Conception et analyse des méthodes d'ordonnancement dans les réseaux 6TiSCH-802.15.4e à mode TSCH pour les applications industrielles de l'Internet des Objets

par

Taieb HAMZA

MÉMOIRE PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
COMME EXIGENCE PARTIELLE À L'OBTENTION DE LA MAÎTRISE
AVEC MEMOIRE EN GÉNIE DES TECHNOLOGIES DE
L'INFORMATION
M. Sc. A.

MONTRÉAL, LE 27 MARS 2019

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE:

M. Georges Kaddoum, Directeur de Mémoire
Département de génie électrique à l'École de technologie supérieure

M. Chamseddine Talhi, Président du Jury
Département de génie logiciel et des TI à l'École de technologie supérieure

M. Sègla Jean-Luc Kpodjedo, membre du jury
Département de génie logiciel et des TI à l'École de technologie supérieure

IL A FAIT L'OBJET D'UNE SOUTENANCE DEVANT JURY ET PUBLIC

LE 26 MARS 2019

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Au terme de la rédaction de ce mémoire, c'est un devoir agréable d'exprimer en quelques lignes la reconnaissance que je dois à tous ceux qui ont contribué de loin ou de près à l'élaboration de ce travail, qu'ils trouvent ici mes vifs respects et ma profonde gratitude.

Je présente mes profonds respects et mes reconnaissances à mon professeur et encadrant M. Georges Kaddoum pour son suivi et pour son énorme soutien, qu'il n'a cessé de me prodiguer tout au long de la période de mon mémoire.

Je remercie tous les membres de ma famille qui m'ont apporté leur soutien et leur aide au cours de ces années d'études, et plus spécialement ma chère mère et mon cher père.

J'adresse aussi mes vifs remerciements aux membres des jurys pour avoir bien voulu examiner et juger ce travail.

Je ne laisserai pas cette occasion passer, sans remercier tous mes collègues au laboratoire LACIME pour leur aide et leurs précieux conseils.

Enfin, mes remerciements à tous ceux qui ont contribué de prés ou de loin au bon déroulement de ce projet.

Conception et analyse des méthodes d'ordonnancement dans les réseaux 6TiSCH-802.15.4e à mode TSCH pour les applications industrielles de l'Internet des Objets

Taieb HAMZA

RÉSUMÉ

La conception des protocoles de la couche MAC dans un RCSF est cruciale en raison des limitations en capacités de traitement et en puissance des capteurs sans fil. La dernière version de l'IEEE 802.15.4, référencée sous le nom de IEEE 802.15.4e, a été publiée par IEEE et décrit le mécanisme du saut de canal par intervalle de temps (TSCH). Par conséquent, le groupe de travail 6TiSCH a publié un algorithme distribué permettant aux nœuds voisins de s'accorder sur un modèle de communication piloté par une fonction d'ordonnancement minimale. Une *slotframe* contient un nombre spécifique d'intervalles de temps qui se répètent et qui sont planifiés en fonction des exigences de l'application et de la topologie de routage. Cet ordonnancement permet aux nœuds capteurs de déterminer quand transmettre ou recevoir des données. Cependant, le standard IEEE 802.15.4e à mode TSCH ne définit pas les spécificités de la planification des intervalles de temps de l'ordonnancement.

Dans ce travail, nous discutons d'abord les techniques de brouillage sophistiquées qui sont appliquées aux réseaux de capteurs sans fil au niveau de la couche MAC ainsi que les contremsures prises pour s'y défendre. Ensuite, nous proposons une fonction minimale d'ordonnancement distirbuée (EMSF) qui est basée sur la fonction d'ordonnancement minimale conforme au standard 802.15.4e à mode TSCH. Pour cette raison, nous introduisons un algorithme distribué, basé sur le processus de Poisson, qui a pour objectif la prédiction des exigences de l'ordonnancement pendant la prochaine *slotframe*. En conséquence, les opérations de négociation entre les paires de nœuds pour s'accorder sur un ordonnancement seront réduites. Par conséquent, EMSF réduit considérablement la surcharge des données échangée, la latence de bout en bout et la longueur de la file d'attente. Les résultats préliminaires de la simulation ont confirmé qu'EMSF surpasse l'algorithme de MSF proposé dans le standard 802.15.4e à mode TSCH.

Mots-clés: Groupe de travail 6TiSCH, IEEE 802.15.4e à mode TSCH, Ordonancement, Processus de Poisson, Prediction.

Design and analysis of scheduling methods for 6TiSCH-802.15.4e TSCH-based networks for industrial applications of Internet of Things

Taieb HAMZA

ABSTRACT

MAC layer protocol design in a WSN is crucial due to the limitations on processing capacities and power of wireless sensors. The latest version of the IEEE 802.15.4, referenced to as IEEE 802.15.4e, was released by IEEE and outlines the mechanism of the Time Slotted Channel Hopping (TSCH). Hence, 6TiSCH working group has released a distributed algorithm for neighbour nodes to agree on a communication pattern driven by a minimal scheduling function. A slotframe contains a specific number of time slots, which are scheduled based on the application requirements and the routing topology. Sensors nodes use the schedule to determine when to transmit or to receive data. However, IEEE 802.15.4e TSCH does not address the specifics on planning time slot scheduling.

