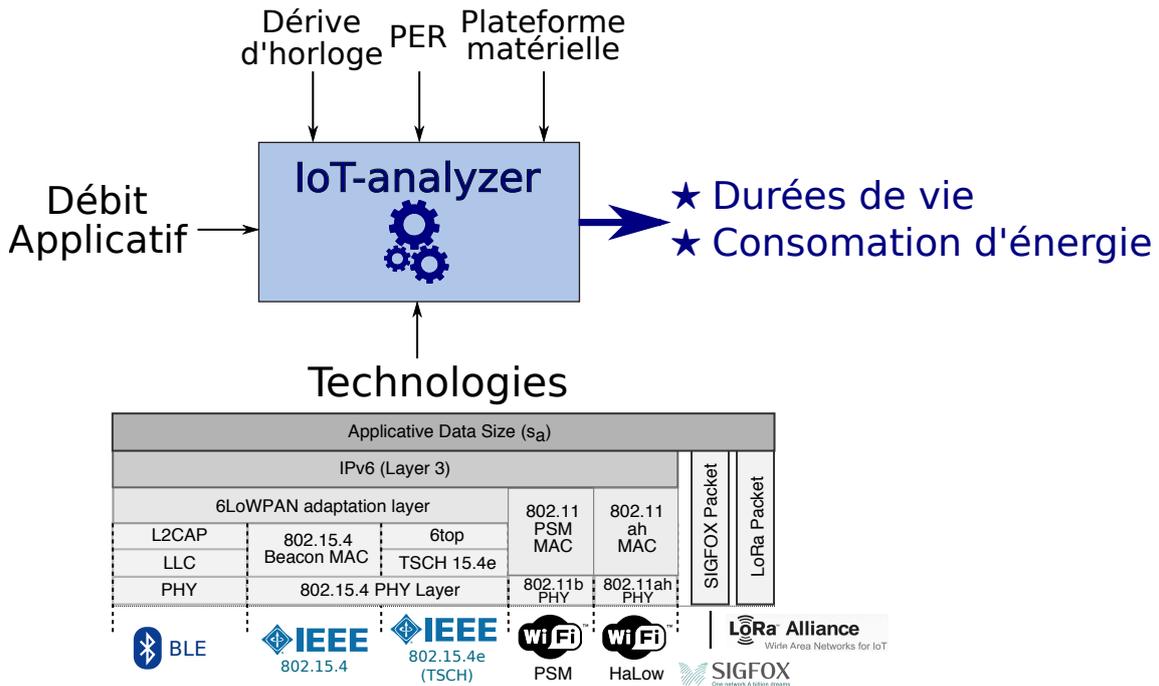


FIGURE 3.1 – Piles protocolaires des technologies comparées, et principe de fonctionnement de l’analyseur utilisé pour la comparaison.

donnée. Nous faisons le postulat suivant : même si une telle plateforme idéale n’existe pas encore, elle représente la meilleure performance énergétique disponible pour une technologie donnée, ce qui permet la comparaison avec d’autres technologies.

Nous présentons dans la suite de cette section les principes de base de notre analyseur qui calcule la consommation d’énergie et la durée de vie des technologies sans fil. Une approche analytique, contrairement à un simulateur, présente l’avantage de diminuer les temps de calcul ainsi que les efforts de développement, tout en offrant une précision suffisante pour une première estimation.

Notons que les fondeurs de puces adoptent une approche similaire, par exemple Linear Technology, qui propose un analyseur pour ses produits [88].

Notre analyseur est développé en Python et disponible sous licence BSD 3 [11] .

3.1.2 Modèle de consommation énergétique

Afin de modéliser la consommation énergétique des différentes plateformes, il est nécessaire de définir les différents modes d’opérations possibles des plateformes matérielles. Pour cette étude, nous supposons que la radio et le microcontrôleur (*Central Processing Unit*, ou CPU) sont les éléments les plus consommateurs des périphériques. Ainsi, seuls ces deux composants seront modélisés pour étudier les performances des technologies MAC. De plus, nous considérons un modèle de consommation d’énergie dans lequel la consommation totale d’énergie est exprimée comme la puissance requise pour chaque état pendant le temps passé dans l’état. Notre modèle suppose ainsi qu’une plateforme IoT fonctionne dans les quatre modes présentés en introduction du chapitre précédent (*Rx*, *Tx*, *Idle* et *Sleep*), qui correspondent aux combinaisons possibles des modes de chaque composant, comme illustrés dans le tableau 1.1, page 27 (nous donnons plus loin les valeurs de la puissance consommée dans chaque état, suivant les plateformes).

Comme mentionné dans le premier chapitre, nous exprimons la consommation d’énergie au cours de l’intervalle t de la manière suivante :

$$E(t) = \sum_S P_S \times d_S, \quad S \in \{Tx, Rx, Idle, Sleep\} \tag{3.16}$$

où P_S est la puissance consommée dans l’état S et d_S la durée dans l’état S au cours de l’intervalle t . d_S

provient de l'analyse du fonctionnement de chaque technologie sans fil présentée dans le chapitre 2.

Nous utilisons les figures 1.5, 1.7, 1.15 et 1.18, présentées au chapitre précédent, qui décrivent les différents modes d'opérations dans le temps d'un nœud qui souhaite transmettre le maximum de données à destination d'une station de base ou d'un nœud plus élevé dans l'arbre de collecte, pour chacune des technologies MAC comparées dans ce chapitre.

Nous supposons, pour chaque technologie, une topologie en étoile avec un lien entre une feuille (nœud de terminaison) contraint en énergie pour lequel nous voulons trouver la durée de vie, et un puits de collecte, alimenté en énergie.

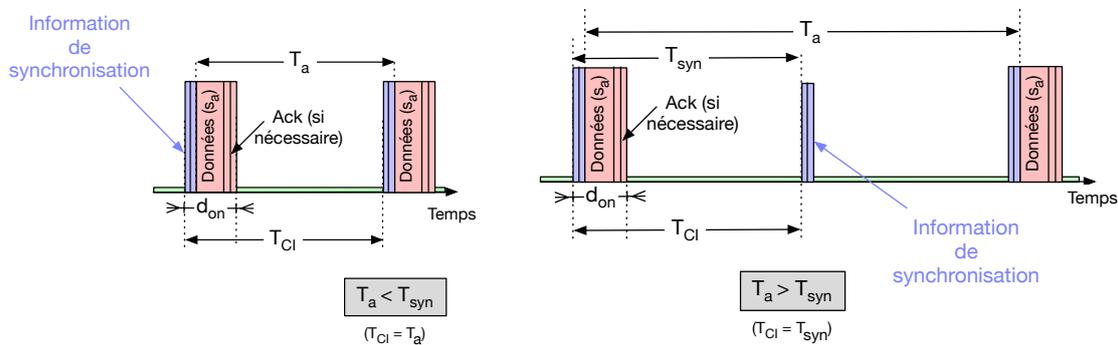
L'analyse prend en compte le trafic applicatif ascendant, tel que les rapports périodiques, à destination du puits. Pour tenir compte de la connectivité de niveau IP, nous considérons une couche d'adaptation 6LoWPAN si nécessaire, ainsi que l'IPv6 au-dessus des couches PHY/MAC de chaque technologie, autant que possible (voir figure 3.1). Toutefois, en raison de la petite taille des paquets applicatifs autorisée, les technologies longue portée prennent en charge les données applicatives directement au-dessus de la couche MAC. De plus, l'analyse suppose un état stationnaire du réseau, c'est-à-dire que le nœud feuille est considéré comme déjà associé au puits (pas d'évaluation de la phase de synchronisation au réseau).

Modèle de batterie utilisé

Nous supposons une énergie de batterie initiale de $E_0 = 13,5 \text{ kJ}$, ce qui correspond à l'énergie disponible avec deux piles AAA (1250 mAh sous 1,5 V chacune), et un modèle simple de batterie avec des fuites E_{leak} de 5 % par an, ainsi qu'une tension de coupure lorsque l'énergie résiduelle atteint 10 % de l'énergie initiale E_0 .

Pour des comparaisons équitables, nous supposons la même batterie pour toutes les technologies, même si une batterie plus petite pourrait être utilisée pour certains protocoles.

3.1.3 Modèle applicatif et calcul de la durée de vie



(a) Le débit applicatif est suffisamment élevé pour maintenir la synchronisation active.

(b) Le débit applicatif est trop faible pour maintenir une synchronisation à lui seul.

FIGURE 3.2 – Principe et variables utilisées lors de la comparaison.

La figure 3.2 illustre la définition des principales variables que nous utilisons pour calculer la consommation d'énergie, tout en analysant le fonctionnement d'un protocole MAC donné pour atteindre un débit maximal. Si un nœud doit fragmenter un paquet, il enverra les fragments dans des trames successives. Les informations de synchronisation correspondent à la communication permettant de maintenir la synchronisation entre deux nœuds, à la manière des balises ou des trames de *poll* du BLE.

Nous définissons ainsi les variables suivantes :

- T_a , la période applicative, c'est-à-dire l'intervalle de temps compris entre deux instants de génération de données par une application.
- s_a , la taille des données applicatives, c'est-à-dire la taille des données générées par l'application à chaque T_a . s_a représentent les données utiles de la couche applicative. Nous ajoutons ensuite les en-têtes de chaque protocole et prenons en compte la fragmentation si nécessaire.
- r_a , le débit applicatif, correspond au débit de données applicatives (en bits/s ou b/s) : $r_a = \frac{s_a}{T_a}$

- T_{syn} , l'intervalle maximal de synchronisation. Cela correspond à l'intervalle au bout duquel un nœud doit se réveiller de manière spécifique pour le maintien d'une synchronisation temporelle. Cette période est nécessaire pour conserver une association active entre les dispositifs. S'il n'y a pas de communication pendant un intervalle plus long que T_{syn} , les nœuds perdent leur association avec le réseau et devront se réintégrer plus tard, au prix d'une plus grande consommation d'énergie.
- T_{ci} , le temps entre deux réveils d'un nœud défini de la manière suivante $T_{ci} = \min(T_a, T_{syn})$. Un transfert de données peut servir à la synchronisation, mais même s'il n'y a pas de données applicatives à envoyer, les nœuds doivent communiquer chaque T_{syn} (voir la partie droite de la figure 3.2b).
- d_{on} , la période active, qui correspond à la durée maximale pendant laquelle un nœud reste éveillé, c'est-à-dire lorsqu'il est dans un état différent de *Sleep* sur l'intervalle T_{ci} .
- L_t , la durée de vie d'un nœud — ou *lifetime*. Elle correspond à la durée de fonctionnement d'un nœud avec une quantité d'énergie initiale disponible E_0 .

Afin d'estimer la durée de vie des appareils pour chaque protocole, nous sélectionnons les paramètres MAC de chaque protocole pour supporter r_a , le débit applicatif, avec une consommation d'énergie minimale, de sorte que nous mesurons la durée minimale d_{on} pour transmettre s_a octets de données. Ainsi, nous choisissons par exemple le paramètre *SD* du 802.15.4 de manière à atteindre la durée d_{on} optimale, pour transmettre s_a octets de données pendant l'intervalle T_a .

Si une technologie ne peut pas prendre en charge une application donnée à cause d'un débit r_a trop élevé, elle n'apparaît pas sur les courbes. Ainsi, la courbe d'une technologie s'arrête lorsque le trafic applicatif correspond à la capacité maximale du protocole donné.

En nous basant sur les variables et les principes de fonctionnement de chaque couche MAC expliqués dans le chapitre 3, nous calculons l'énergie consommée sur l'intervalle de temps T_a avec l'équation 1.1, dans laquelle P_S , la consommation d'énergie dans l'état S , est multipliée par d_S , le temps passé dans l'état S sur l'intervalle T_a .

L'algorithme pour obtenir la durée de vie est présenté dans la figure 3.3 et calcule l'énergie consommée par T_a en fonction du débit d'application r_a et, en dérive L_t , la durée de vie d'un nœud. L_t correspond au nombre d'intervalles T_a qu'un nœud peut exécuter avant que l'énergie initiale E_0 soit épuisée, elle peut s'exprimer en unité de temps en multipliant le nombre d'intervalles T_a supportés par la durée d'un T_a .

3.1.4 Modèle des pertes de paquets

Pour les technologies qui utilisent des retransmissions lorsque le paquet n'est pas correctement reçu, nous prenons en compte les paquets de données non reçus à cause des conditions de transmission imparfaites et/ou des collisions de la manière suivante :

Soit p_{er}^f et p_{er}^r le taux d'erreur de paquet dans la transmission aller (*forward*), et dans le sens retour (*reverse*), respectivement. Un paquet de données envoyé depuis un nœud sera ainsi perdu avec la probabilité p_{er}^f tandis que son acquittement souffrira d'une probabilité de perte p_{er}^r . La probabilité d'un échec de transmission au niveau de la couche MAC est donc :

$$PER = p_{er}^f + (1 - p_{er}^f) \times p_{er}^r. \quad (3.17)$$

Comme p_{er}^r est petit par rapport à p_{er}^f , de par la faible taille des trames d'ACK, la probabilité peut être réduite à $PER = p_{er}^f$. En supposant une probabilité de perte de paquets PER , un nœud effectuera en moyenne le nombre de transmissions suivant pour réaliser une transmission effective :

$$N_{tr} = \frac{1}{1 - PER}. \quad (3.18)$$

Dans la comparaison, nous fixons un niveau de PER donné et, pour ce PER , nous calculons la consommation d'énergie nécessaire pour envoyer N_{tr} trames de données (suivies d'un ACK) pour un paquet généré par la couche applicative. Ainsi, le nombre de transmissions totales prises en compte sera :

$$N_{tr|tot} = N_{tr} \times N_{paquets}. \quad (3.19)$$

Les détails de la fonction `énergieTransmissionDonnées`, dans laquelle nous prenons en compte la consommation d'énergie due aux retransmissions de données dans chaque période d'application T_a , sont présentés sur la figure 3.3 (première fonction de la figure).

```

fonction énergieTransmissionDonnées( $s_a$ , PER) {
     $N_{\text{paquets}} = \lceil \frac{s_a}{\text{TAILLE\_MAX\_PAQUET}} \rceil$ 
    for  $S \in \{Tx, Rx, Idle, Sleep\}$  do
         $d_S = \sum_{i=0}^{N_{\text{paquets}}} d_{S|Transmission\ i} \times N_{tr}$ 
    end for
     $E_{\text{data}} = \sum_S P_S \times d_S$  for  $S \in \{Tx, Rx, Idle, Sleep\}$ 
    return  $E_{\text{data}}$ 
}

fonction énergieConsommée( $T_a$ ,  $T_{\text{syn}}$ ) {
     $E_{\text{data}} = \text{énergieTransmissionDonnées}(s_a, \text{PER})$ 
     $E_{\text{syn}} = \text{énergiePourInfoSynchroPure}$ 
    if  $T_a < T_{\text{syn}}$  then
         $N_{\text{syn}} = 1$ 
    else
         $N_{\text{syn}} = \frac{T_a}{T_{\text{syn}}}$ 
    end if
    if il est possible d'exploiter l'envoi de données pour la synchronisation then
         $N_{\text{dataSyn}} = 1$ 
    end if
     $E_{\text{sleep}} = \text{énergiePendantInactivité}(d_{on}, T_a)$ 
     $E_{\text{tot}} = E_{\text{data}} + (N_{\text{syn}} - N_{\text{dataSyn}}) \times E_{\text{syn}} + E_{\text{sleep}}$ 
    return  $E_{\text{tot}}$ 
}

 $\gamma_{\text{leak}} \leftarrow 5\% \times \frac{t_a}{31536000}$  {1 an = 31 536 000 s}
fonction main( $s_a$ ,  $\gamma_{\text{leak}}$ ) {
     $L_t \leftarrow 0$ 
     $E \leftarrow E_0$ 
     $E_{\text{leak}} \leftarrow E_0 \times \gamma_{\text{leak}}$ 
    while  $E > 10\% E_0$  do
         $E = E - \text{énergieConsommée}(T_a, T_{\text{syn}}) - E_{\text{leak}}$ 
         $L_t = L_t + T_a$ 
    end while
    return  $L_t$ 
}

```

FIGURE 3.3 – Algorithme de calcul de la durée de vie d'un nœud.

3.1.5 Évaluation de l'analyseur

Pour valider notre analyseur, nous avons comparé ses résultats avec l'analyseur de puissance consommée des nœuds SmartMeshIP [88] de TSCH fourni par Linear Technology, en utilisant le mode de puissance la plus faible. Nous avons configuré dans cet analyseur une topologie en étoile avec un nœud envoyant 10 octets à son parent pour chaque période d'application $T_a = 100$ s (appelé *reporting advertising*) avec une stabilité des liens fixée à 100%. L'analyseur suppose une période d'envoi des KA de $T_{\text{syn}} \sim 4.083$ s (~ 25 trames de KA dans un intervalle de 100 s), mais il n'exploite pas les trames de données pour le maintien de la synchronisation temporelle. Dans cette configuration, le courant total moyen des nœuds estimé pour une *slotframe* de liaison descendante de 1024 slots est de $4.3 \mu\text{A}$ pour une tension de 3.6V. Dans une configuration comparable, notre analyseur donne une consommation moyenne de courant de $4.317 \mu\text{A}$, ce qui correspond à une erreur de moins de 1%.

Nous avons également comparé la consommation moyenne d'énergie du 802.15.4 avec la valeur mesurée de la plateforme GREENNET [5] : d'après les mesures, la consommation moyenne de courant est de $3 \mu\text{A}$ pour une valeur de *BO* de 4 (ce qui correspond à un intervalle de *beacon* de $BI = 240$ ms), en exploitant la possibilité qu'un nœud feuille puisse ne pas écouter les trames de *beacon* dans la limite de l'intervalle maximal de temps sans communication fixé par le standard, un ordre de supertrame $SO = 1$ (c'est-à-dire

une durée de supertrame de 15,36 ms), et une période applicative $T_a = 4$ min. Dans une configuration équivalente, notre analyseur annonce une consommation moyenne de $2,46 \mu A$. La différence de 18% est principalement due à une simplification inhérente à notre modèle, qui ne prend pas en compte la consommation d'énergie lors des transitions d'états, ni celle du capteur de température dans les mesures GREENNET.

De plus, nous avons comparé les performances obtenues par notre analyseur pour l'envoi d'une trame de 100 octets avec BLE ou 802.15.4 avec les performances rapportées par Siekkinen *et al.* [80]. Alors que notre analyseur utilise le mode BLE 5.0, nous modifions les paramètres BLE pour qu'ils correspondent à la spécification BLE 4.0 [89], afin de comparer des scénarios équivalents. Nos résultats pour BLE sont 460 Ko/J, tandis que 433 Ko/J ont été rapportés par Siekkinen *et al.* De même, nous trouvons 180 Ko/J pour la norme 802.15.4, alors que Siekkinen *et al.* ont rapporté des valeurs de 168 Ko/J. Dans les deux cas, la différence de résultats est inférieure à 7%.

Dans le but de valider l'analyseur pour les technologies longue portée, nous avons comparé ses valeurs de sortie avec les valeurs mesurées par Martinez *et al.* [81] pour la transmission d'un paquet de données applicatives de 12 octets (ce qui correspond à la taille maximale pour la technologie SIGFOX). Pour établir un scénario proche de ceux de [81], nous avons ajouté un supplément de 2 B au paquet SIGFOX minimal présenté au chapitre précédent (figure 1.22, page 44) parce que SIGFOX impose de toute façon l'utilisation de HMAC pour l'authentification des messages à la couche MAC. [81] ont mesuré une charge de courant de 50 mA pour trois transmissions consécutives de ~ 2 s chacune. En supposant une tension de 3V, cette valeur correspond à 900 mJ d'énergie consommée. Notre analyseur calcule un temps total de transmission de 6,24 s et une consommation d'énergie de 917,28 mJ (moins de 5% de différence). Enfin, nous avons comparé nos résultats avec les mesures LoRa [81] : pour la transmission d'un paquet de données de 15 octets, et en supposant le facteur d'étalement LoRa maximum, la consommation d'énergie mesurée est d'environ 213 mJ. Dans une configuration similaire, nous calculons une énergie consommée de 224 mJ, ce qui correspond à une différence proche de 5%.

L'ensemble de ces comparaisons montrent que notre analyseur calcule des valeurs qui dévient de la plupart des valeurs mesurées publiées avec une faible erreur (comprise entre 5% à 7%), ce qui confirme la cohérence de notre approche.

3.2 Impact des paramètres matériels sur les résultats

Pour comparer les technologies sur une base équitable, nous devons sélectionner les paramètres matériels des plateformes en ce qui concerne les valeurs de leur consommation d'énergie dans les différents états *Rx*, *Tx*, *Sleep* et *Idle*. L'idée est d'identifier les paramètres qui permettent de réduire au minimum la consommation d'énergie pour chacune des technologies.

3.2.1 Paramètres propres à chaque technologie

Technologie	BLE	802.15.4	TSCH	PSM 11	802.11ah		
					10_1	8_2	9_16
Débit PHY (Mb/s)	2	0.25	0.25	11	0.15	7.8	78
Taille max							
Paquet MAC (octets)	245	120	121	1280	1280	1280	1280
SIFS/TIFS (μs)	150	192	192	10	160	160	160
DIFS/LIFS (μs)	-	640	640	50	264	264	264
Trame de <i>beacon</i> (μs)	60	544	-	50.18	3440	295.4	245.5
Trame d'ACK (μs)	44	352	512	10.18	560	240	240
T_{syn} (s)	32	251.66	16.67	$\sim 18h$		~ 5 ans	
d_{on} (ms)@TailleMax	2.58	6.37	10	1.12	75.9	2.56	1.29
Retransmission							
Data/SIFS/ACK à Taille Max (ms)	2.47	5.83	10	1.07	72.5	2.27	1.04

TABLEAU 3.3 – Paramètres utilisés pour les différentes technologies.

Nous commençons par présenter les principaux paramètres des normes dans le tableau 3.3. Le SIFS (Court intervalle entre les trames – ou *Short InterFrame Space*) est un intervalle court utilisé entre la transmission de la trame de donnée et la transmission de la trame d'acquiescement. Cette durée était initialement prévue pour prendre en compte le temps de retournement de la radio matérielle. Le DIFS (Intervalle distribué entre les trames – ou *Distributed InterFrame Space*) quant à lui représente la durée entre deux transmissions successives.

Nous supposons également une valeur de $CCADuration$ 802.15.4 de $128\mu s$, et utilisons les valeurs temporelles TSCH par défaut (voir la table des attributs TSCH-MAC PIB par défaut du standard [9]) :

- durée d'un slot TSCH = 10 ms,
- $tsTxOffset = 2,120$ ms,
- et $tsRxAckDelay = 800\mu s$.

3.2.2 Choix des paramètres

Technologie : Plateforme matérielle	P_{Tx} (mW)	P_{Rx} (mW)	P_{Idle} (mW)	P_{Sleep} (μW)
802.11 PSM : G2M5477 [72]	699.6	170	66	13.2
802.11 PSM : RTX4100 [90]	1050	350	9.1	9.45
802.11 PSM : MAX2830 [91]	699.6	204.6	92.4	66
802.11 PSM : STM SPWF01SA [92]	1135	346.5	85.8	141.9
802.11 PSM : Plateforme Min-Énergie	699.6	170	9.1	9.45
BLE : nRF51822 [93]	37.2	42.3	13.2	7.8
BLE : BLE 112 [94]	97.2	90	27.4	3.24
BLE : BlueNRG [95] ²⁴	31.7	29	7.104	6.4
BLE : Plateforme Min-Énergie	24.11	19.26	4.67	3.24
802.15.4 : GREENNET [5, 96]	25.024	19.26	7.104	5.76
802.15.4 : SmartMeshIP [97, 98]	24.11	20.87	4.67	4.32
802.15.4 : TelosB [87]	76	79	41	15
802.15.4 : Plateforme Min-Énergie	24.11	19.26	4.67	3.24

TABLEAU 3.4 – Puissance consommée par différentes plateformes matérielles dans les états consommateurs (voir la définition de ces états dans le tableau 1.1, page 27).

Nous étudions les performances de plusieurs plateformes matérielles dont les puissances consommées dans les différents modes sont résumées dans le tableau 3.4, et évaluons l'impact de ces paramètres de consommation d'énergie sur la durée de vie. Par plateforme matérielle, nous entendons un SoC (*System on Chip*) composé d'un microcontrôleur et d'un module radio.

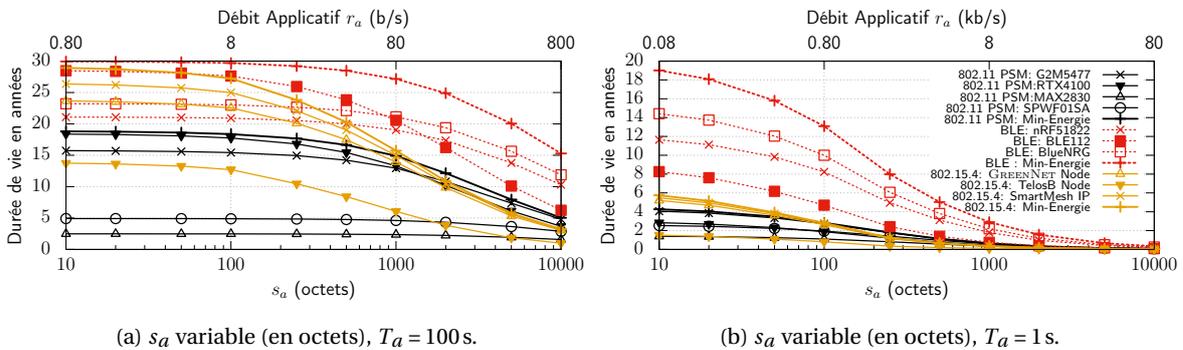


FIGURE 3.4 – Impact de la plateforme matérielle sectionnée sur la durée de vie L_t , pour une énergie initiale $E_0 = 13.5$ kJ. La légende est valide à la fois pour (a) et (b).

24. Si l'on suppose le même CPU STM32 que la plateforme GREENNET

La figure 3.4 compare les durées de vie estimées pour une énergie initiale $E_0 = 13.5\text{kJ}$, une valeur constante de T_a , et une taille de données applicatives générées variant de 10 à 10ko. Cette figure montre un impact important des paramètres matériels : par exemple, la durée de vie des plateformes 802.11b PSM varie de $\sim 1,5$ à ~ 4 ans, pour différents matériels, lorsque $T_a = 1\text{ s}$ (voir Figure 3.4b) et $r_a = 10\text{ o/s}$. L'impact est encore plus important pour $T_a = 100\text{ s}$ (voir Figure 3.4a). De la même manière, on observe d'importantes différences pour les plateformes 802.15.4 et BLE.

Nous pouvons également remarquer que les durées de vie ne sont pas classées par les valeurs de P_{Tx} et P_{Rx} des plateformes matérielles. Par exemple la plupart des matériels 802.11b PSM, fonctionnent plus longtemps que le TelosB, plateforme 802.15.4, bien que la consommation d'énergie du TelosB dans les modes Tx et Rx soit inférieure à celle de toutes les plateformes 802.11b PSM. La consommation d'énergie dans ces états est bien un facteur décisif dans le cas de débits de données élevés (faible T_a), mais, lorsque le débit de données est plus faible (par exemple quand $T_a = 100\text{ s}$), la consommation d'énergie dans les modes Tx et Rx devient moins importante parce que le temps passé avec la radio active devient plus faible. Ainsi, dans le cas de faible trafic, contrairement à la croyance commune, les technologies sont classées, en termes de durée de vie, en fonction de leur valeur de consommation en mode inactif, P_{Sleep} , ce qui s'explique par le fait que les nœuds passent beaucoup plus de temps dans l'état inactif. Par conséquent, les concepteurs de plateforme matérielle devraient, en plus de minimiser la consommation d'énergie dans les modes Tx et Rx , se concentrer également à diminuer la consommation des nœuds lorsqu'ils sont inactifs.

Technologie	BLE	802.15.4	TSCH	802.11ah		
				PSM 11	10_1	8_2
Plateforme	Plateforme Min-Énergie					
$P_{Rayonnée}$ (dBm)	0		18	14		
P_{Tx} (mW)	24.11		699.6	400		
P_{Rx} (mW)	19.26		170	50	130	230
P_{Idle} (mW)	4.67		9.1	30		
P_{Sleep} (μW)	3.24		9.45	7.5		
Sensibilité (dBm)	-93	-97	-85	-98	-69	-58
Portée Théorique						
Modèle à deux rayons (m)	300	400	600	1km	200	100

TABLEAU 3.5 – Paramètres matériels utilisés pour la génération des courbes de comparaison

Pour l'équité de la comparaison, après avoir étudié une gamme de solutions de plateformes matérielles, nous utilisons la valeur de consommation minimale de chacun des modes pour toutes les technologies (dénotée "Min-énergie" dans les tableaux 3.4 et 3.5, ainsi que dans la figure 3.4).

Pour ce faire, nous avons modélisé une plateforme qui offre des performances idéales et l'ensemble de paramètres qui offre le meilleur compromis entre les paramètres P_{Idle} , P_{Tx} et P_{Rx} . Nous partons ici de l'hypothèse que, bien qu'un tel dispositif idéal ne soit pas encore disponible sur le marché, il est possible qu'une plateforme unique offre une telle performance. Les tableaux 3.5 et 3.6 résument les paramètres matériels que nous utilisons pour générer les résultats du reste de l'étude.

De plus, nous avons choisi d'utiliser les mêmes paramètres pour BLE, 802.15.4, et TSCH, car leur conception matérielle est similaire (puissance de transmission et complexité de modulation identiques). Par conséquent, nous adoptons la valeur la plus basse de GREENNET pour P_{Idle} , les valeurs de SmartMeshIP pour P_{Rx} et P_{Idle} , ainsi que la valeur P_{Sleep} la plus basse, celle de BLE 112. Le choix d'une plateforme commune présente l'avantage d'une comparaison équitable des protocoles, plutôt que de comparer les plateformes matérielles.

Cependant, nous ne pouvons pas étendre cette approche à toutes les technologies : nous conservons une grande différence de consommation d'énergie dans les modes Rx et Tx entre les technologies à plus longue portée (802.11b PSM, 802.11ah, LoRa et SIGFOX), et les autres technologies. La différence de puissance rayonnée (0dBm par rapport à 14 et 18dBm) est en effet nécessaire pour atteindre les portées 802.11, et justifie donc une consommation de puissance en émission plus élevée. Différents schémas de modula-

tion expliquent la différence de puissance consommée en réception : la démodulation d'un schéma plus complexe coûte plus d'énergie pour atteindre un taux d'erreur binaire (BER ou *Bit Error Rate*) équivalent.

Pour les technologies à longue portée, nous adoptons les paramètres du tableau 3.6.

Technologie	SIGFOX	LoRa
Débit PHY	100 and 1000 b/s	0.250 and 11 kb/s
Modulation	DBPSK	LoRa (DSSS)
Puissance rayonnée	14 dBm	20 dBm
Sensibilité	-140 dBm	-137 dBm
Portée (Modèle 2-rayons)	~ 11 km	~ 13 km
Plateforme matérielle	Min-énergie ²⁵	Min-énergie ²⁶
P_{Tx}	147 mW	419.6 mW
P_{Rx}	39 mW	44.06 mW
P_{Sleep}	4.32 μ W	4.32 μ W
Temps inter-paquets	0 μ s	Impossible de faire deux transmissions à l'affilée

TABEAU 3.6 – Paramètres utilisés pour SIGFOX et LoRa.

3.3 Comparaison des durées de vie des périphériques

Après avoir soigneusement sélectionné les paramètres matériels des plateformes que nous utilisons pour la comparaison, nous présentons les durées de vie estimées que nous obtenons dans les quatre cas suivants :

1. Évaluation de l'impact du débit d'application en supposant une liaison et une horloge parfaites (sans pertes).
2. Évaluation de l'impact de la variation de la probabilité de pertes, avec une horloge parfaite.
3. Évaluation de l'impact de l'utilisation d'horloges imparfaites, qui génère des dérives d'horloge dans le cas d'une liaison parfaite.
4. Estimation de la durée de vie pour des profils applicatifs typiques, qui prennent en compte à la fois une dérive d'horloge et un PER réalistes.

Enfin, dans une dernière sous-section, nous évaluons la faisabilité avec les différentes technologies des systèmes fonctionnant avec des mécanismes à récupération d'énergie de l'environnement.

3.3.1 Impact de l'applicatif, sans pertes de paquets

La figure 3.5 présente la comparaison de la durée de vie pour des valeurs fixes de la période d'application ($T_a = 1$ jour, 100 s, 1 s et 10 ms), et une variation de la taille des données s_a , qui entraîne une variation du débit r_a . Comme il n'y a pas de perte de paquets, $PER = 0$. Grâce à la séparation de la période applicative T_a et de la taille des données s_a , on peut distinguer la consommation d'énergie d'une application qui génère $s_a = 10$ octets toutes les secondes et une application envoyant $s_a = 600$ octets toutes les minutes, même si la valeur de r_a est identique dans les deux cas ($r_a = 80$ b/s). Dans le premier cas, il y a 60 transmissions de petits paquets par minute, alors que dans le second cas, l'application n'envoie que le nombre minimum de trames, en fonction de la taille maximale des paquets applicatifs de la technologie choisie. Cet effet de mutualisation du surcoût protocolaire explique, par exemple, pourquoi la durée de vie pour un r_a donné (par exemple 80 b/s) est plus élevée dans la figure 3.8c que dans la figure 3.8d pour l'ensemble des technologies présentées.

25. Radio TD1202 [99] avec le microcontrôleur GREENNET et la puissance de consommation de BLE 112 en mode inactif.

26. Radio SX1272 [100] avec le microcontrôleur GREENNET et la puissance de consommation de BLE 112 en mode inactif.

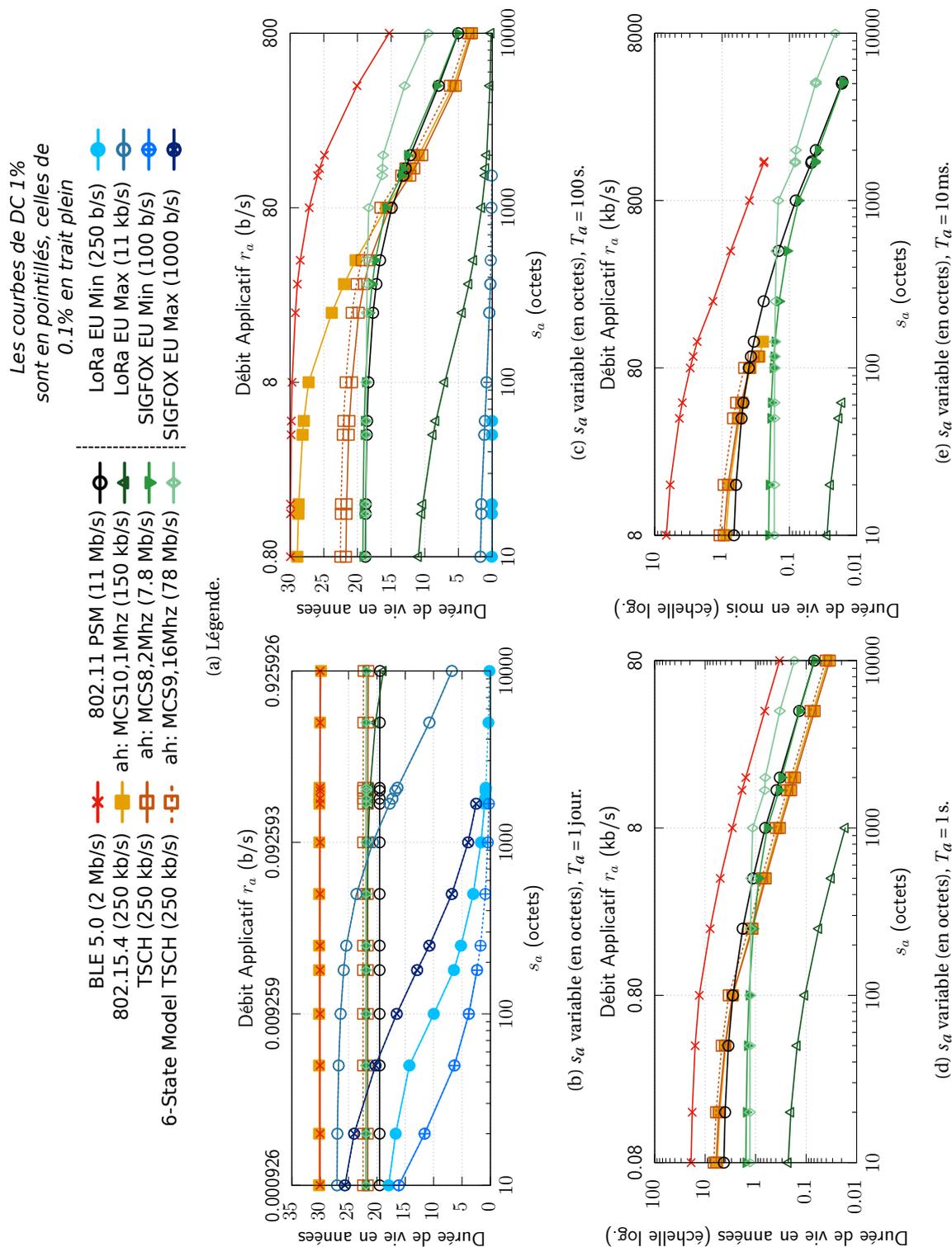


FIGURE 3.5 – Différentes périodes du trafic applicatif, T_a , ainsi que des tailles de données applicatives variables (s_a en octets), permettent de représenter différents débits r_a , et leurs impacts sur la durée de vie des périphériques IoT pour une énergie initiale $E_0 = 13.5$ kJ, un PER nul, et aucune dérive d'horloge. L'axe des abscisses inférieur représente la taille des données applicatives générées (en octets) par intervalle de temps T_a , alors que l'axe des abscisses supérieur affiche le débit de données correspondant r_a (en b/s).

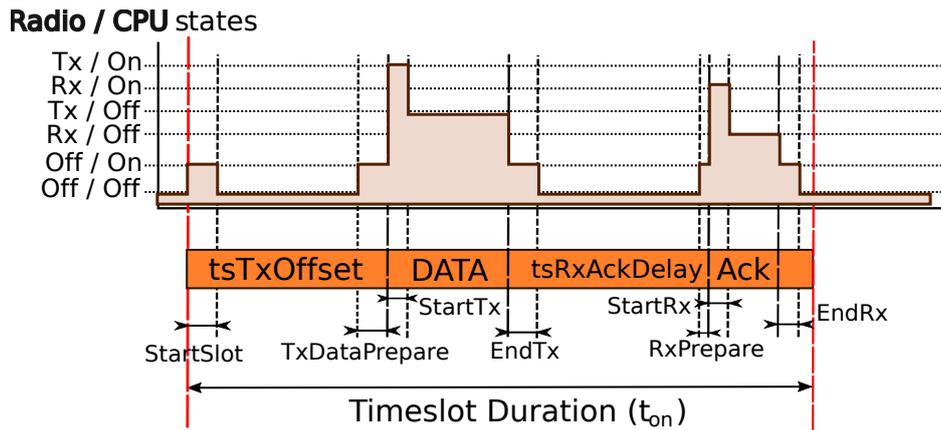


FIGURE 3.6 – Description des différents modes d’opération d’un nœud au sein d’un slot, en utilisant le modèle de consommation à 6 états pour TSCH.

Les figures incluent les résultats pour une technologie TSCH basée sur un modèle plus sophistiqué de consommation d’énergie (nommé modèle TSCH à 6 états, et noté *6-state Model* en lignes pointillées), grâce à deux nouveaux états disponibles, pour modéliser les plateformes 802.15.4 récentes, capables de désactiver leur microcontrôleur lors des transmissions, comme le montre la figure 3.6.

Comme nous l’avons vu au chapitre précédent, pour la bande de fréquences inférieure à 1 GHz, l’ETSI préconise en Europe un certain rapport cyclique en fonction de la sous-bande utilisée (par exemple 0,1% pour 863 – 870 MHz et 1% pour 868 – 868,6 MHz) lorsqu’aucun mécanisme d’agilité en fréquence ou d’écoute avant transmission n’est utilisé. Ainsi, dans notre étude, nous affichons les résultats pour 0.1% avec une ligne pleine et les résultats pour 1% avec une ligne pointillée pour les technologies concernées (SIGFOX et LoRa). La limite de rapport cyclique de 1% détermine la capacité maximale de SIGFOX à 1 kb/s et 100 b/s. De plus, la limitation *marketing* mise en place pour la technologie SIGFOX à 140 paquets par jour ne bride réellement que la capacité de la technologie SIGFOX à 1 kb/s, car elle correspond exactement à la limite de la régulation ETSI (1% d’activité en temps), pour la technologie SIGFOX à 100 b/s.

3.3.1.1 Très faible intensité de trafic ($T_a = 1$ jour)

La figure 3.8b montre la comparaison de la durée de vie pour une très faible intensité du trafic, c’est-à-dire quand la fréquence des données applicatives est de l’ordre de la journée ($T_a = 1$ jour), ce qui correspondant aux cibles d’utilisation des technologies longue portée.

Nous observons que, dans ce contexte, BLE et 802.15.4 obtiennent les meilleures durées de vie. Les performances de LoRa sont également remarquables, avec une durée de vie seulement légèrement plus faible que les normes BLE et 802.15.4 à courte portée.

SIGFOX atteint des durées de vie similaires à celles de LoRa seulement pour des valeurs de r_a extrêmement faibles, ce qui s’explique par le fait que le débit binaire disponible de SIGFOX est 11 fois plus faible que celui de LoRa.

SIGFOX et LoRa consomment plus d’énergie à mesure que la taille des données s_a augmente : leurs durées de vie diminuent parce qu’ils ne peuvent gérer que des paquets de petite taille. Ainsi, la fragmentation se produit souvent, ce qui augmente le surcoût protocolaire, et par conséquent la consommation d’énergie. De plus, pour atteindre les grandes portées ciblées, ils utilisent une puissance de transmission élevée, ce qui contribue à augmenter la consommation d’énergie lorsqu’ils doivent transmettre pendant de longues durées.

Comme le montrent les figures 3.8b et 3.8c, SIGFOX et LoRa souffrent des limitations de capacité dues à la réglementation ETSI. En outre, la réglementation ETSI exige que les nœuds transmettent moins d’un certain nombre de trames consécutives par heure. Ainsi, les nœuds SIGFOX ou LoRa qui doivent transmettre pendant plus de 3,6 s (pour un ratio d’activité de 0,1%) devront retarder la transmission de certains paquets jusqu’à l’heure suivante, ce qui impacte également la latence de transmission. Un nœud SIGFOX nécessite trois transmissions pour envoyer une trame de données. Dans ce cas, la limite de messages par heure conduit à une incapacité de respecter la limitation du rapport cyclique dans la bande de 863 – 870 MHz avec

un débit de 100b/s. Par conséquent, nous recommandons l'utilisation de la sous-bande pour laquelle le DC est de 1%, afin de concurrencer les autres technologies de l'IoT. SIGFOX à 100b/s n'utilise d'ailleurs quasiment pas les bandes de fréquences réglementées à 0.1%.

802.11ah obtient des résultats nettement moins bon que LoRa pour les données de petite taille, alors qu'il devient plus performant que LoRa pour des tailles de données plus élevées. Il est également plus performant que le 802.11b PSM et obtient des performances similaires à celles de TSCH. Notons que toutes les courbes 802.11ah sont superposées dans la figure 3.8c : en rappelant que nous avons utilisé la même valeur P_{Sleep} pour les trois technologies 802.11ah (voir tableau 3.5), nous comprenons la superposition des courbes, car la valeur de P_{Sleep} est le principal facteur impactant la durée de vie, pour une technologie synchrone et des débits applicatifs très faibles (voir paragraphe 3.2.2).

3.3.1.2 Faible intensité de trafic ($T_a = 100$ s)

Pour un trafic applicatif sporadique, BLE atteint la durée de vie la plus longue, suivi de près par les performances de la technologie 802.15.4 pour les petites tailles de données (voir figure 3.8c). Pour les plus grandes valeurs de s_a , on peut observer sur la figure 3.8c que la variante américaine de 802.11ah avec 16 MHz de bande passante devient la solution la plus intéressante après BLE. Dans ces conditions, l'importante surcharge protocolaire des paquets 802.11ah est atténuée par la taille de données maximale par paquet plus importante. Ainsi, pour le 802.11ah, la fragmentation se produit moins souvent que pour le BLE, ce qui diminue le ratio surcoût protocolaire/données transmises, pour le 802.11ah.

Contrairement aux résultats de Tozlu concernant les technologies 802.15.4 et 802.11b PSM [57, 72], la figure 3.8c montre que 802.15.4 consomme moins d'énergie dans la plupart des cas de faible trafic de données (jusqu'à 80b/s) et obtient des durées de vie beaucoup plus longues. Nous pouvons expliquer cette différence de résultat par la consommation d'énergie plus faible de notre plateforme 802.15.4 par rapport à celle de TelosB utilisée dans l'article de Tozlu : comme expliqué précédemment, nous avons choisi la plateforme la plus efficace pour calculer la durée de vie de 802.15.4 (voir tableau 3.5). La durée de vie du 802.11b PSM est supérieure à celle du 802.15.4 uniquement pour les très grands paquets (supérieurs à 1,5 ko), ce qui n'est pas une taille habituelle pour les applications IoT.