In this thesis, we first discuss intelligent MAC layer jamming attacks and the countermeasures taken in the context of WSNs. Then, we propose a distributed Enhanced Minimal Scheduling Function (EMSF) based on the minimal scheduling function, which is compliant with 802.15.4e TSCH. In this vein, we introduce a distributed algorithm based on a Poisson process to predict the following schedule requirements. Consequently, the negotiation operations between pairs of nodes to agree about the schedule will be reduced. As a result, EMSF decreases the exchanged overhead, the end-to-end latency and the packet queue length significantly. Preliminary simulation results have confirmed that EMSF outperforms the 802.15.4e TSCH MSF scheduling algorithm.

Keywords: 6TiSCH WG, IEEE 802.15.4e TSCH, Scheduling, Poisson Process, Prediction.

TABLE DES MATIÈRES

Page

INTRODUCTION 1

CHAPITRE 1 STANDARD 802.15.4E À MODE TSCH 9
1.1 Introduction 9
1.2 Standard IEEE 802.15.4 11
1.2.1 LR-WPAN 11
1.2.2 Composants du LR-WPAN 12
1.2.3 Couche physique 12
1.2.4 Limitations de 802.15.4 13
1.2.5 Attaque de brouillage au niveau de la couche MAC 802.15.4 14
1.3 Standard IEEE 802.15.4e 15
1.3.1 Améliorations fonctionnelles générales 16
1.3.2 Modes de comportement de la couche MAC 17
1.4 Time Slotted Channel Hopping TSCH 19
1.4.1 Aperçu 19
1.4.2 Structure de la *slotframe* 20
1.4.3 Saut de canal 21
1.4.4 Formation du réseau 22
1.4.5 Synchronisation de l'horloge et des nœuds 22
1.5 Ordonnancement dans les réseaux 802.15.4e TSCH 23
1.5.1 Ordonnancement centralisé 24
1.5.2 Ordonnancement distribué 26
1.5.3 Ordonnancement autonome 27
1.6 Groupe de travail IETF 6TiSCH 28
1.6.1 Architecture 29
1.6.2 Couche protocolaire 30
1.7 Conclusion 32

CHAPITRE 2 DÉVELOPPEMENT D'UNE MÉTHODE D'ORDONNANCMENT OPTIMISÉE POUR LES RÉSEAUX TSCH 33
2.1 Introduction 33
2.2 Solution proposée 33
2.2.1 Exigences 34
2.2.2 Contraintes de conception 34
2.2.3 Modèle de réseau et notations 35
2.3 Prédiction de la quantité de données 37
2.3.1 Modèle de génération de paquets à base de processus de Poisson 39
2.3.2 Formulation mathématique 39
2.3.3 Calcul de la probabilité 42
2.3.4 Ajout/Suppression des cellules (6top) 43

2.4 Algorithme d'ordonnancement 45
2.4.1 Calcul de la moyenne 46
2.4.2 Prédiction du nombre de paquets 46
2.4.3 Ajout/Suppression des cellules 47
2.4.4 Exemple d'exécution 48
2.5 Conclusion 49

CHAPITRE 3 SIMULATIONS ET RÉSULTATS 51
3.1 Introduction 51
3.2 Environnement de la simulation 51
3.3 Résultats 54
3.3.1 Taux d'erreur des messages 6p 54
3.3.2 Charge de trafic supplémentaire 56
3.3.3 Temps de latence 57
3.3.4 Taille de la file d'attente 59
3.4 Discussions 60
3.5 Conclusion 61

CONCLUSION ET RECOMMANDATIONS 63

ANNEXE I ARTICLES PUBLIÉS 65

BIBLIOGRAPHIE 78

Liste des tableaux

Tableau 3.1 Paramètres de simulation. 54

LISTE DES FIGURES

Page

Figure 0.1 Diagramme des chapitres. 7

Figure 1.1 Bandes de fréquences de 802.15.4. (Callaway *et al.* (2002)). 13

Figure 1.2 Structure d'une *slotframe* TSCH. 21

Figure 1.3 Architecture globale d'un réseau 6TiSCH. (Dujovne *et al.* (2014)). 30

Figure 1.4 Couche protocolaire de 6TiSCH. (Dujovne *et al.* (2014)). 32

Figure 2.1 Type de données dans un réseau 802.15.4e TSCH. 37

Figure 2.2 Génération des paquets au cours des *slotframes*. 40

Figure 2.3 Distribution d'un processus de Poisson. (Poulsen *et al.* (2011)). 42

Figure 2.4 Couche protocolaire dans le standard 802.15.4e. 43

Figure 2.5 Ajout/suppression des cellules. 44

Figure 2.6 Calcul de la moyenne de paquets générés. 46

Figure 2.7 Prédiction du nombre de paquets. 47

Figure 2.8 Transaction 6p pour ajout et suppression de cellules. 48

Figure 2.9 Exemple d'exécution d'EMSF. 49

Figure 3.1 Architecture de OpenWSN. (Vasiljević & Gardašević (2016)). 52

Figure 3.2 Topologie de réseau. 53

Figure 3.3 Taux d'erreur des messages 6p. 55

Figure 3.4 Charge de trafic supplémentaire. 56

Figure 3.5 Temps de latence de bout en bout. 58

Figure 3.6 Taille de la file d'attente. 59

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

6LowPAN	IPv6 Low power Wireless Personal Area Networks
6TiSCH	IPv6 over the TSCH mode of IEEE 802.15.4e
6TOP	6TiSCH Operation Sublayer Protocol
AMCA	Asynchronous Multi-Channel Adaptation
ASN	Absolute Slot Number
BBR	Backbone Routers
BE	Beacon Enabled
CFP	Contention Free Period
CoAP	Constrained Application Protocol
CPS	Cyber Physical System
CSMA-CA	Carrier Sense Multiple Access with Collision Avoidance
DODAG	Destination Oriented Directed Acyclic Graph
DSME	Deterministic and Synchronous Multi-channel Extension
EB	Enhanced Beacon
EES	Energy Efficient Scheduler
EMSF	Enhanced Minimal Scheduling Function
ETSI	European Telecommunications Standards Institute
ETX	Expected Transmission Count
FastA	Fast Association
FFD	Full Function Devices
GTS	Guaranteed Time Slots
IE	Information Element
IEEE	Institute of Electrical and Electronics Engineers

IETF	Internet Engineering Task Force
IIoT	Industrial Internet of Things
IoT	Internet of Things
IPv6	Internet Protocol version 6
ISA	Industry Standard Architecture
LE	Low-Energy
LLDN	Low Latency Deterministic Network
LQI	Link Quality Indicator
LR-WPAN	Low-Rate Wireless Personal Area Network
MAC	Medium Access Control
ME	Management Entity
MSF	Minimal Scheduling Function
N-BE	Non-Beacon Enabled
PCE	Path Computation Element
PDR	Packet Delivery Ratio
QoS	Quality of Service
RCSF	Réseau de Capteurs Sans Fil
RFD	Reduced Function Devices
ROLL	Routing Over Low-power and Lossy networks
RPL	Routing Protocol for Low-Power and Lossy Networks
RRS	Round Robin Scheduler
RSSI	Received Signal Strength Indication
S-MAC	Sensor Medium Access Control
TDMA	Time Division Multiple Access
TSCH	Time Synchronized Channel Hopping
WPAN	Wireless Personal Area Network

INTRODUCTION

L'idée d'un monde avec des systèmes équipés de capteurs intelligents interconnectés se répand à toutes les domaines de l'industrie tout en permettant aux utilisateurs de prendre des meilleures décisions. Les experts de domaine ont dénommé cette idée "l'Internet des objets (IoT)". L'IoT peut s'appliquer autant à des applications industrielles (IIoT) qu'à des applications personnelles. Parmi les applications industrielles des IIoT on peut nommer l'agriculture intelligente, les villes intelligentes, les usines intelligentes et les réseaux intelligents (*smart grids*). Les applications non-industrielles ou personnelles incluent les maisons intelligentes, les jouets connectés et les appareils de fitness mobiles.

L'Internet des objets industriel (IIoT) est un ensemble de systèmes, d'objets, de plate-formes et d'applications technologiques facilitant la communication et le partage des informations entre les IIoT, les personnes et l'environnement l'entourant. L'adoption vaste des IIoT est rendue possible par la disponibilité accrue et le faible coût des capteurs, des processeurs et d'autres technologies qui captent l'information et contrôlent son accès en temps réel.

De base, les IIoT sont des systèmes industriels interconnectés qui se communiquent pour pouvoir coordonner leurs actions et partager leurs analyses de données dans le but d'améliorer la performance et l'efficacité des usines, ainsi réduisant ou même éliminant les temps d'arrêt et les temps morts où les équipements ne sont pas utilisés. Une application classique peut être un équipement d'usine qui peut détecter des changements infimes, déterminer la probabilité de défaillance d'un composant clé, et planifier son propre maintenance pour éviter les temps d'arrêt imprévus ainsi sauvant des millions de dollars en coûts excédentaires aux entreprises adoptant les IIoT.

Avec l'adoption des ces technologies, le monde s'avance vers l'Industrie 4.0, qui emploie les IoT, les Systèmes Cyber-Physiques (CPS) et les technologies Cloud. Malgré qu'il y a de multiples architectures IoT, qui améliorent l'efficacité des communications, les exigences et le

niveau d'urgence demandent une plus grande efficacité, rapidité et fiabilité, ce qu'on appelle la qualité de service (QoS).

Pour régler ce problème, les réseaux de capteurs sans fil (RCSF) sont apparus et devenus l'une des infrastructures de réseau les plus importantes. Les RCSFs font maintenant partie des applications telles que la surveillance des phénomènes naturels (ex, les volcans et les glaciers) et la surveillance des infrastructures civiles (ex, les ponts et les routes).

Ces exigences strictes en termes de qualité de service des protocoles de communication ont été traditionnellement traitées en faisant des modifications à la norme IEEE 802.15.4 et en proposant des mécanismes externes.

Par conséquent, pour répondre à ces exigences grandissantes du domaine industriel et des systèmes émergents en matière de communication sans fil à faible consommation, à faible portée et à grande fiabilité, l'association des normes de l'institut des ingénieurs électriciens et électroniciens (IEEE-SA) a publié, à l'automne 2012, un amendement du protocole IEEE 802.15.4e. Ce dernier vise à améliorer et à étendre les fonctionnalités du protocole IEEE 802.15.4-2011.

Le mode de fonctionnement appelé TSCH (*Time Synchronized Channel Hopping*) a été conçu pour permettre aux périphériques IEEE 802.15.4e de prendre en charge un large éventail d'applications, notamment les applications industrielles. C'est une technique d'accès au support qui fait recours à la synchronisation temporelle pour obtenir un fonctionnement à faible consommation d'énergie et au saut de canal pour permettre une fiabilité élevée. Un nouveau groupe de travail, appelé 6TiSCH, vient d'être formé au sein de l'IETF (*Internet Engineering Task Force*). Ce dernier vise à lier les capacités IEEE802.15.4e à mode TSCH aux efforts et recommandations de normalisation antérieurs tels que IETF 6LoWPAN et ROLL (*Routing Over Low-power and Lossy networks*).