3.3.1.3 Trafic d'intensité moyenne et de haute intensité ($T_a = 1$ s, 10 ms)

Pour le trafic d'intensité moyenne et élevée (voir les figures 3.8e et 3.8d, où l'axe des ordonnées est logarithmique), BLE obtient la durée de vie la plus longue, bien que l'on s'attende traditionnellement à ce que 802.15.4 soit meilleur. Nous pouvons identifier principalement deux raisons pour ce résultat :

- La couche PHY 802.15.4 est moins efficace que la couche BLE en raison d'un facteur d'étalement de modulation de 8, qui réduit le débit binaire disponible de 802.15.4 de 2Mb/s à 250kb/s, alors que la couche BLE utilise le débit maximal de 2Mb/s. De cette manière, le débit binaire du BLE est 8 fois plus élevé que celui de 802.15.4, ce qui conduit à une consommation d'énergie plus faible.
- La fragmentation en 802.15.4 a lieu lorsque les paquets atteignent une taille de données utiles de 120 octets, alors que BLE commence à fragmenter à 245 octets. Étant donné que le surcoût protocolaire pour une trame de données ne dépend pas de la quantité de données utiles qu'elle transporte, 802.15.4 est moins efficace que BLE pour les paquets plus grands que 120 octets.

Pour valider ces hypothèses, nous avons artificiellement modifié dans l'analyseur le débit binaire de 802.15.4 à 2Mb/s, et réglé la taille maximale des trames des données à 245 octets. Dans cette configuration, 802.15.4 se rapproche de BLE (comme le montre la figure 3.7).

De plus, la figure 3.7 montre la variation en termes de performance énergétique du BLE lorsque l'on sélectionne différents modes spécifiés dans la récente norme Bluetooth 5.0 [6] : le débit de données le plus élevé conduit à une consommation d'énergie plus faible parce qu'il conduit à un temps minimal passé dans les modes consommants de la radio, pour un surcoût protocolaire similaire (voir le format des paquets de données BLE à la page 40).

Nos résultats confirment de plus les travaux de Siekkinen *et al.* [80], qui ont montré que, dans la phase stationnaire, le 802.15.4 transmet moins de données que le BLE pour la même quantité d'énergie. Ils ont également obtenu leurs résultats en utilisant une plateforme matérielle équivalente dans le cas de BLE et de 802.15.4.

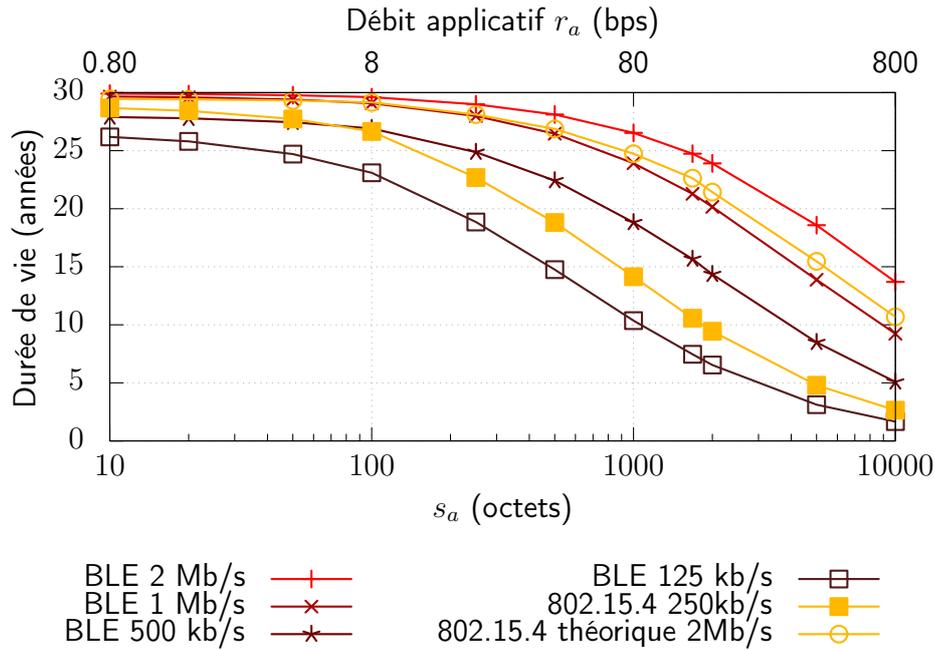


FIGURE 3.7 – Durées de vie estimées pour les différents débits disponibles du standard BLE 5.0, et comparaison avec un 802.15.4 théorique au débit de 2Mb/s (s_a variable, $T_a = 100$ s).

BLE apparaît donc comme la meilleure technologie en termes de consommation d'énergie pour les débits moyens et élevés. Néanmoins, il souffre d'une limitation de capacité vis-à-vis des transferts de grandes tailles de données applicatives. Rappelons que le cas présenté sur la figure 3.8e correspond, pour certaines technologies, à des nœuds allumés en continu, lorsqu'elles fonctionnent à leur capacité maximale (ce qui est matérialisé par la fin de la courbe d'une technologie).

Pour $s_a > 100$ octets, 802.11b PSM surpasse 802.15.4 (cf. figure 3.8e) et devient ainsi la deuxième technologie la plus économe en énergie, et ce jusqu'à une taille de donnée de $s_a > 500$ octets, à partir de laquelle seule la version américaine du 802.11ah (MCS9, 16MHz) avec des débits de modulation élevés et une large bande passante devient la technologie la plus économe en énergie.

Nous observons que les performances de 802.11ah avec une bande passante de canal plus faible, même avec des indices de modulation élevés tels que le débit maximum européen (MCS8, 2MHz, 7,8Mb/s), sont inférieures ou équivalentes à celle du 802.11b PSM en raison de la différence de débit disponible.

Nous pouvons également observer que la durée de vie de 802.11ah 10_1 (MCS 10, 1MHz, 0.15 Mb/s) est de manière récurrente moins longue que celles des autres technologies. Pour obtenir une durée de vie comparable à celle de BLE ou 802.15.4, il faudra abaisser la puissance rayonnée de cette variante de 802.11ah à 0 dBm, ce qui diminuera nécessairement la portée, un des objectifs majeurs du 802.11ah.

3.3.2 Robustesse aux mauvaises conditions de canal

La figure 3.8 présente l'impact d'une probabilité de perte de paquet de $PER = 0,2$ sur la durée de vie des périphériques IoT. Nous avons choisi la valeur de 20% de perte de paquets, car il représente un impact significatif des conditions de canal et des interférences sur la qualité de transmission. Il est aussi par ailleurs généralement adopté comme seuil en deçà duquel une liaison sans fil est considérée comme de mauvaise qualité, et ne sera alors pas utilisée pour réaliser des transmissions. On ne peut constater qu'une légère variation des valeurs absolues de la durée de vie : en effet, les positions relatives des courbes ne changent pas. Par exemple, bien que la durée de vie du BLE pour $r_a = 800$ b/s avec $T_a = 100$ s, passe de ~ 15 ans à $\sim 13,5$ ans, elle reste la plus longue durée de vie, et donc la technologie la moins consommatrice.

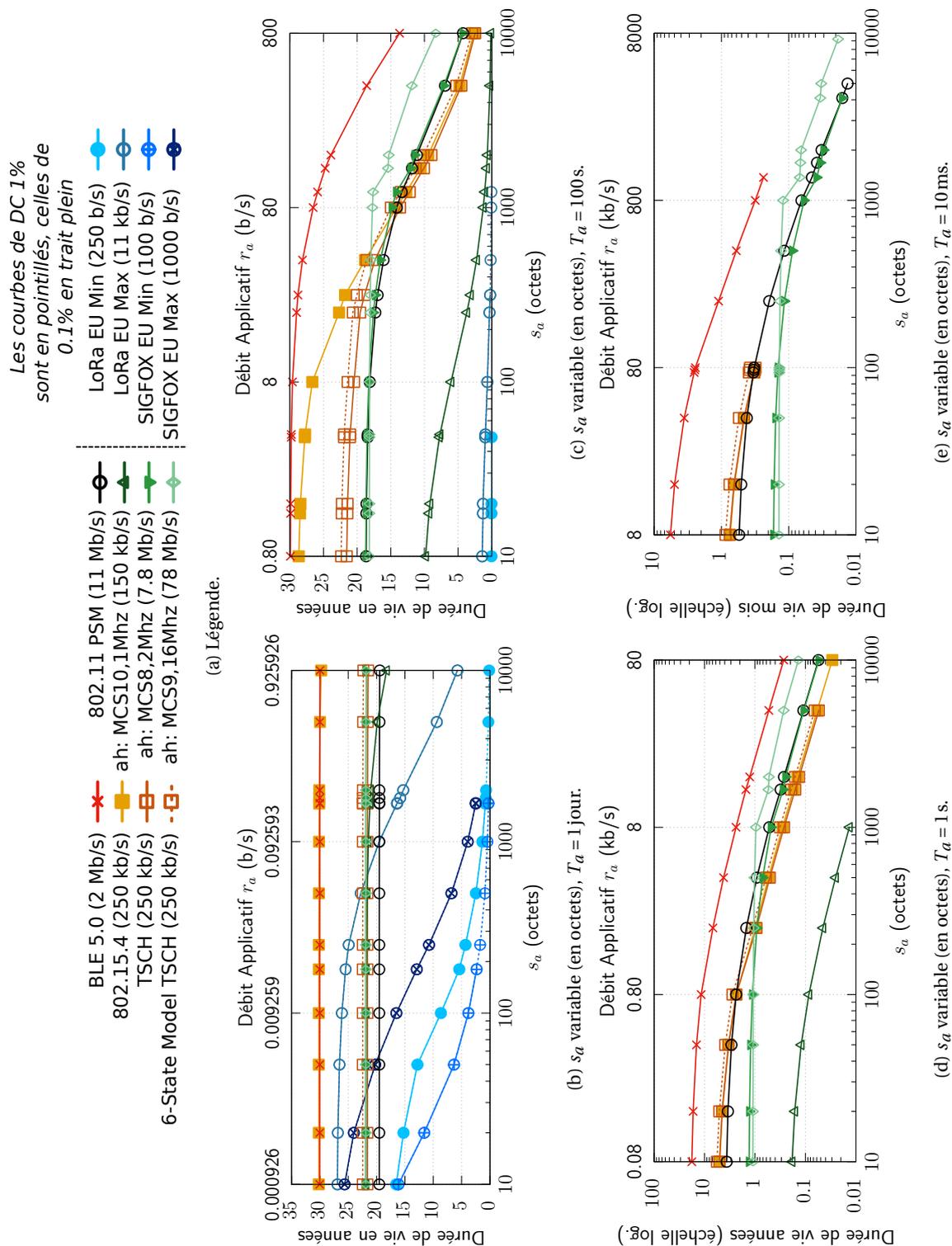


FIGURE 3.8 – Durées de vie pour des débits applicatifs variables (r_a), un taux d'erreur de paquet moyen de 20%, une énergie initiale $E_0 = 13.5$ kJ, et des horloges parfaites.

3.3.3 Impact de la dérive d'horloge

Pour évaluer l'impact de la dérive de l'horloge et de sa prise en charge, nous supposons maintenant que les périphériques ont des horloges imparfaites $C(t)$, dont la dérive est bornée par Δ . Une valeur typique pour les horloges utilisées dans les réseaux de capteurs est de $\Delta = 40$ ppm.

Comme à la fois l'horloge de l'émetteur et du récepteur peuvent dévier de l'horloge parfaite, la différence maximale entre les horloges des deux périphériques (δ_{\max}) est limitée par 2Δ .

Ainsi, comme présenté dans le chapitre précédent (cf. page 52), pour un intervalle T_{ci} tel que $T_{ci} = 100$ s et $\Delta = 40$ ppm, la durée nécessaire pour compenser la dérive d'horloge est de $GT = 8$ ms.

Le cas où l'on suppose des horloges parfaites correspond aux valeurs que nous avons utilisées pour générer les figures 3.5, 3.8 et 3.4.

Les trois autres cas de dérive relative des horloges considèrent le dimensionnement d'un intervalle de garde GT , tolérant une dérive d'horloge relative maximale de $\delta_{\max} = 2\Delta$, ce qui permet de déterminer la durée de l'intervalle de garde GT (voir l'équation 2.6).

La dérive d'horloge peut parfois se compenser pour revenir au cas idéal du point de vue de la consommation d'énergie, comme le montre la figure 2.3 (page 54), le cas $\delta = -2\Delta$ est équivalent au cas idéal. Les figures 3.5, 3.8 et 3.4 nous donnent donc également les valeurs pour $\delta = -2\Delta$.

Pour estimer l'impact de l'intervalle de garde et de la dérive de l'horloge sur la consommation d'énergie, nous avons choisi de générer les résultats de la figure 3.5 pour le pire cas de dérive d'horloge, c'est-à-dire une dérive de $+2\Delta$ (avec $\Delta = 40$ ppm), pour laquelle le nœud récepteur doit rester éveillé pendant l'intervalle le plus long : la figure 3.9 montre les résultats correspondants.

Nous pouvons voir que, pour toutes les technologies synchrones, le temps perdu en mode Rx pendant l'intervalle de garde GT conduit à des durées de vie considérablement plus courtes, plus de 20% plus courtes pour la technologie BLE et $T_a = 1$ jour (environ 23,5 ans contre 30 sans GT). Nous pouvons également observer que, pour une intensité de trafic plus élevée, l'influence de ce facteur est moindre, de sorte que les figures 3.9d et 3.9e sont quasiment identiques aux figures 3.8d et 3.8e. Dans le cas de trafic relativement élevé, c'est-à-dire de faible T_a , cela réduit la longueur de l'intervalle GT , puisque GT est proportionnel à T_{ci} qui dépend lui-même directement T_a .

On peut également noter que l'impact de la dérive de l'horloge sur la durée de vie du TSCH est plus faible que pour les autres technologies : la perte en durée de vie est seulement de 10% pour $T_a = 1$ jour avec le modèle amélioré de TSCH à six états. Cela s'explique par le fait que TSCH spécifie un intervalle de garde par défaut (dans le standard) de 1 ms pour compenser une horloge imparfaite.

Pour une intensité de trafic très basse ($T_a = 1$ jour), LoRa et SIGFOX obtiennent les meilleures durées de vie, car ce sont les mêmes que dans le cas des horloges parfaites, grâce à leur fonctionnement asynchrone : en effet, les périphériques fonctionnant avec ces technologies n'ont pas besoin de se réveiller à l'avance pour compenser la dérive de l'horloge, puisqu'ils n'ont pas de rendez-vous avec le destinataire de la transmission.

Ces résultats invitent à penser à un schéma de synchronisation adaptatif : au lieu de consommer de l'énergie en mode Rx durant l'intervalle de garde et en maintenance de synchronisation via les messages de KA (Maintien vivant — ou *Keep-Alive*), il peut parfois être avantageux pour un nœud de se dissocier, puis de se reconnecter uniquement lorsque cela est nécessaire pour un échange de données. Par exemple, pour maximiser la durée de vie d'un périphérique, un nœud Wi-Fi se déconnecterait du réseau et passerait en mode inactif, puis ne se reconnecterait qu'en cas de besoin.

Néanmoins, il existe déjà des mécanismes de compensation de la dérive d'horloge, de manière réduire la longueur de l'intervalle garde GT , et ainsi la consommation d'énergie liée à la compensation des dérives [101].

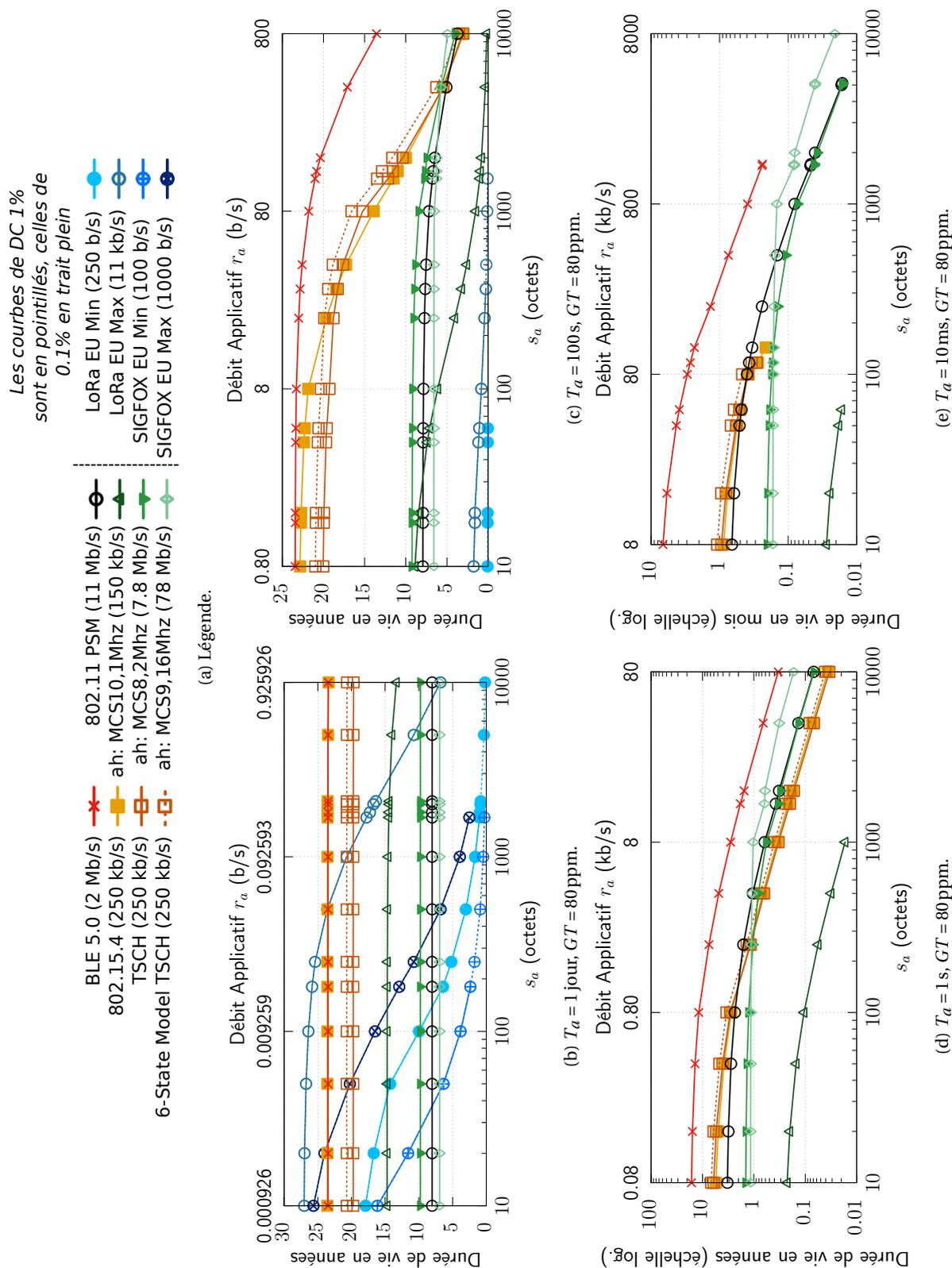


FIGURE 3.9 – Durées de vie pour différents débits applicatifs r_a et une prise en compte d'une dérive d'horloge relative maximale $\delta_{\max} = +2\Delta = 80$ ppm, pour une énergie initiale $E_0 = 13.5$ kJ, et un taux d'erreur de paquet nul (PER = 0).

3.3.4 Durées de vie pour des applications représentatives

L'étude de ce chapitre avait pour but premier d'aider les ingénieurs et les développeurs d'applications IoT à sélectionner la meilleure technologie pour un contexte applicatif donné, en supposant des conditions de fonctionnement optimales d'un dispositif IoT avec deux piles AAA.

Dans le but de faciliter le choix d'une technologie ou d'une autre, nous avons sélectionné des applications représentatives de différents usages de l'IoT, et correspondantes aux différents cas de contextes applicatifs présentés dans ce chapitre.

Ainsi, nous avons choisi, pour le contexte de très bas débits applicatifs, de considérer la mesure du niveau d'enneigement, par exemple utilisée dans le contexte des villes intelligentes. Pour ce type d'application, nous considérons un trafic de 50 octets par jour.

Pour les applications de bas débits, nous considérons le relevé de la température (50 octets de données, en prenant en compte la sécurité) d'une pièce toutes les 100 secondes, dans le but de monitorer l'utilisation correcte d'un système de chauffage d'une maison intelligente, par exemple.

Un usage de débits applicatifs moyens pour l'IoT pourrait être le relevé de la position d'un coureur dans le but de réaliser une analyse de son parcours à la fin de son entraînement. Pour cet usage, nous considérons une remontée d'information de 50 octets par seconde.

Enfin, nous considérons le cas des trafics applicatifs relativement élevés pour des communications IoT en considérant le streaming audio d'un téléphone à destination d'enceintes connectées, par exemple. Pour ce contexte, nous considérons le débit préconisé par la norme de compression audio G.711 de 80 octets toutes les 10ms.

Notre analyseur calcule les durées de vie pour les applications données et leurs débits correspondants (s_a/T_a) : les résultats de nos estimations de durées de vie sont présentées dans le tableau 3.7.

Application (s_a/T_a)	Streaming Audio G. 711 ²⁷ (80 octets/10 ms)	Traqueur d'activité sportive (50 octets/1 s)	Relevé de température (50 octets/100 s)	Mesure du niveau d'enneigement (50 octets/ jour)
BLE	~ 90 jours	10.5 ans	23.5 ans	24 ans
802.15.4	15 jours	5 ans	23 ans	24 ans
TSCH	21 jours	5.5 ans	20.5 ans	20.5 ans
802.11b PSM	15 jours	3 ans	8 ans	8 ans
802.11ah 8_2	5 jours	1 an	9 ans	10 ans
SIGFOX 1%	∅	∅	∅	21 ans
LoRa 1%	∅	∅	1 ans	27 ans

TABLEAU 3.7 – Durées de vie pour des applications représentatives, en supposant une dérive d'horloge maximale de 40ppm, un taux de perte de paquet de 20% (PER = 0.2), et les débits maximaux disponibles en Europe.

3.3.5 Faisabilité des plateformes à récupération d'énergie

Au sein de cette section, nous abordons la question des dispositifs de l'IoT qui récupèrent de l'énergie dans l'environnement qui les entoure (via des panneaux solaires par exemple), et la stocke dans une batterie de faible capacité (environ 20mAh). Nous prenons l'exemple d'un nœud GREENNET, avec un panneau solaire de ~ 18cm² [5] avec les valeurs de puissances collectées de l'environnement (en intérieur et en extérieur) présentées dans le Tableau 3.8.

Environnement	Intensité Lumineuse	Puissance collectée
En extérieur à midi	15% de la radiation solaire à la surface de la terre	240 mW [102]
A l'intérieur, à 5 cm de la source	8000 lx	3.2 mW [103]
A l'intérieur, à 2 m de la source	300 lx	96 μW [103]

TABLEAU 3.8 – Puissances collectées de l'environnement.

27. Norme de compression audio à la base du transport de la voix sur le réseau téléphonique commuté

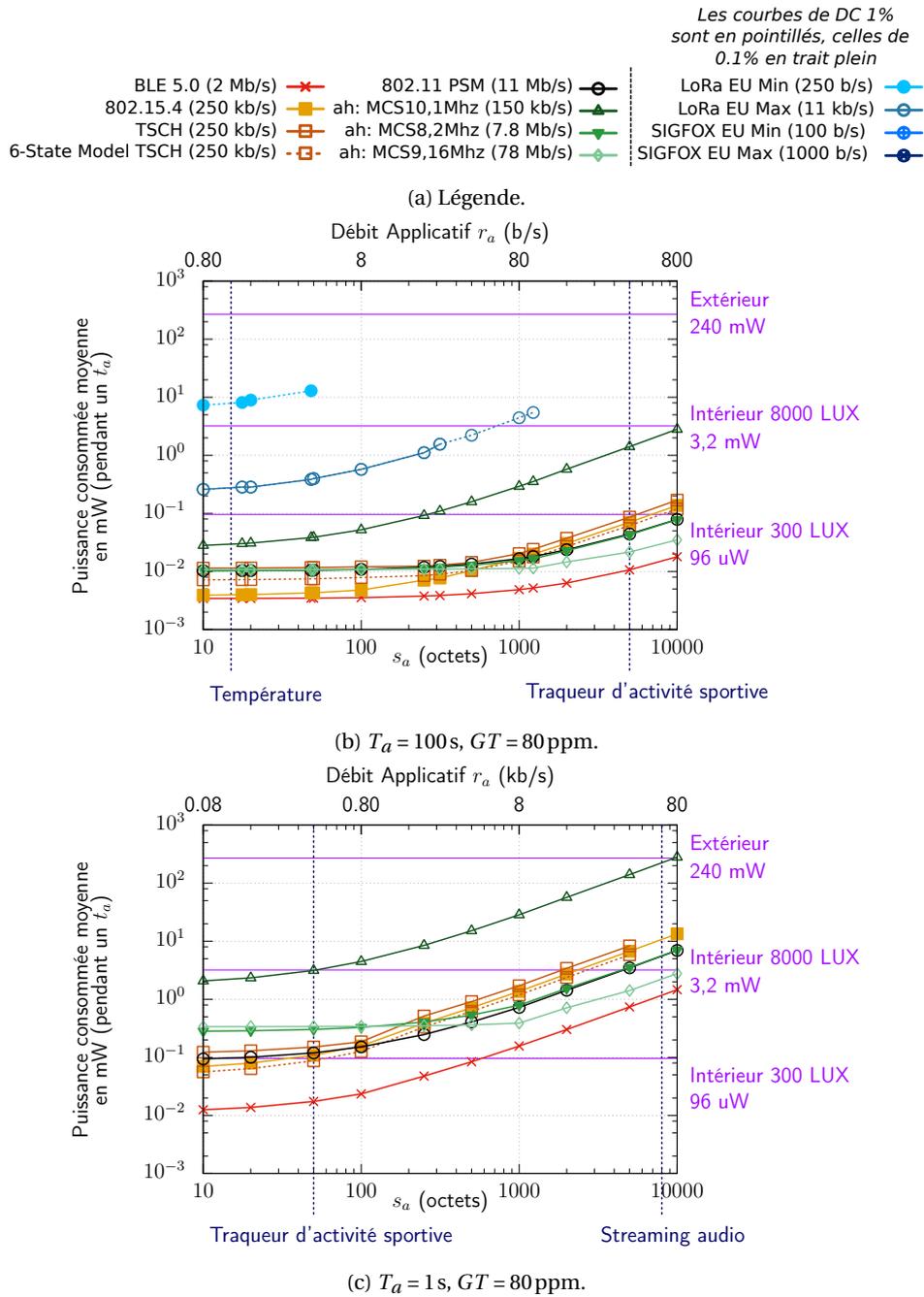


FIGURE 3.10 – Puissance consommée moyenne par T_a , PER = 0.2.

La figure 3.10 présente la puissance moyenne consommée par intervalle T_a pour toutes les technologies étudiées. Sur l'axe des ordonnées de droite, nous avons marqué le niveau de la puissance récupérable pour une intensité lumineuse donnée. Ainsi, toutes les courbes dont les points se situent en dessous de la ligne horizontale spécifiant un niveau de puissance récupérable dans l'environnement sont théoriquement réalisables, car cela signifie qu'un appareil aura récupéré de l'environnement une puissance suffisante pour fonctionner pendant l'intervalle T_a .

Nos résultats montrent que seuls BLE et 802.15.4 peuvent fonctionner avec l'intensité lumineuse de 300lx, une valeur typique en environnement intérieur. L'intensité lumineuse de 8000lx est suffisante pour toutes les autres technologies, mais cela signifie que les appareils doivent être placés à 5 cm de la source lumineuse. De plus, nous pouvons voir que le scénario en plein soleil, à l'extérieur et à midi permet de faire fonctionner toutes les technologies, mais que dans ce cas, les applications seront vulnérables aux variations du niveau d'intensité lumineuse à l'extérieur.

3.3.6 Limites de l'étude

Notre objectif de développement d'un outil analytique conduit à quelques simplifications, mais les comparaisons avec d'autres études montrent une précision suffisante des résultats vis-à-vis du but initial. Cependant, dans le but de prendre en compte l'ensemble des facteurs influants sur la consommation d'énergie, il serait nécessaire de développer un simulateur à part entière.

De plus, la comparaison prend en compte des configurations réseau très diverses, dont les portées varient de quelques mètres (BLE) à plusieurs kilomètres (LoRa, SIGFOX), qui offrent des robustesses de transmission variées, en fonction du système de modulation et de codage inhérent à chaque technologie, et de la puissance de transmission choisie. Nous sommes conscients que la consommation d'énergie n'est qu'un aspect à prendre en compte lors du choix de la bonne technologie pour une application donnée. D'autres paramètres tels que le débit, le délai, la fiabilité, la portée, la puissance rayonnée et le coût de déploiement peuvent également être pris en compte. Par exemple, il n'est pas logique de proposer le déploiement du BLE (courte portée) au lieu du SIGFOX (longue portée) lorsqu'une application nécessite une couverture étendue. De même, il y a toujours un compromis entre la puissance de transmission, la portée et le taux d'erreur binaire qui en résulte.

Cependant, pour sélectionner la meilleure technologie compte tenu des exigences de l'application (performance, consommation d'énergie) dans un contexte de communication spécifique (portée, conditions de canal), il est nécessaire de comprendre les limites énergétiques de chaque technologie et d'identifier les facteurs qui influencent la durée de vie du nœud pour un niveau de performance donné. Le tableau final du premier chapitre (page 48) fournit des renseignements sur ces autres aspects.

Notre analyse fait également quelques simplifications qui impactent la précision des estimations de consommation d'énergie : nous considérons un modèle probabiliste simple de la perte de paquets, un débit binaire constant malgré la variation de la charge applicative, et une topologie à un seul lien.

Le modèle de consommation d'énergie à quatre états n'étant pas suffisamment précis pour certaines technologies, nous avons ajouté un modèle à six états pour le TSCH pour mieux représenter le comportement de cette technologie. Puisque certaines plateformes matérielles ne fonctionnent pas avec le modèle à 6 états (impossibilité d'éteindre le microcontrôleur pendant les transmissions radio), nous ne pouvons cependant pas l'utiliser pour toutes les technologies. Ainsi, d'autres recherches en termes de développement matériel sont encore nécessaires pour tirer parti de l'arrêt du microcontrôleur pendant l'activité radio sans dépenser trop d'énergie lors de la phase de réveil du CPU.

Par ailleurs, le modèle matériel utilisé suppose des transitions instantanées entre les états. Dans la réalité, ces transitions prennent un certain temps, ce qui entraîne une consommation d'énergie supplémentaire.

Enfin, nous utilisons un modèle linéaire de décharge de la batterie, mais le processus réel est plus complexe. Cependant, le modèle linéaire n'est pas très éloigné des modèles récursifs complexes [104] et nous avons compensé cette différence en tenant compte des fuites de courant paramétrables.

Néanmoins, l'analyseur présenté dans ce chapitre donne un moyen d'évaluer rapidement la consommation d'énergie des principaux réseaux sans fil de l'IoT. Il modélise de manière réaliste le comportement des principales couches du protocole et prend en compte plusieurs facteurs importants pour l'évaluation de la consommation d'énergie : le surcoût protocolaire, y compris la fragmentation, l'énergie consommée en état inactif, les pertes de paquets, la synchronisation temporelle et la variation du trafic applicatif.

Conclusion

Ce chapitre a présenté de nombreuses contributions. Tout d'abord, nous étendons à toutes les technologies considérées un modèle énergétique réaliste qui prend en compte le comportement des couches PHY et MAC. De plus, nous développons un analyseur de durée de vie disponible sous forme de logiciel libre, qui constitue un outil permettant d'estimer la durée de vie d'un réseau donné à partir des principaux paramètres matériels et/ou MAC. Nous l'utilisons ensuite pour étudier l'impact des protocoles de couches supérieures (fragmentation, surcharge protocolaire), la synchronisation temporelle et les pertes de paquets sur la durée de vie de différentes plateformes matérielles. Enfin, nous discutons de l'adéquation des technologies pour des applications représentatives et étudions la faisabilité des solutions de récupération d'énergie.

À travers cette étude, nous obtenons ainsi plusieurs nouveaux résultats :

1. En supposant des horloges idéales, nous avons montré que :
 - BLE offre la meilleure durée de vie pour toutes les intensités de trafic dans sa gamme de capacités.
 - Pour une intensité de trafic ultra basse, LoRa offre des performances remarquables, avec une durée de vie très légèrement inférieure à celle des technologies courte portée BLE et 802.15.4.
 - 802.11ah est nettement moins performant que LoRa pour des petites tailles de données applicatives, alors qu'il est meilleur pour des données de plus grande taille.
 - Pour des intensités de trafic faible et moyenne, le classement relatif des technologies est le suivant : BLE, 802.15.4, TSCH et 802.11, ces dernières devenant des solutions intéressantes pour des volumes de données plus importants.
 - Pour une intensité de trafic élevée, 802.11ah avec la plus large bande passante disponible, et 802.11b PSM sont les technologies les moins consommatrices lorsque BLE atteint ses limites de capacité.
2. Lorsque l'on prend en compte les mécanismes pour compenser la dérive d'horloge dans le cas des technologies synchrones, les technologies asynchrones LoRa et SIGFOX permettent d'obtenir les meilleures durées de vie pour un trafic à faible intensité.
3. Contrairement à la croyance répandue selon laquelle la consommation d'énergie en état inactif est négligeable, le P_{sleep} devient un facteur décisif de la durée de vie des périphériques, pour les faibles intensités de trafic, tandis que la consommation dans les états Tx et Rx reste les critères majeurs de consommation pour une intensité de trafic plus élevée.
4. La prise en compte de l'énergie dépensée des retransmissions de trames de données dues à des trames corrompues et aux collisions ne change pas le classement relatif des technologies.
5. Les technologies actuelles à longue portée fonctionnent dans la bande de fréquences sous-1 GHz, avec d'importantes limitations du ratio d'activité imposé par l'ETSI, ce qui réduit leur capacité, et ne leur permet pas de supporter des périphériques à récupération d'énergie.

Au-delà de l'analyse présentée dans cet article, la comparaison a également renforcé notre intérêt pour l'interopérabilité et la coexistence de différentes technologies. Nous avons montré que le coût du maintien de la synchronisation, est parfois trop élevé. Par conséquent, une question de recherche intéressante est d'améliorer la durée de vie du dispositif en proposant un schéma adaptatif pour la synchronisation du temps : la technologie BLE peut déjà transmettre des données dans les paquets ADV, en mode non connecté, il peut donc être bénéfique de se déconnecter d'un maître et fonctionner en mode non connecté pour les applications de trafic rare. Cette approche pourrait également réduire la consommation d'énergie dans d'autres technologies.

Une interopérabilité des modes de communication synchrone et asynchrone permettrait donc de proposer une solution avec les deux propriétés suivantes : à la fois une faible consommation d'énergie des nœuds terminaux, asynchrones, qui ne consomment de l'énergie que lorsque l'envoi de paquet est nécessaire, mais aussi une faible consommation d'énergie aussi du point de vue des nœuds à l'intérieur du réseau (les relais), synchrones, qui pourraient conserver un mode de consommation à faible rapport cyclique.

Ainsi, cette dernière conclusion de notre analyse vient renforcer la nécessité d'une interopérabilité entre deux modes de communication, synchrone et asynchrone, non seulement pour permettre la cohabitation de besoins applicatifs différents, mais aussi pour diminuer la consommation énergétique globale.

Pour ce faire, comme mentionné dans le chapitre Discussion, nous proposons tout d'abord de faire cohabiter ces deux modes de communication (synchrone et asynchrone), au sein d'une même technologie, en permettant à l'un des nœuds synchrones de se désassocier du réseau pour y revenir uniquement s'il a besoin de transférer des données applicatives. Nous avons vu qu'un tel fonctionnement est d'ores et déjà réalisable, mais trop coûteux en énergie du fait du coût d'attache au réseau. C'est pourquoi, dans le chapitre suivant, nous étudions les possibilités de réduire la consommation énergétique lors de cette phase d'attache au réseau.

Réduction du coût d'attache au réseau

Sommaire

3.1 Principe de l'analyse de consommation énergétique	73
3.1.1 Hypothèses et méthodologie	73
3.1.2 Modèle de consommation énergétique	74
3.1.3 Modèle applicatif et calcul de la durée de vie	75
3.1.4 Modèle des pertes de paquets	76
3.1.5 Évaluation de l'analyseur	77
3.2 Impact des paramètres matériels sur les résultats	78
3.2.1 Paramètres propres à chaque technologie	78
3.2.2 Choix des paramètres	79
3.3 Comparaison des durées de vie des périphériques	81
3.3.1 Impact de l'applicatif, sans pertes de paquets	81
3.3.1.1 Très faible intensité de trafic ($T_a = 1$ jour)	83
3.3.1.2 Faible intensité de trafic ($T_a = 100$ s)	84
3.3.1.3 Trafic d'intensité moyenne et de haute intensité ($T_a = 1$ s, 10 ms)	84
3.3.2 Robustesse aux mauvaises conditions de canal	85
3.3.3 Impact de la dérive d'horloge	87
3.3.4 Durées de vie pour des applications représentatives	89
3.3.5 Faisabilité des plateformes à récupération d'énergie	89
3.3.6 Limites de l'étude	91

Introduction

Alors que la première partie de ce rapport nous a permis de faire le bilan des technologies et standards existants pour l'application des réseaux de capteurs sans fil à faible consommation d'énergie, ainsi que de déterminer notre cadre de travail quant à l'interopérabilité, le chapitre précédent a identifié les points forts et les points faibles de chacune des technologies étudiées dans ce manuscrit, du point de vue de leur impact sur la consommation énergétique, et ainsi orienter le choix d'une technologie à un instant donné, lorsque plusieurs technologies sont disponibles. Cette étude a mis en avant le fait que le maintien de la synchronisation au réseau représente une opération coûteuse en énergie, dont le prix n'est pas négligeable dans les contextes de très faible trafic. Ainsi, un nœud aurait parfois plus d'intérêt à se désynchroniser du réseau, et le rejoindre uniquement lorsqu'il aura besoin de communiquer.

Nous proposons ainsi d'exploiter des attaches successives au réseau, à la fois pour éviter de maintenir une synchronisation coûteuse, dans certains contextes applicatifs, mais aussi, car cela permettrait, comme nous l'avons montré dans le chapitre Discussion de réaliser une interopérabilité multimode synchrone et asynchrone au sein d'une même technologie, permettant de relayer un trafic asynchrone avec une faible latence tout en maintenant une consommation globale peu élevée.

Cependant, nous avons vu que coût de la procédure d'attache au réseau est, de la manière standardisée par le mode TSCH du 802.15.4, trop élevé. Il est donc nécessaire de réduire ce coût pour favoriser l'utilisation d'un mode de communication asynchrone.

Pour cette étude, nous avons choisi de s'intéresser aux réductions des coûts d'attache au réseau de la technologie synchrone TSCH. En effet, nous supposons que tous les nœuds du réseau sont également contraints en énergie. Dans une telle situation, TSCH est la seule solution technique qui répartit les cycles d'activités de chacun des nœuds dans le temps et permet à la fois l'exploitation de multiples canaux fréquentiels pour lutter contre les interférences, et le multisaut pour augmenter sa portée.

En effet, si la récente spécification de BLE *Mesh* [105] permet d'apporter un mode de communication multisaut au sein des réseaux BLE, cette spécification repose sur un mécanisme d'inondation du réseau par les messages de communications. Dans cette spécification, les transmissions exploitent un mode de communication asynchrone, qui facilite un mode de faible consommation pour certains périphériques. Cependant, pour mettre en place ce mode de communication asynchrone, les appareils nécessitant un support de faible puissance doivent s'associer à un dispositif disponible de manière continue qui stocke et relaie les messages en leur nom, un *proxy*, à l'aide du concept d'amitié²⁸ (— ou *Friendship* dans la spécification *Mesh*) [105]. Ainsi, ce mode de communication ressemble fortement aux communications Thread, et ne correspond pas au contexte de réseau contraint en énergie en chacun de ses nœuds ciblé ici.

C'est pourquoi nous travaillerons avec la technologie TSCH, bien que ses performances énergétiques à un saut ne soient pas aussi bonnes que celles de BLE, comme l'a montré l'étude du chapitre précédent.

Par ailleurs, vis-à-vis de la technologie synchrone 802.15.4 avec balises, le TSCH fonctionne en mode multicanal, ce qui permet de lutter contre les interférences et les affaiblissements de canal, et le rend ainsi plus robuste [106].

Ainsi, dans ce chapitre, nous proposons deux procédures d'attachement à un réseau TSCH pour réduire la consommation énergétique des nœuds qui tentent de joindre le réseau, sans pour autant reporter cette consommation sur les autres nœuds du réseau.

Pour ce faire, nous présentons tout d'abord un bref état de l'art des protocoles d'attachement à un réseau synchrone. Dans une deuxième section, nous présentons le simulateur utilisé pour l'évaluation des performances des protocoles proposés. Nous poursuivons avec la deuxième contribution de cette thèse : une procédure d'attachement à un réseau TSCH basée sur l'utilisation de séquences mathématiques et nous montrons qu'une telle solution garantit de faibles consommations d'énergie pour des délais d'attaches acceptables. Enfin, la dernière section présente notre troisième contribution, une procédure d'attachement au réseau fondée sur i) l'envoi d'information de synchronisation au sein des trames de données du réseau, et ii) l'ordonnancement soigneux des trames de synchronisation (EB) afin de garantir de bonnes performances, mesurées à l'aide de notre simulateur de la phase d'attache.

28. Le principe de *Friendship* de la spécification *mesh* est très similaire au fonctionnement du parent Thread, présenté à la page 35

4.1 Les protocoles de découverte de voisinage

Alors que les procédures d'attachement ont beaucoup été étudiées dans le cadre des communications mobiles, *ad hoc*, les travaux existants pour l'IoT se sont principalement intéressés à la phase d'opération du réseau.

La phase d'attachement, bien que très consommatrice, a rarement été étudiée dans le contexte des réseaux de capteurs. Cette procédure a en revanche été intensivement étudiée dans le contexte des protocoles de découverte de voisinage (ND (Découverte de voisinage — ou *Neighbour Discovery*)) pour les réseaux mobiles à mobiles, moins contraints en énergie.

Cette section présentera donc les solutions existantes de la littérature pour la phase d'attache à un réseau TSCH, suivi d'une brève présentation des protocoles de ND existants dans la littérature des communications mobiles, dans le but de les adapter par la suite aux réseaux TSCH, et ainsi présenter des protocoles plus performants que les solutions existantes.

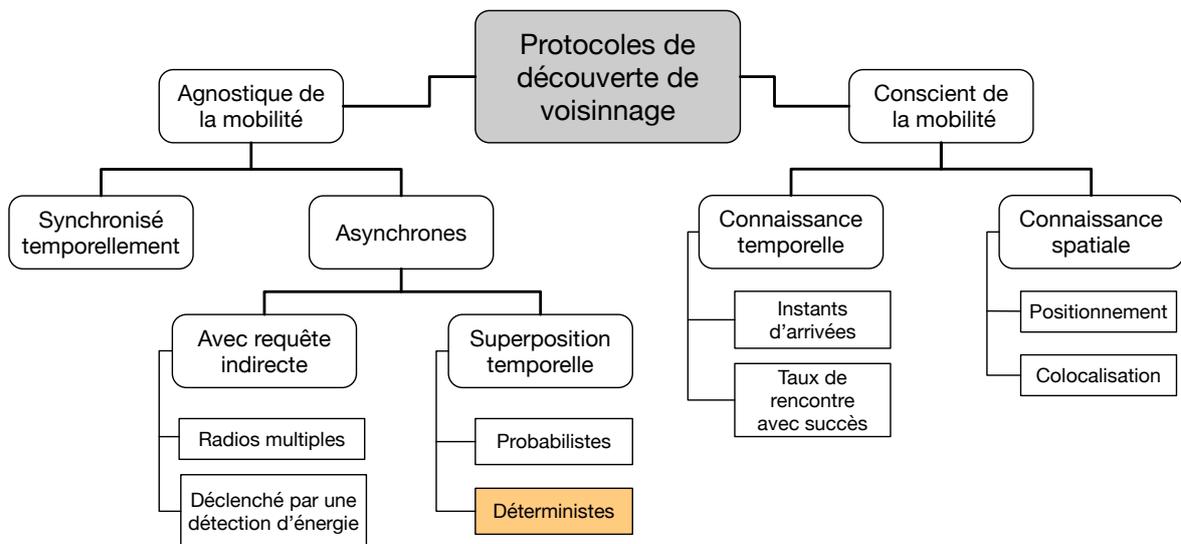


FIGURE 4.1 – Taxonomie des protocoles de découverte de voisinage dans le cas de réseaux opportunistes pour l'IoT. Figure extraite de l'article de Pozza *et al.* [107], puis traduite. Nous avons identifié la catégorie dans laquelle s'insèrent nos travaux en colorant la case correspondante.

La figure 4.1 présente la taxonomie proposée dans l'étude de Pozza *et al.* [107]. Cette taxonomie classe les protocoles de ND dans le contexte des réseaux opportunistes, c'est-à-dire des réseaux pour lesquels il n'y a quasiment pas d'infrastructure, et où les nœuds communiquent directement entre eux, en exploitant les possibilités de communications dès lors qu'elles sont réalisables. Un bon exemple de réseau opportuniste serait par exemple la traversée d'une ville par un véhicule connecté : le véhicule serait en communication de manière opportuniste avec chacun des autres nœuds du réseau (par exemple un feu tricolore, ou bien des places de parking connectées).

Dans le contexte des réseaux opportunistes, les processus de ND représentent une large proportion de l'activité du réseau. Ainsi, le niveau de consommation d'une telle procédure est un des critères déterminants dans le choix du protocole de ND à déployer. Alors que la plupart des solutions troquent la latence du processus pour économiser de l'énergie lors de la phase d'attachement, un tel prix n'est pas tolérable dans le contexte des réseaux opportunistes. Cette étude présente donc des protocoles de ND qui ne réduisent pas leur consommation énergétique au prix d'une augmentation de la latence. C'est cet aspect qui nous intéresse dans le contexte de réseau TSCH, duquel on voudrait pouvoir facilement se désynchroniser/resynchroniser pour ne pas conserver une communication active au prix d'une synchronisation coûteuse.

Ainsi, après une sous-section dédiée aux solutions existantes de protocoles d'attaches au réseau TSCH, nous présenterons deux des protocoles de ND de l'étude de Pozza *et al.* [107] qui se rapprochent le plus de notre contexte, en se positionnant dans la branche des protocoles agnostiques de mobilité, de type asynchrone, dont le principe repose sur la superposition temporelle de deux périodes d'activité, notre cadre de travail des réseaux TSCH.

4.1.1 Procédures d'attachement dédiées à la technologie TSCH

Le protocole d'attache décrit dans le standard 802.15.4 [9], et préconisé dans le brouillon²⁹ de l'IETF proposant un mode de fonctionnement minimal pour un réseau TSCH [38], est, nous l'avons vu dans le chapitre Discussion (cf. page 61) coûteux en énergie.

C'est pour cette raison que de nouvelles propositions dédiées aux réseaux TSCH ont été faites dans la littérature. Nous présentons ici la proposition de Duy *et al.* [2], qui repose sur l'utilisation de la logique floue pour déterminer le nombre de trames d'EB (Balise améliorée — ou *Enhanced Beacon*) à envoyer par nœud. Nous présenterons également la deuxième proposition de protocole de ND dédiée aux réseaux TSCH trouvée dans la littérature, le mode de remplissage des cellules dédiées à l'envoi d'EB de manière aléatoire proposé par Vogli *et al.* [3].

4.1.1.1 RapidJoin

L'approche proposée par Duy *et al.* [2] permet à tous les membres du réseau d'envoyer plusieurs EB (N_{EB}) par *slotframe*, avec $N_{EB} \geq 1$.

Le protocole proposé divise une *slotframe* en deux parties : une première partie dédiée à l'envoi d'EB, et une seconde pour l'envoi des trames de données.

Chaque nœud qui annonce sa présence au réseau en envoyant des EB utilise la partie dédiée aux balises pour envoyer N_{EB} sur des canaux physiques consécutifs, en sélectionnant aléatoirement le premier canal physique de la séquence.

Le nombre d'EB envoyés par chaque synchroniseur, N_{EB} dépend d'un système à logique floue qui prend en compte le nombre de voisins dans le réseau et optimise le nombre d'EB envoyés par chaque nœud.

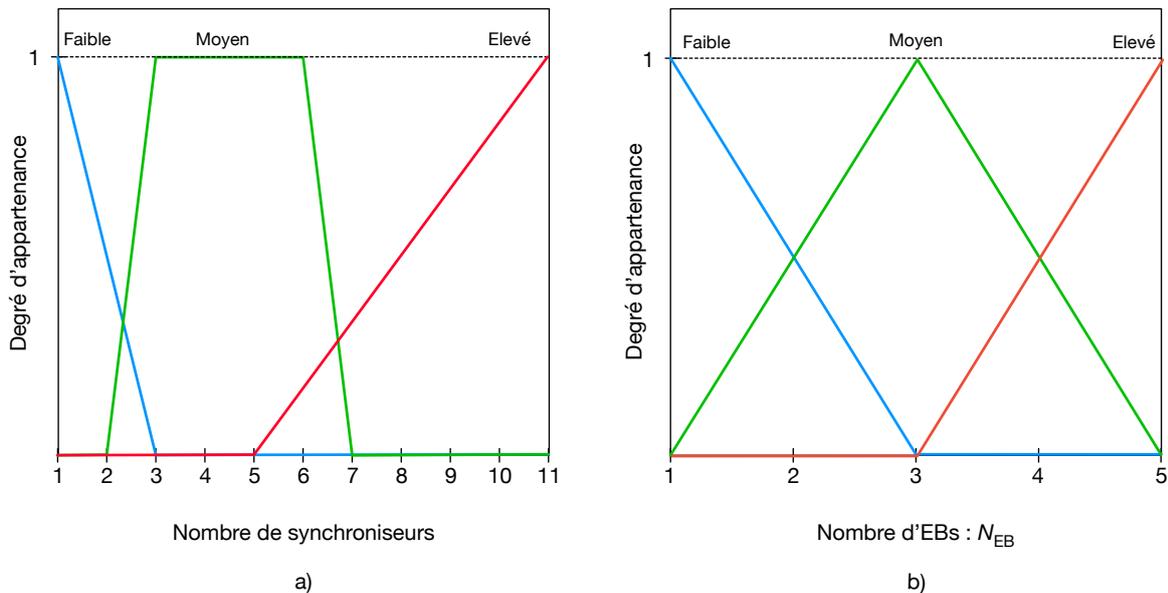


FIGURE 4.2 – Principe de fonctionnement du système à logique floue exploitée par *Rapid Join*, l'algorithme d'ordonnement de l'envoi des EB proposé par Duy *et al.* [2].

Le principe de cette logique floue est présenté sur la figure 4.2 : en fonction du nombre N_{neigh} de nœuds voisins qui envoient aussi des balises, un nœud TSCH sera classifié grâce à la fonction de logique floue présentée sur la gauche de la figure 4.2 comme ayant un nombre de voisins N_{neigh} faible, moyen ou élevé.

Ensuite, le système à logique floue confronte ce résultat à la règle de type IF-THEN qui peut être résumée de la façon suivante :

- **SI** un nœud a un nombre de voisins **faible**, **ALORS** il doit envoyer un nombre **élevé** d'EB par *slotframe*,
- **SI** un nœud a un nombre de voisins **moyen**, **ALORS** il doit envoyer un nombre **moyen** d'EB,
- **SI** un nœud a un nombre **élevé** de voisins, **ALORS** il doit envoyer un **faible** nombre d'EB par *slotframe*.

29. draft

Enfin, le système flou traduit la notion de nombre d'EB (faible, moyenne ou élevée) en valeur numérique N_{EB} grâce à la fonction floue présentée sur la droite de la figure 4.2.

Ce système, s'il permet de s'adapter en fonction de la densité du réseau, ne garantit aucune valeur maximale pour d_{join} et peut aboutir à ce qu'un nœud qui tente de rejoindre le réseau n'y parviennent pas, les canaux physiques sur lesquels les nœuds envoient leurs EB étant sélectionnés de manière aléatoire.

4.1.1.2 Random Vertical / Random Horizontal

Vogli *et al.* [3] ont proposé d'améliorer le protocole d'attache au réseau TSCH existant en prenant en compte la dimension fréquentielle dans le design de leur proposition, mais aussi en allouant des slots dédiés à l'envoi de trame d'EB de manière absolue, c'est-à-dire que même si aucun EB n'est réellement transmis pendant ce slot, il n'est pas possible pour un nœud du réseau d'utiliser ce slot dans le but de transférer des trames de données.

Dans leur proposition, initialement, seul le premier créneau temporel de la *slotframe* est dédié à l'envoi d'EB. Afin d'envoyer des EB sur tous les canaux fréquentiels, leur proposition met en place l'utilisation de *multislotframe*, définies comme la succession de r répétitions de la *slotframe* dans le temps. Un nœud du réseau envoie une seule trame d'EB par *multislotframe*. Deux schémas d'envois des EB sont proposés : le remplissage aléatoire vertical, et l'horizontal.

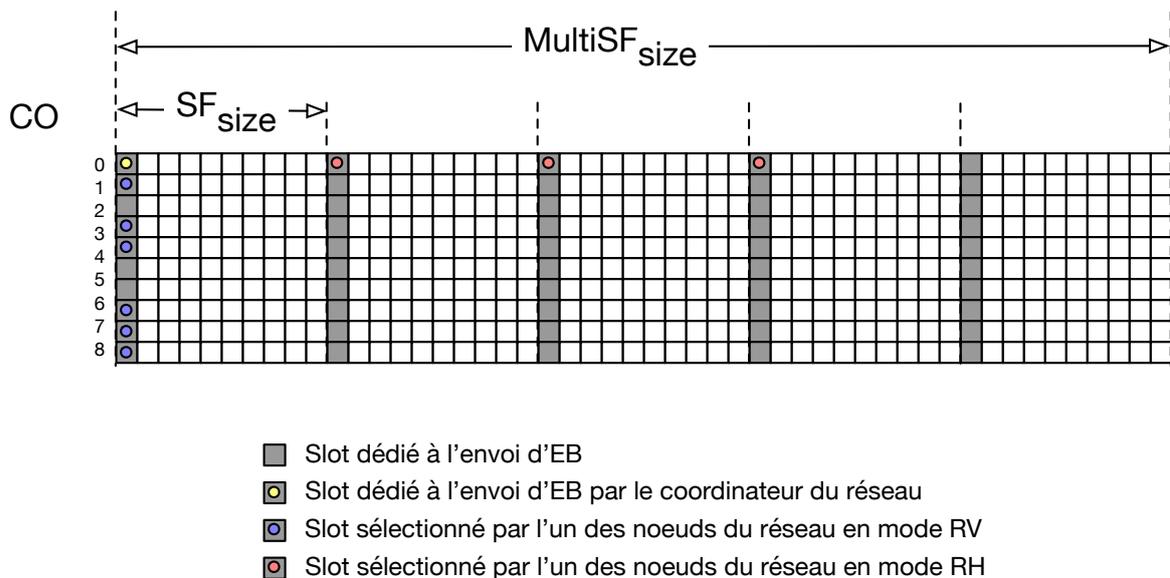


FIGURE 4.3 – Principe de fonctionnement de l'algorithme d'ordonnancement d'envoi des EB RV/RH proposé par Vogli *et al.* [3].

Remplissage aléatoire vertical (RV). Dans ce mode, le coordinateur du réseau transmet ses EB dans le premier slot de la *multislotframe* en utilisant le canal logique $CO = 0$. Tout nouveau nœud synchronisé doit transmettre ses propres EB dans le même créneau temporel, mais sélectionne le canal logique aléatoirement. Ainsi, chaque nœud remplira aléatoirement une cellule dans la première colonne de la structure représentée à la figure 4.3. Cependant, il est possible de perdre des trames d'EB à cause des collisions. En effet, RV ne garantit pas qu'une cellule sera choisie par un unique nœud.

Remplissage aléatoire horizontal (RH). Le coordinateur transmet aussi ses EB dans le premier slot de la *multislotframe*, sur le canal logique $CO = 0$. En revanche, un nouveau nœud synchronisé dans le réseau sélectionnera aléatoirement parmi l'un des slots dédiés aux EB disponibles de la *multislotframe*, en conservant le canal logique $CO = 0$. De cette façon, la première rangée des slots dédiés à l'envoi de trame de EB sera remplie de manière horizontale, et aléatoirement, comme illustré sur la figure 4.3. De la même manière que dans RV, il n'y a aucune garantie que deux nœuds ou plus ne transmettront pas dans le même slot.

Ces propositions de Vogli *et al.* [3], exploitent des probabilités statistiques, qui permettent d'obtenir de bonnes performances moyennes, mais l'aspect non déterministe de cette approche ne permet pas de garantir une latence, ou une quantité d'énergie maximale utilisée pour la phase d'attache au réseau. De plus, deux nœuds du réseau peuvent sélectionner le même slot d'envoi de leur EB, et, s'ils sont à portée l'un de l'autre, engendrer des collisions non gérées par Vogli *et al.*.

4.1.2 Protocoles de ND classiques

Tous les schémas de ND présentent un compromis entre l'énergie consommée, représentée par le temps passé en mode actif d_{on} , et le délai d'attache au réseau, d_{join} : plus de slots d'endormissement impliquent une consommation d'énergie amoindrie, mais correspondent à une latence d'attache au réseau plus élevée que lorsque le nœud J balaye un canal en mode Rx en continu.

Alors que les protocoles de ND présentés à la section précédente sont spécifiques à l'usage avec la technologie TSCH, et donc multicanaux, dans cette section, nous présentons des protocoles de ND classiques, qui sont principalement monofréquentiels. Nous présenterons le fonctionnement de Searchlight-S, un protocole de ND éprouvé, ainsi que deux récentes propositions, B-Nihao, qui repose sur le principe de "Parler plus pour écouter moins" [108], et une approche dans laquelle le temps est considéré de manière continue, et non découpé en slots [109].

4.1.2.1 Searchlight

Bakht *et al.* [110] proposent en 2012 un protocole dans lequel la consommation à la fois des nœuds du réseau N et des nœuds J , qui tentent de rejoindre ce réseau, ont un fonctionnement par intermittence, dans le but d'économiser de l'énergie. Ainsi, contrairement aux propositions existantes pour TSCH, le nœud J ne reste pas en mode Rx pendant toute la durée du processus d'attache au réseau.

Searchlight-S découpe le temps en slot, et, comme la plupart des protocoles de ND classiques [111, 112, 113, 114, 115], utilise des slots d'activité qui ont à la fois une période d'écoute et une période de transmission. De cette manière, les nœuds peuvent se découvrir mutuellement, ce qui est nécessaire dans le contexte des communications directes mobile à mobile.

Searchlight-S Searchlight-S en version "rayée" (*striped* — d'où le S) est un protocole efficace énergétiquement, éprouvé et entièrement déterministe. Il repose sur l'utilisation de séquences *quorum* [113].

Dans cette variante, un nœud se réveille seulement deux fois au cours d'une trame de t slots consécutifs, de sorte que le ratio d'activité du réseau et du nœud qui souhaite s'attacher est de $\frac{2}{t}$. Ainsi, il est possible de choisir la taille de la trame t pour obtenir un ratio d'activité cible, à la fois pour un nœud du réseau et le nœud J .

Un slot actif est constitué de l'envoi d'une trame de synchronisation au début et à la fin du slot, les deux envois étant séparés par une période d'écoute. Les deux slots actifs de Searchlight-S, identiques du point de vue de l'activité, sont répartis dans la première moitié de la trame de la manière suivante :

- Le premier slot actif est un slot d'ancrage (A) : il correspond toujours au premier slot de la trame, et est actif à chaque trame.
- Le second slot actif est un slot qui sert de sonde, *probe* en anglais (P) : ce second slot actif se "déplace" à chaque itération de la trame, dans le but de trouver le slot d'ancrage du nœud voisin qu'il tente de découvrir.

Dans le mode à sondage systématique, le slot de sonde P prend initialement place juste après slot d'ancrage A, c'est-à-dire que cela correspond au second slot de la trame. Le slot de sonde P sélectionne ensuite le slot suivant pour chaque répétition de la trame. Par conséquent, les slots P correspondent au troisième créneau de la trame lors la deuxième répétition, le quatrième à la troisième répétition, et ainsi de suite.

Pour gérer le non-alignement des slots, Searchlight-S utilise la solution proposée dans Disco [111] qui consiste à envoyer des balises non pas une fois par slot, mais à la fois au début et à la fin d'un slot.

Bakht *et al.* [110] montrent de plus que, comme les nœuds transmettent deux balises par slot et choisissent leurs emplacements de sondage P de façon consécutive, il est possible de n'insérer qu'un slot P sur

deux sans impacter les performances. Ainsi, dans Searchlight-S, les slots P ne sont sélectionnés qu'un slot sur deux dans la trame, et cela suffit pour réaliser la découverte de voisinage, dans le cas où les deux nœuds qui se découvrent ne sont pas parfaitement synchronisés.

Pour couvrir le cas de l'alignement parfait des trames, Bakht *et al.* ont proposé d'augmenter légèrement la taille des slots d'activité et de compenser par une réduction de la taille du prochain slot inactif.

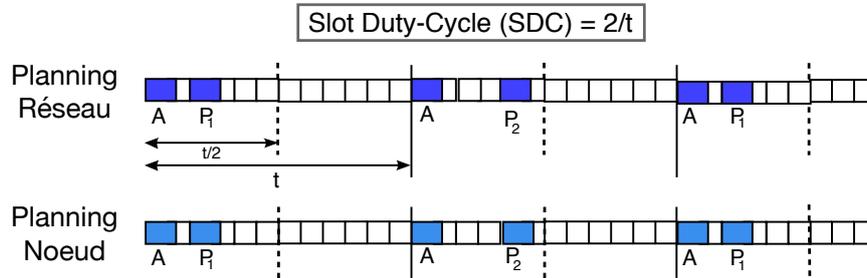


FIGURE 4.4 – Principe du protocole de ND SL-S.

La figure 4.4 illustre le principe de SL-S. Dans cet exemple, le motif Searchlight-S est répété après deux trames et les emplacements actifs sont légèrement plus grands que les autres emplacements. Les slots suivants sont réduits pour compenser cette augmentation.

Searchlight-R Searchlight-R est une version aléatoire (*random* en anglais, d'où le R) de Searchlight-S dans laquelle le slot de *Probe* est sélectionné aléatoirement, dans le but d'augmenter les performances moyennes du protocole, à la manière du protocole **Birthday** [114], totalement aléatoire, mais statistiquement performant. Cette version ne possède par conséquent aucun des avantages du déterminisme, ce qui implique que le coût énergétique d'attache au réseau ne peut être borné.

4.1.2.2 Nihao

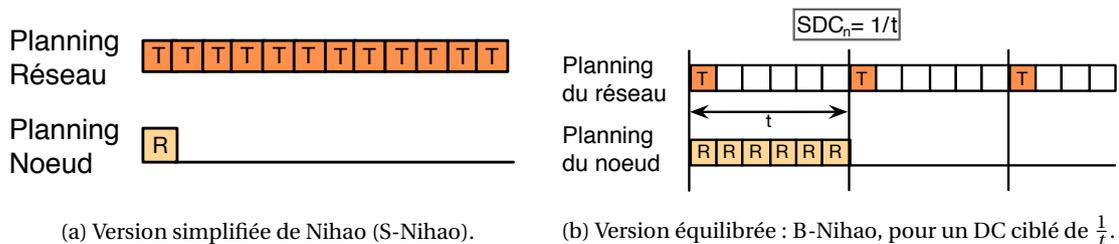


FIGURE 4.5 – Principe de fonctionnement de Nihao. T représente un slot de transmission, R un slot de réception.

Nihao [108] adopte un modèle différent, plus proche de la problématique des réseaux TSCH, en définissant des slots dédiés à la réception ou à l'émission. Dans un slot d'émission, un nœud envoie une balise courte au début du slot. Dans un slot de réception, un nœud active sa radio en mode *Rx* pendant la durée totale du slot.

Dans la variante S-Nihao (pour Nihao simple), un nœud transmet des balises dans chaque slot, ce qui conduit à un délai de jointure d_{join} et un temps en *Rx* de la durée d'un seul slot dans le pire des cas (voir figure 4.5a).

Dans sa version étendue appelée B-Nihao (pour Nihao équilibré ou *balanced* en anglais) les nœuds envoient une balise tous les t slots et le nœud qui tente de s'attacher écoute pendant les t slots consécutifs. La figure 4.5b présente le concept de B-Nihao.

Ainsi, on remarque que le fonctionnement actuel du TSCH repose sur le protocole B-Nihao, avec t correspondant à l'intervalle entre l'envoi des balises, T_{adv} .

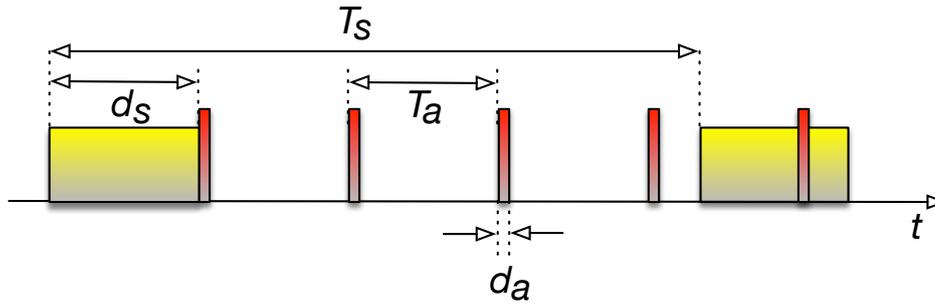


FIGURE 4.6 – Modèle de découverte en temps continu : un nœud annonceur envoie des balises à chaque intervalle T_a et le nœud J balaye le canal à chaque intervalle T_s . Le balayage dure pendant l'intervalle d_s et les balises durent d_a .

4.1.2.3 Découverte périodique par intervalle

Kindt *et al.* [109] ont présenté un modèle de découverte à intervalle périodique, en temps continu, au contraire des solutions présentées précédemment qui découpaient le temps en slot, dans lequel les nœuds ont des rôles différenciés. Ce modèle a notamment été créé dans le but de formaliser, et d'estimer les performances du protocole de ND utilisé dans le BLE [116].

La figure 4.6 montre les principaux paramètres du modèle de découverte en temps continu : un nœud annonceur envoie des balises chaque intervalle T_{adv} et le nœud qui souhaite rejoindre le réseau balaye le canal avec une période T_s .

Le balayage dure pendant l'intervalle d_s et les balises durent d_a .

Ils ont également proposé plusieurs variantes de protocoles de découverte. L'intervalle de balayage T_s est choisi pour obtenir un rapport cyclique donné. Une telle période maximale est nécessaire pour couvrir la possibilité que le nœud n'entende aucun des annonceurs du réseau.

Kindt *et al.* [109] ont montré qu'avec des périodes et des fenêtres de balayage optimisées, dans une solution nommée PI-0M, le schéma proposé surpasse tous les protocoles à intervalles de temps connus tels que Disco [111], U-Connect [113], Searchlight [110] et les codes basés sur les ensembles de différences [117].

4.1.3 Ensemble de différence cyclique

Meng *et al.* [117] proposent d'exploiter des CDS (Ensemble de différences cyclique — ou *Cyclic Difference Set*) parfaits dans le but de créer des ordonnanceurs symétriques, à la fois pour les nœuds qui sont dans le réseau, mais aussi pour les nœuds qui cherchent à rejoindre le réseau.

Présentation des ensembles cycliques de différences. Les CDS sont des séquences mathématiques définies de la manière suivante [118] :

Définition 1 Un v, k, λ -ensemble de différence $D = \{d_1, d_2, \dots, d_k\}$ est une collection de résidus k modulo v telle que, pour tout résidu $\beta_{ij} \not\equiv 0 \pmod{v}$, la congruence

$$d_i - d_j \equiv \beta_{ij} \pmod{v}$$

a exactement λ paires solutions (d_i, d_j) , avec $d_i, d_j \in D$ [118].

En d'autres termes, $D = \{d_1, d_2, \dots, d_k\}$ est un ensemble de nombres positifs k (plus petit que v), qui respecte la propriété suivante : toutes les différences $\beta_{ij} \pmod{v}$ (pour $i \neq j$) sont différentes. Certaines séquences sont considérées comme parfaites, et garantissent de plus que l'apparition d'une valeur de β_{ij} est unique, comme le précise la deuxième définition.

Définition 2 Un CDS parfait est un CDS pour lequel $\lambda = 1$ [118].

De plus, ces séquences mathématiques étant des cycles, elles ont des propriétés qui se transmettent lors de la rotation du cycle sur lui-même.

Définition 3 *Étant donné l'ensemble de différences $D = \{d_1, d_2, \dots, d_k\}$, pour tout entier α , l'ensemble $\{d_1 + \alpha, d_2 + \alpha, \dots, d_k + \alpha\} = D + \alpha$ considéré modulo ν est aussi un ensemble de différences; on l'appelle un décalage de l'ensemble original D [118].*

Ainsi, dans le schéma de ND proposé, les propriétés suivantes des décalages des séquences de CDS parfaites sont exploitées :

- Chaque différence de 0 à $\nu - 1$ se produit dans entre exactement k décalages.
- Deux décalages différents d'un même CDS possèdent $\lambda = 1$ élément en commun.

Ces séquences CDS parfaites garantissent que pour tout décalage α , entier ou non, pour lequel le nœud arrivant dans le réseau commence sa séquence d'écoute, il y aura un chevauchement de $\lambda = 1$ slot actif du réseau et d'un slot d'écoute du nœud qui souhaite s'attacher au réseau.

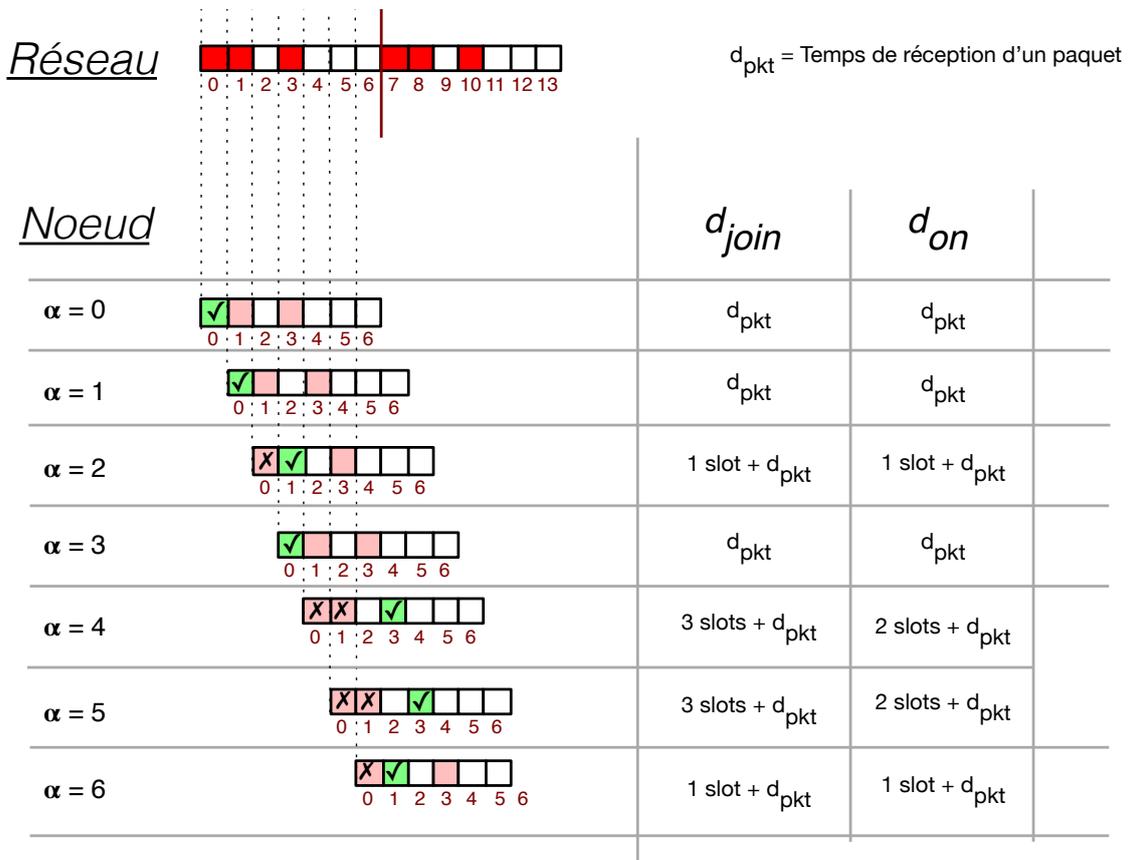


FIGURE 4.7 – Exemple pour le 7,3,1-CDS parfait : $\{0,1,3\}$.

Exemple d'application à la phase de découverte. La figure 4.7 montre un exemple de découverte de voisinage à l'aide d'une séquence de CDS parfait, le 7,3,1-CDS : $\{0, 1, 3\} \text{ mod } 7$ [118]. Le temps est découpé en slots assemblés en trames qui se répètent périodiquement. La taille de la séquence de slot est définie grâce au paramètre ν du CDS (7 dans ce cas). Les slots actifs sont sélectionnés dans la séquence CDS elle-même : ici, les slots 0, 1 et 3 sont actifs alors que les autres slots sont inactifs. Les nœuds qui sont déjà dans le réseau possèdent le même ordonnancement d'activité de slots que les nœuds qui tentent de rejoindre le réseau : on parle de protocole d'attache symétrique.

La figure 4.7 montre différents décalages α du nœud arrivant dans le réseau, correspondants aux différents instants possibles auxquels le nœud se réveille et démarre sa phase d'écoute, comme nous l'avons mentionné dans l'introduction de ce chapitre. Notons que par souci de lisibilité de la figure, les décalages α prennent des valeurs de slots entiers. Meng *et al.* ont montré que l'utilisation de slots d'activité à doubles transmissions permet de résoudre la problématique d'un décalage non entier, au prix d'un faible surcoût de consommation énergétique.

Nous présentons également sur la figure le délai entre le réveil du nouveau nœud et son attache au réseau, d_{join} , ainsi que le temps total d'activité avant de réussir à s'attacher au réseau, d_{on} .

L'exemple montre les principaux avantages d'une séquence CDS :

- au pire cas, le délai d_{join} est borné par l'indice du dernier slot actif de la séquence CDS (augmenté de 1 si l'on commence l'indexation à 0),
- les CDS permettent à la fois au réseau et au nouveau nœud de répartir les instants d'activité (transmission ou réception d'une trame de synchronisation), et les instants de sommeil (radio désactivée), pour économiser de l'énergie.
- les nœuds qui tentent de rejoindre le réseau n'ont pas à supporter de longues périodes de balayage continues, ce qui évite de drainer toute l'énergie disponible lors de la phase d'attache au réseau, dans le cas des nœuds fonctionnant avec un mécanisme de récupération d'énergie.

Ainsi, dans l'exemple, on remarque que la séquence CDS garantit de rejoindre le réseau en moins de quatre slots, l'indice du dernier slot actif étant 3. On peut aussi remarquer que pour $\alpha = 2$, le troisième slot du nouvel arrivant, celui numéroté 2 car on commence à 0, reste inactif sans affecter le délai d'attache au réseau. De cette manière, l'utilisation des séquences CDS diminue le temps d_{on} sans pour autant changer la durée d_{join} , ce qui n'est pas le cas dans le standard actuel TSCH, pour lequel nous observons l'égalité $d_{\text{join}} = d_{\text{on}}$,

Pendant, la génération des séquences de CDS parfait n'est pas aisée, et le plus faible ratio d'activité DC que nous pouvons atteindre avec les CDS proposés par [Meng et al.](#) est de l'ordre de 5 slots actifs pour une trame de 100 slots.

4.2 Simulateur de la procédure d'attachement au réseau

Alors qu'il existe un simulateur pour les réseaux TSCH dans la phase TSCH opérationnelle, [119, 120], nous n'avons pas trouvé de simulateurs dédiés à l'évaluation des performances d'un protocole de ND dans le contexte de réseaux TSCH.

Ainsi, nous avons créé un simulateur à événements discrets, développé en Python qui imite la procédure d'attache d'un nœud qui tente de rejoindre un réseau TSCH opérationnel, à une date de réveil t_0 , décalée par rapport à l'heure du réseau d'un offset α .

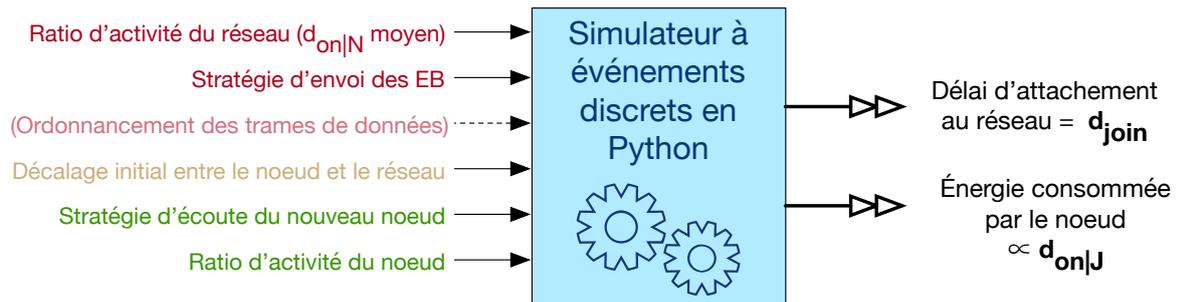


FIGURE 4.8 – Schéma de principe du simulateur de procédure d'attache au réseau.

Notre simulateur, que l'on peut schématiser de la manière présentée sur la figure 4.8, calcule le délai d'attache d_{join} et le temps passé en réception $d_{\text{on}|R_x}$ du nœud arrivant dans le réseau, pour chacun des décalages initiaux possibles α entre le réseau et le nœud. Les décalages α peuvent être définis avec granularité de $100\mu\text{s}$.

Ainsi, pour un décalage initial donné α , la figure 4.8 présente les paramètres qui influencent la consommation d'énergie et le délai d'attache au réseau :

- La stratégie d'écoute du nœud J ,
- La stratégie d'envoi de EB pour les nœuds N ,
- Le ratio d'activité radio (DC) du nœud J , $DC_J = \frac{d_{\text{on}|J}}{d_{\text{join}}}$.

- Le ratio d'activité radio des nœuds émetteurs d'EB, $DC_N = \frac{d_{on|N}}{\nu}$, où ν est la période de l'ordonnancement des nœuds du réseau, et $d_{on|N}$ le temps passé avec la radio active pendant la durée ν , du point de vue des nœuds du réseau.

Notons que l'activité du nœud qui souhaite rejoindre le réseau consiste uniquement à écouter l'activité du réseau pendant les slots d'écoute. Ainsi, du point de vue du nœud qui tente de rejoindre le réseau à proximité (noté J pour *joining* dans la suite), on a : $d_{on|J} = d_{on|Rx}$.

De la même manière, les nœuds qui sont dans le réseau, dans le cadre de la procédure d'attache au réseau, vont uniquement envoyer des trames de synchronisation EB, et leur activité liée à la procédure de ND se résumera à cette activité de transmission. Ainsi, pour les nœuds du réseau (notés N pour *network* dans la suite), nous avons : $d_{on|N} = d_{on|Tx}$.

Nous utilisons les temps d'activité de la radio d_{on} (Rx ou Tx) comme métriques de la consommation d'énergie. En effet, bien que nous ayons vu dans le chapitre précédent que les valeurs de consommation des plateformes dans le mode *Sleep* est un facteur déterminant de la consommation d'énergie, il n'est pas impactant dans le cadre de l'évaluation d'un protocole de ND : si nous souhaitons comparer les performances de différents protocoles de ND, une comparaison à plateformes égales est nécessaire, par souci d'équité entre les solutions. C'est pourquoi nous ne comparerons ici que la part de consommation d'énergie qui provient de l'activité de la radio.

De plus, comme les travaux récents de modèles réalistes de consommation énergétique [13, 81, 5], nous supposons que la consommation d'énergie de radio varie linéairement avec le temps pendant lequel la radio reste dans les modes de consommation Rx ou Tx , la pente n'étant différente que pour des plateformes matérielles différentes. Comme nous supposons que l'on évalue les performances sur une même plateforme matérielle, alors la consommation d'énergie est proportionnelle au temps passé avec la radio en activité d_{on} .

Ce simulateur sera utilisé pour évaluer les performances des deux protocoles de ND que nous proposons dans les sections suivantes.

4.3 Singer-Cyclic Difference Set

4.3.1 Présentation du schéma de ND proposé

Adaptation du protocole de ND basé sur les CDS à un réseau TSCH. Pour appliquer les CDS au problème de la découverte d'un voisinage TSCH, supposons que chacun des nœuds du réseau N envoie des balises aux emplacements définis par une séquence CDS, sur un canal fréquentiel dédié. Supposons de plus, que le nœud qui souhaite rejoindre le réseau écoute uniquement aux emplacements qui suivent la même séquence CDS, et qu'il connaisse le canal fréquentiel dédié à la procédure d'attache au réseau.

Nous proposons que chacun des nœuds se comporte de la même manière qu'en mode de communication TSCH classique lors des slots dédiés à la procédure d'attache du réseau : les nœuds du réseau envoient des trames d'EB au sein des slots dictés par la séquence CDS, mais l'envoi en lui-même se produit à la manière d'une transmission de trame TSCH, c'est-à-dire un slot d'*advertisement* classique (cf. page 30).

Le comportement du nœud qui souhaite rejoindre le réseau lors d'un slot indiqué comme actif par la séquence de CDS sera le même que celui d'un nœud qui souhaite rejoindre le réseau de la manière standardisée actuellement : il écoute en continu pour toute la durée du slot.

De ce fait, alors que la proposition de Meng *et al.* [117] résolvait la problématique d' α non entiers via l'utilisation de slots à doubles transmissions entrecoupées d'une partie en mode de réception, nous devons mettre en place un mécanisme pour gérer ces situations de non-alignement de slots provenant d'un réveil aléatoire d'un nœud.

Nous résolvons les décalages non entiers grâce à l'observation suivante : un décalage initial α non entier n'implique pas d'impossibilité de rejoindre le réseau, car il est toujours possible de se ramener dans le cas d'un décalage entier, au prix d'une légère dégradation des performances. Nous proposons de ne rien mettre en place pour gérer les valeurs d' α non entières, car celles-ci n'empêchent pas le bon fonctionnement de la procédure d'attache au réseau.

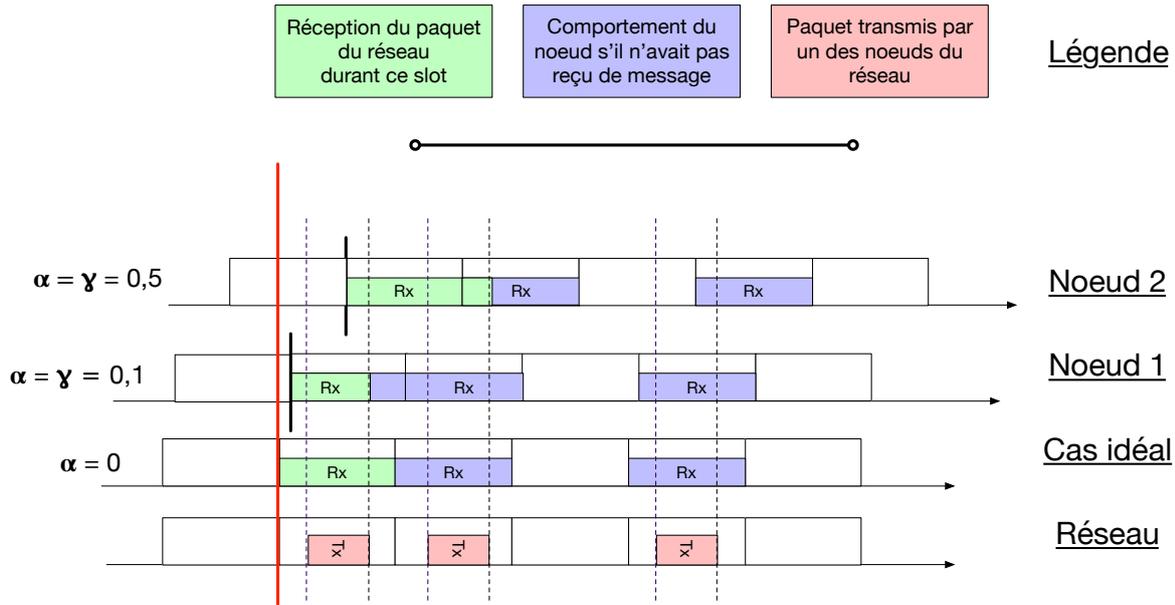


FIGURE 4.9 – Gestion du non-alignement des slots dans le cas des CDS.

En effet, on suppose qu'un paquet TSCH est envoyé $TsTxOffset = 1,120$ ms après le début d'un slot – c'est le comportement par défaut. Si nous décomposons le décalage entre l'ordonnement de la transmission des EB et celui du nœud qui tente de rejoindre le réseau de la manière suivante, $\alpha = \alpha_{int} + \gamma$ avec $\gamma = \alpha - Ent(\alpha)$, et $\alpha_{int} = Ent(\alpha)$; alors, deux cas se présentent, comme le montre la figure 4.9 :

- soit la valeur de γ est plus petite que $TsTxOffset$, et dans ce cas, le nœud arrive "un peu en retard", mais ne manque pas la transmission. C'est le cas du nœud 1 de la figure 4.9. Dans cette situation, on a $\alpha \sim \alpha_{int}$, avec un faible gain en termes de temps d'activité de la radio : d_{on} est réduit de γ .
- soit la valeur de γ est plus grande que $TsTxOffset$ (cas du nœud 2 sur la figure 4.9), auquel cas la transmission est manquée. En revanche, comme nous avons $\gamma > TsTxOffset$, cela implique qu'au lieu de terminer la période d'écoute à la fin du slot d'activité du réseau, le nœud qui tente de rejoindre le réseau écoutera le début du slot suivant, pendant une durée plus grande que $TsTxOffset$ (puisqu'il est décalé de plus que cette valeur). Dans ce cas, il pourra recevoir le début d'une trame dans le slot suivant, et ne pas éteindre sa radio comme prévu à la fin de son slot, dans le but de recevoir pleinement la transmission. Ainsi, dans le cas où $\gamma > TsTxOffset$, on a $\alpha \sim (\alpha_{int} + 1)$.³⁰

Cette logique ne tient cependant pas pour tous les schémas de ND. Elle ne peut s'appliquer que dans les cas où le protocole garantit qu'il existe une superposition entre les ordonnancements du nœud qui se réveille et du réseau pour le décalage $\alpha_{int} + 1$.

Par ailleurs, nous supposons que la dérive d'horloge observée lors de l'écoute du nœud est négligeable, car les délais d'attache d_{join} que nous considérons sont de l'ordre de la dizaine de secondes. En outre, si nécessaire, le nœud peut se réveiller légèrement plus tôt que le prochain slot programmé : la durée de l'intervalle de garde nécessaire dépend directement du temps passé en état de veille et de la précision de l'horloge matérielle, de la même manière que nous l'avons vu dans le chapitre précédent

Extension à des DC plus faibles : exploitation des séquences de Singer.

Les S-CDS (Séquence de CDS générée à partir des séquences de Singer) sont des CDS dont la construction est basée sur l'utilisation de cycles de Singer [121]

Définition 4 Pour tout nombre premier q , il existe un ensemble S-CDS de Singer construit de la manière suivante.

Soit

$$G = \frac{GF(q^{n+2})^*}{GF(q)^*}$$

30. Nous avons pris l'exemple de deux slots d'activités consécutifs du réseau, mais la même logique s'applique si le slot d'activité commune du réseau et du nœud était plus tardif.

où $\text{GF}(q)$ est le champ de Galois d'ordre q , et $\text{GF}(q)^*$ est le groupe multiplicatif de l'élément non nul.

L'ensemble $D = \{x \in G \mid \text{Tr}_{q^{n+2}/q}(x) = 0\}$ est un Singer v, k, λ -CDS, défini avec les paramètres suivants :

$$v = \frac{q^{N+1} - 1}{q - 1}, k = \frac{q^N - 1}{q - 1}, \lambda = \frac{q^{N-1} - 1}{q - 1},$$

où $\text{Tr}_{q^{n+2}/q} : \text{GF}(q^{n+2}) \rightarrow \text{GF}(q)$ est la fonction trace telle que :

$$\text{Tr}_{q^{n+2}/q}(x) = x + x^q + \dots + x^{q^{n+1}}$$

dans laquelle on utilise la valeur $N = 2n$.

Nous proposons d'utiliser les séquences S-CDS de paramètre $N = 2$ qui ont une valeur de q suffisamment grande pour atteindre de faibles DC de fonctionnement, c'est-à-dire une faible consommation d'énergie : pour cela, nous sélectionnons les séquences S-CDS correspondantes parmi la base de La Jolla [122], une base accessible en ligne et qui répertorie les différents CDS existants.