La norme 802.15.4e à mode TSCH ne définit pas comment construire un ordonnancement, mais elle définit seulement comment l'exécuter. L'un des principaux défis auxquels sont confrontés les administrateurs de ce type de réseau consiste à maximiser leurs profits en termes de temps de latence, vu la nature d'implémentation des réseaux installés dans un environnement industriel. Cela nécessite l'intégration d'un algorithme d'ordonnancement distribué qui s'adapte aux changements de topologie ou de charge de trafic de réseau tout en satisfaisant aux exigences de service en termes de temps de latence (crucial dans les RCSFs industriels), du charge de trafic supplémentaire et de la taille de la file d'attente.

0.1 Problématique

Le principal défi des RCSFs industriels est de minimiser le temps de latence vu la nature critique des événements physiques détectés par les noeuds capteurs. En d'autre termes, ce défi consiste à réduire le taux de paquets de contrôle échangés entre les noeuds dans l'objectif de déterminer leurs besoins en bande passante.

Le mode TSCH a été conçu pour permettre aux noeuds capteurs, implémentant le protocole 802.15.4, de prendre en charge un large éventail d'applications, y compris les applications industrielles. Ce mode consiste en une technique d'accès au support de communication en utilisant une synchronisation temporelle entre les noeuds afin d'obtenir un niveau opérationnel à faible consommation d'énergie. D'autre part, TSCH implémente le saut de fréquence pour permettre au réseau d'atteindre un niveau de fiabilité avancé.

L'amendement 802.15.4e est le dernier standard proposé par IEEE pour les RCSFs à faible consommation d'énergie. Cette norme est implémentée dans un environnement industriel ayant des exigences élevées en terme de fiabilité, de disponibilité et de sécurité. Dans cet environnement, le déploiement des capteurs en parallèle avec des équipements métalliques entraîne la dégradation du signal dûe aux interférences, ainsi bloquant l'utilisation d'un seul canal pour

la communication. Toutefois, le mode TSCH, implémenté dans le standard 802.15.4e, permet une meilleure agilité lors de l'utilisation des ressources de communication en offrant une plus grande fiabilité au réseau.

Le mode TSCH se concentre uniquement sur le fonctionnement de la couche MAC. De ce fait, le groupe de travail 6TiSCH a été formé par l'organisation de standardisation IETF pour lier les capacités IEEE802.15.4e à mode TSCH aux efforts et recommandations de normalisation antérieurs tels que 6LoWPAN et ROLL. Le groupe 6TiSCH propose une architecture basée sur des normes *open source* dans l'objectif d'atteindre des performances élevées de niveau industriel en terme de temps de latence, de fiabilité et de consommation d'énergie. Cependant, le groupe ne définit pas comment planifier et programmer l'envoi et la réception des trames entre les noeuds du réseau et durant les slots de temps qui définissent l'ordonnancement.

Ce travail vise à élaborer un algorithme d'ordonnancement qui permet de répondre aux besoins des RCSFs industriels en termes de temps de latence. En d'autres termes, il vise à réduire la quantité de trafic de contrôle échangée entre les noeuds pour déterminer leurs besoins en bande passante. La solution proposée se base sur un algorithme distribué qui permet à chaque paire de noeuds de déterminer le nombre de cellules nécessaires pour échanger leurs paquets. Selon le mode de fonctionnement du réseau, l'enchaînement et le déroulement des événements a été modélisé en un processus de Poisson. L'algorithme s'appuie sur un calcul de probabilité qui a pour objectif de prédire le nombre de cellules nécessaires pour une paire de noeuds dans une étape postérieure. Ceci va minimiser le nombre de paquets de contrôle échangés et ainsi réduire le temps de latence (Hamza & Kaddoum (2019)).

0.2 Objectif et méthodologie

L'objectif principal de ce mémoire est de présenter une solution performante permettant d'ordonnancer efficacement la communication entre les noeuds dans un réseau 802.15.4e à mode

TSCH. La solution proposée prend en considération la nature des événements imprévisibles et soudains qu'un noeud capteur pourrait détecter dans les réseaux industriels. D'autre part, l'algorithme, défini dans ce mémoire, permet aux noeuds de déterminer leur ordonnancement d'une manière distribuée, ce qui permet d'éviter considérablement la surcharge des paquets générés et réduire, par conséquent, le temps de latence.

Afin d'atteindre cet objectif, nous allons procéder comme suit :

- Nous allons commencer par décrire les concepts de base du standard 802.15.4 ainsi que les amendements apportés qui forment le nouveau standard 802.15.4e. Ensuite, nous allons décrire le mode de fonctionnement TSCH ainsi que l'effort d'intégration, réalisé par IETF, afin d'associer ce mode avec une couche protocolaire introduisant les fonctionnalités d'IPv6. Après cela, nous allons présenter les techniques d'ordonnancement les plus citées dans la littérature.
- Nous allons ensuite formuler la problématique en un modèle mathématique qui décrit le mode d'échange des paquets dans un réseau 802.15.4e-TSCH. Ensuite, nous allons proposer une solution qui a pour objectif de définir un ordonnancement distribué avec un temps de latence réduit.
- Finalement, nous allons réaliser des simulations pour tester notre solution et nous allons comparer les résultats trouvés avec l'algorithme MSF qui est défini par défaut dans le standard 802.15.4e-TSCH. Les paramètres d'évaluation que nous allons utiliser sont : le taux d'erreurs de messages 6p, la charge de trafic supplémentaire, la taille de la file d'attente et le temps de latence.

0.3 Publications

- Le travail présenté dans ce mémoire a été accepté pour la publication dans la conférence " IEEE Wireless Communications and Networking Conference (WCNC 2019, Marrakech) "

sous le nom de " Enhanced Minimal Scheduling Function for IEEE802.15.4e TSCH Networks ".

- Une étude bibliographique a été faite aussi sur les attaques de brouillage au niveau de la couche MAC et a été objet d'une publication dans " IEEE Vehicular Technology Conference (VTC-Fall 2016, Montréal) " sous le nom de " A Survey on Intelligent MAC Layer Jamming Attacks and Countermeasures in WSNs ".

0.4 Plan du mémoire

Ce mémoire est organisé comme suit : Le premier chapitre définit les notions de base et le mode fonctionnement des RCSFs 802.15.4e à mode TSCH. Nous détaillons aussi les modes d'ordonnancement les plus cités dans la littérature.

Le deuxième chapitre est une formulation mathématique du problématique suivie par une méthode d'ordonnancement comme solution à la problématique.

Le troisième chapitre présente les simulations ainsi que les résultats obtenus. Chaque paramètre d'évaluation étudié a été comparé avec la méthode d'ordonnancement implémentée par défaut dans les réseaux 802.15.4e-TSCH.

La figure 0.1 est une représentation graphique du plan de ce mémoire.

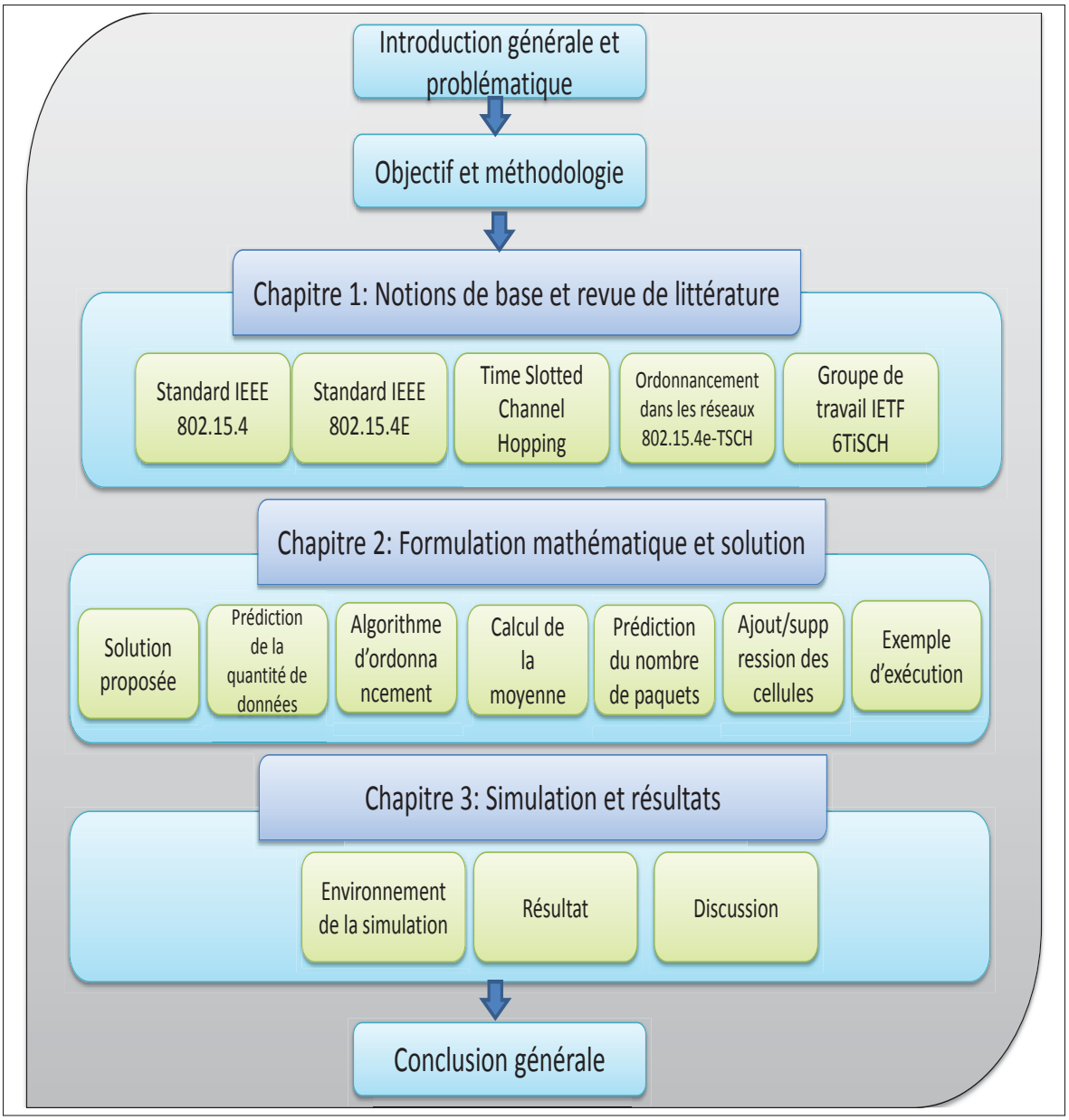


Figure 0.1 Diagramme des chapitres.