Par exemple, le CDS de taille $v = 3783$, correspond à la séquence suivante de $k = 62$ slots actifs [122] :

{0, 1, 73, 159, 205, 343, 427, 507, 549, 568, 734, 791, 845, 876, 879, 884, 981, 1010, 1058, 1108, 1164, 1170, 1177, 1179, 1197, 1207, 1260, 1307, 1469, 1572, 1589, 1647, 1663, 1707, 1742, 1820, 1824, 1996, 2064, 2257, 2401, 2493, 2515, 2602, 2616, 2640, 2661, 2710, 2861, 2873, 3081, 3107, 3148, 3214, 3362, 3385, 3417, 3592, 3603, 3628, 3668, 3732} \bmod 3783

Cette séquence d'exemple correspond à un ratio de slots actifs de $\frac{62}{3783} \approx 1.64\%$.

4.3.2 Évaluation des performances de la proposition S-CDS

Nous utilisons le simulateur présenté au paragraphe 4.2 (cf. page 102) pour estimer les performances de notre proposition. Cependant, puisque la stratégie des S-CDS n'utilise pas les trames de données, nous n'avons pas besoin de donner ce paramètre au simulateur, et ne le prendrons pas en compte dans l'évaluation des performances de la séquence S-CDS.

4.3.2.1 Point de départ de la comparaison

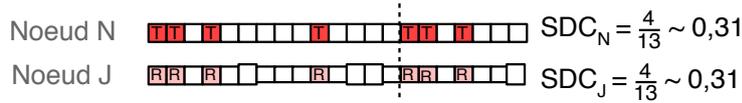
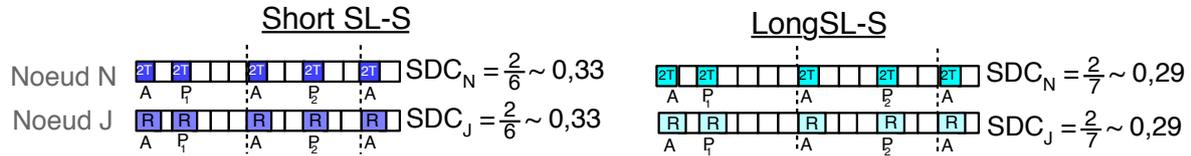
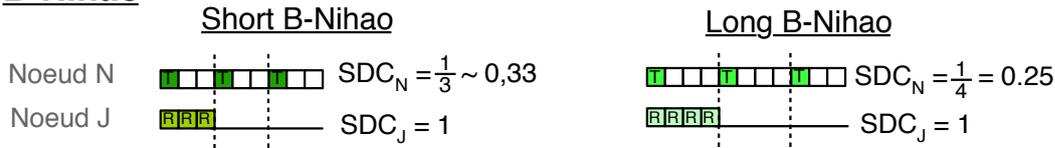
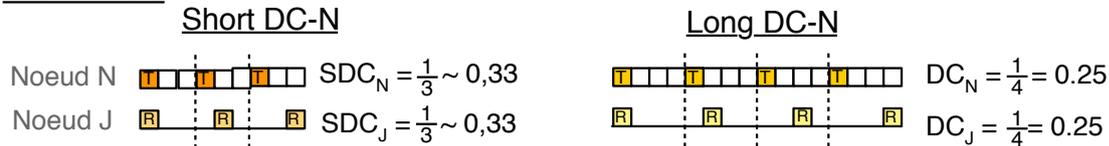
Afin de comparer le protocole de ND reposant sur l'usage des cycles de Singer aux autres protocoles de ND, nous définissons une nouvelle métrique, *SDC* (Cycle de charge en nombre de slots — ou *Slot Duty Cycle*), qui mesure le ratio de slots actifs sur la taille de la séquence.

Ainsi, dans le cas des CDS, nous avons : $SDC = \frac{k}{v}$.

Alors que la solution proposée de S-CDS est symétrique, ce n'est pas le cas de toutes les procédures d'attache au réseau. C'est pourquoi nous distinguerons les valeurs CDS_J et CDS_N , respectivement les valeurs de *SDC* du nœud qui veut rejoindre le réseau, et d'un nœud qui envoie les trames d'EB.

Notons que la radio d'un nœud N n'est pas en mode *Tx* pendant toute la durée d'un slot TSCH (cf. présentation du protocole TSCH au sein du chapitre 3), ce qui entraîne une distinction entre les valeurs SDC_N , le ratio d'activité en termes de slots, et DC_N , le ratio d'activité réelle de la radio. Du point de vue du nœud J en revanche, le fait de considérer le ratio d'activité de la radio DC_J , ou bien le ratio d'activité en termes de slots, SDC_J , ne changera pas les performances de la plupart des protocoles étudiés puisqu'un nœud J reste en mode *Rx* pendant toute la durée d'un slot. Seul le protocole SL-S augmente la taille des slots de *Rx*, et on pourra donc observer une différence entre SDC_J et DC_J pour cette technologie.

La figure 4.10 présente les détails des stratégies d'envoi de EB et d'écoute avec lesquelles nous comparons le schéma proposé, S-CDS. Les slots notés T représentent la transmission d'un EB dans un slot d'*advertisement* et ceux notés R représentent une écoute du nœud J pendant la durée totale du slot. Tous les protocoles comparés à S-CDS sont adaptés pour fonctionner dans les réseaux TSCH. L'adaptation consiste tout d'abord à différencier le rôle de l'émetteur et du récepteur pour chaque stratégie, car la plupart des propositions de la littérature utilisent des slots constitués d'une période de transmissions suivie d'une période de réception. Dans notre adaptation, comme nous l'avons proposé pour le schéma des S-CDS, pendant leurs slots actifs, les nœuds du réseau transmettent des EB selon le plan de transmission classique

S-CDS = {0,1,3,9} [13]

Searchlight-S

B-Nihao

DC-Nihao

 FIGURE 4.10 – Principe de comparaison de la proposition S-CDS à l'existant : on positionne les protocoles sur un pied d'égalité, le paramètre SDC_N

de TSCH. Le nœud J active sa radio au début d'un slot et surveille l'activité pendant toute la durée de ce créneau, fixée à 10 ms (valeur par défaut du standard).

Les slots de Searchlight-S sont légèrement plus longs que ceux des autres protocoles en raison de leur manière de gérer le non-alignement des slots. De plus, alors que tous les autres schémas ne transmettent qu'un seul EB par slot, ce qui suit strictement le modèle de slot TSCH, nous avons modifié le modèle de slot d'émission pour Searchlight-S afin qu'un nœud envoie deux EB consécutifs.

La figure 4.10 présente les schémas de comparaison que nous utilisons pour un SDC_N de 31% (une valeur arbitraire élevée, pour comprendre le principe), ce qui correspond à la séquence S-CDS $\{0, 1, 3, 9\} \bmod 13$ [122].

Pour que tous les schémas soient mis sur un pied d'égalité, nous fixons le rapport cyclique SDC_N pour les nœuds du réseau à la même valeur pour tous les schémas considérés.

Cependant, pour certains protocoles, il est impossible d'obtenir une valeur identique de SDC_N : par exemple, les ordonnancements SL-S sont de la forme $SDC_N = \frac{2}{t}$ alors que ceux de B-Nihao sont de la forme $SDC_N = \frac{1}{t}$, où t est la période des ordonnancements.

Ainsi, $SDC_{N|S-CDS}$, le SDC pour une séquence S-CDS donnée, ne peut être exactement égal à la valeur de $SDC_{N|SL-S}$, ni celle de $SDC_{N|Nihao}$. Pour résoudre ce problème, nous définissons deux nouveaux schémas (aussi présentés sur la figure 4.10) en sélectionnant les deux valeurs de SDC_N qui se rapprochent le plus de $SDC_{N|S-CDS}$: une légèrement plus faible (désignée par "long") et une plus élevée (désignée par "short").

Enfin, dans B-Nihao, le nœud J recherche continuellement des EB, ce qui ne convient pas à la plupart des périphériques du fait de leur faible capacité en énergie. Ainsi, nous définissons une nouvelle variante de B-Nihao, que l'on nomme DC-Nihao dans laquelle le nœud J limite son balayage à un slot consécutif, et il distribue les t slots nécessaires à la découverte parmi t répétitions de la séquence de taille t , ce qui conduit à un protocole de ND symétrique, c'est-à-dire que $SDC_J = SDC_N$, de la même manière que SL-S et S-CDS, comme le montre la figure 4.10.

4.3.2.2 Résultats pour une valeur de SDC_N fixée

Cette section présente les résultats pour un ensemble S-CDS qui conduit à une valeur de SDC_N proche de 1,6%, ce qui correspond à l'utilisation de la séquence de S-CDS présentée à la page 105.

La figure 4.11a présente DC_N , le ratio d'activité radio d'un nœud qui envoie des trames d'EB. Nous pouvons remarquer que toutes les valeurs sont similaires, à l'exception de Searchlight-S, ce qui s'explique par le lien étroit entre les paramètres DC_N et SDC_N pour la plupart des schémas : le rapport cyclique radio DC_N réel est équivalent à la valeur SDC_N de 1,6%, divisée par le temps d'activité radio en mode Tx au cours d'un slot. En utilisant les valeurs par défaut du standard 802.15.4 [9], c'est-à-dire environ 4,256ms passées en mode Tx pour des slots de 10ms, on peut retrouver la valeur obtenue avec l'expression :

$$DC_N = SDC_N \times \frac{4,256}{10} \sim 0,68\%$$

Rappelons que cette expression ne peut pas être appliquée à Searchlight-S, puisqu'un nœud SL-S transmet deux balises au sein d'un même slot, ce qui donne une valeur deux fois plus élevée de $DC_N \sim 1.36\%$.

Les figures 4.11c et 4.11b présentent les performances des schémas considérés pour une valeur de SDC_N fixée à 1,6% : le délai d'attache au réseau et le temps passé en mode Rx d'un nœud J . Comparé au protocole B-Nihao (Short-Nihao et Long-Nihao), le S-CDS atteint un niveau de consommation énergétique équivalent.

En ce qui concerne le délai d_{join} , B-Nihao est nettement plus rapide, car le comportement du nœud J est de balayer un canal en continu, ce qui garantit une découverte rapide. En revanche, comme cela a déjà été mentionné, il peut être impossible pour les nœuds alimentés par des piles ou utilisant des dispositifs de récupération d'énergie de supporter une telle activité, notamment quand le dispositif qui est censé envoyer les trames d'EB n'est plus à portée radio du nœud J .

Nous pouvons également constater que, pour un délai d'attache d_{join} équivalent, Searchlight-S consomme plus d'énergie que le schéma proposé (plus de temps passé en mode Rx), à la fois dans le cas moyen et le pire cas.

En outre, par rapport à la version à rapport cyclique de Nihao, DC-Nihao, notre proposition consomme une quantité d'énergie similaire, mais elle permet d'obtenir un délai légèrement plus court, encore une fois dans les cas moyens et les pires.

Pour résumer, notre proposition permet soit de réduire la consommation d'énergie, soit de raccourcir les délais d'attache au réseau par rapport aux autres protocoles de ND, dans le cas de faible valeur SDC_N .

4.3.2.3 Courbes de tendance en fonction du DC

Pour comparer les schémas avec des valeurs différentes de rapport cyclique, nous avons effectué la simulation pour des S-CDS de différentes tailles (voir figures 4.12, 4.13, et 4.14). Les courbes pour les versions courtes et longues de Searchlight-S, B-Nihao, et DC-Nihao correspondent en réalité au même protocole, de telle sorte qu'elles apparaissent comme une seule courbe, où les points des versions "short" et "long" ne font qu'ajouter de la précision à la courbe de tendance.

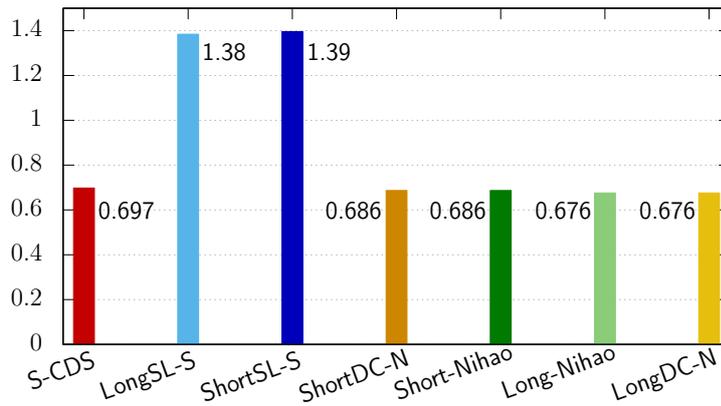
Nous pouvons vérifier grâce à la figure 4.12 que le rapport cyclique du nœud J , imposé par une stratégie d'envoi d'EB donnée, varie linéairement avec le SDC_N du réseau pour tous les schémas, à l'exception du protocole qui impose un balayage continu continu (B-Nihao). Pour ce protocole, le rapport cyclique du nœud J est de 100% pour B-Nihao. Les nœuds Searchlight-S passent plus de temps en mode Rx en raison d'une augmentation de la taille des slots d'écoute, ce qui conduit à une augmentation de la valeur DC_N .

Les figures 4.13 et 4.14 présentent les performances des différents protocoles de ND envisagés, pour des valeurs de SDC_N descendant jusqu'à 1%.

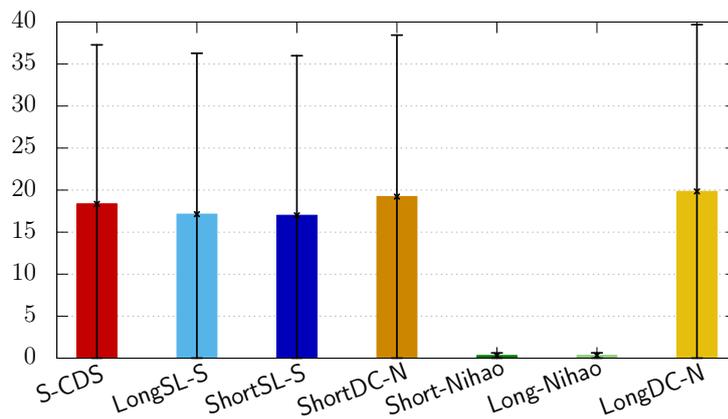
Nous pouvons voir que notre proposition se traduit par une faible consommation énergétique (faible temps passé en Rx) et un délai d'attache (d_{join}) court, dans le cas de SDC_N très faibles.

Ces valeurs basses de ratios d'activités sont les DC visés dans le contexte des WSN (rappelons que 1% de SDC correspond à un DC réel de la radio de 0,5%). Il est de plus possible d'atteindre une valeur de SDC_N proche de 0,3% pour la séquence S-CDS avec les paramètres $\nu = 86143$ et $k = 294$, ce qui peut même correspondre aux DC visés par les plateformes fonctionnant avec des systèmes à récupération d'énergie.

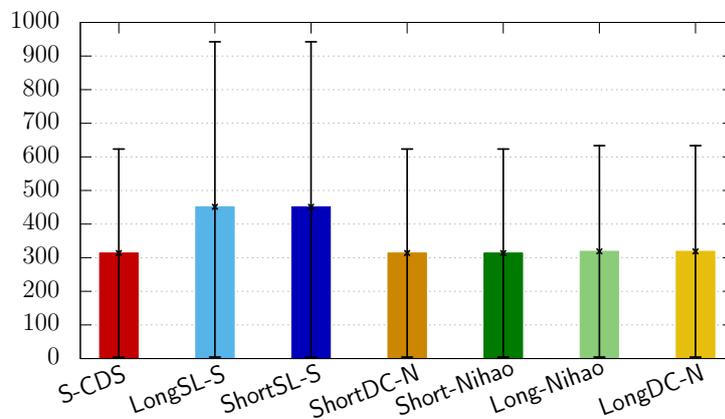
Enfin, les figures 4.13 et 4.14 confirment les résultats précédents : plus le rapport cyclique est bas, plus notre proposition est avantageuse.



(a) Ratio d'activité moyen de la radio d'un nœud du réseau DC_N (%).



(b) Délai d'attache au réseau d_{join} (s) : moyen/min/max.



(c) Temps passé en Rx par le nœud J (ms) : moyen/min/max.

FIGURE 4.11 – Performances des différents protocoles de ND pour $SDC_N \approx 1,6$

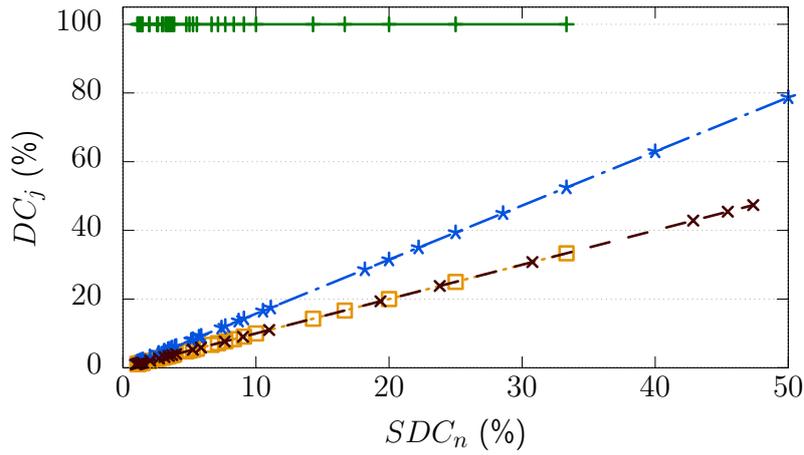


FIGURE 4.12 – DC radio du nœud J (DC_j) en fonction du DC d'un nœud N en termes de slots SDC_N .

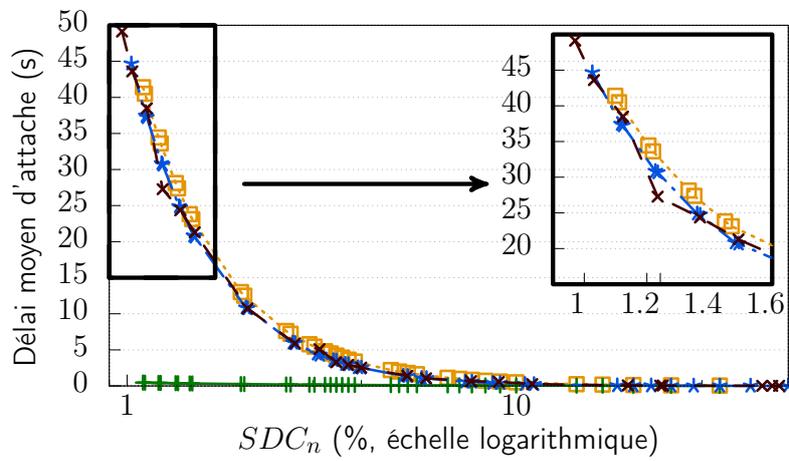


FIGURE 4.13 – Délai d'attache pour les protocoles de ND considérés en fonction du DC d'un nœud N en termes de slots SDC_N .

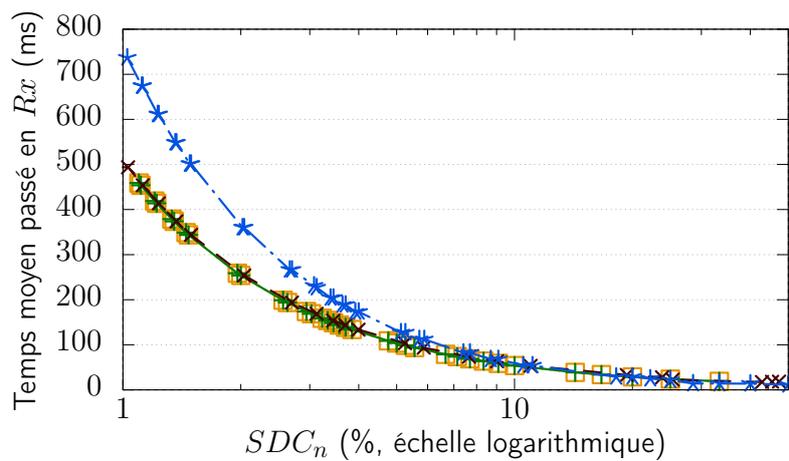


FIGURE 4.14 – Temps passé en Rx par le nœud J des protocoles de ND considérés en fonction du DC d'un nœud N en termes de slots SDC_N .

DC-Nihao $\text{---}\square\text{---}$ B-Nihao $\text{---}\text{+}\text{---}$ Searchlight $\text{---}\ast\text{---}$ S-CDS $\text{---}\times\text{---}$

FIGURE 4.15 – Légende pour les figures 4.12, 4.13, et 4.14.

4.3.3 Conclusion sur les S-CDS

Le protocole proposé, S-CDS, distribue dans le temps les slots actifs des nœuds J , de façon à ce qu'un nœud qui tente de rejoindre le réseau ne souffre pas de longues périodes de balayage, contrairement au mécanisme existant. Les principaux avantages de notre proposition résident dans la nature symétrique de la consommation entre les nœuds du réseau et les nœuds qui tentent de rejoindre le réseau, dans la garantie de rejoindre le réseau en une durée finie si l'on est à portée du réseau, mais également dans la garantie d'une consommation moyenne constante pendant la phase d'attache, qu'on soit ou non à portée d'un nœud annonceur.

La comparaison que nous venons de mener montre que le protocole S-CDS obtient des performances équivalentes à Searchlight-S en ce qui concerne le délai d'attache au réseau d_{join} , alors qu'il consomme moins d'énergie (plus faible durée en mode Rx) et qu'il réduit l'occupation du canal (2 fois moins de transmissions).

De plus, S-CDS atteint un niveau de consommation d'énergie faible, et similaire à celui de la version B-Nihao, tout en réduisant légèrement le délai moyen d'attache au réseau. Ainsi, le S-CDS semble une bonne alternative au fonctionnement actuel (qui repose sur les principes de B-Nihao) pour les périphériques du réseau qui possèdent de fortes contraintes en termes d'énergie tels que les nœuds alimentés par un mécanisme de récupération d'énergie.

Cependant, nous avons conscience que le schéma proposé possède majoritairement quatre inconvénients :

- Premièrement, il n'exploite pas la fonctionnalité multicanal apportée par la technologie TSCH puisque nous proposons d'utiliser un canal dédié à la procédure d'attache au réseau. Nous pouvons étendre la logique des S-CDS à un environnement multicanal en allouant des canaux fréquentiels différents à des nœuds voisins, mais cela signifie que, dans le pire cas, les performances présentées plus haut devront être multipliées par le nombre de canaux utilisés par le réseau, N_c . Par ailleurs, la solution proposée peut-être directement appliquée aux solutions monocanales que nous avons évoquées dans le premier chapitre, par exemple le mode *beacon* du 802.15.4, ou le 802.11b PSM.
- Deuxièmement, cette procédure implique un ratio d'activité symétrique entre le nœud du réseau et le nœud qui souhaite joindre ce réseau. Cela signifie que le DC utilisé par le réseau est choisi à l'initialisation de celui-ci et qu'un nœud qui cherche à le rejoindre devra fonctionner avec ce DC. Ainsi, on devra dimensionner le réseau en fonction du nœud le plus contraint en énergie, au détriment des nœuds moins contraints, qui pourraient utiliser eux des séquences de DC plus élevés, pour diminuer le délai d_{join} .
- Troisièmement, il est impossible pour les périphériques de générer des séquences S-CDS, du fait de la complexité de génération de ces dernières : l'utilisation des S-CDS requiert donc une connaissance *a priori* de ces séquences, qui devront être stockées dans la mémoire non volatile des nœuds.
- Enfin, dans ce travail, nous avons supposé que chacun des nœuds N à l'intérieur du réseau devait envoyer des trames d'EB en suivant le principe S-CDS, sans exploiter le fait que plusieurs nœuds du réseau peuvent se répartir la charge demandée par la réalisation de la séquence S-CDS, et ainsi diminuer la consommation globale liée à la phase d'attache au réseau.

4.4 DataJoin

Partant du constat que la proposition S-CDS n'exploite pas directement la capacité multicanale de TSCH, et que la complexité de génération des séquences S-CDS rend une telle proposition difficile à déployer pour les réseaux de capteurs, nous proposons une seconde solution, directement applicable à un réseau TSCH, qui repose sur les principes suivants :

- Nous supposons que le nœud J connaît les paramètres de base du réseau TSCH, en particulier la valeur la plus longue de T_{adv} , $T_{adv|max}$, pour en dériver $d_{on|max}$, la période la plus longue pendant laquelle il devra sonder un canal à la recherche d'une trame d'EB.
- Nous adoptons la stratégie B-Nihao/PI-0M de balayage d'un canal donné pendant au moins l'intervalle T_{adv} .
- Nous réduisons la consommation d'énergie du nœud J en analysant le trafic de données habituel pendant le balayage afin d'obtenir des informations sur la date de la prochaine trame de EB sur le canal,
- Nous construisons les ordonnancements TSCH de manière à attribuer des zones de l'ordonnancement exclusivement réservées pour l'envoi de trames EB, et nous répartissons de manière uniforme au sein d'une slotframe la transmission des informations sur la date du prochaine EB, dans le but de diminuer $T_{adv|max}$.

Afin de profiter d'un trafic de données déjà en cours, nous proposons d'ajouter les informations sur le temps attendu des prochaines trames d'EB à l'intérieur des trames d'Enh-ACK, déjà utilisées par ailleurs dans les réseaux TSCH. 802.15.4-2015 [9] définit deux types d'ACK : les Imm-ACK et Enh-ACK. Les trames d'Enh-ACK peuvent contenir un ou plusieurs IE (Elément d'information — ou *Information Element*), permettant par exemple d'envoyer des informations de synchronisation. En particulier, l'IE RIT est utilisé dans la méthode d'accès RIT pour notifier l'heure prévue d'un rendez-vous pour la transmission des trames de données. Nous proposons d'utiliser cet IE et plus particulièrement son champ "Date du rendez-vous" (en nombre de multiples de 10 symboles), dans les trames d'Enh-ACK afin d'annoncer la date prévue de la prochaine transmission d'EB sur le même canal physique que l'Enh-ACK, en prenant en compte les EB envoyés par les nœuds voisins proches de l'émetteur du Enh-ACK.

De cette manière, lorsqu'un nœud qui souhaite rejoindre le réseau recherche une balise et entend une telle trame d'Enh-ACK, il peut s'endormir et ne se réveiller que quelques instants avant la date annoncée de la trame d'EB, réduisant ainsi sa consommation d'énergie. La figure 4.16 présente les principes temporels du schéma proposé.

4.4.1 Fonctionnement détaillé de DataJoin

Pour obtenir de meilleures performances, la stratégie de planification des trames de balises doit répartir uniformément les EB et les Enh-ACK au sein de la *slotframe*. Dans cette section, nous détaillons les prin-

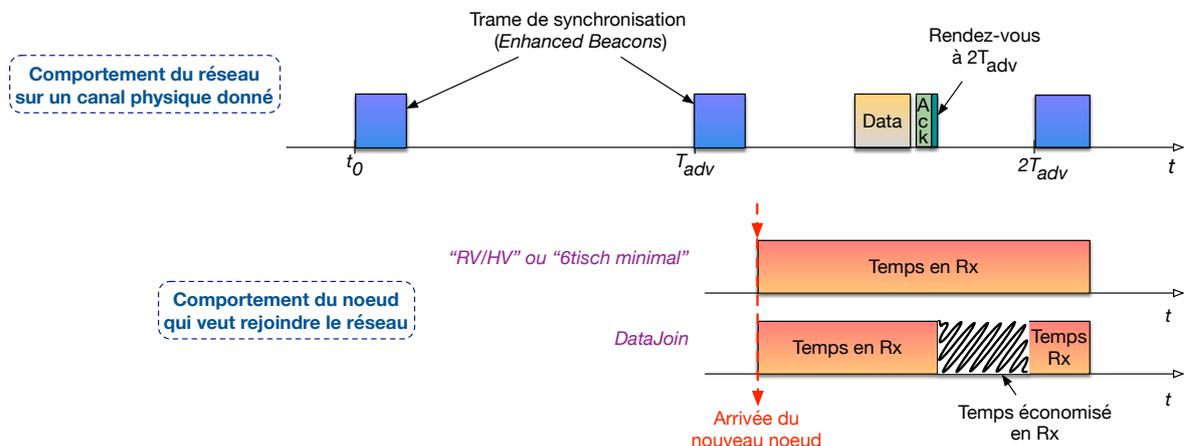


FIGURE 4.16 – Schéma de principe de fonctionnement de DataJoin.

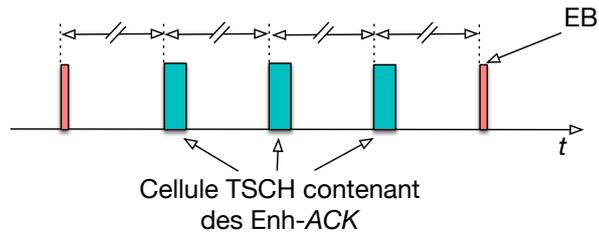


FIGURE 4.17 – Répartition de l'allocation des cellules d'Enh-ACK afin de minimiser la consommation énergétique.

cipes du schéma d'ordonnancement et illustrons son fonctionnement avec un exemple.

4.4.1.1 Contraintes d'envoi des EB

Nous souhaitons tenir compte de la diversité de canal de TSCH et ainsi répartir les EB sur différents canaux, dans le but que le réseau fournisse à un nœud J donné les meilleures dispositions temporelles et fréquentielles vis-à-vis des trames balises et des Enh-ACK, en exploitant les activités des nœuds dans le voisinage. Cela implique de prendre en compte les aspects temporels, fréquentiels et spatiaux lors de l'ordonnancement des EB.

Aspect temporel. Un nœud J qui souhaite rejoindre le réseau se réveille et commence à balayer un canal sélectionné de manière aléatoire.

Comme le montre la figure 4.16, lorsqu'il entend une trame Enh-ACK contenant l'élément IE qui précise la date attendue de la prochaine trame d'EB, il s'endort et ne se réveille que quelques instants avant le début de la transmission de l'EB. Pour faire face à la dérive de l'horloge, le nœud se réveille légèrement plus tôt que la date de l'EB annoncée. Comme indiqué dans le chapitre précédent, la durée de l'intervalle de garde dépend directement du temps passé en état de sommeil et de la précision de l'horloge matérielle.

La figure 4.17 montre comment la construction de l'ordonnancement peut contribuer à réduire la consommation d'énergie : pour un pourcentage de slots contenant des Enh-ACK donné, le protocole d'allocation de ces cellules peut les répartir uniformément sur l'intervalle d'envoi des trames de balises T_{adv} , de sorte qu'un nœud qui tente de rejoindre le réseau passe le moins de temps possible en mode Rx à la recherche d'une trame d'EB.

Aspect fréquentiel. Les réseaux TSCH utilisent le saut de fréquence, ce qui signifie que les nœuds du réseau doivent planifier soigneusement l'envoi des séquences de balise sur tous les canaux.

S'il ne reçoit ni un EB ni un Enh-ACK au bout du temps T_{advmax} , un nœud J considère que le canal qu'il est en train d'écouter n'est pas utilisé et passe à l'écoute d'un autre canal, et ce jusqu'à ce qu'il entende une transmission.

Ainsi, pour obtenir les meilleures performances, nous devons répartir uniformément les transmissions d'EB sur tous les canaux. Comme pour le protocole RV/RH de Vogli *et al.* [3], nous adoptons la notion de *multislotframe* afin de garantir un minimum d'une trame de balise sur chaque canal, dans un intervalle de temps donné. Une *multislotframe* correspond à r répétitions temporelles d'une *slotframe*.

Pour la construction de la *multislotframe*, nous avons défini la règle suivante : à chaque répétition de la *multislotframe*, un même décalage de fréquence logique CO (pour **C**hannel **O**ffset) doit être mappé sur les mêmes canaux physiques CH (pour **C**Hannel) aux mêmes instants, par rapport au début de la *multislotframe*.

Nous pouvons satisfaire à cette condition en choisissant r comme le plus petit commun multiple entre la taille de la *slotframe*, SF_{size} , et HS , la longueur de la séquence de sauts de fréquence.

Pour gérer le cas où un nœud J n'est à portée que d'un seul nœud N , chaque nœud N appartenant au réseau envoie des balises sur chaque canal de la séquence de sauts au sein d'une *multislotframe*.

Aspect spatial. Étant donné que le nœud J qui tente de rejoindre le réseau ne peut recevoir des trames que d'une partie des nœuds du réseau dans un voisinage donné, un nœud N à l'intérieur du réseau peut et doit programmer l'envoi de ses balises de manière non conflictuelle par rapport à ses voisins, soit en

écoutant les emplacements des slots dédiés à l'envoi de EB de ses voisins, soit en exploitant les informations transmises via l'ordonnancement commun TSCH.

Les nœuds sélectionnent leurs cellules avec des balises en fonction des informations de l'ordonnancement de leurs voisins, et des voisins de leurs voisins. Ces informations sont obtenues grâce aux IE de type *Slotframe and Link*, envoyés dans les trames d'EB par défaut. Ainsi, un nœud, après avoir rejoint le réseau TSCH, sélectionne ses propres cellules d'*advertisement*, en fonction des informations qu'il a obtenues sur les cellules d'*advertisement* des nœuds l'entourant, récupérées lors de sa synchronisation au réseau TSCH. De cette manière, nous garantissons qu'il n'y ait pas de collisions avec les nœuds voisins, mais également avec les nœuds en situation de nœuds cachés.

Ainsi, dans notre proposition, les décisions d'ordonnancement des EB ne dépendent que des ordonnancements locaux des nœuds et de ce qu'ils entendent dans leur voisinage, car nous n'utilisons pas de négociation entre les nœuds. Cette gestion locale garantit non seulement l'absence de collisions dans un voisinage donné, mais assure également que le schéma est extensible pour les réseaux plus grands.

4.4.1.2 Allocation des cellules de EB et de Enh-ACK avec *DataJoin*

Les règles de sélection d'une cellule pour l'envoi des balises utilisées par *DataJoin* sont les suivantes :

1. Nous définissons $N_z = \text{seq}_{\text{length}}$ zones temporelles dédiées à l'envoi des EB au sein de la *multislotframe*, afin de transmettre les EB de manière uniformément répartie dans le temps.
2. Un nœud N transmet un EB par canal physique au cours d'une *multislotframe*.
3. Phase Déterministe : Si un nœud N transmet un Enh-ACK sur un canal physique donné, il sélectionne la zone d'EB optimale pour l'envoi de sa trame de balise sur ce canal : la zone d'EB choisie doit respecter au mieux une distribution uniforme entre les trames d'Enh-ACK et les EB sur ce canal physique (cf. figure 4.17).
4. Phase aléatoire : Si aucun Enh-ACK n'est transmis sur un canal physique, les nœuds N sélectionnent aléatoirement une zone d'EB pour la transmission de leur EB sur ce canal physique.
5. Lors de l'ordonnancement des EB, tous les nœuds du réseau N sélectionnent d'abord leurs slots d'EB pour optimiser leurs transmissions vis-à-vis de l'envoi de leurs Enh-ACK pendant la phase déterministe (règle numéro 3); ensuite, ils placent au hasard leurs slots de balises dans les slots restants (règle numéro 4).
6. Dans le cas où le slot sélectionné pour l'envoi d'EB est déjà occupé par une transmission dans le voisinage, le nœud N sélectionnera l'ASN suivant sur le même canal physique pour envoyer son EB.
7. Un nœud ne peut allouer qu'une cellule TSCH pour un même ASN : il ne peut être occupé dans deux transmissions sur des canaux différents pour un même instant.

Avec les informations obtenues lors de l'ordonnancement d'un nœud N , les nœuds peuvent, en plus d'annoncer leur date d'envoi des balises sur le canal physique, annoncer la date de la prochaine transmission d'EB par un de leurs voisins (toujours sur le même canal physique).

Exemple d'allocation

La figure 4.19 présente l'ordonnancement résultant de l'exemple d'application de *DataJoin* pour la topologie de 60 nœuds décrite sur la figure 4.18. L'exemple suivant se concentre dans le voisinage des nœuds 29, 30 et 33, tels que présenté sur la figure 4.18 et coloré de manière à identifier leurs affectations dans la *multislotframe* de la figure 4.19.

Pour l'exemple, nous utilisons 5 canaux physiques et une *multislotframe* de 30 slots. La figure 4.19 montre les différentes cellules disponibles de la *multislotframe*, son découpage en *slotframe*, ainsi que les attributions des cellules aux nœuds impliqués dans les communications.

Les numéros de canaux physiques CH sont indiqués dans le coin supérieur gauche des cellules. Ainsi, pour retrouver les cellules qui sont attribuées sur un canal physique donné, il faut suivre une séquence de cellules diagonales (par exemple, toutes les cellules marquées de 5 pour le $CH = 5$).

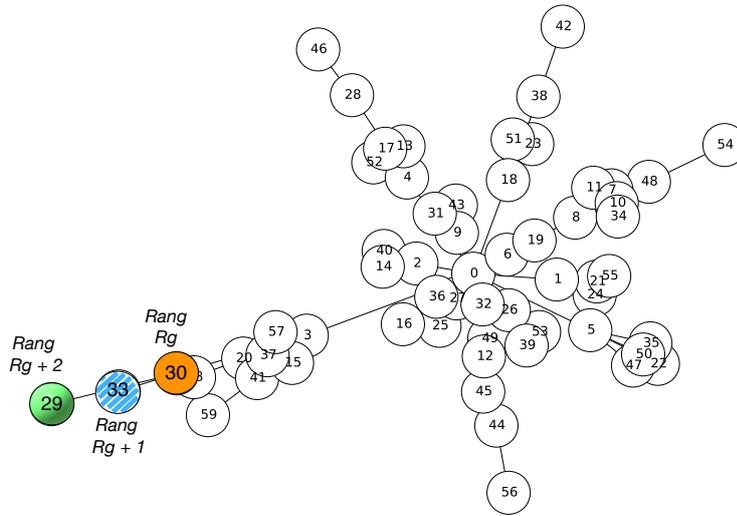


FIGURE 4.18 – Exemple de topologie simulée pour évaluer les performances du protocole *DataJoin*.

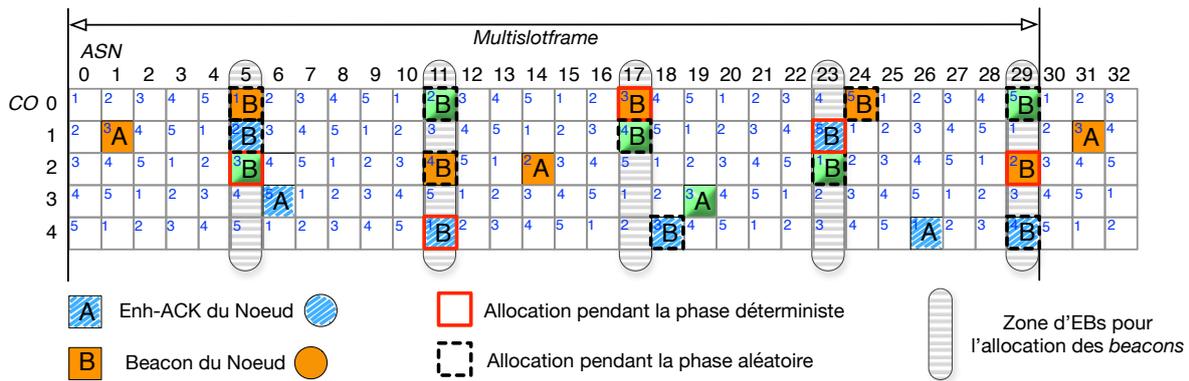


FIGURE 4.19 – Exemple d'un ordonnancement TSCH avec 5 canaux logiques (CO), 5 canaux physiques (numéros dans les cellules), et une *multislotframe* de 30 slots.

La phase déterministe. Au cours de la phase déterministe, les nœuds qui envoient les balises sélectionnent la zone d'EB optimale qui se rapproche du cas idéal tel que présenté sur la figure 4.17, où les cellules actives sont uniformément réparties, et ce pour chacun des canaux physiques de la *multislotframe*. Les cellules entourées de lignes rouges sur la figure 4.19 représentent l'allocation réalisée pendant la phase déterministe.

On peut ainsi noter que le nœud orange 30 attribue des cellules d'EB sur les canaux physiques 3 et 2 (resp. (17,0) et (29,2)) durant cette phase.

De plus, on remarque que ce ordonnancement répond à l'exigence de la figure 4.17 : l'Enh-ACK de la cellule (1,1) (resp. (14,2)) et la trame d'EB de la cellule (17,0) (resp. (29,2)) sont équidistants dans la *multislotframe*.

Les nœuds du voisinage sélectionnent ensuite leurs cellules d'envoi de EB en fonction de leur ordonnancement sur ce canal. Par exemple, le nœud vert 29 choisit d'envoyer sa balise dans la cellule (5,2) sur le canal physique 3.

La phase aléatoire. Après la phase déterministe, les nœuds sélectionnent au hasard leurs cellules d'EB parmi les cellules/zones d'EB disponibles restantes sur un canal physique donné. La figure 4.19 présente ces allocations aléatoires avec des cellules entourées de lignes pointillées jaunes.

Pendant cette phase aléatoire, le nœud orange 30 sélectionne au hasard les cellules (5,0) et (11,2) pour envoyer ses EB sur les canaux physiques 1 et 4, respectivement. La seule possibilité restante pour envoyer un EB sur le canal physique 5 est la zone d'EB à l'ASN 23. Cependant, il y a déjà un EB dans cette cellule (celui du nœud bleu 33).

Dans ce cas, le nœud orange 30 sélectionne la cellule suivante disponible sur le même canal physique, ce qui correspond ici à la cellule (24,0).

L'aspect spatial. Dans *DataJoin*, les nœuds peuvent annoncer la date du prochain EB transmis par un de leurs voisins sur le même canal physique. Cependant, même si le nœud N qui annonce la date de l'EB de son voisin se situe à portée du nœud J , le voisin du nœud N peut se situer hors de portée de ce nœud J . Le nœud J se réveillera alors pour une trame de balise qu'il ne recevra pas. Si cela se produit, nous proposons que le nœud J continue de balayer jusqu'à ce qu'il reçoive une trame d'EB ou un autre Enh-ACK. On risque alors de retomber dans une situation pour laquelle *DataJoin* n'apporte aucun gain de consommation d'énergie, mais notre proposition ne perd pas en performance.

Pour illustrer cette situation, supposons un nœud qui tente de s'attacher au réseau dans le coin inférieur gauche de la figure de topologie (cf. figure 4.18). Supposons de plus qu'il se réveille à l'ASN 24. Il commence alors à surveiller le canal physique 1, aléatoirement sélectionné, et ne peut entendre que les nœuds bleu 33 et vert 29.

Il reçoit un Enh-ACK du nœud bleu 33 qui indique la date du prochain EB du nœud orange 30 sur le canal physique 1, celui de l'ASN 35 (cellule (5,0) dans la prochaine *multislotframe*, non représentée sur la figure). Le nœud J s'endort donc à la suite de ce Enh-ACK et ne se réveille que pour l'ASN 35. Cependant, il ne reçoit pas la trame d'EB, puisque le nœud orange 30 n'est pas dans sa zone de communication. Il doit donc continuer à surveiller le canal physique 1 jusqu'à la réception de la trame d'EB dans la cellule (11,4), à l'ASN 41.

4.4.2 Évaluation des performances

4.4.2.1 Environnement de test

Pour évaluer les performances du schéma *DataJoin*, qui dépend fortement de l'ordonnement des trames des données, nous avons tout d'abord utilisé le simulateur 6TiSCH à temps discret proposé par Watteyne *et al.* [120], contenant l'implémentation du protocole Stripe [56], pour obtenir l'ordonnement distribué des transmissions de trames de données pour une topologie aléatoire, et donc la distribution de l'envoi des trames d'Enh-ACK au cours de la *slotframe*, que nous utiliserons comme paramètre d'entrée du simulateur présenté à la page 102.

Nous utilisons le protocole Stripe pour générer l'ordonnement des transmissions des trames de données dans deux cas extrêmes de trafics applicatifs : un débit de données élevé de 1 paquet applicatif par seconde et par nœud, et cas de faible débit de données avec 1 paquet pour 1000 secondes (environ un paquet toutes les demi-heures). Les paquets de données envoyés sont de taille maximale ($s_a = 116$ octets, paquet PHY de 127 octets).

Connaissant les cellules dédiées à l'envoi des données, nous réalisons l'ordonnement des cellules d'EB avec le schéma *DataJoin* pour obtenir un ordonnancement global du réseau, que l'on donne en entrée du simulateur de la procédure d'association à un réseau TSCH présenté au paragraphe 4.2 (cf. page 102).

Nous comparons *DataJoin* avec un schéma que l'on nomme *Baseline*, et qui correspond quasiment à un ordonnancement des balises suivant le protocole de ND *Random Vertical* (RV), tel que présenté par Vogli *et al.* [3] (cf. paragraphe 4.1.1.2). Nous avons légèrement amélioré RV pour le rendre directement comparable à *DataJoin* :

- Nous nous assurons que les balises sont réparties sur tous les canaux physiques (la proposition originale de RV sélectionne effectivement les cellules en fonction des canaux logiques et non sur les canaux physiques)

- Nous assurons qu'il n'y a pas de collision de balise.

En termes de consommation d'énergie du point de vue de la consommation du réseau, l'ordonnement sur des canaux logiques ou physiques est strictement équivalent.