CHAPITRE 1

STANDARD 802.15.4E À MODE TSCH

1.1 Introduction

L'adoption à grande échelle d'appareils embarqués à faible consommation permet l'interconnexion de tous les objets intelligents à l'Internet, mettant l'attention sur l'Internet des Objets. Selon Cisco, il y aurait 50 milliards d'objets IoT d'ici 2020, alors que Huawei estime qu'il y en aurait plus de 100 milliards d'ici 2025 (Huawei-Technologies (2015)). L'adoption des objets IoT va aider les villes intelligentes à réduire le trafic routier, à économiser de l'énergie, et à réduire la pollution (Zanella *et al.* (2014)).

Alors que les premiers utilisateurs achetaient des objets IoT pour le loisir, les applications modernes sont plus exigeantes en termes de fiabilité de communication. En effet, les transmissions radio sont en train de remplacer les connexions par câble. Cependant, les nœuds capteurs d'un réseau doivent maintenant interagir en temps réel, donc on s'attend à une interconnexion fiable et ponctuelle entre les objets.

Les experts du domaine prévoient qu'il y aurait une adoption très répandue de l'Internet des Objets Industriels (IIoT) dans plusieurs domaines clés. Par exemple, l'agriculture intelligente permet d'exploiter une infrastructure radio en temps réel pour observer une serre ou un champ (Ye *et al.* (2013)). L'Industrie 4.0 s'attend à utiliser l'Internet des Objets pour rendre la chaîne industrielle plus flexible (Hermann *et al.* (2016)). Son objectif est de transformer tous les dispositifs de la chaîne d'approvisionnement et de fabrication en dispositifs radio autonomes. L'intégration de nombreux capteurs et actionneurs radio dans l'automatisation des maisons intelligentes permet de réduire leur consommation d'énergie (Khajenasiri *et al.* (2017)). L'ETSI (*European Telecommunications Standards Institute*) a détaillé les exigences en termes de délai, de fiabilité et de volume de trafic pour différentes applications dans les villes intelligentes (ETSI). À cette fin, des systèmes en temps réel sont nécessaires, englobant le système d'exploitation, l'application et les protocoles de communication.

Les piles protocolaires pour les réseaux radio à faible puissance tentent de mettre en œuvre des approches de service à court cycle : un nœud doit éteindre son antenne radio la plupart du temps pour économiser de l'énergie. La couche contrôle d'accès au support (MAC) est chargée de décider le temps auquel un nœud est autorisé à transmettre afin d'éviter les collisions (deux transmissions s'imbriquent) et d'alerter quand le noeud récepteur n'est pas en écoute. S-MAC a été l'un des premiers algorithmes à programmer des transmissions au niveau de la couche liaison (Ye *et al.* (2002)). Dans ce protocole, les nœuds capteurs voisins doivent envoyer leurs temps de réveil précis prévu afin de pouvoir échanger des paquets. Plus tard, les approches d'échantillonnage à préambule ont proposé de réduire les surcharges de trafic généraux en obligeant les noeuds émetteurs à annoncer, via un préambule, leurs prochains temps de transmission (Polastre *et al.* (2004)). Lorsqu'un nœud détecte un préambule, il doit rester éveillé pour recevoir la prochaine trame. Cependant, ces approches doivent lutter contre le problème de terminal caché et reposent sur les méthodes classiques d'accès par contention, étant donc incapables de fournir des garanties strictes pour l'accès au moyens de communication.

L'industrie a ensuite poussé l'interopérabilité en créant des standards. La norme IEEE 802.15.4-2006 (IEEE (2006)) a proposé une approche synchronisée dans le temps, combinant un accès aléatoire durant la première partie et des timeslots dédiés pour les transmissions en temps réel au cours de la dernière partie. Néanmoins, un nombre important de collisions continuent de se produire pendant la partie d'accès aléatoire, et ont un impact très négatif sur la latence et sur la fiabilité (Abdeddaim *et al.* (2013)). Plus récemment, les amendements apportés à cette norme (IEEE (2018)) ont été axés sur l'amélioration de la fiabilité et de l'efficacité énergétique. Ces approches reposent pour la plupart sur un Accès Multiple par Répartition dans le Temps (TDMA) associé à un mécanisme de synchronisation.

En outre, il a été prouvé que le saut de canal lent permet de lutter efficacement contre le bruit en bande étroite, chose très courante dans les environnements industriels (Watteyne *et al.* (2009)). Une trame est transmise via un seul canal physique, mais les autres paquets (et les retransmissions possibles) utilisent un canal différent, suivant une séquence pseudo-aléatoire. Bien que ces protocoles MAC se concentrent sur une fiabilité élevée et une faible latence, ils requièrent

tous des transmissions ordonnancées. Les transmetteurs brouilleurs ne devraient pas être autorisés à émettre simultanément pour éliminer les collisions. De même, la planification doit être correctement mesurée afin d'optimiser la latence de bout en bout, qui est un indicateur de performance essentiel pour la plupart des réseaux industriels (Gaillard *et al.* (2014)).

1.2 Standard IEEE 802.15.4

Il existe plusieurs protocoles de communication sans fil qui supportent différents types d'applications, telles que les communications vidéo, vocales et data. Chacun de ces protocoles met un compromis entre les propriétés telles que le débit, la latence, l'efficacité énergétique et la couverture radio visant des scénarios d'application bien définis. Les réseaux de capteurs sans fil n'imposent généralement pas d'exigences strictes en termes de bande passante, mais ils requièrent une consommation d'énergie minimale afin de prolonger la durée de vie du réseau dans son ensemble. Il est important de répondre aux exigences de QoS telles que l'efficacité énergétique et la rapidité d'exécution afin d'atteindre les objectifs principaux des protocoles et des technologies des RCSFs.

1.2.1 LR-WPAN

Au cours de la dernière décennie, plusieurs normes visant les communications sans fil à faible puissance ont été définis pour répondre aux besoins en matière de qualité de service des communications industrielles (Lu *et al.* (2002), van Dam & Langendoen (2003)). IEEE 802.15.4 (IEEE (2003)) est l'une de ces normes répandues qui a été publiée pour la première fois en 2003 pour les WPAN (Wireless Personal Area Networks).

Le protocole ne définit que la couche physique et la couche d'accès au canal de communication alors que quelques propositions, telles que les protocoles ZigBee (ZigBee-Alliance (2005)) ou RPL (Winter *et al.* (2012)), ont été établis pour compléter la pile protocolaire de communications.

1.2.2 Composants du LR-WPAN

Dans la norme IEEE 802.15.4 (IEEE (2006)), les périphériques peuvent être classés en dispositifs entièrement fonctionnels (FFD) et dispositifs à fonctions réduites (RFD) :

- Les noeuds routeurs FFD (Full Function Devices) pour transférer des données via un routage multi-saut. Ces noeuds coordonnent également l'ensemble d'autres fonctionnalités du réseau.
- Les noeuds terminaux RFD (Reduced Function Devices) qui impliquent une pile de protocoles légers et économiques.

Le coordinateur PAN est un FFD qui agit en tant que contrôleur principal auquel d'autres périphériques peuvent être associés. Il est responsable de la synchronisation du temps dans tout le réseau. Parfois, un FFD peut également agir en tant que coordinateur fournissant des services de synchronisation et d'acheminement locaux à ses voisins. Chaque coordinateur doit être associé à un coordinateur PAN et celui-ci forme son propre réseau s'il ne trouve pas d'autres réseaux à proximité. Le dispositif à fonction réduite (RFD) est généralement le noeud final d'un réseau IEEE 802.15.4. Un RFD est destiné à des applications extrêmement simples, telles qu'un commutateur de lumière ou un capteur infrarouge passif, qui sont généralement synchronisées avec un coordinateur et qui n'ont pas de fonctionnalités de routage.

1.2.3 Couche physique

Comme le montre la Figure 1.1, le protocole IEEE 802.15.4 fonctionne dans trois bandes de fréquences différentes : 2,4 GHz (avec 16 canaux), 915 MHz (avec 10 canaux) et 868 MHz (avec un seul canal) (Callaway *et al.* (2002)). Le débit de transmission de données varie également en fonction des bandes utilisées. La bande 2,4 GHz permet d'atteindre un débit de 250 Kbps. Les bandes de fréquences 915 MHz et 868 MHz peuvent garantir des débits allant jusqu'à 40 et 20 Kbps respectivement. La couche physique est responsable de l'activation et de la désactivation de l'émetteur-récepteur radio, de la mesure de la qualité de liaison, de l'éva-

luation et de la sélection du canal avec prévention de collision. Tandis que la *Contention Free Period* (CFP) est équipée de 7 *Guaranteed Time Slots* (GTS), qui sont utilisés par les nœuds nécessitant une bande passante garantie.

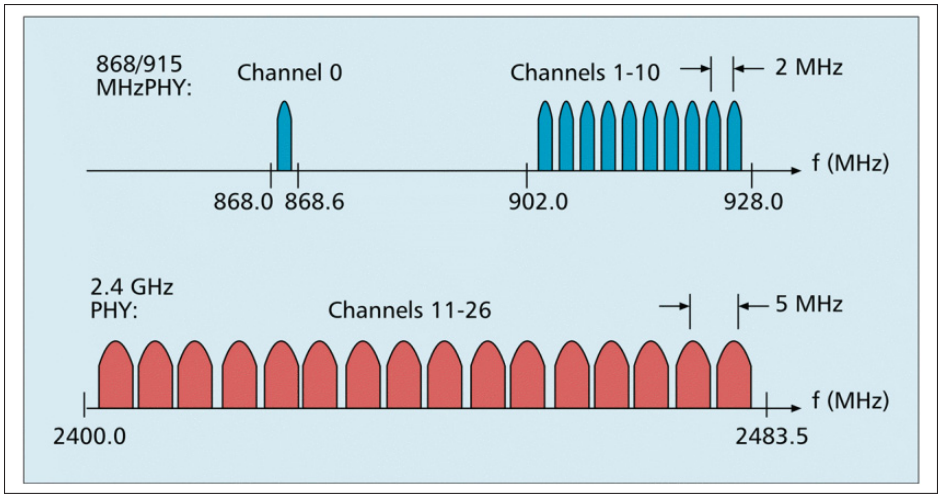


Figure 1.1 Bandes de fréquences de 802.15.4. (Callaway *et al.* (2002)).