En revanche, pour ce qui est de la consommation d'énergie des nœuds qui souhaitent s'attacher au réseau, le schéma *Baseline* est meilleur qu'RV puisqu'un nœud J est certain de recevoir une trame d'EB sur n'importe lequel des canaux dans un délai plus court, car tous les nœuds N envoient des EB sur tous les canaux. En supposant que la taille de la *multislotframe* et la longueur de la séquence de saut seq_{length} sont premier entre eux, RV garantit qu'un nœud J rejoigne le réseau sur un canal donné en une répétition de la *multislotframe*.

Cependant, s'ils ne sont pas premier entre eux, un nœud qui cherche à rejoindre le réseau peut avoir besoin de changer de canal d'écoute parce qu'il ne reçoit aucun EB sur le canal d'écoute sélectionné aléatoirement en premier lieu : il peut de plus ne pas entendre tous les nœuds qui envoient des EB, de telle sorte que RV peut ne pas aboutir à l'attache du nœud J au réseau, dans certains cas. De plus, étant donné que nous utilisons un ordonnancement qui garantit aucune collision avec les trames d'EB des nœuds voisins, le système *Baseline* obtient aussi de meilleures performances que RV en ce qui concerne la consommation énergétique des nœuds du réseau.

Ainsi, nous comparons les performances de notre proposition à celle que l'on a nommée *Baseline*, qui surpasse elle-même la proposition de Vogli *et al.* [3]. Nous simulons une topologie aléatoire de 60 nœuds et supposons des liaisons stables et parfaites. Nous construisons les ordonnancements des nœuds du réseau et du nœud qui tente de rejoindre suivant les deux stratégies *DataJoin* et *Baseline*. Ensuite, le simulateur présenté au paragraphe 4.2 (cf. page 102) est utilisé pour calculer les performances de *DataJoin* et *Baseline* en termes de consommation d'énergie et de délai d'attache. Les résultats présentés correspondent aux valeurs moyennes calculées sur toutes les valeurs possibles du paramètre α (avec un pas de 100 μ s).

Pour une topologie aléatoire donnée, nous sélectionnons deux positions représentatives du nœud J : le "meilleur" et le "pire" cas. La meilleure position correspond à la situation dans laquelle le nœud qui tente de rejoindre le réseau entend un grand nombre de nœuds du réseau (haute densité de nœuds). Le cas le plus défavorable correspond à une faible densité de nœuds dans la zone de portée du nœud J , par exemple un nœud qui voudrait rejoindre le réseau à côté des terminaisons du réseau.

4.4.2.2 Résultats

La figure 4.20 présente une comparaison des performances des propositions *Baseline* et *DataJoin* pour deux positions des nœuds J . La première position, qui représente le meilleur cas, est choisie comme une position où le nœud qui veut rejoindre le réseau est à portée d'un relativement grand nombre de nœuds N . La seconde, qui représente le pire cas, est sélectionnée de manière à ce que le nœud J n'entende qu'un petit nombre de nœuds N du réseau TSCH. Nous avons sélectionné un nombre de voisins à portée du nœud J de $N_{neigh} = 15$ pour le meilleur cas et $N_{neigh} = 2$ pour le pire cas. De plus, la comparaison évalue *DataJoin* et *Baseline* dans deux contextes applicatifs différents : un débit de trafic élevé et un plus faible.

Rappelons que le délai de jointure d_{join} et le temps Rx sont proportionnels à la consommation d'énergie du nœud qui tente de rejoindre le réseau, et seront donc considérés comme des indicateurs de la consommation énergétique.

La figure 4.20 confirme que pour une stratégie de réseau donné, *DataJoin* et *Baseline* obtiennent des résultats similaires en ce qui concerne le délai d'attache d_{join} (voir la colonne de gauche de la figure 4.20) et le ratio d'activité moyen en Tx (voir la colonne de droite de la figure 4.20). On s'attendait à ce que les mesures soient similaires dans les deux cas, car les deux solutions utilisent le même ordonnancement d'envoi des trames d'EB, et que seule la stratégie d'attache des nœuds J diffère.

La colonne centrale de la figure 4.20 montre que les nœuds J qui utilisent la stratégie *DataJoin* passent moins de temps en Rx , donc consomment moins d'énergie que *Baseline*, et ce dans toutes les configurations (le meilleur et le pire cas, avec un trafic de données faible ou élevé). Par exemple, les nœuds "Best-*DataJoin*" passent deux fois moins de temps en mode Rx (0,5s vs. 1s) que les nœuds "Best-*Baseline*", pour un trafic de données élevé (voir Fig.4.20a). Notez que le gain de temps en Rx de *DataJoin* est plus élevé dans le meilleur cas que dans le pire des cas, car le meilleur cas correspond à un voisinage plus dense, donc la probabilité d'entendre un Enh-ACK est plus forte, ce qui améliore encore les performances de *DataJoin*.

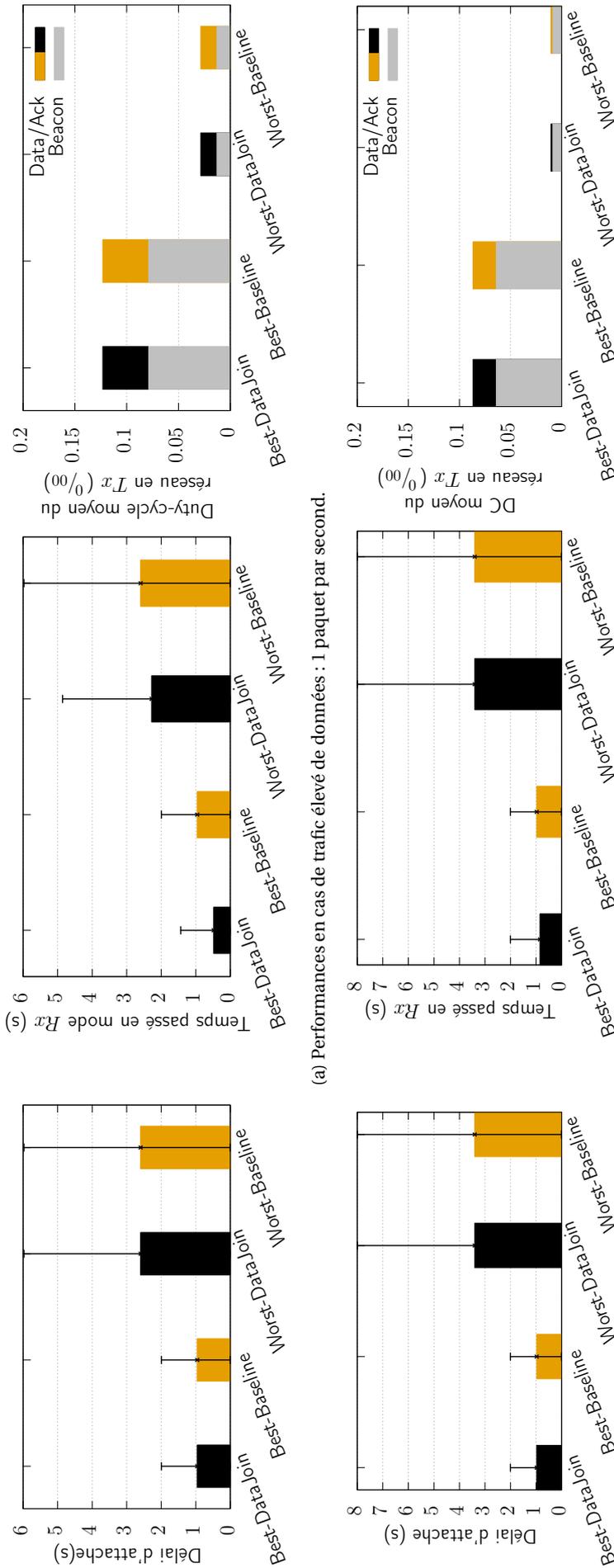
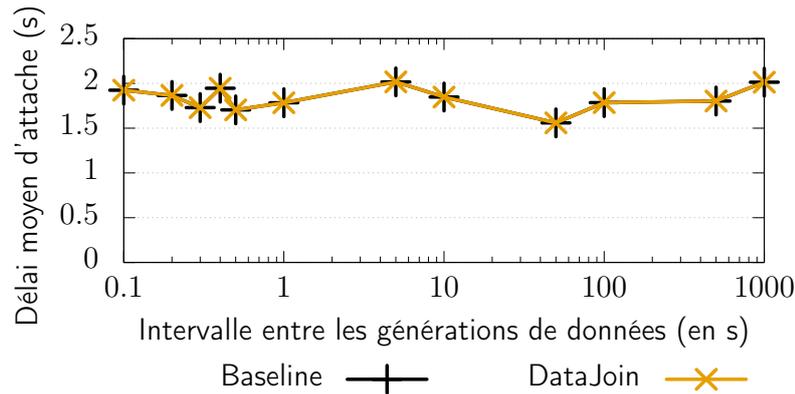
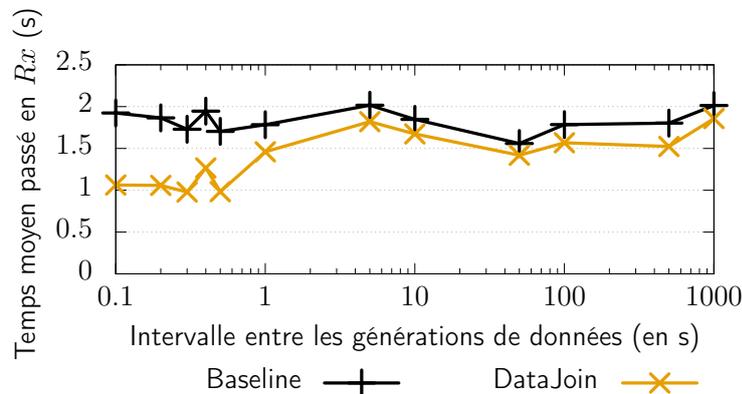


FIGURE 4.20 – Comparaison des performances *Baseline* et *DataJoin* pour deux positions d'un nœud qui tente de rejoindre le réseau : dans un endroit du réseau dense (*Best*) et dans un endroit épars (*Worst*), pour un trafic d'intensité faible et un trafic de forte intensité.

Rappelons que nous utilisons le temps R_x et T_x comme métriques proportionnelles à la consommation d'énergie. Les simulations montrent que dans tous les cas, *DataJoin* surpasse le schéma de *Baseline* en termes de consommation d'énergie du point de vue du nœud J , pour une consommation d'énergie du point de vue du réseau et un délai d'attache similaires. De plus, nous observons que le gain de performance de *DataJoin* augmente avec le taux de cellules contenant des Enh-ACK : alors qu'en cas de trafic de forte intensité, *DataJoin* passe 2 fois moins de temps en R_x que *Baseline* pour un faible trafic de données (voir Fig. 4.20b), "Best-*DataJoin*" ne passe que 10% moins de temps en R_x que pour le cas "Best-*Baseline*" (0,9 s vs. 1 s).



(a) Délai d'attache, d_{join} .



(b) Temps passé en mode R_x , d_{on} .

FIGURE 4.21 – Comparaison de performances pour une période de paquet variable (en s).

Pour confirmer cette tendance, nous avons évalué *DataJoin* pour des débits de données variables, toujours pour une topologie aléatoire de 60 nœuds. La figure 4.21 présente les valeurs moyennées sur 50 positions aléatoires différentes du nœud J , pour un même ordonnancement de données de réseau obtenu grâce au simulateur 6TiSCH étendu avec le protocole Stripe.

Nous avons fixé la taille de la *multislotframe* à 1000 slots et nous supposons un réseau qui utilise cinq canaux logiques pour cinq canaux physiques. La figure 4.21 montre un temps d'activité en R_x plus court, donc une consommation d'énergie plus faible, pour le schéma *DataJoin*, pour un délai d'attache moyen identique. Les délais d'attache sont identiques puisque la proposition *DataJoin* ne permet pas de réduire ce délai, mais d'en limiter la consommation énergétique associée.

Cela confirme également la tendance observée sur la figure 4.20 : le gain de *DataJoin* augmente lorsque les débits de données augmentent, ce qui est compréhensible, du fait de l'augmentation des trames de Enh-ACK.

Conclusion

Dans ce chapitre, nous avons dans un premier temps brièvement rappelé les protocoles de ND existants.

Nous avons ensuite proposé deux nouveaux protocoles d'attache à un réseau TSCH. À l'aide d'un simulateur de la phase d'attache à un réseau TSCH, nous avons validé les performances de ces deux propositions, en montrant qu'elles permettent de réduire la consommation d'énergie du nœud qui tente de rejoindre le réseau en une durée minimale, sans pour autant induire une augmentation de la consommation des nœuds au sein du réseau.

Alors que la première solution permet d'atteindre des DC très bas de manière déterministe, la deuxième solution permet de réduire la consommation des nœuds qui tente de rejoindre le réseau de manière statistique, en fonction du lieu d'apparition de ce nœud, ainsi que de l'organisation du trafic de données au sein du réseau TSCH.

De plus, la deuxième solution, bien que moins bonne en termes de performance énergétique pure – le temps moyen en Rx est plus élevé que dans le cas de l'utilisation de CDS – est plus simple à déployer, est multicanal, et ne demande aucune connaissance *a priori* des nœuds qui tentent de s'attacher au réseau existant. C'est pourquoi nous recommanderons plutôt cette solution dans le cadre d'un réseau TSCH.

De plus, nous montrons que les durées passées en mode Rx pour rejoindre le réseau sont, dans les deux solutions proposées de l'ordre de la seconde. Il semble ainsi désormais envisageable de considérer un mode de communication non synchronisé, dans lequel les nœuds se désassocient pour ne s'attacher au réseau que lorsqu'ils auront besoin d'effectuer une communication.

En effet, si l'on considère une valeur de temps passé en Rx de $\sim 2s$ pour la procédure d'attache, on peut l'utiliser et la remplacer dans l'équation 2.14, pour déterminer la nouvelle fréquence de données $T_{a|seuil}$ à partir de laquelle un nœud aurait intérêt à se désynchroniser. On peut ainsi réécrire l'équation 2.14 de la manière suivante :

$$T_{a|seuil} = T_{syn} \times \left(\frac{d_{Rx|join}}{d_{syn}} + 1 \right), \quad (4.20)$$

où $d_{Rx|join}$ représente le temps passé en mode Rx pour la procédure d'attache au réseau.

En gardant les mêmes hypothèses que dans le chapitre Discussion pour le coût de la synchronisation ($T_{syn} = 12.5s$ et $d_{syn} = 1.5ms$), et en utilisant une valeur moyenne de $1.5s$ pour *DataJoin* (cf. figure 4.11c) et de $400ms$ pour la stratégie S-CDS (cf. figure 4.14), on trouve que notre procédure de synchronisations/désynchronisations successives devient intéressante dès que la période des données à envoyer T_a est plus faible que :

- $T_{a|seuil} \approx 3345,83s$, soit environ une heure dans le cas du protocole S-CDS. On peut noter que ce résultat est excellent, mais, comme nous l'avons mentionné, il ne prend pas en compte l'aspect multicanal des réseaux TSCH.
- $T_{a|seuil} = 12512,5s$, soit environ trois heures et demie dans le cas du protocole *DataJoin*. Ce résultat est $10\times$ moins important que celui existant dans les réseaux TSCH tels que suggérés par le brouillon *6tisch-minimal* [38].

Ainsi, nous avons créé un moyen de rendre deux modes de communication interopérables au sein d'une même technologie TSCH : un mode de communication synchrone, le TSCH classique, avec un mode de communication asynchrone, les synchronisations/désynchronisations successives rendues possibles par les améliorations proposées pour la phase initiale dans laquelle un nœud tente de rejoindre le réseau.

Comme nous l'avons vu dans le deuxième chapitre du présent manuscrit, la situation d'un nœud qui tente de rejoindre un réseau synchrone, et celle d'un nœud asynchrone qui tente de communiquer avec le réseau synchrone sont similaires. Ainsi, nous proposons dans le chapitre suivant d'utiliser l'ordonnanceur intelligent de notre seconde proposition, *DataJoin*, pour ajouter des slots dédiés à la réception de trames de trafic asynchrone contraint en latence de type Thread dans les ordonnancements des nœuds d'un réseau TSCH synchrone. L'utilisation de cet ordonnanceur intelligent permettra de garantir l'acheminement dans une latence maximale, de la même manière qu'il garantit un délai d'attache au réseau maximal.

Application à l'interopérabilité Google Thread/TSCH

Sommaire

4.1 Les protocoles de découverte de voisinage	95
4.1.1 Procédures d'attachement dédiées à la technologie TSCH	96
4.1.1.1 RapidJoin	96
4.1.1.2 Random Vertical / Random Horizontal	97
4.1.2 Protocoles de ND classiques	98
4.1.2.1 Searchlight	98
4.1.2.2 Nihao	99
4.1.2.3 Découverte périodique par intervalle	100
4.1.3 Ensemble de différence cyclique	100
4.2 Simulateur de la procédure d'attachement au réseau	102
4.3 Singer-Cyclic Difference Set	103
4.3.1 Présentation du schéma de ND proposé	103
4.3.2 Évaluation des performances de la proposition S-CDS	105
4.3.2.1 Point de départ de la comparaison	105
4.3.2.2 Résultats pour une valeur de SDC_N fixée	107
4.3.2.3 Courbes de tendance en fonction du DC	107
4.3.3 Conclusion sur les S-CDS	110
4.4 DataJoin	111
4.4.1 Fonctionnement détaillé de <i>DataJoin</i>	111
4.4.1.1 Contraintes d'envoi des EB	112
4.4.1.2 Allocation des cellules de EB et de Enh-ACKavec <i>DataJoin</i>	113
4.4.2 Évaluation des performances	115
4.4.2.1 Environnement de test	115
4.4.2.2 Résultats	116

Introduction

Dans le but d'acheminer des trafics asynchrones contraints en latence tout en conservant une faible consommation d'énergie, nous avons suggéré dans le deuxième chapitre de réaliser une optimisation conjointe MAC de deux technologies, l'une synchrone et l'autre asynchrone. Dans ce chapitre, nous proposons une interopérabilité d'un réseau synchrone TSCH et d'un réseau asynchrone Thread, permettant ainsi d'acheminer un trafic asynchrone contraint en latence grâce au réseau TSCH voisin. Une telle interopérabilité permet effectivement d'utiliser un réseau synchrone TSCH dans lequel chacun des nœuds fonctionne par intermittence plutôt qu'un dispositif constamment disponible en mode *Rx* (comme c'est le cas dans Thread) pour acheminer le trafic contraint en latence d'un nœud asynchrone contraint en énergie.

Comme ces deux technologies fonctionnent avec la même couche PHY, nous n'avons pas ici à nous soucier des différentes contraintes physiques liées à différentes modulations, et une seule radio par nœud est suffisante pour que les nœuds aient la possibilité d'exploiter deux modes MAC.

De plus, même si les couches MAC Thread et TSCH sont distinctes, elles respectent toutes les deux le format des trames MAC de commandes et de données du standard 802.15.4 [9] : les nœuds pourront donc échanger ce type de trames, au niveau MAC, pour réaliser l'interopérabilité.

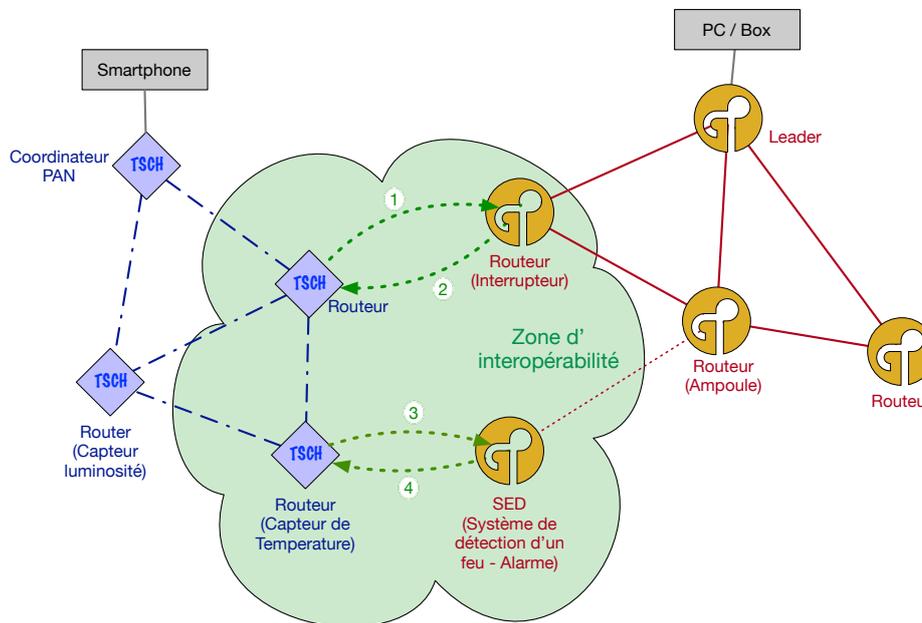


FIGURE 5.1 – Cohabitation de réseaux Thread et TSCH.

La figure 5.1 représente deux réseaux, l'un utilisant le mode MAC synchrone TSCH, l'autre implémentant la technologie Thread. Sur cette figure, nous avons reporté un nuage d'interopérabilité, qui représente la zone dans laquelle l'interopérabilité des deux technologies peut être exploitée. Dans cette zone, on peut identifier quatre types de communications, correspondants à celles que nous avons identifiées de manière générique dans le chapitre deux :

1. Un nœud synchrone TSCH souhaitant communiquer avec un routeur asynchrone constamment disponible Thread,
2. Un routeur asynchrone disponible Thread souhaitant communiquer avec un nœud synchrone du réseau TSCH,
3. Un nœud synchrone TSCH souhaitant envoyer un paquet à un nœud asynchrone contraint en énergie du réseau Thread.
4. Un nœud asynchrone contraint Thread souhaitant communiquer à destination d'un nœud synchrone TSCH.

En effet, les autres liens (représentés en bleu ou en rouge) ne sont pas liés à l'interopérabilité puisqu'ils représentent des communications d'un nœud TSCH à destination d'un nœud TSCH, ou d'un nœud Thread vers un autre nœud Thread.

Dans le deuxième chapitre (cf. page 65), nous avons mis en avant le fait que la mise en place des deux premiers types de communications ne présente pas de difficultés techniques, en raison de la nature constamment disponible du nœud asynchrone alimenté en énergie. Nous avons de plus discuté de la façon de réaliser les communications à destination du nœud asynchrone contraint en énergie, qui repose sur l'utilisation d'un nœud *proxy* responsables des communications à destination de ce nœud

Dans ce chapitre, nous nous focalisation sur le quatrième type de communication : la transmission initiée de manière asynchrone d'un trafic contraint en latence, au sein d'un réseau synchrone disponible par intermittence. Ainsi, nous proposons de remplacer le rôle du nœud asynchrone alimenté par une activité soigneusement planifiée des nœuds synchrones dans le but de diminuer la consommation d'énergie globale.

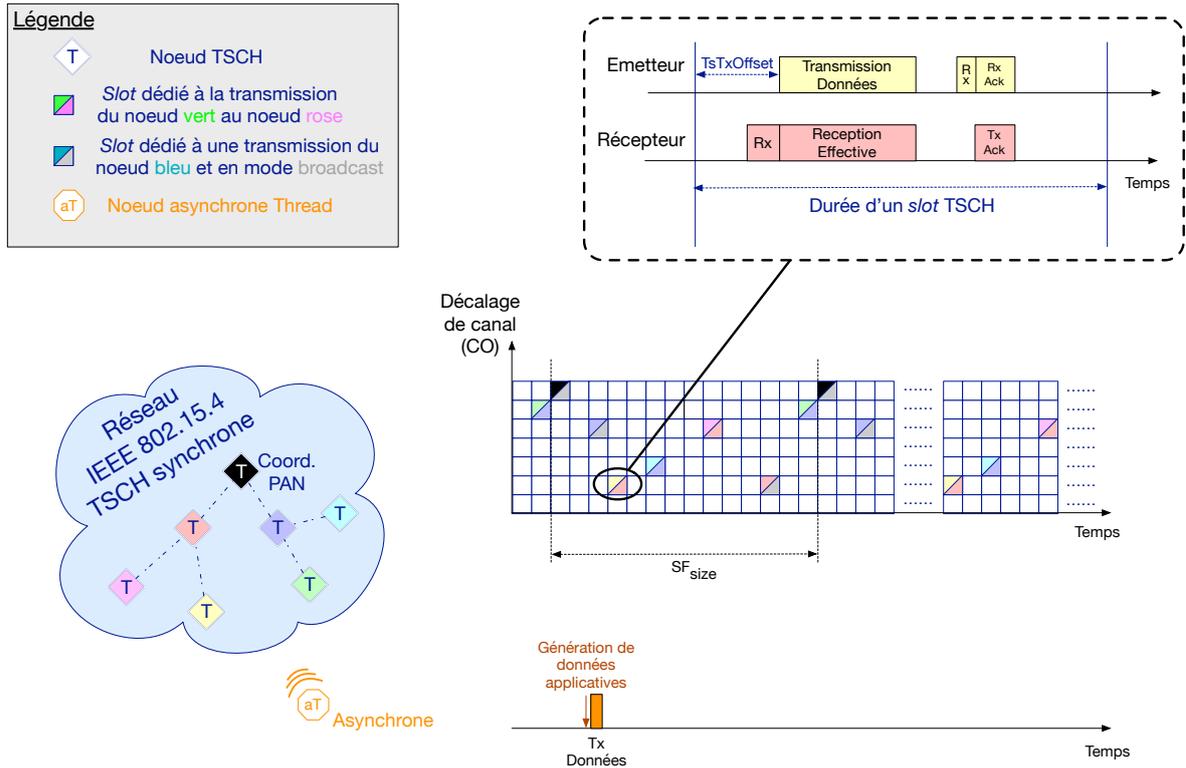


FIGURE 5.2 – Présentation graphique de la problématique : comment modifier les comportements des nœuds TSCH contraints en énergie et du nœud Thread, également contraint, pour acheminer un trafic de nature asynchrone dans une latence L_{max} ?

La figure 5.2 présente graphiquement cette problématique : pour une latence donnée L_{max} , comment modifier les comportements des nœuds TSCH et du nœud asynchrone contraint Thread pour garantir qu'un paquet généré par le nœud contraint soit transmis au cœur de réseau TSCH avec une latence L telle que $L < L_{max}$.

L représente le temps mis pour la communication point à point du nœud asynchrone contraint à destination d'un nœud synchrone, L_{AtoS} , auquel s'ajoute la latence liée aux transmissions du paquet au sein du réseau TSCH, L_{TSCH} .

L_{TSCH} est par définition très fortement liée à la nature de l'algorithme d'ordonnancement des transmissions au sein du réseau TSCH. Keshavarzian *et al.* [57] présentent une étude des différentes latences induites par un réseau multisaut, et proposent un système pour les minimiser, à la fois pour les trafics montants et descendants. Dans ces travaux, nous supposons que les moyens mis en œuvre pour acheminer le trafic au cœur du réseau synchrone reposent sur une problématique propre aux réseaux synchrones, non étudiés ici.

De ce fait, nous travaillerons à contraindre la valeur L_{AtoS} telle que :

$$L_{AtoS} < L_{obj} \tag{5.21}$$

avec $L_{obj} = L_{max} - L_{TSCH}$, où L_{TSCH} est supposé connu.

Pour ce faire, nous proposons d'apporter des modifications dans le comportement des nœuds synchrones du réseau TSCH, en ajoutant dans leurs ordonnancements des slots dédiés à l'acheminement du trafic du nœud asynchrone contraint. Nous modifions également le comportement du nœud contraint pour qu'il insère ses communications dans l'ordonnancement TSCH, sans pour autant perturber le fonctionnement correct du réseau synchrone.

Ainsi, plusieurs problématiques se posent pour permettre ces communications :

- Comment doivent se comporter les nœuds du réseau TSCH au sein des slots dédiés à l'interopérabilité?
- Comment doit se comporter le nœud asynchrone contraint lors de la génération de données à destination du réseau TSCH?
- Comment ordonnance-t-on, en temps et en fréquence, les slots d'interopérabilité au sein d'un réseau TSCH, de manière à respecter les contraintes de latence de l'équation 5.21, ainsi que les contraintes énergétiques de chacun des nœuds?
- Quel est le coût d'une telle modification, à la fois en termes de latence, et d'énergie consommée, par le nœud asynchrone et par les nœuds TSCH?

La suite du chapitre s'organise de la façon suivante : dans un premier temps, nous présentons le comportement des nœuds TSCH et du nœud Thread lorsque leurs activités respectives se superposent. Nous remarquons que la période de l'activité des nœuds du réseau synchrone et la nature asynchrone du trafic à acheminer correspondent à la description des mécanismes d'économie d'énergie du standard 802.15.4 (CSL et RIT). Ainsi, nous proposons d'utiliser des comportements similaires aux modes CSL et RIT pour la mise en place de la communication du nœud Thread contraint à destination d'un nœud TSCH également contraint.

Nous discutons de ces deux approches, et montrons que dans notre cadre de travail, la solution exploitant les mécanismes RIT est plus appropriée, car moins perturbatrice du réseau TSCH. De plus, la possibilité de coupler l'envoi d'information RIT aux trames d'EB représente un avantage non négligeable de la solution RIT.

Dans une deuxième section, nous expliquons le principe d'ordonnancement des slots de communication d'interopérabilité au sein du réseau TSCH, de manière à s'assurer qu'ils se produisent suffisamment fréquemment pour garantir la latence L_{AtoS} . Comme le mode RIT permet d'exploiter les trames d'EB pour transporter les informations de sollicitation RIT, nous proposons d'utiliser et d'améliorer la proposition du chapitre précédent, *DataJoin*, pour réaliser l'ordonnancement des envois d'EB contenant les informations RIT.

Enfin, dans une dernière section, nous évaluerons l'impact des solutions proposées, en termes de latence L_{AtoS} , d'énergie consommée par le nœud asynchrone contraint, par les nœuds du réseau TSCH, et l'énergie globalement consommée avec ou sans notre proposition, pour la prise en charge du trafic asynchrone contraint en latence.

5.1 Principe de fonctionnement

Les deux premières problématiques présentées ci-dessus sont relatives au comportement de chacune des entités lors de la superposition de deux périodes d'activité : dans une telle situation, comment les nœuds TSCH et Thread contraints doivent se comporter pour que la transmission ait effectivement lieu.

Nous avons vu dans le deuxième chapitre qu'il existe différents mécanismes MAC pour répartir l'activité d'un nœud dans le temps. Les techniques utilisées par les technologies présentées dans le premier chapitre s'organisent en quatre catégories, comme le montre la figure 2.1 (cf. page 51) : les mécanismes synchrones fonctionnant à l'aide de rendez-vous de communication (comme le mode TSCH), les mécanismes asynchrones exploitant des préambules (le mode CSL), ou des trames de sollicitation (le mode RIT), et enfin les mécanismes asynchrones de transmissions supposant un récepteur constamment disponible (le mode non *beacon*).

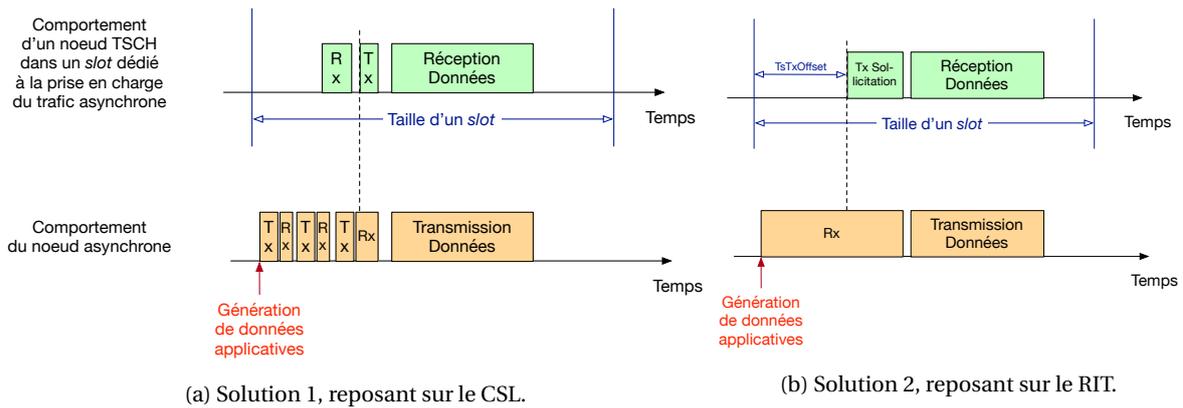


FIGURE 5.3 – Détail du fonctionnement au sein d'un slot d'interopérabilité.

Nous remarquons que ces quatre modes de communication MAC existent au sein du standard 802.15.4. Ainsi, le format des trames MAC décrit par la figure 1.2 (cf. page 28) sera décodable par n'importe quel module radio disponible certifié 802.15.4, dans chacun des quatre modes de communication mentionnés.

Dans cette partie, nous proposons d'exploiter les deux modes de fonctionnement asynchrone reposant sur l'échange de trames de préambule ou de trames de sollicitation au service de l'interopérabilité des deux modes de communication synchrone et asynchrone.

En effet, nous remarquons que la période de la *slotframe* du mode TSCH permettrait de réaliser périodiquement la détection de trames de préambules pour la communication avec le nœud asynchrone contraint fonctionnant en mode CSL. À la génération de données applicatives, ce dernier devrait alors modifier son comportement pour envoyer des trames de préambules, permettant ainsi l'établissement d'une communication entre les deux nœuds en exploitant les principes du mode CSL du standard 802.15.4.

Un raisonnement similaire est réalisable pour la période des trames de *RIT Data Request* et l'exploitation du mode RIT du 802.15.4.

5.1.1 Modifications apportées vis-à-vis des standards existants

Une telle modification du fonctionnement du comportement des nœuds TSCH ne perturbera ni leurs modes opératoires ni celui de leurs voisins synchrones TSCH, puisqu'on ne modifie pas les contraintes temporelles de l'ordonnancement TSCH, et que les slots alloués par un nœud pour la réception d'une trame de type asynchrone sont considérés par les voisins de ce nœud comme n'importe quel autre slot de communication de l'ordonnancement qui ne les concerne pas. La modification consiste donc en l'ajout de slots dédiés à la communication avec le nœud asynchrone dans l'ordonnancement TSCH. Suivant que l'on choisisse d'exploiter le mode RIT ou le mode CSL, ces slots correspondront à des cellules TSCH classiques de types *Rx* ou *Tx*.

Du point de vue du nœud asynchrone contraint en revanche, la modification est plus ample : dans la spécification Thread, les nœuds contraints réalisent une procédure de CSMA/CA avant d'envoyer leurs trames de données dès la génération de données applicatives (cf. figure 1.3, page 29). Pour permettre la communication à destination d'un des nœuds du réseau synchrone, disponible uniquement par intermittence, il est nécessaire de modifier ce comportement, dans le but de se rapprocher du comportement d'un nœud RIT ou CSL lors de la génération de paquet applicatif.

5.1.1.1 Mode CSL

Si l'on considère une interopérabilité exploitant le mode CSL, nous proposons d'ajouter dans l'ordonnancement des nœuds TSCH des cellules de type *Rx*, permettant aux nœuds TSCH d'être périodiquement disponibles pour la réception de trames de préambule.

De ce fait, pour exploiter le mode CSL, il faut modifier le comportement du nœud Thread pour qu'il envoie des trames de préambule pendant un intervalle plus grand que la période des slots d'écoute CSL du

réseau TSCH. De cette manière, on peut garantir qu'au moins un des nœuds du réseau TSCH reçoive le préambule, et se rende ainsi disponible pour la transmission de la trame de données. Dans le mode CSL, les préambules sont constitués de trame de type générique, ou *multipurpose*, contenant un IE donnant la date du prochain rendez-vous de communication. Ainsi, si l'on exploite ce mode de communication, le nœud asynchrone contraint impose ses contraintes temporelles en indiquant la date de sa transmission de données, et les nœuds du réseau TSCH devront s'y adapter pour recevoir la transmission asynchrone.

Un tel mode de fonctionnement implique cependant de perturber le réseau synchrone TSCH de deux manières :

- Premièrement, l'envoi de trames de préambules par le nœud asynchrone contraint vient temporairement perturber le fonctionnement du réseau TSCH sur le canal utilisé par ce dernier, puisqu'il agit comme un brouilleur.
- Deuxièmement, le réseau synchrone TSCH, qui possède déjà des contraintes temporelles par sa nature synchrone, devra se plier aux exigences temporelles des nœuds asynchrones, indiquées dans les trames de préambule. Ainsi, les nœuds TSCH, pour se rendre disponible à l'instant requis par le nœud asynchrone, vont temporairement ne pas suivre leur ordonnancement TSCH, au risque de perdre la synchronisation du réseau.

Alors que la première contrainte paraît inhérente à l'utilisation du mode CSL, il est possible de mitiger l'impact de la seconde contrainte. Pour ce faire, nous proposons de mettre en place un mécanisme de CSL reposant sur l'échantillonnage de préambule amélioré par l'utilisation des trames d'acquiescement.

On peut en effet envisager qu'un nœud du réseau TSCH envoie un ACK dès la réception du préambule, de manière à accélérer la procédure d'envoi du paquet du nœud asynchrone, et surtout garantir que cet envoi se produise au sein du slot sélectionné par le réseau comme un slot dédié à la communication asynchrone. Le fonctionnement d'un tel slot est détaillé par la figure 5.3a, en présentant les modes d'opération d'un nœud du réseau TSCH, ainsi que celui du nœud asynchrone Thread.

5.1.1.2 Mode RIT

La figure 5.3b présente le fonctionnement proposé si l'on exploite le mode RIT. Dans ce contexte, un nœud du réseau TSCH envoie des trames de RIT *Data Request* au début d'une cellule TSCH dédiée à la communication avec le nœud asynchrone contraint, et se rend ensuite disponible pour la réception potentielle d'une trame de données.

Dans ce mode, à la génération d'une trame de données, le nœud asynchrone se positionne en mode *Rx* de manière continue, jusqu'à la réception d'une trame de sollicitation, avant d'y répondre avec la donnée applicative à transmettre.

Ce mode de fonctionnement possède l'avantage de ne pas perturber le fonctionnement temporel du TSCH, puisque ce sont les nœuds du réseau TSCH qui choisissent la date d'envoi des trames asynchrones, via l'envoi de leurs trames de sollicitations.

Couplage du RIT avec les trames d'EB. Par ailleurs, un autre avantage du mode de communication RIT est que l'envoi des trames RIT *Data Request* n'est pas nécessaire pour envoyer les informations de sollicitation. En effet, comme l'IE RIT peut être embarqué dans une trame d'EB, il est possible de coupler l'envoi des balises avec l'envoi de ces informations de sollicitation.

Un tel couplage permettrait de rendre quasiment négligeable le surcoût énergétique lié à la fonctionnalité d'interopérabilité : plutôt que d'allouer un slot de communication entier, pour garantir l'envoi d'une trame MAC de sollicitation par *slotframe* au minimum, l'exploitation des trames d'EB permettrait de réduire le surcoût à seulement 4 octets transmis sur l'air, ajoutés au sein de la trame d'EB.

Ce couplage permet de plus de rendre égales la période d'envoi des balises T_{adv} et la période des trames de sollicitation T_{poll} . De ce fait, que ce soit pour le mécanisme d'attache au réseau, ou pour la transmission d'interopérabilité, coupler les deux trames permet à l'un des mécanismes d'exploiter les performances, en termes de latence, requises par l'autre mécanisme, tout en négligeant le surcoût énergétique.

Néanmoins, comme la trame d'EB occupe quasiment la totalité du temps disponible pour la transmission de données au sein d'un slot TSCH, il n'est pas possible de réaliser le slot d'interopérabilité de la manière présentée sur la figure 5.3b. Il faut alors dédier un slot de réception plus tard dans l'ordonnancement TSCH pour la réception potentielle de la trame de données : une cellule TSCH de type *Rx* classique.

Comme la date de la prochaine période d'écoute du nœud émetteur de la trame de sollicitation RIT est incluse dans cette trame, il est possible d'indiquer une date ultérieure, correspondante à un des slots libres de l'ordonnancement TSCH du nœud émetteur du RIT, alloué à la volée par ce dernier comme un slot de réception classique TSCH.

5.1.2 Discussion

Le choix de la technologie RIT ou CSL est motivé par plusieurs éléments. Le premier, nous l'avons vu peut venir de l'effet perturbateur du mode CSL sur l'ordonnancement du TSCH : dans le mode exploitant les principes du CSL, les contraintes temporelles seront imposées par le nœud asynchrone, qui, par définition, ne connaît aucun des besoins temporels du réseau synchrone.

Cependant, le mode CSL permettrait de mutualiser les activités de différents nœuds TSCH au sein d'une même cellule dédiée à la communication asynchrone : en effet, comme le montre la figure 5.3a, dans ces cellules, le fonctionnement des nœuds TSCH est de se tenir en mode Rx pour la réception potentielle d'une trame de préambule. On peut ainsi mutualiser la ressource d'interopérabilité du point de vue du réseau. De ce fait, des nœuds à des positions géographiques distinctes peuvent sélectionner le même slot et ainsi étendre la zone de portée capable de gérer un trafic asynchrone contraint en latence.

Au contraire, une telle mutualisation n'est pas réalisable si l'on utilise le mode de communication RIT : dans ce mode, les nœuds du réseau transmettent des trames de sollicitation au sein des slots d'interopérabilité. La mutualisation de ce slot impliquerait donc des collisions potentielles des trames de sollicitations.

Si l'on considère le cas où le réseau est constitué d'un grand nombre de nœuds asynchrones, pour un plus petit nombre de nœuds synchrones, la probabilité d'occurrence de ces transmissions asynchrones par nœud synchrone est élevée. Dans ce cas, il peut être intéressant de répartir la charge représentée par ses communications d'interopérabilité parmi les nœuds du réseau synchrone à portée du nœud asynchrone. Une telle opération est possible en mutualisant les slots d'interopérabilité, et en sélectionnant le nœud synchrone sollicité à l'aide d'un algorithme de type CSMA/CA sur l'envoi de la trame d'ACK du préambule. Cependant, il est également possible de répartir la charge de ces nœuds de manière temporelle, en augmentant l'intervalle entre les slots dédiés à la communication asynchrone de chacun des nœuds du réseau TSCH, et en les échelonnant dans le temps.

Si, en revanche, un très petit nombre de nœuds asynchrones communiquent à destination d'un grand nombre N_T de nœuds synchrones, ces N_T nœuds n'ont pas spécialement d'intérêt à mutualiser leurs efforts. Dans ce cas, l'exploitation du mode RIT apparaît comme plus pertinente, car moins perturbatrice : rappelons que dans le mode CSL, le nœud asynchrone transmet des trames de préambules perturbant ainsi un canal physique pendant la durée totale de transmission des préambules. Au contraire, le nœud asynchrone RIT est simplement en mode Rx jusqu'à réception d'une trame de sollicitation, ce qui ne vient pas interférer avec les communications du réseau TSCH pendant la durée entre la génération de données asynchrones et la réception de la trame de sollicitation.

Dans la suite de nos travaux, nous choisissons de travailler avec le mode RIT, moins perturbateur et plus avantageux dans le contexte d'un petit nombre de nœuds contraints à proximité d'un réseau synchrone TSCH. On peut également citer l'étude de Fujiwara *et al.* [123], qui montre que le mode RIT consomme moins d'énergie que le mode CSL.

En utilisant de plus le couplage des informations RIT dans les trames d'EB, l'impact de notre proposition d'interopérabilité sur le fonctionnement du réseau TSCH est très faible, et ne représente pas de difficulté d'implémentation, puisqu'il correspond quasiment au mode de fonctionnement classique d'un réseau TSCH.

Il est cependant nécessaire de soigneusement planifier l'envoi des informations de sollicitation RIT (incluse dans les EB) de manière à garantir des performances en termes de la latence et de consommation d'énergie, à la manière de la planification de l'envoi des trames d'EB au sein d'un réseau TSCH présenté au chapitre précédent.

5.2 Ordonnancement des slots d'interopérabilité

5.2.1 Principe de l'ordonnancement

Afin de garantir l'acheminement du trafic asynchrone dans une certaine latence L_{obj} , il est nécessaire de soigneusement planifier les slots d'envoi des informations RIT au sein des ordonnancements TSCH.

Nous proposons d'exploiter l'ordonnanceur présenté au chapitre précédent pour allouer les slots d'EB contenant les informations nécessaires au fonctionnement asynchrone RIT (si besoin, le principe de fonctionnement de l'ordonnanceur exploité par *DataJoin* est décrit à la page 113).

À la manière de *DataJoin* et afin de garantir une équirépartition des transmissions de trames d'EB sur un même canal, nous conservons l'utilisation de N_z zones d'*advertisement*, des zones temporelles dédiées à la transmission d'EB. Nous conservons également la transmission d'une trame d'EB par canal physique et par *slotframe*, ce qui implique que $N_z = N_c$.

Cependant, nous modifions le fonctionnement de l'ordonnancement *DataJoin* concernant les cinq points suivants :

- Au lieu d'organiser le temps en *multislotframe* (des multiples de la *slotframe*), nous imposons que la taille de la *slotframe* soit un multiple du nombre de canaux utilisés par le réseau, afin d'obtenir que le canal physique correspondant à la cellule (0,0) de l'ordonnancement correspondent toujours au même canal physique (par exemple le canal physique 1 dans le cas de l'ordonnancement présenté sur la figure 5.4). Ainsi, nous imposons $SF_{size} = \kappa \cdot N_c$, où N_c correspond au nombre de canaux physiques utilisés par le réseau (3 sur la figure 5.4), et κ appartient aux entiers naturels, défini par le concepteur réseau en fonction des besoins applicatifs et de la consommation d'énergie.
- On ne réalise plus l'allocation des cellules d'EB des nœuds en fonction de leur rang dans l'arbre de routage, mais on réalise l'ordonnancement au moment où ils apparaissent dans le réseau. De cette manière, notre protocole est autonome et ne suppose pas d'organisation en arbre par un protocole de niveau 3.
- Contrairement à *DataJoin*, n'ayant pas de lien avec les transmissions de données, l'ordonnanceur d'EB qui contiennent les informations liées aux transmissions RIT ne prend pas en compte l'ordonnancement des trames de données pour ses décisions d'allocations. Plus exactement, il prend en compte le fait que les slots soient libres ou non dans le but d'éviter les collisions, mais ne considère pas la nature du slot (données ou *advertisement*).
- Nous supposons que le coordinateur PAN impose, pour l'ensemble du réseau l'ordre des canaux fréquentiels physiques à suivre pour l'envoi des EB au sein de la *slotframe*, dans le but d'éviter des conflits au niveau des canaux fréquentiels physiques sélectionnés aléatoirement, comme nous avons pu en rencontrer dans la présentation de *DataJoin*.
- Nous ajoutons également le critère suivant dans le choix du canal sur lequel un nœud doit émettre ses EB : quand tous les canaux utilisés par le réseau sont déjà couverts par ses voisins, l'extension de la portée spatiale des trames d'EB devient le critère secondaire de sélection des slots d'EB. Pour ce faire, le nœud sélectionne le canal physique sur lequel il doit envoyer ses balises comme celui pour lequel il reçoit l'EB avec le plus faible RSSI (Mesure de la puissance en réception d'un signal reçu — *Received Signal Strength Indication*), parmi les EB de ses voisins. Ainsi, on permet l'extension de la portée spatiale de la zone de réception des trafics asynchrones.
- Afin de garantir l'absence de collisions lors de l'envoi des trames d'EB contenant les informations RIT, y compris dans la situation de nœuds cachés, nous supposons que les nœuds obtiennent non seulement l'ordonnancement TSCH de leurs voisins, mais également celui des voisins de leurs voisins, grâce aux informations des ordonnancements contenues dans les trames EB classiques des voisins.

5.2.2 Exemple d'ordonnancement

La figure 5.4 présente l'élaboration de l'ordonnancement des transmissions des trames de EB au fur et à mesure de la construction du réseau dont la topologie est représentée en haut de la figure.

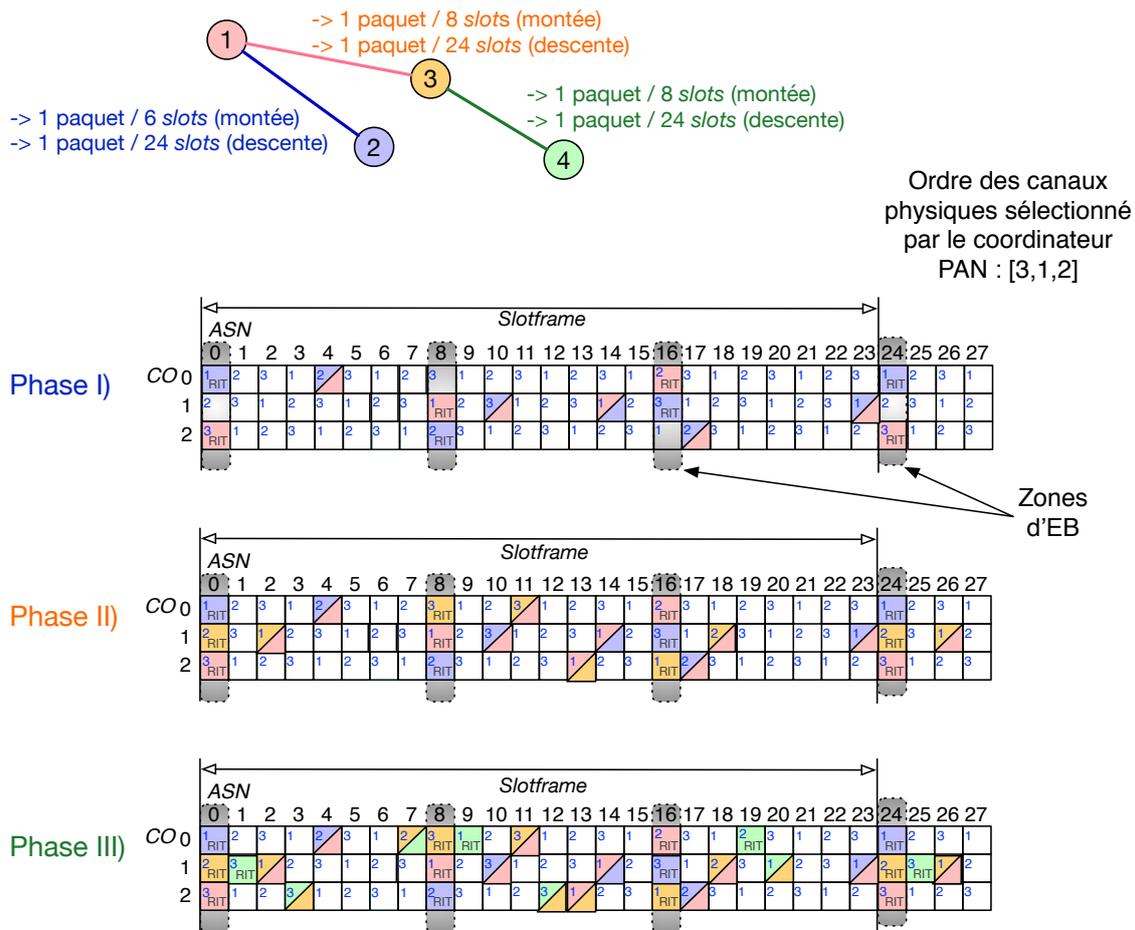


FIGURE 5.4 – Principe de l'allocation des slots d'EB contenant les informations d'interopérabilité.

Sur cette figure, nous avons représenté la *slotframe* de la manière classique pour les réseaux TSCH : en abscisse nous avons représenté le temps, découpés en *timeslot* dont l'ASN est reporté sur la ligne du haut de la *slotframe*. En ordonnée sont représentés les canaux logiques, identifiés par leur décalage en fréquence CO.

Ainsi, chaque case de la *slotframe* représente une cellule TSCH. Les cellules blanches représentent des cellules inactives. Les cellules colorées et dénotées par RIT correspondent aux cellules d'*advertisement* du nœud de la couleur de la cellule. Ainsi la cellule (0,0) correspond à l'envoi d'une trame d'EB par le nœud bleu 2. Les cellules divisées en diagonale et de deux couleurs représentent les cellules de communication de données entre deux nœuds, identifiés par leur couleur : par exemple, la cellule (4,0) permet la communication de données depuis le nœud bleu 2 à destination du nœud rouge 1. Ces cellules de données sont allouées de manière aléatoire dans l'ordonnancement TSCH, le nombre de cellules allouées permettant de couvrir les besoins applicatifs mentionnés sur le schéma présentant la topologie utilisée pour la réalisation de l'exemple, en haut de la figure 5.4.

De plus, le canal physique utilisé par chaque cellule TSCH est indiqué par le chiffre en haut à gauche de chaque case. Ainsi, pour suivre l'activité temporelle sur un canal physique donné, il faut lire la *slotframe* en diagonale.

Enfin, sur la figure, les zones d'envoi des trames d'EB sont grisées sur l'ensemble des canaux fréquentiels, comme on peut le voir à l'ASN 0.

Dans la section suivante, nous proposons d'analyser en détail la construction de l'ordonnancement correspondant à la situation proposée par la représentation topologique du haut de la figure, en découpant la formation d'un tel réseau en plusieurs phases.

Phase 0

Supposons que le nœud rouge 1 soit le coordinateur PAN du réseau, dont l'ordonnancement des trames

d'EB a été choisi de manière à envoyer une trame d'EB dans chaque zone d'EB, sur un canal physique différent dans chaque zone, de manière à respecter la première règle. L'ordre dans lequel les canaux physiques seront sollicités est décidé aléatoirement par le coordinateur PAN (le nœud rouge 1 dans notre exemple, et est reporté en haut à droite de la figure 5.4.

Phase I

Cette phase correspond à l'attache au réseau du nœud bleu 2 : après avoir entendu un EB du coordinateur PAN, le nœud bleu 2 alloue des ressources, non seulement dédiées à sa communication avec le nœud rouge 1, mais aussi pour la signalisation du réseau à un potentiel nœud souhaitant rejoindre le réseau, à travers les slots d'*advertising*. L'allocation des slots pour la transmission des trames de données est réalisée par l'ordonnanceur des données (non étudié ici), et répartit le nombre de slots actifs en fonction du débit applicatif nécessaire. Pour réaliser ses transmissions d'EB, le nœud bleu 2 choisit aléatoirement le canal physique initial sur lequel envoyer son EB dans la première zone d'EB parmi les canaux non encore occupés par les nœuds du voisinage. Pour savoir les canaux déjà utilisés dans le voisinage, le nœud 2 utilise les informations d'ordonnement obtenues dans la trame d'EB du nœud rouge 1 qu'il a reçu pour se synchroniser au réseau. Dans cet exemple, seul le canal 3 est utilisé, et le nœud bleu 2 choisit alors le canal physique 1 aléatoirement parmi les canaux physiques disponibles (2,1).

Il doit ensuite respecter l'ordre des canaux fréquentiels physiques utilisés par le coordinateur PAN pour envoyer ses trames d'EB dans les autres zones. Dans l'exemple de la figure 5.4, le nœud bleu 2 sélectionne initialement le canal physique 1, et suit ensuite la séquence de canaux du coordinateur pour l'envoi des trames de balises, [3,1,2], qu'il suit de manière circulaire : dans la deuxième zone d'EB, il enverra sur le canal fréquentiel physique 2 et dans la troisième, le canal fréquentiel physique 3.

Phase II

Au cours de cette seconde phase, le nœud orange 3 rejoint le réseau grâce aux trames d'EB du nœud rouge 1, qui devient son parent. Ainsi, au niveau de la topologie logique, le nœud orange 3 est directement relié au coordinateur PAN. De ce fait, le nœud orange 3 ajoute dans l'ordonnement des slots de communication de données pour permettre des échanges avec le nœud rouge 1, en fonction du débit applicatif requis par ce lien. Par ailleurs, le nœud orange 3 doit également ajouter dans l'ordonnement ses propres trames d'*advertisement*. Pour ce faire, il procède de la même manière que le nœud bleu 2 : il doit tout d'abord sélectionner aléatoirement parmi les canaux fréquentiels physiques disponibles, celui sur lequel il transmettra son EB dans la première zone d'EB. Étant donné que seul le canal physique 2 est disponible il suit la séquence de canaux suivante : [2,3,1] pour l'envoi des EB dans les zones dédiées.

Phase III

Lorsque le nœud vert 4 entre dans le réseau, il tente de s'ajouter dans l'ordonnement à la manière des nœuds 2 et 3. Cependant, lors de la première étape, alors qu'il doit sélectionner le canal fréquentiel physique sur lequel il doit envoyer sa trame d'EB, et parce qu'il connaît l'ordonnement des voisins via les informations contenues dans le ou les EB reçus, le nœud vert 4 réalise que tous les slots de la zone d'EB sont occupés.

De manière à augmenter spatialement la couverture des EB, il mesure les RSSI des EB de ses voisins, et choisit de transmettre ses EB sur les mêmes canaux fréquentiels que le nœud dont il reçoit les signaux avec le plus faible RSSI (le nœud rouge 1 dans l'exemple).

Cependant, pour éviter les collisions, le nœud vert 4 utilise le slot de temps consécutif à celui de la zone d'EB. Cela conduit, si le réseau devient très dense, à une zone de la *slotframe* de plus d'un slot de large dédiée à la transmission des EB, ce qui rejoint l'idée des travaux de Duy *et al.* [2], présentée au chapitre précédent (cf. page 96), et exploitant des zones de la *slotframe* dédiées à l'*advertisement*.

Par ailleurs, on remarque sur la figure 5.4 que pour la troisième zone d'EB, le nœud vert 4, s'il diffère sa transmission d'EB d'un slot de temps et qu'il respecte la séquence du coordinateur du PAN, devrait envoyer son EB dans la cellule ($CO=2$, $ASN=17$). Cependant, cette cellule est déjà réservée pour le trafic de données de trafic montant du nœud bleu 2 au nœud rouge 1. Dans ce cas, la transmission de l'EB du nœud vert 4 est de nouveau décalée, et ce jusqu'au prochain slot libre sur le canal physique concerné. Dans l'exemple, le nœud vert 4 décale donc sa transmission jusqu'à l'ASN 19.

De cette manière, on construit un ordonnancement qui garantit l'absence de collision, et qui permet également de garantir la transmission d'une trame de sollicitation (couplée à un EB) par nœud, par *slotframe* et par canal fréquentiel physique. On peut donc borner, de manière théorique, les performances de notre proposition d'interopérabilité, ainsi qu'en évaluer les performances moyennes.

5.3 Évaluation des performances théoriques

Dans cette section, nous proposons d'analyser les performances théoriques obtenues par notre schéma d'interopérabilité, à la fois en termes de délai de transmission et de consommation d'énergie, dans le but de valider l'intérêt et les limites d'une telle solution.

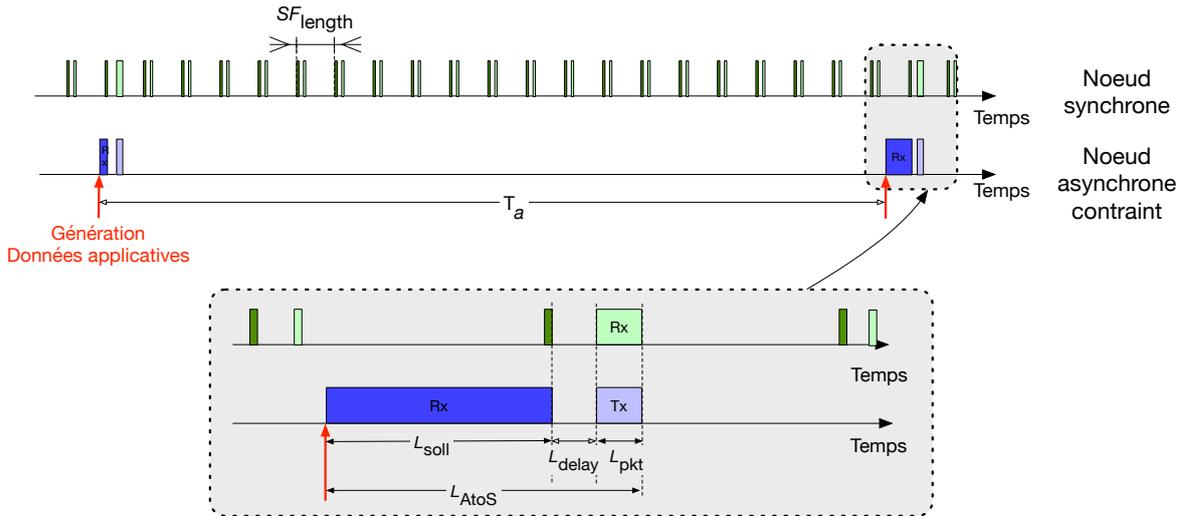


FIGURE 5.5 – Comportement temporel des nœuds du réseau sur un canal fréquentiel donné.

La figure 5.5 présente le comportement temporel d'un nœud du réseau synchrone et d'un nœud asynchrone à portée du nœud synchrone, de manière à visualiser notre proposition de façon globale. Sur cette figure, on retrouve l'envoi périodique de trames de sollicitation RIT au sein des trames d'EB suivi d'une courte écoute pour vérifier la présence ou non de trafic asynchrone, sur la ligne de comportement d'un nœud synchrone TSCH. Cette figure représente le fonctionnement temporel sur un canal donné. Ainsi, comme les nœuds TSCH envoient des trames de sollicitation une fois par *slotframe* et par canal fréquentiel, l'intervalle de transmission des informations RIT, sur un canal, vaut SF_{length} .

Le nœud asynchrone quant à lui transmet des données de manière sporadique, périodique ou non. Pour simplifier le raisonnement, nous supposons que le trafic du nœud asynchrone est généré d'une manière périodique et rare, c'est-à-dire avec une valeur de T_a grande.

Dans notre proposition, nous supposons que le nombre de canaux logiques est égal au nombre de canaux physiques, ce qui se traduit par $N_z = N_c$. De cette manière, on peut garantir que les zones d'EB équiréparties se produisent avec une période T_z telle que

$$T_z = \frac{SF_{\text{length}}}{N_c}, \quad (5.22)$$

où SF_{length} la durée (en ms) de la *slotframe* : $SF_{\text{length}} = SF_{\text{size}} \times ST$ (Durée d'un slot – ou *Slot Time*), avec ST la taille d'un slot, en millisecondes, et SF_{size} la taille de la *slotframe* en nombre de slots.

Métriques utilisées. Comme décrit dans la section 5.1.1.2, l'envoi de la trame de sollicitation est couplé à l'envoi des trames d'EB de manière à minimiser le surcoût lié à l'envoi des trames de sollicitation. De ce fait, évaluer les performances de la communication d'interopérabilité est similaire à l'évaluation des performances des protocoles d'attache au réseau présentée au chapitre précédent.

Ainsi, nous évaluons les performances de notre protocole d'attache vis-à-vis des métriques suivantes, similaires à celles du chapitre précédent :

- L_{AtoS} , la latence subie par le trafic asynchrone à un saut.
- E_{TSCH} , le surcoût d'énergie dépensée par un nœud du réseau synchrone par *slotframe* pour mettre en place les communications asynchrones dans le réseau TSCH, qui dépend majoritairement du nombre de canaux fréquentiels utilisés dans le réseau et de la taille de *slotframe*.
- E_{Thread} , l'énergie dépensée par le nœud asynchrone sur l'intervalle T_a , qui dépend quant à elle uniquement des latences L_{soll} et L_{delay} , si l'on suppose que la donnée transmise est de taille constante.

Comme les intervalles sur lesquels nous mesurons l'énergie dépensée par le nœud asynchrone contraint et par les nœuds TSCH ne sont pas les mêmes (T_a pour le nœud asynchrone et SF_{length} pour les nœuds du réseau synchrone), et donc pas comparables, nous proposons de ramener les deux valeurs de consommation en exprimant la consommation moyenne, par seconde de chacun des nœuds. Les valeurs moyennes par seconde seront notées $\overline{E_{\text{TSCH}}}$ et $\overline{E_{\text{Thread}}}$.

Afin de mesurer l'impact de notre proposition, nous allons mesurer les consommations des nœuds avec et sans la mise en place de notre proposition, dénotée c , pour la solution classique et p pour la prise en charge de notre proposition. Ainsi, dans la dernière partie de cette section, nous tracerons des courbes comparatives de la consommation d'énergie moyenne par seconde des nœuds synchrones, d'un nœud asynchrone, et la consommation d'énergie globale du réseau, avant et après la mise en place de notre solution. Nous tracerons également la courbe de latence du trafic asynchrone L_{AtoS} , permettant de mesurer les performances de notre proposition : quel est le gain d'énergie globale de la mise en place de notre solution, et au prix de quelle latence.

5.3.1 Latence du trafic asynchrone L_{AtoS}

L_{AtoS} représente le temps écoulé entre la génération des données et la transmission effective de ces données au réseau synchrone. Ce délai correspond à la somme du temps mis pour recevoir la trame de sollicitation depuis le réseau, L_{soll} , la durée nécessaire entre les trames de sollicitation et le créneau de disponibilité annoncé par le récepteur, que l'on nommera L_{delay} , et la durée du paquet à transmettre L_{pkt} , tel que $L_{\text{AtoS}} = L_{\text{soll}} + L_{\text{delay}} + L_{\text{pkt}}$.

L_{pkt} dépendant de la quantité d'octets à transmettre, et étant commune pour tous les scénarios étudiés, nous présentons les résultats suivants en supposant une valeur nulle de L_{pkt} .

Ainsi, la latence L_{AtoS} est fonction des deux latences L_{delay} et L_{soll} . La latence L_{delay} ne dépend que de la disponibilité des cellules de l'ordonnancement TSCH du nœud synchrone. Le délai L_{soll} en revanche, dépend, pour une taille de *slotframe* donnée, du nombre de zones d'EB (et donc du nombre de canaux N_c), ainsi que du nombre de nœuds TSCH à portée du nœud asynchrone. Nous montrerons que les valeurs de L_{soll} minimale et maximale sont directes, alors que la détermination de la valeur $L_{\text{soll}|moy}$, plus complexe, constituera le troisième paragraphe de cette section.

Détermination du délai L_{delay} . Un nœud TSCH qui envoie des informations de sollicitation au sein de ses EB doit inclure dans ces informations un des slots disponibles de l'ordonnancement, dans lequel il se tiendra prêt pour recevoir une potentielle transmission d'un nœud asynchrone voisin.

Un tel slot est alloué dynamiquement par l'ordonnanceur du nœud TSCH, en fonction des créneaux disponibles au moment de l'émission de la trame d'EB (cf. page 126). Ainsi, en fonction de la densité des ressources utilisées au sein de l'ordonnancement TSCH, ce slot peut se situer au début ou la fin de la *slotframe*.

Dans le pire cas, l'ordonnanceur ne trouve un slot disponible qu'à la fin de la *slotframe*, et on a $L_{\text{delay}|worse} = SF_{\text{length}} - ST$. Dans le meilleur des cas, le slot consécutif à l'envoi de la trame de sollicitation est alloué pour la transmission d'interopérabilité, et on a $L_{\text{delay}|best} = ST$, avec ST la durée d'un slot temporel.

Enfin, la valeur moyenne de latence entre la transmission de l'EB contenant les informations RIT et le slot d'interopérabilité vaut

$$L_{\text{delay}|moy} = \frac{SF_{\text{length}}}{2}. \quad (5.23)$$

Cette valeur correspond à un choix aléatoire du slot d'interopérabilité parmi les slots libres de la *slotframe*.

Il est par ailleurs possible d'envisager d'allouer le slot le plus proche de l'envoi d'EB, pour optimiser la latence du trafic asynchrone, et se rapprocher du cas optimal présenté ci-dessus, où le slot dédié à l'envoi de la trame d'interopérabilité est celui consécutif à l'envoi de la trame de sollicitation.

5.3.2 Détails de la latence de réception de la trame de sollicitation L_{soll}

5.3.2.1 Valeurs minimale et maximale de L_{soll}

Dans le pire cas, un nœud asynchrone n'est à portée que d'un seul nœud du cœur de réseau synchrone, et se réveille sur le canal sur laquelle ce voisin vient d'envoyer sa trame de sollicitation, un court instant (ε) après le début de la transmission de la trame d'EB contenant les informations RIT.

Dans ce cas, le nœud devra attendre la prochaine itération de la *slotframe* pour entendre de nouveau la trame d'EB sur le canal qu'il a sélectionné. Ainsi, dans le pire cas L_{soll} vaut :

$$L_{\text{soll}|worse} = SF_{\text{length}} - \varepsilon. \quad (5.24)$$

Dans le meilleur des cas, le nœud asynchrone se réveille un court instant (ε') avant le début de l'envoi de la trame d'EB, et sur le canal physique utilisé par le nœud à sa portée qui envoie la trame de sollicitation. Dans ce cas, le nœud asynchrone la reçoit donc dès son réveil, et L_{soll} vaut :

$$L_{\text{soll}|best} = \varepsilon'. \quad (5.25)$$

Ainsi, nous avons une latence totale pour le trafic asynchrone bornée par la somme de L_{delay} et L_{soll} , dans le pire et dans le meilleur cas :

$$L_{\text{delay}|best} + L_{\text{soll}|best} \leq L_{\text{AtoS}} \leq L_{\text{delay}|worse} + L_{\text{soll}|worse} \quad (5.26)$$

$$ST \leq L_{\text{AtoS}} \leq 2 \times SF_{\text{length}}. \quad (5.27)$$

Le cas moyen du délai L_{soll} dépend fortement du nombre de voisins TSCH à portée du nœud asynchrone, N_{neigh} , ainsi que des choix d'ordonnancement de ces voisins.

Dans le paragraphe suivant, nous proposons d'établir la valeur de latence moyenne que le trafic asynchrone pourra éprouver, qui dépend directement du nombre de voisins N_{neigh} , mais aussi de l'intervalle entre les zones d'EB, T_z , et du nombre de canaux physiques utilisés par le réseau N_c .

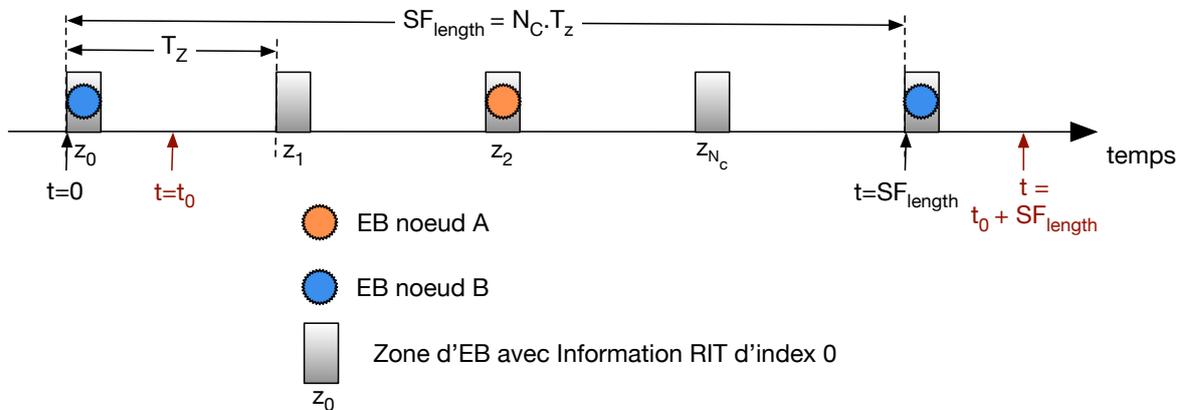


FIGURE 5.6 – Représentation temporelle du point de vue du nœud asynchrone (sur un canal physique donné).

La figure 5.6 présente le point de vue d'un nœud asynchrone, c'est-à-dire un point de vue monocanal, d'un nœud qui se réveille à la date t_0 . Sur la figure, nous avons représenté les envois d'EB de deux nœuds A et B du réseau synchrone, répartis dans le temps de la manière présentée dans la section relative à l'ordonnancement.

5.3.2.2 $L_{\text{soll}|moy}$: Cas particuliers

Cas $N_{\text{neigh}} = 1$

Si l'on suppose que le nœud asynchrone ne peut entendre que le nœud orange A, on peut écrire le délai de réception d'une trame de sollicitation $L_{\text{soll}} = n_1 T_z - t_0$, où n_1 est l'indice de la zone d'EB utilisée par le seul voisin à portée pour envoyer son EB (ici, $n_1 = 2$, car c'est la zone z_2 qui est utilisée), et t_0 mesuré par rapport au début de la *slotframe* courante. Ainsi, dans le cas de la figure 5.6, on a $L_{\text{soll}} = 2T_z - t_0$.

En moyennant sur les différents instants t_0 de réveil du nœud asynchrone par rapport à l'avancement dans la *slotframe*, on obtient :

$$L_{\text{soll|moy}} = \frac{1}{SF_{\text{length}}} \left[\int_0^{n_1 T_z} (n_1 T_z - t) \cdot dt + \int_{n_1 T_z}^{SF_{\text{length}}} (SF_{\text{length}} - t + n_1 T_z) \cdot dt \right]. \quad (5.28)$$

Cas $N_{\text{neigh}} = 2$

Supposons maintenant que le nœud asynchrone puisse entendre à la fois le nœud orange A et le nœud bleu B.

Sur l'exemple de la figure 5.6, le nœud bleu B envoie ses trames d'EB au premier indice de zone et le nœud orange A sélectionne la zone d'indice 2. Rappelons ici que ce choix est aléatoire, comme nous l'avons expliqué au paragraphe concernant l'ordonnancement : le choix du canal physique sur lequel est transmise la trame d'EB de la première zone est réalisé de manière aléatoire parmi les canaux non utilisés par les voisins dans cette zone et, pour les autres zones, il faut suivre la séquence de canaux fréquentiels imposée par le coordinateur.

Ainsi, d'après la définition du coefficient binomial, si on a N_z zones d'EB, et qu'on entend 2 voisins, il existe $\binom{2}{N_z}$ combinaisons d'ordonnancement des EB parmi les N_z zones disponibles, et la probabilité que la distribution des EB corresponde à celle présentée sur la figure 5.6 vaut $\frac{1}{\binom{2}{N_z}}$.

Dans les cas où ce sont ces zones z_0 et z_2 d'EB occupées, le délai moyen $L_{\text{soll|moy}}$ vaut :

$$L_{\text{soll|moy}}(n_1, n_2) = \frac{1}{SF_{\text{length}}} \times \sum_{i=0}^1 \left[\int_{n_i T_z}^{n_{i+1} T_z} (n_{i+1} T_z - t) dt \right] + \int_{n_2 T_z}^{SF_{\text{length}}} (SF_{\text{length}} - t + n_1 T_z) dt, \quad (5.29)$$

en utilisant $n_0 = 0$, et n_x l'indice de la zone de la x -ième trame de sollicitation envoyée sur le canal. Nous utilisons la notation n_x pour souligner le fait qu'une trame provienne du voisin A ou du voisin B n'influe pas sur la valeur de L_{soll} : seul l'ordre d'apparition des ces trames est important. En effet, il est important de savoir que la 2^{ème} zone est occupée, mais pas de savoir si c'est l'EB provenant du nœud A ou du nœud B qui l'occupe.

5.3.2.3 $L_{\text{soll|moy}}$: Généralisation

Cas $N_{\text{neigh}} < N_z$

On peut alors généraliser au cas où le nœud asynchrone est à portée de N_{neigh} voisins de la manière suivante.

Soit φ le n -uplet représentant une distribution des zones d'EB exploitées par le voisinage en portée radio tel que $\varphi = (n_0, n_1, \dots, n_{N_{\text{neigh}}})$. Dans l'exemple de la figure 5.6, on a représenté le cas $\varphi = (0, 2)$.

Soit N_φ le nombre de combinaisons différentes de valeurs de φ , tel que : $N_\varphi = \binom{N_z}{N_{\text{neigh}}}$.

On peut alors écrire la latence moyenne de réception d'une trame de sollicitation lorsque le nœud asynchrone est à portée de N_{neigh} voisins :

$$L_{\text{soll|moy}}(N_{\text{neigh}}) = \frac{1}{N_\varphi} \times \sum_{\varphi=\varphi_0}^{\varphi_{N_\varphi}} L_{\text{soll}}(\varphi) \text{ avec} \quad (5.30)$$

$$L_{\text{soll}}(\varphi) = \frac{1}{SF_{\text{length}}} \times \left[\sum_{i=0}^{N_{\text{neigh}}-1} \int_{n_i T_z}^{n_{i+1} T_z} (n_{i+1} T_z - t) dt + \int_{n_{N_{\text{neigh}}} T_z}^{SF_{\text{length}}} (SF_{\text{length}} - t + n_1 T_z) dt \right]. \quad (5.31)$$

Cas $N_{\text{neigh}} \geq N_z$

Supposons maintenant qu'un nœud entende en moyenne au moins autant de voisins que le nombre de canaux utilisés par le réseau, c'est dire que $N_{\text{neigh}} \geq N_z$.

Si le nombre de voisins à portée du nœud asynchrone est égal au nombre de canaux physiques utilisés dans le réseau ($N_{\text{neigh}} = N_c$), toutes les zones d'EB au sein d'une *slotframe* seront remplies sur l'ensemble des canaux fréquentiels utilisés, du point de vue du nœud asynchrone ultra- contraint, ce qui implique que ce dernier entende au minimum une trame d'EB contenant des informations RIT par intervalle T_z .

Si $N_{\text{neigh}} > N_c$, alors nous avons vu que les décisions d'ordonnancement consistent à décaler l'envoi d'EB d'un slot temporel, et utiliser la dimension fréquentielle de manière à étendre la portée spatiale de la

zone de gestion des trafics asynchrones (cf. dernière règle d'ordonnancement, page 128). Nous supposons que le gain, en termes de latence, lié à l'ajout de ce slot est négligeable, puisqu'il est accolé à un slot également dédié à l'envoi d'informations RIT. Ainsi, dans la suite, nous supposons que l'intervalle entre les messages de sollicitation reçus par le nœud asynchrone vaut T_z pour toutes les valeurs de $N_{\text{neigh}} \geq N_c$, ce qui implique un délai moyen de :

$$L_{\text{soll|moy}} = \frac{T_z}{2}. \quad (5.32)$$

5.3.2.4 Nombre moyen de voisins à portée radio

Le délai moyen d'acheminement d'un trafic asynchrone dépend fortement du nombre de voisins N_{neigh} à portée radio du nœud asynchrone (cf. équations 5.31 et 5.30) .

Dans ce paragraphe, nous nous proposons de déterminer, pour une surface carrée de coté ℓ donnée, l'espérance du nombre de voisins d'un nœud asynchrone, en supposant une portée radio R en mètre, et un nombre N_T de nœuds constituant le réseau synchrone TSCH.

Nous avons choisi de modéliser un déploiement aléatoire de N_T nœuds répartis dans une aire de jeu carrée, car c'est de cette manière que le simulateur 6TiSCH [120], utilisé dans les travaux précédents, simule le fonctionnement du réseau.

La probabilité qu'un des N_T nœuds soit à portée du nœud asynchrone aléatoirement positionné dans l'aire de jeu correspond à la probabilité que ce nœud, placé aléatoirement au sein de l'aire de jeu, appartienne aussi à la surface de portée radio du nœud asynchrone, c'est-à-dire la probabilité qu'un nœud appartenant à l'aire de jeu carrée de coté ℓ appartienne également à la partie incluse dans l'aire de jeu d'un cercle de rayon R , centré sur le nœud asynchrone.

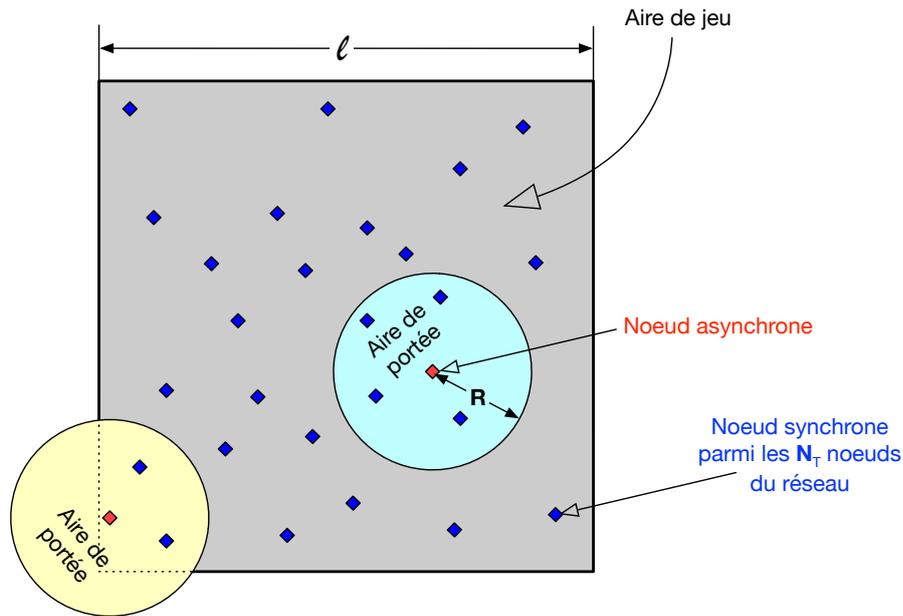


FIGURE 5.7 – Schéma présentant le calcul de l'espérance du nombre de voisins.

Le schéma 5.7 présente cette problématique ainsi que les noms des variables utilisées dans la suite.

Si nous supposons que le nœud asynchrone a pour coordonnées (x_0, y_0) , nous distinguons deux cas :

1. Quand (x_0, y_0) est tel que $\begin{cases} R \leq x_0 \leq (\ell - R) \\ R \leq y_0 \leq (\ell - R) \end{cases}$. Cette situation correspond à celle représentée en bleu sur le schéma 5.7, c'est-à-dire lorsque l'aire de portée est totalement incluse dans l'aire de jeu.
2. Quand $|\ell - x_0| < R$ ou $|\ell - y_0| < R$, c'est à dire quand le nœud asynchrone est proche des bords de l'aire de jeu, et qu'alors la totalité de l'aire de portée n'est pas incluse dans l'aire de jeu.

Dans le second cas, on peut cependant supposer qu'il existe une autre aire de jeu voisine dans laquelle la même quantité de nœuds N_T a été déployée de manière aléatoire, l'aire de jeu n'ayant pas de correspon-

dance physique réelle vis-à-vis du déploiement de réseau. C'est pourquoi nous considérerons pour l'étude que tous les cas sont similaires (on néglige les effets de bords).

En considérant que l'aire de portée est totalement incluse dans l'aire jeu, on peut déterminer la probabilité qu'un nœud du réseau synchrone appartienne à l'aire de portée, parce qu'elle correspond à la situation où un nœud appartenant à l'aire de jeu de surface $A_{\text{jeu}} = \ell^2$ appartiennent également au cercle de portée de surface $A_{\text{portée}} = \pi \times R^2$. Ainsi, la probabilité qu'un nœud de coordonnées (x, y) ($x, y \in [R, \ell - R]$) appartienne au cercle de portée peut s'écrire :

$$p((x, y) \in A_{\text{portée}} | (x, y) \in A_{\text{jeu}}) = \frac{A_{\text{portée}}}{A_{\text{jeu}}} \tag{5.33}$$

$$= \frac{\pi \times R^2}{\ell^2}. \tag{5.34}$$

Ainsi, la variable X_1 qui décrit le fait que le premier nœud tiré aléatoirement soit voisin du nœud asynchrone suit une loi de Bernoulli de paramètre $p = \frac{\pi \times R^2}{\ell^2}$, et il en est de même pour chacun des nœuds N_T .

Comme le positionnement des N_T nœuds est aléatoire, cela correspond à un tirage aléatoire de N_T événements indépendants : on a donc X_1, X_2, \dots, X_{N_T} une collection de N_T variables suivant une loi de Bernoulli de paramètre p , indépendantes et identiquement distribuées. Ainsi, la somme N_{neigh} de ces variables est aussi une variable aléatoire : elle suit une loi binomiale de la forme

$$N_{\text{neigh}} = \sum_{k=1}^{N_T} X_k \sim \mathcal{B}(N_T, p). \tag{5.35}$$

Nous pouvons donc en déduire l'espérance de cette variable qui modélise le nombre de voisin du nœud asynchrone :

$$E(N_{\text{neigh}}) = N_T \times p \tag{5.36}$$

$$= N_T \times \frac{\pi \times R^2}{\ell^2}. \tag{5.37}$$

De cette équation, nous pouvons générer la courbe de l'espérance du nombre de voisins en fonction des paramètres N_T et R .

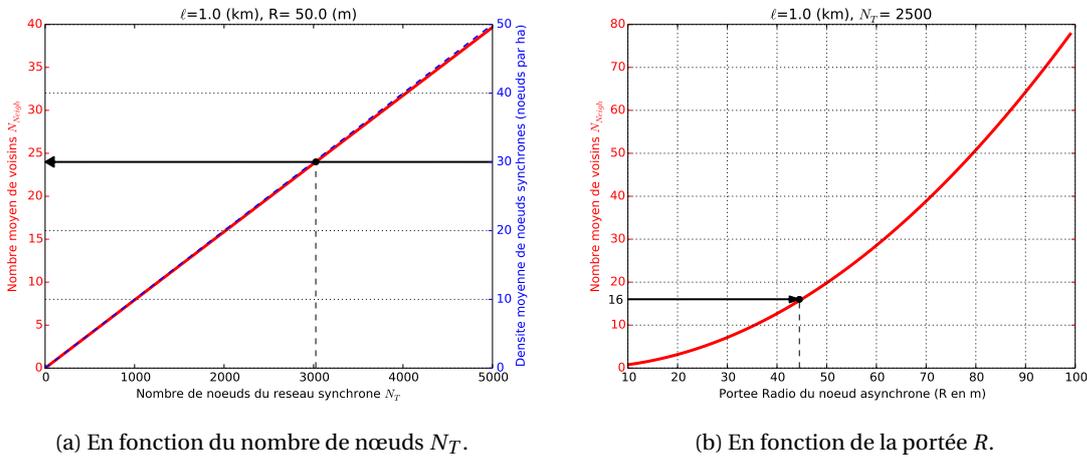


FIGURE 5.8 – Espérance du nombre moyen de voisins.

Sur ces courbes, nous observons que le nombre de voisins moyens est relativement élevé : par exemple, pour un réseau synchrone d'une densité moyenne de 30 nœuds par hectare (30000 nœuds par km^2), la courbe 5.8a montre que l'espérance moyenne de nœuds voisins correspond à une valeur de ~ 24 nœuds, pour une portée radio de 50m. Sur la courbe 5.8b, on peut voir que pour une portée supérieure à 50m, un nœud asynchrone peut espérer avoir plus de 16 voisins pour une densité de nœuds synchrones de 2500 nœuds par km^2 (ce qui correspond à 25 nœuds par hectare).

Ainsi, comme la portée annoncée du TSCH est de l'ordre d'une centaine de mètres et que la densité de 2500 nœuds par km^2 correspond à un déploiement d'un réseau de capteurs peu dense, nous supposons par la suite que le nombre de voisins N_{neigh} sera plus grand que le nombre de canaux physiques existants $N_{c|\text{max}} = 16$.

5.3.2.5 Espérance de la latence moyenne avec $N_{\text{neigh}} \geq N_c$

Nous venons de montrer qu'il est possible de considérer $N_{\text{neigh}} \geq N_c$. On peut alors obtenir le délai moyen d'acheminement d'un trafic asynchrone au sein du réseau synchrone en utilisant les valeurs obtenues aux équations 5.23 et 5.32 :

$$L_{\text{AtoS}}|_{\text{moy}} = L_{\text{soll}}|_{\text{moy}} + L_{\text{delay}}|_{\text{moy}} = \frac{T_z + SF_{\text{length}}}{2}. \quad (5.38)$$

Si l'on reporte dans cette équation $T_z = \frac{SF_{\text{length}}}{N_z}$ (équation 5.22), on obtient :

$$L_{\text{AtoS}}|_{\text{moy}} = SF_{\text{length}} \times \frac{1 + N_c}{2 \cdot N_c}. \quad (5.39)$$

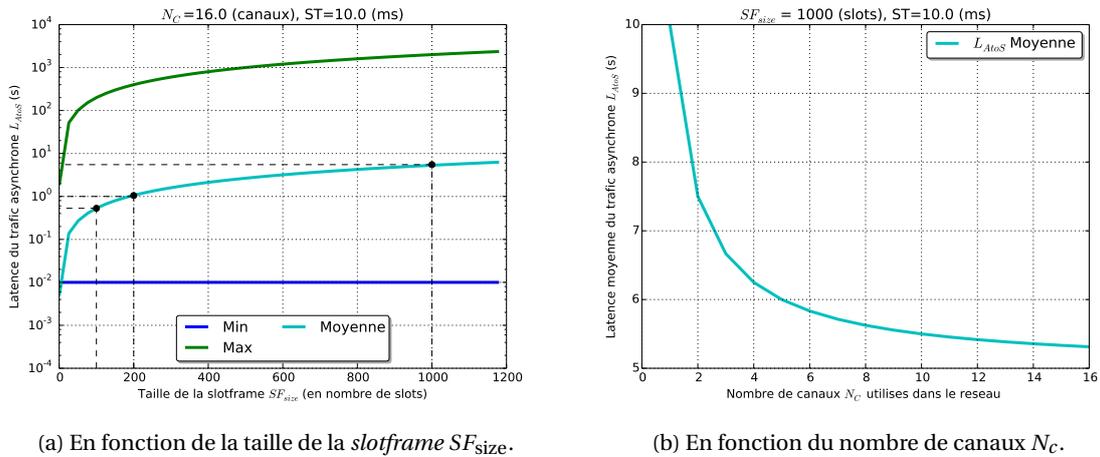


FIGURE 5.9 – Latence moyenne du trafic asynchrone (avec approximation).

La figure 5.9 présente les courbes de la latence subie par le trafic asynchrone, en fonction du nombre de canaux fréquentiels utilisés par le réseau N_c , et la durée de la *slotframe*, SF_{length} . Sur ces courbes, nous pouvons voir que l'ordre de grandeur, en termes de latence, d'acheminement des trafics à un saut dépend fortement, comme nous pouvions l'attendre, du nombre de canaux fréquentiels utilisés dans le réseau, ainsi que de la taille de la *slotframe*.

Si l'on regarde la courbe de latence subie par le trafic asynchrone dans le cas d'une *slotframe* de 1000 slots, on remarque sur la figure 5.9b que la latence subie diminue avec l'augmentation du nombre de canaux utilisés.

Ce résultat s'explique par le fait que, plus l'on augmente le nombre de canaux utilisés pour une taille de *slotframe* donnée, plus on va diminuer l'intervalle T_z (cf. équation 5.22), la distance entre deux transmissions de trame de sollicitation. Ainsi, on comprend que plus les trames de sollicitation sont rapprochées, plus la latence subie par le trafic asynchrone est faible.

Sur la figure 5.9a, on remarque que pour une *slotframe* d'une centaine de slots, on trouve une valeur de latence moyenne L_{AtoS} d'environ 530ms. Si la taille de la *slotframe* augmente, les délais deviennent plus importants : pour une *slotframe* de 1000 slots, la latence moyenne avoisine les 5,5s, une valeur de latence relativement élevée pour les trafics de types asynchrones.

On peut également lire cette courbe dans l'autre sens, en fixant une contrainte de latence, par exemple à 1 s, et en déduisant la taille de *slotframe* qu'il faudrait utiliser, pour un réseau synchrone exploitant 16 canaux. Ainsi, notre proposition permet de garantir une latence de 1 s pour une taille de *slotframe* d'environ 200 slots.

Cependant, cet objectif de latence ne dépend que du concepteur du réseau : en effet, nous avons vu que le fait d'augmenter la taille de la *slotframe* sans pour autant faire varier le nombre de zones d'EB revient à augmenter l'intervalle T_z , et donc augmenter la latence de traitement du trafic asynchrone. Ainsi, il revient au concepteur de faire le compromis entre la latence de traitement des trafics asynchrones et la longueur de la *slotframe* du réseau.

Ce compromis existe déjà par ailleurs dans le réseau purement synchrone : alors que les ratios d'activité des nœuds sont de l'ordre d'un slot sur mille, on ne peut espérer transférer, en moyenne une trame à ce nœud en moins de 500 slots : pour palier cette contrainte de latence, il est en effet nécessaire d'augmenter la disponibilité du récepteur, ce qui revient à augmenter sa consommation d'énergie.

De la même manière, il est possible de réduire la latence subie par le trafic asynchrone en diminuant la taille de la *slotframe*, et donc en diminuant l'intervalle entre les trames de sollicitation T_z , mais cela impactera nécessairement la consommation d'énergie des nœuds émetteurs de telles trames. Ainsi, la section suivante analysera les performances en termes de consommation d'énergie à la fois pour le nœud asynchrone et pour chacun des nœuds du réseau synchrone, engendrée par la prise en charge d'un mode de communication asynchrone au sein du réseau synchrone.

5.3.3 Analyse de la consommation d'énergie

Comme dans les précédents chapitres, nous utilisons un modèle de consommation d'énergie simple, mais réaliste qui repose sur la consommation d'énergie d'un nœud dans différents états matériels. Nous reprenons donc l'équation 1.1 du chapitre 4, où S est un état matériel donné :

$$E(t) = \sum_S P_S \times d_S. \quad (5.40)$$

Nous reprenons également les modes matériels présentés dans le tableau 1.1 (cf. page 27) ainsi que les valeurs utilisées par la plateforme de consommation minimale que nous avons présentée page 79 : $P_{Tx} = 24.11 \text{ mW}$, $P_{Rx} = 19.26 \text{ mW}$, $P_{Idle} = 4.67 \text{ mW}$, et $P_{Sleep} = 3.24 \mu\text{W}$.

À partir de ces valeurs, nous pouvons donc déduire, à la fois l'énergie moyenne consommée par les nœuds synchrones TSCH et le nœud Thread asynchrone pour la prise en charge du trafic asynchrone contraint en latence par le réseau synchrone existant.

La figure 5.10 présente le fonctionnement détaillé des différents acteurs des réseaux avec et sans notre proposition. Par simplicité, seule l'activité liée à notre proposition est représentée : on ne montre pas ici les communications de données, ou de commandes nécessaires au fonctionnement classique du réseau.

Sur cette figure, nous avons représenté, dans chacun des cas, le comportement temporel d'un des nœuds TSCH, c'est-à-dire la projection de son activité dans le temps, tous canaux fréquentiels compris. La première ligne des deux figures 5.10a et 5.10b représente cette activité, la couleur utilisée en arrière-plan correspondant au canal fréquentiel sur lequel le nœud est actif au cours du temps.

La troisième ligne représente le comportement du nœud asynchrone contraint au cours du temps sur le canal de communication de son choix (sur la figure, nous avons représenté ce canal de couleur grise). Comme le comportement du nœud asynchrone dépend de l'activité du réseau synchrone TSCH et du nœud asynchrone constamment disponible à portée du nœud asynchrone contraint, nous avons représenté sur la ligne intermédiaire des deux figures 5.10a et 5.10b la projection temporelle de l'activité sur le canal sélectionné.

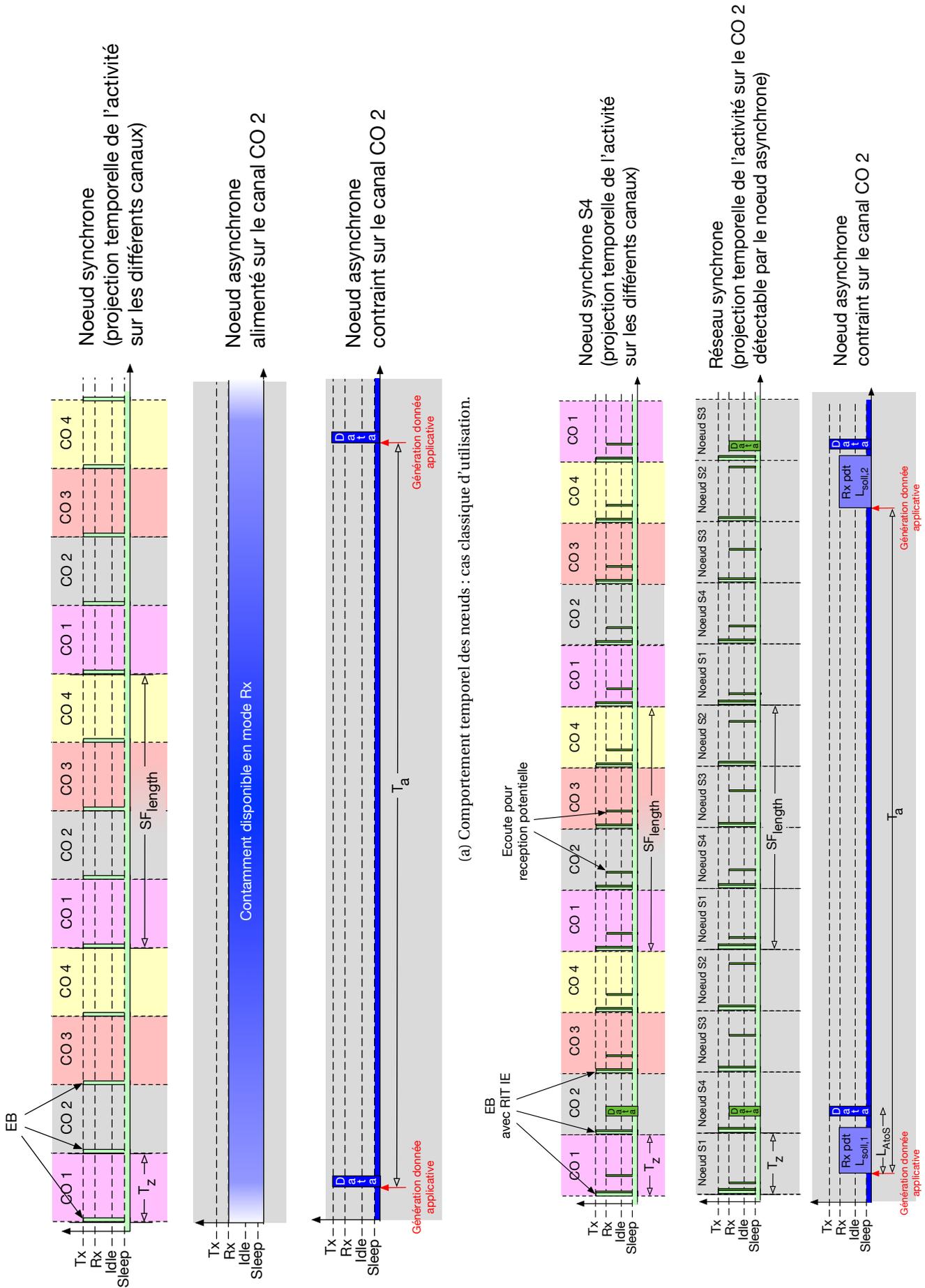
Sur la figure, nous avons également reporté le mode d'activité matériel des nœuds au cours du temps (*Rx*, *Tx*, *Idle* et *Sleep*) de manière à présenter les détails de la consommation d'énergie calculée à l'aide de l'équation 5.40.

Enfin, pour générer la figure, nous avons supposé un réseau constitué d'une dizaine de nœuds synchrones, donc quatre sont à portée radio du nœud asynchrone contraint. Nous avons également supposé un fonctionnement de réseau synchrone avec $N_c = 4$ canaux fréquentiels, ce qui explique le découpage de la *slotframe* en quatre zones de longueur $T_z = \frac{SF_{\text{length}}}{N_c}$.

5.3.3.1 Point de vue du nœud asynchrone contraint

Mode classique. Comme le montre la figure 5.10a, dans le cas d'une communication asynchrone classique, le nœud contraint consomme en moyenne, sur la durée de l'intervalle T_a :

$$E_{\text{Thread}}(c) = d_{\text{data}} \times P_{Tx} + (T_a - d_{\text{data}}) \times P_{\text{Sleep}}. \quad (5.41)$$



(b) Comportement temporel des nœuds : notre proposition.

FIGURE 5.10 – Description détaillée du comportement temporel des différents acteurs avant et après la prise en charge d'un trafic asynchrone au sein d'un réseau synchrone.

Mode d'insertion dans un réseau synchrone. La figure 5.10b résume les modes d'opération du nœud asynchrone à la suite d'une génération de données sont les suivants : une première période dans le mode *Rx* jusqu'à réception d'une trame de sollicitation, suivie d'une courte période d'inactivité en mode *Idle* en attendant la disponibilité du récepteur, et enfin la transmission de la trame en mode *Tx* pendant la durée de la trame.

Si l'on reprend les notations de la figure 5.5, on peut donc écrire la consommation d'énergie par intervalle de temps T_a du nœud asynchrone de la manière suivante :

$$E_{\text{Thread}}(p) = L_{\text{soll}} \times P_{\text{Rx}} + L_{\text{delay}} \times P_{\text{Idle}} + d_{\text{data}} \times P_{\text{Tx}} + (T_a - L_{\text{AtoS}}) \times P_{\text{Sleep}}. \quad (5.42)$$

On peut alors utiliser les valeurs de L_{soll} et L_{delay} obtenues au paragraphe précédent pour obtenir les différentes valeurs de latences globales, suivant que l'on considère le pire cas, le cas moyen, ou le meilleur cas.

Ainsi, on obtient, dans le pire cas :

$$E_{\text{Thread|worse}}(p) = SF_{\text{length}} \times P_{\text{Rx}} + SF_{\text{length}} \times P_{\text{Idle}} + d_{\text{data}} \times P_{\text{Tx}} + (T_a - 2 \cdot SF_{\text{length}} - d_{\text{data}}) \times P_{\text{Sleep}}.$$

De la même manière, on peut mesurer la consommation d'énergie minimale dans le meilleur cas, correspondant à : $E_{\text{Thread|best}}(p) = ST \times P_{\text{Rx}} + d_{\text{data}} \times P_{\text{Tx}} + (T_a - ST - d_{\text{data}}) \times P_{\text{Sleep}}$.

Enfin, on peut écrire l'énergie consommée en moyenne par le nœud asynchrone sur l'intervalle T_a de la manière suivante :

$$E_{\text{Thread|moy}}(p) = \frac{SF_{\text{length}}}{2 \cdot N_c} \times P_{\text{Rx}} + \frac{SF_{\text{length}}}{2} \times P_{\text{Idle}} + d_{\text{data}} \times P_{\text{Tx}} + (T_a - \frac{SF_{\text{length}}}{2} (1 + N_c) + d_{\text{data}}) \times P_{\text{Sleep}}. \quad (5.43)$$

Pour une longueur de trame donnée, on trouve la durée de la trame en utilisant le débit de la technologie utilisée. Dans l'interopérabilité proposée, nous utilisons la modulation O-QPSK du 802.15.4, à 250 kb/s, et le format de trame de paquet du standard 802.15.4. Dans cette situation, la transmission d'une trame applicative de 10 octets dure 864 μs . Ainsi, nous utilisons $d_{\text{data}} = 864 \mu\text{s}$.

De plus, nous supposons que le trafic applicatif du nœud asynchrone contraint se produit avec deux périodes différentes, $T_a = 1 \text{ min}$ et $T_a = 1 \text{ jour}$. On peut alors générer les courbes de consommation du nœud asynchrone, en fonction du nombre de canaux utilisés par le réseau N_c , et la taille de la *slotframe*, en distinguant la consommation avec et sans la prise en charge du trafic asynchrone.

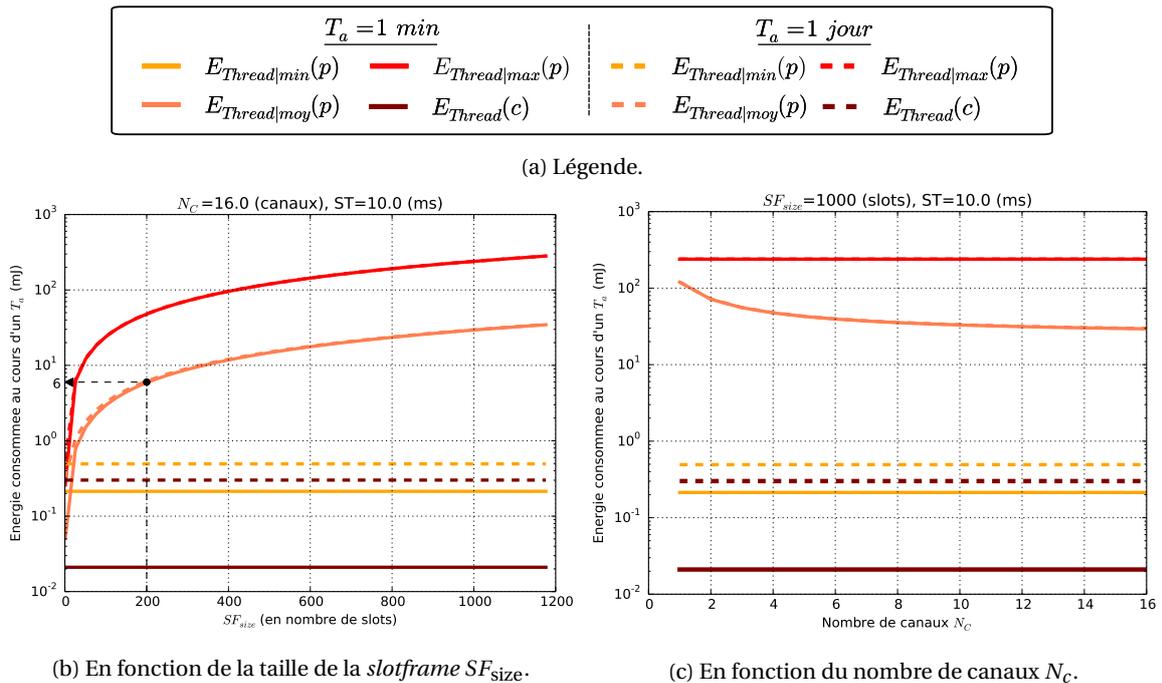


FIGURE 5.11 – Énergie consommée par le nœud asynchrone sur un intervalle T_a , avec et sans la prise en charge du trafic asynchrone (respectivement noté p et c), pour deux valeurs différentes de T_a : $T_a = 1 \text{ min}$ et $T_a = 1 \text{ jour}$.

On remarque sur la figure 5.11c que, comme dans le cas de latence, la valeur d'énergie consommée diminue lorsque l'on augmente le nombre de canaux physiques dans le réseau asynchrone.

On aurait pu s'attendre à ce que le nœud asynchrone consomme plus d'énergie pour trouver les nœuds disponibles sur un plus grand nombre de canaux (comme il ne change pas de canal fréquentiel lors de sa transmission de donnée). Cependant, notre proposition d'interopérabilité détermine le nombre d'EB à envoyer par nœud et par *slotframe* en fonction du nombre de canaux physiques. Cela permet d'exploiter la diversité fréquentielle, non seulement pour atténuer l'impact des conditions physiques sur les transmissions, mais aussi pour diminuer la latence et l'énergie consommée par le nœud asynchrone.

Par ailleurs, on remarque que pour une *slotframe* de 200 slots, ce qui correspond à une latence de $L_{\text{AtoS}} = 1$ s (cf. figure 5.9a), le nœud asynchrone devra consommer une valeur moyenne d'environ 6 mJ par intervalle de T_a , y compris dans le cas d'une valeur de T_a de l'ordre de la journée. Afin de nous donner un ordre d'idée de ce que représente 6 mJ, nous utilisons les valeurs d'énergie récupérées par la plateforme GREENNET. Nous avons vu au cours de notre première contribution (chapitre 3, cf. page 89) qu'une telle plateforme pouvait récupérer de son environnement une puissance de $96 \mu\text{W}$. Ainsi, en une minute, une telle plateforme pourra collecter une énergie de 5,76 mJ, comparable à la dépense d'énergie du nœud asynchrone de 6 mJ par intervalle de temps T_a .

5.3.3.2 Point de vue du réseau synchrone

Du point de vue du réseau, les pires ou meilleurs cas de consommation énergétique sont quasiment identiques. En effet la consommation moyenne des nœuds TSCH n'est que faiblement impactée par la présence ou non de trafic asynchrone : la majorité de la dépense d'énergie provient de l'envoi périodique d'informations RIT et des zones d'écoute pour une réception asynchrone potentielle, et non du temps passé pour la réception du paquet de données en lui-même.

Mode classique. Pour déterminer la consommation d'énergie moyenne d'un des nœud du réseau, nous proposons d'exploiter la figure 5.10a pour décrire le comportement d'un nœud au cours du temps, et en déduire sa consommation d'énergie par *slotframe*. Nous définissons d_{EB} , la durée d'une trame d'EB classique de 3 ms. Ainsi, l'énergie dépensée au cours d'une *slotframe* par un nœud du réseau synchrone dans le cas d'une communication classique vaut :

$$E_{\text{TSCH|moy}}(c) = N_c \times d_{\text{EB}} \times P_{\text{Tx}} + (SF_{\text{length}} - N_c \times d_{\text{EB}}) \times P_{\text{Sleep}}. \quad (5.44)$$

En prenant en charge la communication asynchrone. Si l'on prend en charge la gestion du trafic asynchrone, l'énergie consommée par un des nœuds TSCH au cours d'une *slotframe* devient :

$$E_{\text{TSCH|moy}}(p) = N_c \times [d_{\text{EBwRIT}} \times P_{\text{Tx}} + d_{\text{tout}} \times P_{\text{Rx}}] + [SF_{\text{length}} - N_c \times (d_{\text{EBwRIT}} + d_{\text{tout}})] \times P_{\text{Sleep}}, \quad (5.45)$$

avec d_{EBwRIT} la durée de transmission d'une trame d'EB contenant des informations RIT, et d_{tout} la durée pendant laquelle le nœud du réseau synchrone écoute une potentielle réponse d'un nœud asynchrone.

Cependant, cette équation n'est pas représentative, car elle décrit la consommation des nœuds du réseau, comprenant l'envoi de la trame d'EB, nécessaire au fonctionnement classique du réseau synchrone TSCH.

C'est pourquoi il nous semble intéressant de présenter le surcoût énergétique purement lié à la communication d'interopérabilité au cours d'une *slotframe*, qui peut s'écrire de la manière suivante :

$$\Delta E_{\text{TSCH|moy}}(p) = N_c \times (d_{\text{RIT IE}} \times P_{\text{Tx}} + d_{\text{tout}} \times P_{\text{Rx}}). \quad (5.46)$$

Dans cette équation, nous pouvons remplacer les durées $d_{\text{RIT IE}}$ et d_{tout} par les valeurs classiques du standard TSCH.

Ainsi, la durée d'envoi des informations RIT au sein de la trame d'EB est directement calculée en fonction de la taille, en octet, de l'élément d'information RIT IE et du débit de la modulation TSCH :

$$d_{\text{RIT IE}} = \frac{4 \text{ octets} * 8}{250 \text{ kb/s} * 10^3} \approx 128 \mu\text{s}.$$

d_{tout} correspond au temps passé en mode Rx par un nœud du réseau TSCH dans l'attente d'une potentielle transmission d'un nœud asynchrone. Cette durée peut être définie suivant les besoins de consommation énergétique. Dans le but de rester au plus proche du standard 802.15.4, nous proposons d'utiliser la valeur classiquement utilisée pour les slots dédiés en mode Rx , $tsRxWait$ du standard. Ainsi, nous utilisons la valeur $d_{\text{tout}} = tsRxWait = 2200 \mu\text{s}$.

Ainsi, en utilisant ces valeurs dans l'équation 5.46, pour l'utilisation de $N_c = 16$ canaux fréquentiels, notre proposition implique une surconsommation par $slotframe$ d'environ $678 \mu\text{J}$.

Toujours dans le but de nous donner un ordre d'idée de ce que représente cette consommation d'énergie, nous utilisons la valeur d'énergie récupérée par la plateforme GREENNET présentée au chapitre 3 (cf. page 89) : une puissance de $96 \mu\text{W}$ pour un système situé en intérieur, à environ 2 m d'une source lumineuse. Ainsi, le système à récupération d'énergie de la plateforme GREENNET permet de collecter en environ 7 s le surplus d'énergie nécessaire par $slotframe$ à la mise en place de notre solution.

On remarque que la surconsommation énergétique par $slotframe$ liée à notre proposition ne dépend que du nombre de canaux utilisés dans le réseau. Cela s'explique par le fait que le nombre de RIT IE envoyés au cours d'une $slotframe$ par notre proposition se calcule uniquement en fonction du nombre de canaux disponibles ($N_z = N_c$) : en effet, quelle que soit la longueur de la $slotframe$, notre proposition ajoute uniquement N_c informations RIT par $slotframe$, donc la surconsommation d'énergie par $slotframe$ reste identique.

Si les consommations par $slotframe$ sont constantes, les consommations moyennes globales, c'est-à-dire les énergies moyennes dépensées par seconde, dépendent du nombre d'itérations de la $slotframe$ au cours d'une seconde, et donc de la taille de cette dernière.

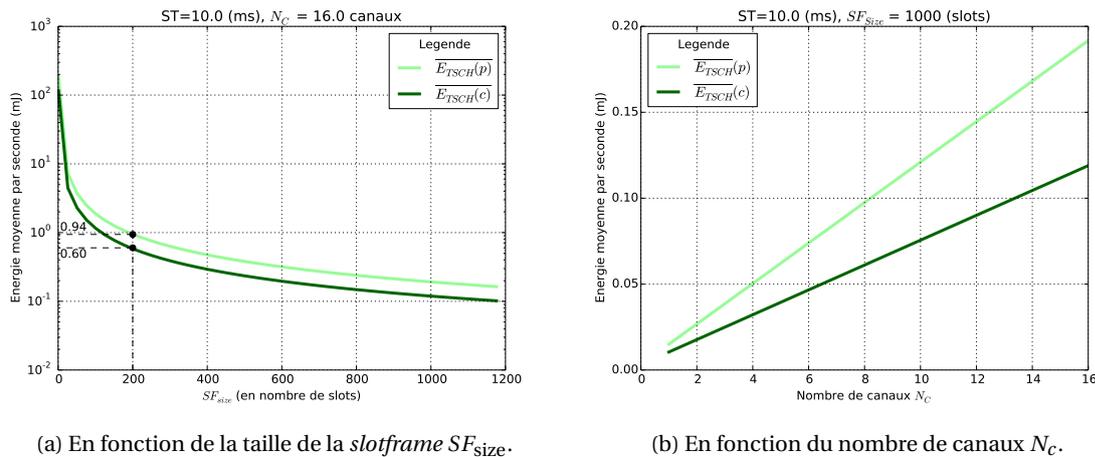


FIGURE 5.12 – Énergie moyenne consommée par seconde par un nœud du réseau synchrone avec ou sans la prise en charge du trafic asynchrone à proximité.

Ainsi, nous présentons sur la figure 5.12 les courbes de consommation moyenne, par seconde d'un nœud synchrone du réseau TSCH, avec ou sans la prise en charge du trafic asynchrone à proximité ($\overline{E_{\text{TSCH}}(c)}$ et $\overline{E_{\text{TSCH}}(p)}$). Sur ces courbes, l'écart entre les deux courbes représente donc la surconsommation liée à notre proposition.

Grâce à ces courbes, on peut vérifier la surconsommation moyenne par seconde requise par un nœud synchrone du réseau TSCH pour garantir une prise en charge du trafic asynchrone avec une latence de 1 s. En effet, si l'on se reporte à la figure 5.9a, on remarque que cela correspond à une $slotframe$ constituée de 200 slots. La figure 5.12a permet ensuite de mesurer que pour une consommation sans prise en charge du trafic asynchrone d'environ 0.6 mJ par seconde. Pour la prise en charge d'un trafic asynchrone avec une latence garantie de 1 s, notre proposition implique une consommation moyenne d'un nœud asynchrone de 0,94 mJ par seconde. Ainsi, la surconsommation moyenne de notre proposition représente une dépense d'énergie de 0,34 mJ par seconde. Si l'on multiplie cette valeur par la durée de la $slotframe$, on retombe bien sur la valeur de surconsommation déduite de l'équation 5.46 en utilisant $N_c = 16$ canaux fréquentiels de $\sim 0,68 \text{ mJ}$.

De plus, comme attendu, la figure 5.12b traduit le lien proportionnel direct entre le nombre de canaux sur lesquels les nœuds TSCH devront envoyer les trames de sollicitation et la surconsommation énergétique liée à notre proposition.

5.3.3.3 Discussion

Dans cette section, nous plaçons les valeurs numériques de ces courbes dans le contexte global. Comme lors de la phase d'attache au réseau, on retrouve le compromis présenté par les trois dimensions suivantes :

- Consommation d'un nœud TSCH du réseau synchrone, à multiplier par le nombre de nœuds N_T si l'on souhaite mesurer la consommation globale du réseau.
- Consommation du nœud asynchrone contraint.
- Latence du trafic asynchrone L_{AtoS} .

Pour diminuer la consommation d'énergie des nœuds du réseau, il est nécessaire que la taille de la *slotframe* soit importante, de manière à augmenter la longueur de l'intervalle T_a , et ainsi diminuer le ratio d'activité DC_{TSCH} de ces nœuds.

En effet, le DC des nœuds du réseau est défini par leur activité au cours du temps, que l'on peut décrire grâce à la figure 5.10b :

$$DC_{TSCH}(p) = \frac{N_c \times (d_{EBWRIT} + d_{tout})}{SF_{length}}. \quad (5.47)$$

Dans cette équation, il apparaît clairement que le DC des nœuds du réseau TSCH est inversement proportionnel à la durée de la *slotframe*, SF_{length} .

En revanche, l'augmentation de la taille de la *slotframe* va représenter un surcoût en termes de latence, et donc un surcoût énergétique du point de vue asynchrone, comme le montre les figures 5.9a et 5.11b. Il revient donc au designer du réseau de réaliser le compromis entre la latence subie par le trafic asynchrone (et donc l'énergie dépensée par le nœud asynchrone), et la surconsommation induite à chacun des nœuds synchrones TSCH pour prendre en compte ce trafic.

Le compromis consommation/latence. Pour l'aider dans ce choix, nous proposons, de regrouper, sur la même courbe, la latence subie par le trafic asynchrone, la consommation moyenne à la seconde du nœud asynchrone pour acheminer un trafic asynchrone de période $T_a = 1$ min et $T_a = 1$ jour, ainsi que l'énergie moyenne consommée par un des nœuds synchrones du réseau TSCH, en fonction de la taille de la *slotframe*, pour un réseau synchrone constitué de $N_T = 100$ nœuds utilisant $N_c = 16$ canaux fréquentiels.

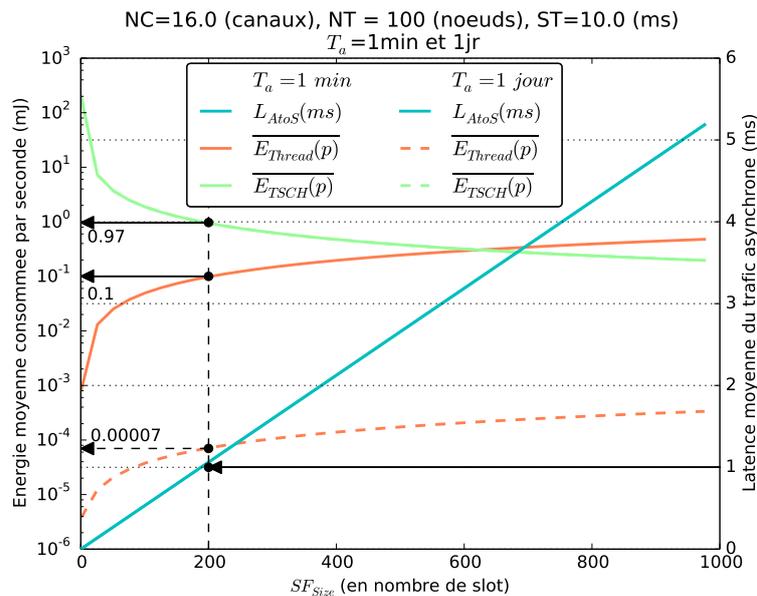


FIGURE 5.13 – Énergie moyenne consommée par seconde par un nœud du réseau synchrone TSCH, par le nœud asynchrone Thread, et la latence subie par le trafic asynchrone en fonction de la taille de la *slotframe*.

La figure 5.13 présente les résultats obtenus. Sur cette figure, on retrouve ainsi les valeurs d'exemple précédemment mentionnées : pour garantir une latence de traitement du trafic asynchrone de l'ordre de la seconde, il faut utiliser une *slotframe* de 200 slots. Pour une telle *slotframe*, la consommation moyenne par seconde d'un nœud synchrone TSCH est de l'ordre du millijoule, alors que la consommation moyenne du nœud Thread, par seconde est quasiment dix fois plus petite (0.1 mJ), dans le cas d'un trafic applicatif asynchrone généré par minute. Dans le cas d'un trafic asynchrone généré par heure, la consommation du nœud asynchrone est encore plus faible puisque ce dernier passe moins de temps dans les phases d'activités pour l'envoi de données.

On peut également confirmer le compromis énoncé plus haut entre la latence du trafic asynchrone, et la consommation d'énergie des nœuds TSCH : moins on contraint le trafic asynchrone en latence, plus on économise de l'énergie au niveau des nœuds synchrones TSCH, au prix d'une augmentation de la consommation du nœud Thread.

Impact global de notre solution. De manière à représenter l'impact global de notre proposition, nous présentons les courbes de consommation globale du réseau avec et sans la prise en charge du trafic asynchrone. Pour ce faire, nous utilisons l'exemple représenté sur les figures 5.10a et 5.10b, pour lequel nous proposons de prendre en compte un réseau constitué de $N_T = 10$ nœuds TSCH. Nous supposons que ce réseau TSCH utilise 4 canaux fréquentiels ($N_c = 4$), et supposons également un nœud asynchrone contraint en portée portée radio de 4 nœuds synchrones du réseau TSCH ($N_{\text{neigh}} = N_c = 4$).

Dans cet exemple, nous souhaitons mesurer l'intérêt de notre proposition, en comparant la consommation globale lors d'une utilisation classique $\overline{E_{tot}(c)}$, et avec celle notre proposition $\overline{E_{tot}(p)}$.

Dans le cas classique. Dans la situation actuellement standardisée, l'énergie globale consommée par seconde par l'ensemble des nœuds de la situation décrite précédemment se décompose en trois pôles :

- l'énergie consommée par seconde par l'ensemble des nœuds du réseau synchrone TSCH,
- l'énergie dépensée par seconde par le nœud asynchrone contraint,
- et l'énergie dépensée par seconde par le nœud asynchrone constamment disponible.

Bien que le nœud constamment disponible soit alimenté en énergie, nous prenons sa consommation d'énergie dans ce calcul pour évaluer le gain de consommation énergétique que nous pouvons réaliser par la mise en place de notre solution. Ainsi, on peut écrire :

$$\overline{E_{tot}(c)} = N_T \times \overline{E_{\text{TSCH}}(c)} + \overline{E_{\text{Thread}}(c)} + \overline{E_{\text{alim}}(c)}. \quad (5.48)$$

En utilisant les équations 5.41 et 5.44, et avec $\overline{E_{\text{alim}}(c)} = P_{\text{Rx}}$, on peut tracer les courbes de consommation de chacun des pôles de dépense énergétique, ainsi que la dépense totale d'énergie engendrée par la prise en charge du trafic asynchrone par un nœud Thread alimenté constamment disponible.

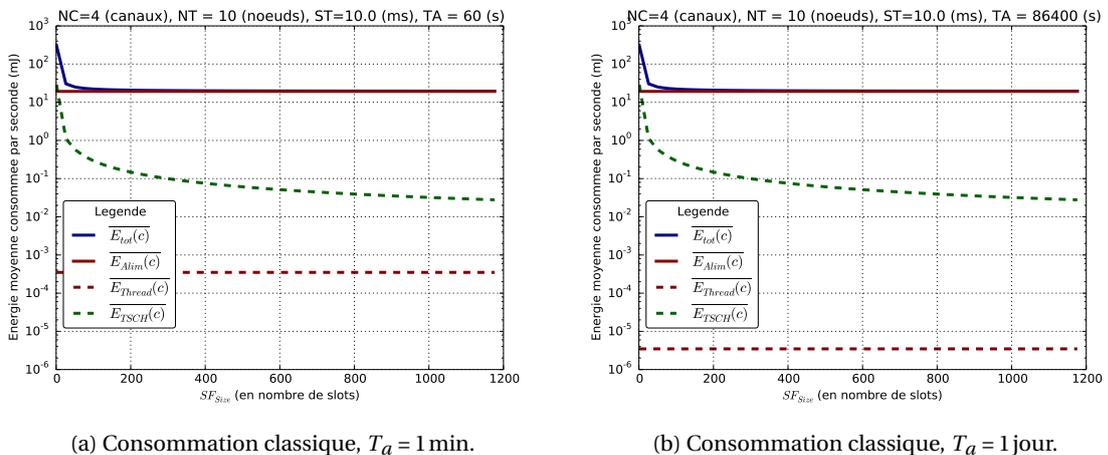


FIGURE 5.14 – Consommation globale dans la proposition classique, et détail de la consommation d'un nœud TSCH, du nœud Thread alimenté, et du nœud Thread contraint.

La figure 5.14 montre clairement que, quelle que soit la période du trafic asynchrone T_a , la majorité de la dépense énergétique globale par seconde, représentée en trait plein bleu sur la figure, provient de la consommation énergétique du nœud Thread constamment disponible.

Avec notre proposition. Nous proposons de remplacer le pôle de consommation énergétique lié à la présence du nœud alimenté Thread par une faible surconsommation appliquée à chacun des nœuds du réseau synchrone TSCH. Ainsi, dans notre proposition, l'énergie globale consommée peut s'écrire :

$$\overline{E_{tot}(p)} = N_T \times \overline{E_{TSCH}(p)} + \overline{E_{Thread}(p)}. \quad (5.49)$$

Dans cette équation, nous pouvons également reporter les valeurs de consommation moyenne des nœuds du réseau TSCH avec la mise en place de notre solution, $\overline{E_{TSCH}(p)}$ (cf. équation 5.45), ainsi que celle du nœud asynchrone contraint avec les modifications de son comportement pour exploiter le réseau TSCH voisin, $\overline{E_{Thread}(p)}$ (cf. équation 5.43), puis tracer la courbe de la consommation moyenne totale, et de consommation de chacun des nœuds sur la figure 5.15.

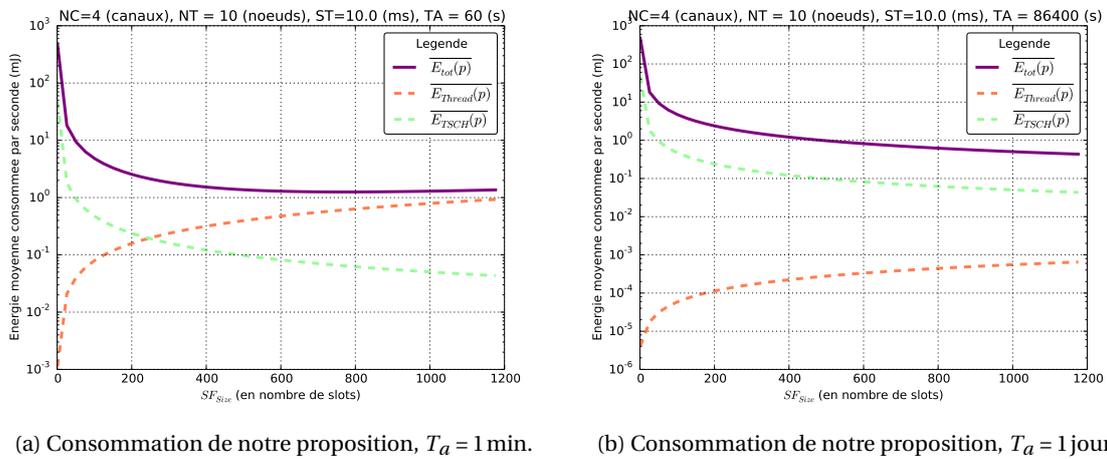


FIGURE 5.15 – Consommation globale de notre proposition, et détail de la consommation d'un nœud TSCH et du nœud Thread.

Sur ces courbes, nous avons représenté la consommation d'un nœud TSCH pour la comparer à celle du nœud Thread, mais, dans le calcul de la consommation d'énergie globale, nous avons considéré le nombre de nœuds N_T du réseau comme le montre l'équation 5.49.

Sur la figure 5.15a, qui considère un trafic applicatif d'une période $T_a = 1$ min, on peut noter que la consommation d'énergie globale, représentée en trait plein, n'est pas majoritairement constituée de l'une ou l'autre des entités du réseau, un nœud TSCH ou le nœud Thread, puisque lorsque l'on considère des petites tailles de *slotframe*, ce sont les nœuds du réseau qui consomment plus que le nœud Thread, alors qu'en augmentant la taille de la *slotframe*, le nœud Thread consomme plus d'énergie que l'un des nœuds TSCH.

La figure 5.15b nous montre que pour un trafic asynchrone rare ($T_a = 1$ jour), la consommation de chacun des nœuds TSCH reste, en moyenne, beaucoup plus élevée que la consommation moyenne d'un nœud asynchrone, quelle que soit la taille de la *slotframe*. Cela s'explique par le fait que l'on considère la consommation moyenne des nœuds par seconde, compensant ainsi les longues périodes d'activités en mode *Rx* du nœud Thread (liées à l'espacement de l'envoi des trames de sollicitations) par une plus longue période passée en mode *Sleep* au cours du T_a .

Ainsi, on peut tracer les courbes d'énergie moyenne consommée avant et après la mise en place de notre solution, si l'on considère le même exemple que celui présenté précédemment sur les figures 5.10b et figures 5.10a ($N_T = 10$, $N_{neigh} = N_c = 4$).

La figure 5.16 présente ainsi les performances globales de notre proposition, pour deux valeurs de période du trafic asynchrone $T_a = 1$ min et $T_a = 1$ jour. Sur ces courbes, on peut voir que notre solution obtient de meilleures performances de consommation énergétique globale que la solution existante, dès lors que la *slotframe* mesure plus de 25 slots, quelle que soit la valeur de T_a considérée. Cependant, notre solution,

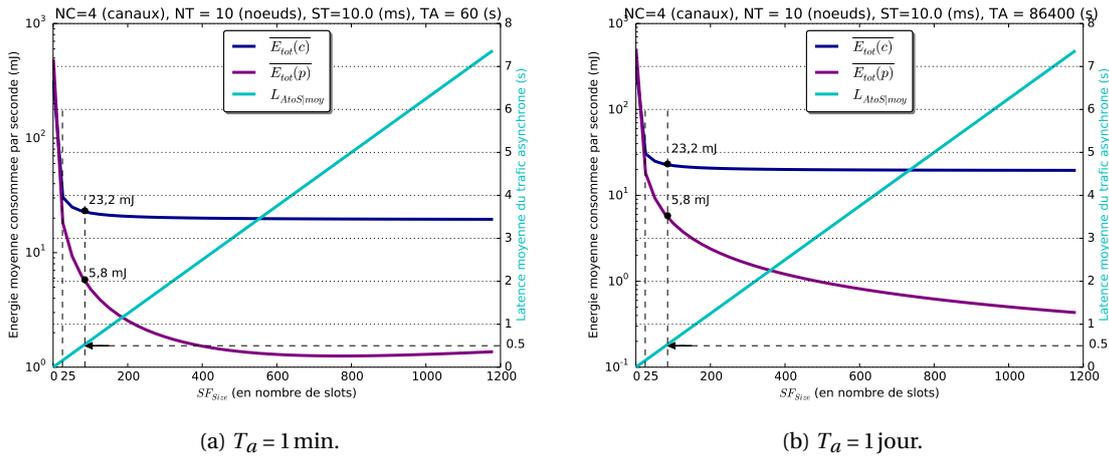


FIGURE 5.16 – Évaluation globale des performances.

si elle permet d'économiser l'énergie globale consommée, se traduit par une augmentation de la latence de prise en charge du trafic asynchrone L_{AtoS} . En effet, la solution classique, même si elle consomme plus, car elle exploite un nœud constamment disponible en mode Rx , garantie une latence L_{AtoS} nulle, puisque le nœud Thread alimenté est disponible dès le réveil du nœud Thread contraint. Notre solution, si l'on relâche légèrement la contrainte de latence, permet de réaliser des économies d'énergie globale non négligeables : sur la courbe 5.16a, on peut par exemple remarquer que pour un trafic applicatif asynchrone apparaissant une fois par minute, notre solution permet de diviser par quatre la consommation globale en garantissant une latence moyenne de moins de 500 ms.

Cette logique est identique dans le cas d'un trafic applicatif asynchrone plus rare, même si notre proposition consomme encore moins d'énergie lorsque l'on considère la génération d'un trafic asynchrone par jour : contrairement à la solution classique, notre proposition consomme moins d'énergie en l'absence de transmission de paquets asynchrones, comme les nœuds ne sont en mode Rx qu'une courte durée pour détecter la présence de paquet et éteignent leurs radios si aucun paquet n'est détecté.

5.4 Conclusion

Dans ce dernier chapitre, nous avons décrit un mode d'opération pour mettre en place une interopérabilité d'une technologie synchrone TSCH et d'une autre technologie asynchrone Thread, dont le trafic est contraint en latence.

Pour ce faire, nous avons utilisé les mécanismes MAC du standard 802.15.4 RIT, non-*beacon* et TSCH de manière simultanée. Nous avons proposé ainsi d'ajouter dans les ordonnancements des nœuds synchrones TSCH l'envoi d'information de type RIT, permettant de mettre en place une interopérabilité avec le nœud Thread, lui aussi fonctionnant avec le mécanisme RIT pour la communication d'interopérabilité. Nous avons également utilisé une version améliorée de l'ordonnanceur présenté dans le chapitre précédent pour garantir la transmission d'informations RIT sur chaque canal et par chacun des nœuds au cours d'une *slotframe*.

Nous avons ensuite évalué analytiquement les performances de la stratégie proposée, et montrons que notre solution est non seulement possible en exploitant le mécanisme RIT du standard 802.15.4, mais permet également de réduire globalement la consommation d'énergie en remplaçant les nœuds asynchrones constamment actifs en mode Rx par une activité judicieusement répartie dans le temps et sur les différents canaux des nœuds d'un réseau TSCH à proximité, en relâchant légèrement la contrainte de latence. Ainsi, si on tolère une latence de 500 ms, notre proposition permet de dépenser globalement quatre fois moins d'énergie.

Nous avons conscience que cette étude ne constitue qu'une étape préliminaire dans la réalisation d'une interopérabilité effective : nous n'avons par exemple pas pris en compte le surcoût de consommation lié à l'exploitation de la présence d'un réseau synchrone à proximité par de nombreux nœuds asynchrones exploitant le réseau synchrone TSCH simultanément. Une telle situation générera, dans la proposition

actuelle, des collisions dont l'impact pourrait devenir non négligeable dans le cas d'une forte densité de nœuds asynchrone. Ainsi, notre étude nécessite d'être prolongée dans cette direction.

Par ailleurs, il est nécessaire de tester notre proposition dans un environnement réel, en exploitant sur des plateformes capables de gérer les deux modes de communication : le mode natif synchrone TSCH ou asynchrone Thread, ainsi que le mode d'interopérabilité, exploitant le mode RIT.

Ces travaux représentent néanmoins les performances que l'on peut espérer atteindre lors du déploiement de cette interopérabilité, et permettent de valider l'intérêt de mettre en place une telle solution.

De plus, le déploiement de la solution proposée ne modifie quasiment pas les standards et modes de fonctionnement existants, ce qui devrait faciliter le développement d'une telle interopérabilité.

Conclusion générale

Les travaux présentés s'inscrivent dans la continuité du projet GREENNET, mené par ST Microélectronique, dont l'objectif était de créer une pile protocolaire utilisant une couche MAC 802.15.4 dédiée aux plateformes à récupération d'énergie.

À la suite du projet GREENNET, l'essor rencontré par les différentes technologies dédiées aux WSN, a conduit au développement de plateformes capables d'opérer dans deux technologies différentes, adaptées aux contextes de transmission.

De telles plateformes ouvrent la porte à la conception de réseaux multitechnologies, que nous proposons d'exploiter dans le but de réduire la consommation d'énergie globale.

Après avoir décrit les principales technologies de l'IoT dans un premier chapitre, nous les avons classifiées en fonction des mécanismes MAC qu'elles exploitent. Nous avons ensuite présenté le lien entre le contexte applicatif (latence et fréquence de la génération de données) et le mécanisme MAC consommant le moins d'énergie.

Nous avons alors analysé les technologies exploitant un mécanisme MAC synchrone : elles sont les plus adaptées aux trafics applicatifs périodiques dont les intervalles entre les générations de données sont courts. En effet, la dérive d'horloge entraîne un coût de maintien d'une synchronisation active trop élevé dans le cas de trafics périodiques rares.

De plus, nous avons remarqué que la gestion des trafics applicatifs rares contraints en latence repose, dans les solutions existantes, sur l'utilisation d'une plateforme constamment active en mode *Rx*.

Nous proposons ainsi d'exploiter les plateformes multitechnologies pour constituer un réseau synchrone dans lequel chaque nœud répartit son activité dans le temps pour globalement économiser de l'énergie pour remplacer le rôle du dispositif constamment disponible utilisé pour acheminer des trafics asynchrones contraints en latence.

Résumé de la contribution

Nous avons tout d'abord proposé un système de comparaison des différentes solutions, en comparant toutes les technologies sur un pied d'égalité.

Pour ce faire, nous avons considéré la même quantité de données applicatives au niveau de la troisième couche du modèle OSI, et déterminé de manière analytique la durée de vie de plateformes opérant dans chacune des technologies étudiées.

Les principaux résultats de cette étude sont les suivants : parmi les technologies synchrones, le BLE obtient les meilleures durées de vie tant que son débit lui permet d'acheminer le trafic imposé. Les solutions 802.11 ne deviennent intéressantes que lorsque le trafic acheminé est de taille très importante. Les technologies asynchrones LoRa et SIGFOX obtiennent des durées de vie comparables à celles des technologies synchrones dans un contexte de très bas débit applicatif. Si l'on prend en compte la dérive d'horloge que les nœuds synchrones peuvent subir entre deux intervalles de communication, les technologies asynchrones deviennent les plus performantes lorsque l'on considère une faible quantité de données applicatives envoyées par jour. Cela peut s'expliquer par le fait que, pour des données relativement rares, il faut dimensionner un intervalle de garde, qui consomme de l'énergie en mode *Rx*, pour prendre en compte la dérive d'horloge sans pour autant manquer les rendez-vous de communications synchrones.

Nous avons également montré que, si l'on considère de faibles débits applicatifs, les temps passés dans les modes de consommation radio (*Rx* et *Tx*) sont négligeables devant le temps passé en mode *Sleep*. Ainsi, contrairement à ce qui est communément admis, la valeur de la consommation matérielle dans ce mode *Sleep* devient le critère le plus impactant sur les durées de vie des capteurs, dans le cas de faibles densités de trafic.

De plus, nous avons montré que les technologies qui fonctionnent dans la bande de fréquence sous-1 GHz sont fortement limitées en raison du rapport cyclique d'activité imposé par l'ETSI : ainsi, ces technologies ne sont pas recommandées pour l'acheminement d'une forte quantité de données.

Enfin, nous avons montré que le fonctionnement avec une plateforme à récupération d'énergie n'est pour l'instant pas réalisable pour les technologies asynchrones ; seuls le BLE et le 802.15.4 peuvent espérer

fonctionner avec ces systèmes.

Ces résultats nous ont permis d'identifier les principaux avantages et inconvénients de chacune des solutions : si le BLE obtient les meilleures performances lorsque l'on ne prend pas en compte la dérive d'horloge, c'est au prix d'une portée relativement faible, et d'un coût de maintien de la synchronisation active.

Nous avons en effet montré que le coût de maintien d'une synchronisation active est parfois démesuré par rapport à la consommation liée au trafic à acheminer. Ainsi, dans le contexte des WSN, qui ne doivent parfois remonter des valeurs de capteurs qu'une seule fois par jour, le maintien d'une association active est parfois trop coûteux.

Dans ce contexte, il peut être plus intéressant de se détacher du réseau pour le rejoindre en cas de besoin, en fonction du coût de la procédure d'attache au réseau. Ainsi, dans le but de favoriser l'utilisation d'un mode de communication non synchronisé, nous avons proposé de diminuer le coût de la procédure d'attache.

Comme le TSCH est la seule technologie synchrone, multisaut et multicanal par construction, qui permet d'économiser de l'énergie en chacun des nœuds du réseau, nous avons proposé de s'intéresser à cette technologie. Ainsi, dans le cinquième chapitre, nous avons présenté deux approches pour réduire le coût de cette procédure d'attache à un réseau TSCH. En effet, la procédure actuellement standardisée d'attache au réseau est naïve et très coûteuse en énergie, ce qui décourage l'utilisation d'un mode de communication asynchrone, reposant sur une succession d'associations/ désassociations du réseau.

Nous avons proposé deux solutions : la première repose sur l'exploitation de séquences mathématiques dont la propriété est d'étaler les périodes d'activités dans le temps, tout en minimisant l'impact sur la latence de la procédure, ce qui permet de diminuer le coût global énergétique de cette phase d'attache au réseau. Grâce au développement d'un simulateur des performances de la phase d'attache à un réseau TSCH, nous avons montré que le protocole d'attache proposé obtient de meilleures performances, soit en termes de latence, soit en termes de consommation d'énergie globale, que les protocoles d'attache classiquement utilisés dans les réseaux de capteurs.

Cependant, cette méthode nécessite que tous les nœuds aient une connaissance *a priori* des séquences mathématiques, que l'on ne peut générer aisément. Nous avons alors proposé une seconde solution d'attache au réseau, plus simple en termes de déploiement.

Cette deuxième méthode, *DataJoin*, exploite les trames d'acquiescement des communications TSCH pour y ajouter des éléments d'informations : nous avons proposé d'inclure dans ces trames d'acquiescement la date d'envoi de la prochaine trame de synchronisation sur le même canal physique que celui utilisé pour l'envoi de la trame d'ACK. Nous avons montré que, couplée à un ordonnancement soigneux des balises, cette solution permet d'obtenir des délais identiques, pour une durée d'activité en mode *Rx* considérablement réduite.

Ainsi, cette seconde proposition, si elle ne réduit pas la latence d'attache au réseau, en réduit le coût énergétique, et rend possible l'exploitation de synchronisations/désynchronisations successives dans le but d'une interopérabilité multimode : synchrone et asynchrone, au sein de la même technologie.

Par ailleurs, nous avons souligné que, lors de la procédure d'attache au réseau, la situation du nœud qui tente de rejoindre le réseau synchrone dans le but d'y acheminer des données applicatives est très proche de la situation d'un nœud asynchrone qui souhaiterait acheminer des données au sein d'un réseau synchrone à proximité.

C'est pourquoi nous avons proposé, dans un dernier chapitre, d'exploiter l'ordonnanceur intelligent de *DataJoin* pour l'envoi de trames de sollicitation à destination d'un potentiel nœud asynchrone, permettant ainsi d'acheminer un trafic asynchrone au sein d'un réseau synchrone avec une latence bornée.

Nous avons également quantifié et évalué les performances de cette dernière proposition, qui montrent la faisabilité et prouvent l'intérêt d'une telle proposition.

Travaux futurs

Au cours des travaux menés durant cette thèse, nous avons sélectionné les différentes directions à donner à nos recherches. Cependant, il nous paraît important de mentionner que ces choix, s'ils nous ont permis de constituer une première étape de travail dans le domaine de l'interopérabilité, ne constituent qu'une première étape en matière d'interopérabilité multiMAC. Ainsi, des travaux sont encore nécessaires quant à l'exploitation des plateformes capables d'utiliser plusieurs technologies en parallèle.

Tout d'abord, lors de la comparaison des technologies existantes, nous avons choisi de ne pas prendre en compte l'impact de la modulation physique, en termes de robustesse et de portée des transmissions. Il est évidemment possible d'harmoniser les résultats que nous avons présentés en divisant par la portée de chacune des technologies, et ainsi obtenir la consommation énergétique d'une technologie par bit applicatif et par mètre de portée, mais de tels résultats ne prennent pas en compte le surcoût imposé par les technologies multisauts pour atteindre des portées équivalentes à celles obtenues par les technologies longues portées de types LoRa ou SIGFOX. Il paraît donc pertinent de mener une étude comparative qui estimerait le coût global des technologies multisauts, en comparaison des coûts de communication à longue portée, pour une taille de donnée applicative donnée. Les travaux de Lampin [124] constituent une première étape dans ce sens, et pourraient ainsi être couplés à notre étude, permettant ainsi de rendre la comparaison encore plus réaliste.

Par ailleurs, il est également pertinent de complexifier le modèle de comparaison des piles IoT, de manière à prendre en compte la phase d'attache au réseau, qui, comme nous l'avons vu, représente une partie non négligeable de la consommation énergétique d'un nœud.

Toujours en vue de complexifier la comparaison, rappelons que notre étude ne prend actuellement en compte que la consommation énergétique de la feuille, ce qui ne traduit pas les différences de consommation globale entre les réseaux. Il est en est de même pour la prise en compte des réseaux maillés et multisauts. Prendre en compte la consommation globale du réseau au lieu de celle de la feuille uniquement, permettrait de comparer de manière plus raffinée, les technologies synchrones et asynchrones.

Enfin, vis-à-vis de cette comparaison, on peut également parfaire le modèle de consommation d'énergie, à la fois en prenant en compte l'énergie perdue dans les changements d'états matériels, et en prenant des modèles de décharge de batteries plus réalistes.

Ensuite, au cours de nos travaux sur la réduction des coûts de la procédure d'attache dans le but d'exploiter des associations/désassociations successives, nous ne prenons pas en compte le fait que les nœuds qui ont déjà rejoint le réseau ont également obtenus des informations sur son fonctionnement, que l'on pourrait exploiter dans le but d'une interopérabilité multimode synchrone et asynchrone.

Récemment, les travaux de Claeys *et al.* [125] présentent un mécanisme d'exploitation des données obtenues lors de la phase précédente de synchronisation au réseau dans le but de le rejoindre rapidement. Ainsi, au lieu de répéter la procédure d'attache totale, il nous semble pertinent d'étudier la solution proposée au service d'une interopérabilité multimode synchrone/asynchrone constituée de synchronisations et désynchronisations successives.

Par ailleurs, cette logique peut également s'étendre au cadre générique d'interopérabilité proposé dans le dernier chapitre, en supposant que le nœud asynchrone puisse exploiter certaines des informations recueillies sur le réseau synchrone, lors d'une précédente interaction avec ce réseau, de manière à encore diminuer le coût de la communication d'interopérabilité.

Le cadre d'interopérabilité synchrone/asynchrone que nous avons proposé, s'il est générique, n'est pas décrit de manière précise dans le contexte de couches PHY différentes. En effet, en choisissant de travailler avec la même couche physique, nous avons, dans un premier temps, écarté les problématiques liées aux formats des trames utilisées, celles liées à la gestion des contraintes de temps matériels pour changer d'une technologie vers une autre, ainsi que la mise en place d'une solution pour gérer la différence de débit entre les différentes technologies, notamment vis-à-vis de la quantité de paquets à envoyer via une communication d'interopérabilité.

Par ailleurs, dans ces travaux, nous avons proposé deux types d'interopérabilité : à la fois une interopérabilité multimode avec l'exploitation d'associations/ désassociations successives, et une interopérabilité multitechnologie, en permettant la communication entre des nœuds Thread et TSCH.

Cependant, l'intérêt de la solution multitechnologie vis-à-vis d'une interopérabilité multimode reste à évaluer. En effet, l'exploitation des multimodes correspond à une réduction du coût de déploiement et de maintien de l'infrastructure, mais ne permet en revanche pas d'exploiter différentes couches PHY, ce qui peut être nécessaire en matière de portée, de robustesse et de diversité des bandes de fréquences utilisées.

Dans le but de comparer ces deux approches, le déploiement des différentes solutions proposées, dans un environnement réel, de plateformes capables d'exécuter de manière concurrente les deux piles protocolaires Thread et TSCH dans un premier temps, et des technologies de couches PHY différentes ensuite, ainsi que de plateformes capables d'exploiter des synchronisations/désynchronisations successives via les différentes procédures d'attache au réseau proposées, permettrait à la fois de valider l'étude sur l'interopérabilité menée dans cette thèse, et de comparer les approches multimodes et multitechnologies.

Enfin, ces travaux ayant été menés en parallèle de l'évolution de la standardisation de la cinquième génération de standards pour la téléphonie mobile, il nous semble intéressant de mentionner que le standard de la 5G est censé couvrir une multitude de contextes applicatifs, à la fois courte et longue portée, à la fois bas débit, dans le contexte de l'IoT, et très haut débit, de l'ordre de la dizaine de gigabits par seconde.

Nous pensons alors que différents modes de communications de 5G vont nécessairement cohabiter, dont probablement des modes synchrones et asynchrones. Notre proposition pourrait alors être étudiée dans ce cadre.

Bibliographie

- [1] THREAD GROUP : Thread : A New Wireless Networking Protocol for the Home, 2017. URL <http://threadgroup.org/>. Accessed on 2017-07-27. 5, 19, 35
- [2] Thang Phan DUY, Thanh DINH et Younghan KIM : A Rapid Joining Scheme based on Fuzzy Logic for Highly Dynamic IEEE 802.15.4e TSCH Networks. *International Journal of Distributed Sensor Networks*, 12(8):1550147716659424, 2016. 6, 96, 130
- [3] Elvis VOGLI, Giuseppe RIBEZZO, Luigi Alfredo GRIECO et Gennaro BOGGIA : Fast Join and Synchronization Schema in the IEEE 802.15.4e MAC. In *IEEE WCNC Workshops*, pages 85–90, 2015. 6, 96, 97, 98, 112, 115, 116
- [4] Andrew S. TANENBAUM et David J. WETHERALL : *Computer Networks*. Pearson, 2011. 17
- [5] Liviu-Octavian VARGA, Gabriele ROMANIELLO, Malisa VUCINIC, Michel FAVRE, Andrei BANCIU, Roberto GUIZZETTI, Christophe PLANAT, Pascal URARD, Martin HEUSSE, Franck ROUSSEAU *et al.* : Green-Net : an Energy Harvesting IP-enabled Wireless Sensor Network. *IEEE Internet of Things Journal*, 2, 2015. 17, 72, 73, 77, 79, 89, 103
- [6] BLUETOOTH SIG : Specification of Bluetooth Core v5.0, décembre 2016. 19, 38, 84
- [7] LORA ALLIANCE : LoRa Alliance Website, 2015. URL <http://lora-alliance.org/>. Accessed on 2017-1-10. 19, 72
- [8] IEEE 802 LAN/MAN Standards COMMITTEE *et al.* : 802.11-2016 - IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. <http://ieeexplore.ieee.org/document/7786995/>, 2016. 19
- [9] IEEE : IEEE Standard for Low-Rate Wireless Networks, 2015. 19, 30, 33, 35, 61, 64, 79, 96, 107, 111, 122
- [10] Elodie MORIN, Mickael MAMAN, Roberto GUIZZETTI et Andrzej DUDA : Comparison of the Device Lifetime in Wireless Networks for the Internet of Things. *IEEE Access*, 5:7097–7114, 2017. 20
- [11] Elodie MORIN : IoT-Analyzer. <https://gricad-gitlab.univ-grenoble-alpes.fr/morinelot/iot-analyzer>, mars 2017. 20, 73, 74
- [12] Elodie MORIN, Mickael MAMAN, Roberto GUIZZETTI et Andrzej DUDA : Singer Cyclic Difference Sets for an Energy-Efficient Joining Scheme. In *13th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications.*, 2017. 20
- [13] Xavier VILAJOSANA, Qin WANG, Fabien CHRAIM, Thomas WATTEYNE, Tengfei CHANG et Kristofer S.J. PISTER : A Realistic Energy Consumption Model for TSCH Networks. *IEEE Sensors Journal*, 14(2):482–489, 2014. 27, 72, 73, 103
- [14] IEEE : Standard for Local and Metropolitan Area Networks – Part 15.4e : Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1 : MAC sublayer, 2012. 30, 32
- [15] Maria-Rita PALATTELLA, Nicola ACCETTURA, Mischa DOHLER, Luigi Alfredo GRIECO et Gennaro BOGGIA : Traffic Aware Scheduling Algorithm for Reliable Low-Power Multi-Hop IEEE 802.15.4e Networks. In *IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pages 327–332. IEEE, 2012. 31
- [16] Antoni MORELL, Xavier VILAJOSANA, José López VICARIO et Thomas WATTEYNE : Label Switching over IEEE802.15.4e Networks. *Transactions on Emerging Telecommunications Technologies*, 24(5):458–475, 2013. 31

- [17] Nicola ACCETTURA, Maria Rita PALATTELLA, Gennaro BOGGIA, Luigi Alfredo GRIECO et Mischa DOHLER : DeTAS : A Decentralized Traffic Aware Scheduling Technique Enabling IoT-compliant Multi-hop Low-power and Lossy Networks. *In Second IEEE WoWMoM Workshop on the Internet of Things : Smart Objects and Services, IoT-SoS*, pages 1–6, 2013. 31
- [18] Joseph POLASTRE, Jason HILL et David CULLER : Versatile Low Power Media Access for Wireless Sensor Networks. *In Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems*, pages 95–107. ACM, 2004. 33
- [19] Abdelmalik BACHIR, Dominique BARTHEL, Martin HEUSSE et Andrzej DUDA : Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks. *In 2006 IEEE International Conference on Communications*, volume 7, pages 3365–3370. IEEE, 2006. 33
- [20] Amre EL-HOIYDI : Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. *In 2002 IEEE International Conference on Communications (ICC). Conference Proceedings.*, volume 5, pages 3418–3423. IEEE, 2002. 33
- [21] Michael BUETTNER, Gary V YEE, Eric ANDERSON et Richard HAN : X-MAC : A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks. *In Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, pages 307–320. ACM, 2006. 33
- [22] Adam DUNKELS : The ContikiMAC Radio Duty Cycling Protocol. Rapport technique, Swedish Institute of Computer Science, 2011. 33
- [23] Heping WANG, Xiaobo ZHANG, Farid NAÏT-ABDESSELAM et Ashfaq KHOKHAR : DPS-MAC : An Asynchronous MAC Protocol for Wireless Sensor Networks. *High Performance Computing–HiPC 2007*, pages 393–404, 2007. 33
- [24] SangSoon LIM, SungHo KIM, JaeJoon CHO et Sunshin AN : Medium Access Control with an Energy-Efficient Algorithm for Wireless Sensor Networks. *In Personal Wireless Communications*, pages 334–343. Springer, 2006. 33
- [25] Stefan MAHLKNECHT et Michael BOCK : CSMA-MPS : a minimum preamble sampling MAC protocol for low power wireless sensor networks. *In IEEE International Workshop on Factory Communication Systems, 2004. Proceedings.*, pages 73–80. IEEE, 2004. 33
- [26] E-YA LIN, Jan M RABAEY et Adam WOLISZ : Power-efficient rendez-vous schemes for dense wireless sensor networks. *In 2004 IEEE International Conference on Communications*, volume 7, pages 3769–3776. IEEE, 2004. 33, 34
- [27] Carlos CORDEIRO et Kiran CHALLAPALI : C-MAC : A Cognitive MAC Protocol for Multi-Channel Wireless Networks. *In 2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pages 147–157. IEEE, 2007. 33
- [28] Pei HUANG, Li XIAO, Soroor SOLTANI, Matt W MUTKA et Ning XI : The Evolution of MAC Protocols in Wireless Sensor Networks : A Survey. *IEEE Communications Surveys & Tutorials*, 15(1):101–120, 2013. 33
- [29] Prabal DUTTA, Stephen DAWSON-HAGGERTY, Yin CHEN, Chieh-Jan Mike LIANG et Andreas TERZIS : Design and Evaluation of a Versatile and Efficient Receiver-Initiated Link Layer for Low-Power Wireless. *In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 1–14. ACM, 2010. 34
- [30] Yanjun SUN, Omer GUREWITZ et David B JOHNSON : RI-MAC : A Receiver-Initiated Asynchronous Duty Cycle MAC Protocol for Dynamic Traffic Loads in Wireless Sensor Networks. *In Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, pages 1–14. ACM, 2008. 34
- [31] N. KUSHALNAGAR, G. MONTENEGRO et C. SCHUMACHER : IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs) : Overview, Assumptions, Problem Statement, and Goals. RFC 4919, RFC Editor, août 2007. URL <http://www.rfc-editor.org/rfc/rfc4919.txt>. <http://www.rfc-editor.org/rfc/rfc4919.txt>. 35, 41

- [32] Gary Scott MALKIN : RIP Version 2. STD 56, RFC Editor, novembre 1998. URL <http://www.rfc-editor.org/rfc/rfc2453.txt>. <http://www.rfc-editor.org/rfc/rfc2453.txt>. 36
- [33] Maria Rita PALATTELLA, Nicola ACCETTURA, Xavier VILAJOSANA, Thomas WATTEYNE, Luigi Alfredo GRIECO, Gennaro BOGGIA et Mischa DOHLER : Standardized Protocol Stack for the Internet of (Important) Things. *Communications Surveys & Tutorials, IEEE*, 15(3):1389–1406, 2013. 36
- [34] APPLE : HomeKit. <https://developer.apple.com/homekit/>, juin 2016. (Accessed on 11/28/2017). 36
- [35] OPEN CONNECTIVITY FOUNDATION : IoTivity. <https://www.iotivity.org/>, février 2017. (Accessed on 11/28/2017). 36, 37
- [36] OPEN CONNECTIVITY FOUNDATION (OCF) : Open Connectivity Foundation : Unlocking the Massive Opportunity in the Internet of Things. <https://openconnectivity.org/>, février 2017. (Accessed on 11/28/2017). 36
- [37] Pascal THUBERT : An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4. Internet-Draft draft-ietf-6tisch-architecture-13, Internet Engineering Task Force, novembre 2017. URL <https://datatracker.ietf.org/doc/html/draft-ietf-6tisch-architecture-13>. Work in Progress. 38
- [38] X. VILAJOSANA, K. PISTER et T. WATTEYNE : Minimal 6TiSCH Configuration. Internet Draft draft-ietf-6tisch-minimal-21, IETF, février 2017. URL <https://tools.ietf.org/html/draft-ietf-6tisch-minimal-21>. 38, 96, 119
- [39] Q. WANG et X. VILAJOSANA : 6top Protocol (6P). Internet Draft draft-ietf-6tisch-6top-protocol-03, IETF, octobre 2016. URL <https://tools.ietf.org/html/draft-ietf-6tisch-6top-protocol-03>. 38
- [40] 6LO WORKING GROUP IETF : 6lo Status Pages, 2015. URL <https://tools.ietf.org/wg/6lo/>. Accessed on 2017-1-10. 38, 41
- [41] G. MONTENEGRO, N. KUSHALNAGAR, J. HUI et D. CULLER : Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944, RFC Editor, septembre 2007. URL <http://www.rfc-editor.org/rfc/rfc4944.txt>. <http://www.rfc-editor.org/rfc/rfc4944.txt>. 41
- [42] IEEE : IEEE Standard 802.11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999. 41, 72
- [43] IEEE : IEEE P802.11ah/D10.0 Draft Standard for Information Technologies Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks Specific Requirements – Part 11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications – Amendment 6 : Sub 1 GHz License, september 2016. 42, 43, 72
- [44] IEEE : IEEE Standard 802.11 : Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 4 : Enhancements for Very High Throughput for Operation in Bands below 6 GHz, 2014. 42
- [45] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE : Electromagnetic compatibility and Radio spectrum Matters (ERM); Radio Frequency Identification Equipment operating in the band 865 MHz to 868 MHz with power levels up to 2 W. http://www.etsi.org/deliver/etsi_en/302200_302299/30220801/01.04.01_40/en_30220801v010401o.pdf, juillet 2011. (Accessed on 01/12/2017). 43, 44, 47
- [46] Evgeny KHOROV, Andrey LYAKHOV, Alexander KROTOV et Andrey GUSCHIN : A Survey on IEEE 802.11ah : an Enabling Networking Technology for Smart Cities. *Computer Communications*, 2014. 43

- [47] J. KIM : RAW Assignment follow up. <http://mentor.ieee.org/802.11/dcn/13/11-13-0510-01-00ah-raw-assignment-follow-up.ppt>, novembre 2013. (Accessed on 11/30/2017). 43
- [48] Orod RAEESI, Juho PIRSKANEN, Ali HAZMI, Jukka TALVITIE et Mikko VALKAMA : Performance Enhancement and Evaluation of IEEE 802.11ah Multi-Access Point Network Using Restricted Access Window Mechanism. *In International Conference on Distributed Computing in Sensor Systems (DCOSS), 2014 IEEE*, pages 287–293. IEEE, 2014. 43
- [49] Claire GOURSAUD et Jean-Marie GORCE : Dedicated Networks for IoT : PHY/MAC State of the Art and Challenges. *EAI Endorsed Transactions on Internet of Things*, 2015. 44, 72
- [50] SEMTECH, AN : AN1200.22, LoRa Modulation Basics, mai 2015. 46
- [51] PYCOM : FiPy - Five Network Development Board with LTE-M, LoRa, Sigfox, WiFi and Bluetooth. <https://pycom.io/product/fipy>, avril 2017. (Accessed on 11/29/2017). 50
- [52] Pere TUSET, Xavier VILAJOSANA et Thomas WATTEYNE : OpenMote+ : a Range-Agile Multi-Radio Mote. *In International Conference on Embedded Wireless Systems and Networks (EWSN)*, pages 333–334, Graz, Austria, février 2016. ACM. URL <https://hal.inria.fr/hal-01239662>. 50, 57
- [53] VOLANSYS | NXP : Modular IoT Gateway. <https://volansys.com/kw41z/>, novembre 2017. (Accessed on 11/14/2017). 50, 57
- [54] ST MICROELECTRONIQUE : Discovery kit for IoT node, multi-channel communication with STM32L4. http://www.st.com/content/ccc/resource/technical/document/data_brief/group1/ab/8e/cb/09/e6/fc/42/9a/DM00347837/files/DM00347837.pdf/jcr:content/translations/en.DM00347837.pdf, novembre 2017. (Accessed on 11/14/2017). 50, 57
- [55] Ricardo C. CARRANO, Diego PASSOS, Luiz C.S. MAGALHAES et Celio V.N. ALBUQUERQUE : Survey and Taxonomy of Duty Cycling Mechanisms in Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, 16(1):181–194, 2014. 51, 52, 55
- [56] Iacob JUC, Olivier ALPHAND, Roberto GUIZZETTI, Michel FAVRE et Andrzej DUDA : Stripe : a Distributed Scheduling Protocol for 802.15.4e TSMC Networks. Research Report RR-LIG-54, Laboratoire d’Informatique de Grenoble, septembre 2017. URL <https://hal.inria.fr/hal-01585904>. Les rapports de recherche du LIG - ISSN : 2105-0422. 55, 115
- [57] Abtin KESHAVARZIAN, Huang LEE et Lakshmi VENKATRAMAN : Wakeup Scheduling in Wireless Sensor Networks. *In Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 322–333. ACM, 2006. 55, 84, 123
- [58] Jon POSTEL : Internet Protocol. STD 5, RFC Editor, septembre 1981. URL <http://www.rfc-editor.org/rfc/rfc791.txt>. <http://www.rfc-editor.org/rfc/rfc791.txt>. 57
- [59] SILICON LABS : Wireless Gecko IoT Connectivity Portfolio. <https://www.silabs.com/products/wireless/wireless-gecko-iot-connectivity-portfolio>, novembre 2017. (Accessed on 11/13/2017). 57
- [60] NXP : Kinetis KW40Z-2.4 GHz Dual Mode : BLE and 802.15.4 Wireless Radio Microcontroller (MCU) based on ARM Cortex-M0+ Core. <https://www.nxp.com/products/microcontrollers-and-processors/arm-based-processors-and-mcus/kinetis-cortex-m-mcus/w-serieswireless-conn.m0-plus-m4/kinetis-kw40z-2.4-ghz-dual-mode-ble-and-802.15.4-wireless-radio-microcontroller-mcu-based-on-arm-cortex-m0-plus-core:KW40Z>, novembre 2017. (Accessed on 11/13/2017). 57

- [61] ARROW ELECTRONICS : Arrow Electronics accelerates IoT endpoint creation with new ARIS EDGE platform. <https://www.arrow.com/en/research-and-events/articles/arrow-electronics-accelerates-iot-endpoint-creation-with-new-aris-edge-platform>, septembre 2017. (Accessed on 11/13/2017). 57
- [62] Andras FARAGO, Andrew D MYERS, Violet R SYROTIUK et Gergely V ZARUBA : Meta-MAC protocols : Automatic combination of MAC protocols to optimize performance for unknown conditions. *IEEE Journal on Selected Areas in Communications*, 18(9):1670–1681, 2000. 59
- [63] Christian DOERR, Michael NEUFELD, Jeff FIFIELD, Troy WEINGART, Douglas C SICKER et Dirk GRUNWALD : MultiMAC - an adaptive MAC framework for dynamic radio networking. *In First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, 2005. DySPAN 2005*, pages 548–555. IEEE, 2005. 59
- [64] Sufi Tabassum GUL : *Optimization of multi-standards software defined radio equipments : a common operators approach*. Thèse de doctorat, Université Rennes 1, 2009. 59
- [65] Tae-Suk KIM, Taerim PARK, Mo SHA et Chenyang LU : Toward MAC Protocol Service over the air. *In Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 451–457. IEEE, 2012. 59
- [66] Pieter DE MIL, Peter RUCKEBUSCH, Jeroen HOEBEKE, Ingrid MOERMAN et Piet DEMEESTER : Pluralis-MAC : a generic multi-MAC framework for heterogeneous, multiservice wireless networks, applied to smart containers. *EURASIP Journal on Wireless Communications and Networking*, 2012(1):166, 2012. 59
- [67] Xi ZHANG, Junaid ANSARI, Guangwei YANG et Petri MÄHÖNEN : TRUMP : Efficient and Flexible Realization of Medium Access Control Protocols for Wireless Networks. *IEEE Transactions on Mobile Computing*, 15(10):2614–2626, 2016. 60
- [68] Junaid ANSARI, Xi ZHANG, Andreas ACHTZEHN, Marina PETROVA et Petri MAHONEN : Decomposable MAC framework for highly flexible and adaptable MAC realizations. *In 2010 IEEE Symposium on New Frontiers in Dynamic Spectrum (DySPAN)*, pages 1–2. IEEE, 2010. 60
- [69] Qiang GUO, Jie ZHU et Xianghua XU : An adaptive multi-criteria vertical handoff decision algorithm for radio heterogeneous network. *In IEEE International Conference on Communications (ICC), 2005*, volume 4, pages 2769–2773. IEEE, 2005. 60
- [70] David C. HARRISON, Winston K.G. SEAH et Ramesh RAYUDU : Rare Event Detection and Propagation in Wireless Sensor Networks. *ACM Computing Surveys (CSUR)*, 48(4):58, 2016. 67
- [71] IEEE : Standard for Local and Metropolitan Area Networks – Part 15.4 : Low-Rate Wireless Personal Area Networks (LR-WPANs), 2003. 72
- [72] Serbulent TOZLU : Feasibility of Wi-Fi Enabled Sensors for Internet of Things. *In 7th International Wireless Communications and Mobile Computing Conference (IWCMC), 2011*, pages 291–296, juillet 2011. 72, 79, 84
- [73] SIGFOX COMPAGNY : SIGFOX Website, 2015. URL <http://www.sigfox.com/en/>. Accessed on 2017-1-10. 72
- [74] A. AL-FUQAHA, M. GUIZANI, M. MOHAMMADI, M. ALEDHARI et M. AYYASH : Internet of Things : a Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376, novembre 2015. ISSN 1553-877X. 72
- [75] T. KIM, I. H. KIM, Y. SUN et Z. JIN : Physical Layer and Medium Access Control Design in Energy Efficient Sensor Networks : An Overview. *IEEE Transactions on Industrial Informatics*, 11(1):2–15, février 2015. ISSN 1551-3203. 72
- [76] Abdelmalik BACHIR, Mischa DOHLER, Thomas WATTEYNE et Kin K LEUNG : MAC Essentials for Wireless Sensor Networks. *IEEE Communications Surveys & Tutorials*, 12(2):222–248, 2010. 72

- [77] Quan WANG et Jin JIANG : Comparative Examination on Architecture and Protocol of Industrial Wireless Sensor Network Standards. *IEEE Communications Surveys & Tutorials*, 2016. 72
- [78] Taejoon KIM, Il Han KIM, Yanjun SUN et ZhongYi JIN : Physical Layer and Medium Access Control Design in Energy Efficient Sensor Networks : an Overview. *IEEE Transactions on Industrial Informatics*, 11(1):2–15, 2015. 72
- [79] B. B. OLYAEI, J. PIRSKANEN, O. RAEESI, A. HAZMI et M. VALKAMA : Performance Comparison Between Slotted IEEE 802.15.4 and IEEE 802.11ah in IoT Based Applications. *In IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 332–337, octobre 2013. 72
- [80] Matti SIEKKINEN, Markus HIIENKARI, Jukka K. NURMINEN et Johanna NIEMINEN : How Low Energy is Bluetooth Low Energy? Comparative Measurements with ZigBee/802.15.4. *In WCNC 2012 Workshop on Internet of Things Enabling Technologies*, 2012. 72, 78, 84
- [81] Borja MARTINEZ, Màrius MONTÓN, Ignasi VILAJOSANA et Joan Daniel PRADES : The Power of Models : Modeling Power Consumption for IoT Devices. *IEEE Sensors Journal*, 15(10):5777–5789, 2015. 72, 73, 78, 103
- [82] Qin WANG, Mark HEMPSTEAD et Woodward YANG : A Realistic Power Consumption Model for Wireless Sensor Network Devices. *In Proceedings of the Third Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON 2006, September 25-28, 2006, Reston, VA, USA*, pages 286–295, 2006. 73
- [83] Jian LI et Prasant MOHAPATRA : Analytical Modeling and Mitigation Techniques for the Energy Hole Problem in Sensor Networks. *Pervasive and Mobile Computing*, 3(3):233–254, 2007. 73
- [84] Qin WANG et Woodward YANG : Energy Consumption Model for Power Management in Wireless Sensor Networks. *In 2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pages 142–151. IEEE, 2007. 73
- [85] Hai-Ying ZHOU, Dan-Yan LUO, Yan GAO et De-Cheng ZUO : Modeling of Node Energy Consumption for Wireless Sensor Networks. *Wireless Sensor Network*, 3(1):18, 2011. 73
- [86] Baoqiang KAN, Li CAI, Lei ZHAO et Yongjun XU : Energy Efficient Design of WSN Based on an Accurate Power Consumption Model. *In 2007 International Conference on Wireless Communications, Networking and Mobile Computing*, pages 2751–2754. IEEE, 2007. 73
- [87] Joseph POLASTRE, Robert SZEWCZYK et David CULLER : Telos : Enabling Ultra-Low Power Wireless Research. *In Proc. of the Fourth International Symposium on Information Processing in Sensor Networks (IPSN)*., pages 364–369. IEEE, 2005. URL <http://tinyurl.com/hh6g6al>. 73, 79
- [88] DUST NETWORKS : Dust Networks SmartMesh Power and Performance Estimator, . URL <http://www.linear.com/docs/42452>. Accessed on 2017-1-10. 74, 77
- [89] Specification of the Bluetooth System v4.0, juin 2010. 78
- [90] RTX : RTX4100 Datasheet DS1, 2015. URL <http://tinyurl.com/oeekop6>. Accessed on 2017-1-10. 79
- [91] MAXIM INTEGRATED : MAX2830 2.4GHz to 2.5GHz 802.11g/b RF Transceiver with PA and Rx/Tx/Diversity Switch, 2015. URL <http://tinyurl.com/q3xyynj>. Accessed on 2017-1-10. 79
- [92] ST MICROELECTRONIQUE : SPWF01SA Datasheet, 2015. URL <http://tinyurl.com/or94b9l>. Accessed on 2015-10-16. 79
- [93] NORDIC SEMICONDUCTOR : nRF51822 Product Specification v3.1, 2015. URL <http://tinyurl.com/npuckqy>. Accessed on 2017-1-10. 79
- [94] BLUEGIGA : BLE112 Datasheet. URL <http://tinyurl.com/o8p8spj>. Accessed on 2017-1-10. 79

- [95] ST MICROELECTRONIQUE : BlueNRG Datasheet. URL <http://tinyurl.com/pyl5ms3>. Accessed on 2017-1-10. 79
- [96] P. URARD, G. ROMAGNELLO, A. BANCIU, J.C. GRASSET, V. HEINRICH, M. BOULEMNAKHER, F. TODESCHNI, L. DAMON, R. GUIZZETTI, L. ANDRE et A. CATHELIN : A Self-Powered IPv6 Bidirectional Wireless Sensor and Actuator Network for Indoor Conditions. *In Symposium on VLSI Circuits*, pages C100–C101. IEEE, 2015. 79
- [97] Thomas WATTEYNE, Lance DOHERTY, Jerome SIMON et Kris PISTER : Technical Overview of Smart-Mesh IP. *In Proc. of the Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, pages 547–551. IEEE, 2013. 79
- [98] DUST NETWORKS : SmartMeshIP Datasheet, . URL <http://tinyurl.com/gqyeu5v>. Accessed on 2017-1-10. 79
- [99] TELECOM DESIGN : TD1202 Datasheet, 2015. URL <http://tinyurl.com/oclnxyo>. Accessed on 2017-1-10. 81
- [100] SEMTECH : SX1272 LoRa Datasheet. URL <http://tinyurl.com/qbbzn3d>. Accessed on 2017-1-10. 81
- [101] Marcin BRZOWSKI, Hendrik SALOMON et Peter LANGENDOERFER : On Efficient Clock Drift Prediction Means and Their Applicability to IEEE 802.15.4. *In Proc. of the IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC)*, pages 216–223. IEEE, 2010. 87
- [102] Vijay RAGHUNATHAN, Aman KANSAL, Jason HSU, Jonathan FRIEDMAN et Mani SRIVASTAVA : Design Considerations for Solar Energy Harvesting Wireless Embedded Systems. *In Proceedings of the 4th International Symposium on Information Processing in Sensor Networks*, page 64. IEEE Press, 2005. 89
- [103] Gabriele ROMANIELLO : *Energy Efficient Protocols for Harvested Wireless Sensor Networks*. Theses, Université Grenoble Alpes, mars 2015. URL <https://tel.archives-ouvertes.fr/tel-01149450>. 89
- [104] Joseph RAHMÉ, Nicolas FOURTY, Khaldoun AL AGHA et Adrien van den BOSSCHE : A Recursive Battery Model for Nodes Lifetime Estimation in Wireless Sensor Networks. *In Proceedings of IEEE WCNC, Sydney, Australia, 18-21 April 2010*, pages 1–6, 2010. 91
- [105] BLUETOOTH SIG : Specification of Bluetooth Mesh Profile v1.0, juillet 2017. 94
- [106] Thomas WATTEYNE, Steven LANZISERA, Ankur MEHTA et Kristofer SJ PISTER : Mitigating Multipath Fading through Channel Hopping in Wireless Sensor Networks. *In 2010 IEEE International Conference on Communications (ICC)*, pages 1–5. IEEE, 2010. 94
- [107] Riccardo POZZA, Michele NATI, Stylianos GEORGOULAS, Klaus MOESSNER et Alexander GLUHAK : Neighbor Discovery for Opportunistic Networking in Internet of Things Scenarios : A Survey. *IEEE Access*, 3:1101–1131, 2015. 95
- [108] Ying QIU, Shining LI, Xiangsen XU et Zhigang LI : Talk More Listen Less : Energy-Efficient Neighbor Discovery in Wireless Sensor Networks. *In IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016. 98, 99
- [109] Philipp H. KINDT, Daniel YUNGE, Gerhard REINERTH et Samarjit CHAKRABORTY : Griassdi : Mutually Assisted Slotless Neighbor Discovery. *In Proc. 16th ACM/IEEE IPSN*, pages 93–104, 2017. 98, 100
- [110] Mehedi BAKHT, Matt TROWER et Robin Hilary KRAVETS : Searchlight : Won't you be my neighbor? *In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, pages 185–196. ACM, 2012. 98, 100
- [111] Prabal DUTTA et David CULLER : Practical Asynchronous Neighbor Discovery and Rendezvous for Mobile Sensing Applications. *In Proc. 6th ACM SenSys*, pages 71–84. ACM, 2008. 98, 100

- [112] Arvind KANDHALU, Karthik LAKSHMANAN et Rangunathan Raj RAJKUMAR : U-Connect : A Low-Latency Energy-Efficient Asynchronous Neighbor Discovery Protocol. *In Proc. 9th ACM/IEEE IPSN*, pages 350–361, 2010. 98
- [113] Wai-Shing LUK et Tien-Tsin WONG : Two New Quorum Based Algorithms for Distributed Mutual Exclusion. *In Proc. 17th ICDCS*, pages 100–106. IEEE, 1997. 98, 100
- [114] Michael J. MCGLYNN et Steven A. BORBASH : Birthday Protocols for Low Energy Deployment and Flexible Neighbor Discovery in Ad Hoc Wireless Networks. *In Proc. 2nd ACM Mobihoc*, pages 137–145, 2001. 98, 99
- [115] Rong ZHENG, Jennifer C. HOU et Lui SHA : Optimal Block Design for Asynchronous Wake-Up Schedules and Its Applications in Multihop Wireless Networks. *IEEE Trans. on Mobile Computing*, 5(9):1228–1241, 2006. 98
- [116] Philipp KINDT, Marco SAUR et Samarjit CHAKRABORTY : Neighbor discovery latency in BLE-like duty-cycled protocols. *arXiv preprint arXiv:1509.04366*, 2015. 100
- [117] Tong MENG, Fan WU et Guihai CHEN : On Designing Neighbor Discovery Protocols : a Code-Based Approach. *In Proc. IEEE INFOCOM*, pages 1689–1697, 2014. 100, 101, 102, 103
- [118] LD BAUMERT : Cyclic Difference Sets. *Lecture Notes in Mathematics*, 182, 1971. 100, 101
- [119] Contiki-OS TEAM : The COOJA Simulator. https://github.com/contiki-os/contiki/wiki/An-Introduction-to-Cooja#The_COOJA_Simulator, novembre 2017. (Accessed on 11/23/2017). 102
- [120] Thomas WATTEYNE, Kazushi MURAOKA, Nicola ACCETTURA et Xavier VILAJOSANA : The 6TiSCH Simulator. <https://bitbucket.org/6tisch/simulator/src>, août 2015. (Accessed on 11/23/2017). 102, 115, 135
- [121] James SINGER : A Theorem in Finite Projective Geometry and Some Applications to Number Theory. *Transactions of the American Mathematical Society*, 43(3):377–385, 1938. 104
- [122] Dan GORDON : La Jolla Difference Set Repository, 2017. URL <https://ccrwest.org/diffsets.html>. 105, 106
- [123] Jun FUJIWARA, Ryota OKUMURA, Keiichi MIZUTANI, Hiroshi HARADA, Sota TSUCHIYA et Takuya KAWATA : Ultra-low power MAC protocol complied with RIT in IEEE 802.15. 4e for wireless smart utility networks. *In 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6. IEEE, 2016. 127
- [124] Quentin LAMPIN : *Urban Wireless Sensor Networks : Applications, Characterization, and Protocols*. Thèse de doctorat, in French, INSA de Lyon, 2014. 151
- [125] Timothy CLAEYS, Franck ROUSSEAU, Bernard TOURANCHEAU et Andrzej DUDA : Clock Drift Prediction for Fast Rejoin in 802.15. 4e TSCH Networks. *In 2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9. IEEE, 2017. 151