1.2.4 Limitations de 802.15.4

Les performances du protocole MAC 802.15.4, à la fois en mode *Beacon Enabled* (BE) et en mode *Non-Beacon Enabled* (NBE), ont été minutieusement étudiées dans le passé (Daidone *et al.* (2014), Guglielmo *et al.* (2016), Yazdi *et al.* (2014)). En conséquence, un certain nombre de limitations et d'anomalies ont été identifiées, dont on peut citer :

- Délai illimité : Étant donné que le protocole MAC 802.15.4, en modes BE et NBE, repose sur un algorithme CSMA-CA (*Carrier Sense Multiple Access with Collision Avoidance*), aucun délai de temps maximal a été fixé afin que les données atteignent leurs destinations finales.
- Fiabilité de communication limitée : Le protocole MAC 802.15.4 en mode BE a un taux de livraison très faible, même lorsque le nombre de nœuds n'est pas très élevé. Ceci est principalement dû à l'inefficacité de l'algorithme CSMA-CA utilisé pour l'accès au canal.

Un comportement similaire peut également se produire dans le mode NBE lorsqu'un grand nombre de nœuds commencent à émettre simultanément des paquets (par exemple, lors de la détection d'un événement).

- Pas de protection contre les interférences et le brouillage multi-chemin : Ce sont des phénomènes très courants dans les réseaux de capteurs sans fil. Contrairement aux autres technologies de réseau sans fil telles que Bluetooth (Bluetooth), ISA 100.11a (ISA) et WirelessHART (HART), le protocole MAC 802.15.4 utilise un canal unique et ne possède pas de mécanisme de saut de fréquence intégré pour atténuer les effets négatifs des interférences et de brouillage multi-chemin. Par conséquent, le réseau est sujet à de fréquentes instabilités et peut également s'effondrer.
- Nœuds de relais alimentés : La norme 802.15.4 prend en charge les topologies à un seul saut (en étoile) et à plusieurs sauts (pair à pair). Le mode BE peut être utilisé pour former un *Personal Area Network* (PAN) multi-saut avec une topologie en arbre où les nœuds intermédiaires n'ont pas besoin de garder l'état actif en permanence. Toutefois, la définition de topologies à saut multiples en mode 802.15.4 BE requiert des mécanismes complexes de synchronisation et de planification des beacons non spécifiés par la norme (Yeh & Pan (2014)). Pour surmonter ces limitations, dans de nombreuses applications, les nœuds relais intermédiaires des réseaux multi-saut 802.15.4 gardent leur radio allumée en permanence, entraînant une consommation d'énergie importante.

Pour ces raisons, la norme 802.15.4 ne convient pas à de nombreux scénarios critiques, dans lesquels les applications ont des exigences strictes en termes de rapidité et de fiabilité.

1.2.5 Attaque de brouillage au niveau de la couche MAC 802.15.4

Les réseaux de capteurs sans fil (WSN) constituent la majeure partie de l'IoT (Mainetti *et al.* (2011)). Ils sont utilisés dans de nombreux domaines tels que la santé, l'agriculture, l'environnement et la détection des catastrophes naturelles. Les caractéristiques fondamentales des RCSFs les rendent vulnérable à des attaques en raison de la nature de fonctionnement de ses

composants (communication sans fil). Cela les expose à des attaques passives et actives, qui varient de leur nature et de leurs objectifs. Les communications sans fil sont exposées à l'écoute et à l'interception des signaux. Le brouillage radio est l'une des attaques qui peuvent se produire dans presque tous les environnements (Mpitziopoulos *et al.* (2009)).

Le protocole MAC, proposé par le standard 802.15.4, est vulnérable aux attaques de brouillage intelligents. De ce fait, quelques améliorations ont été proposées dans la littérature afin d'optimiser son fonctionnement. Toutefois, certains de ces protocoles sont incapables de livrer aucun paquet dans le cas où ils sont exposés à un brouilleur. Dans le cas de réseaux à grande échelle avec des contraintes d'énergie élevées, les nœuds sont programmés pour communiquer et se mettre en veille simultanément, comme dans le cas des protocoles basés sur TDMA (Althobaiti & Abdullah (2015)). L'objectif de ces protocoles est d'éviter les écoute attentive et en état de repos, qui constituent les principales sources de perte d'énergie dans les RCSFs. Cependant, un tel ordre temporel peut introduire un modèle de communication qui permet à l'attaquant de prédire le prochain cycle de communication. Ceci peut être exploité par un nœud malveillant afin de lancer une attaque à efficacité énergétique (Wood *et al.* (2007)). En conséquence, les signaux bloqués coïncident avec les paquets envoyés par des nœuds de réseau légitimes. Le nœud malveillant pourra exploiter le modèle temporel de communication et bloquer l'envoi de paquets légitimes, tout en transmettant des pulsions de brouillage réduites. Par conséquent, il atteint la même efficacité énergétique que les nœuds du réseau et devient plus difficile à repérer (Hamza *et al.* (2016)).

1.3 Standard IEEE 802.15.4e

Le standard IEEE 802.15.4 est la référence pour les réseaux de capteurs sans fil RCSFs. Il définit le fonctionnement des réseaux locaux sans fil à faible débit (LR-WPAN) et spécifie la couche physique et la couche contrôle d'accès au support (IEEE (2006)). Ces réseaux sont bien connus pour leur facilité d'installation, leur coût extrêmement faible, leur fonctionnement à courte portée, leur transfert fiable des données et leur autonomie raisonnable, tout en maintenant une pile de protocoles simple et flexible. Plusieurs études ont analysé les performances du

standard IEEE 802.15.4 dans les RCSFs. Néanmoins, de nombreuses limitations et faiblesses ont été identifiées, qui rendent ce standard inadéquat aux applications critiques. Ces derniers, fonctionnant généralement dans des environnements difficiles, ont des exigences strictes en termes de fiabilité, de latence, d'efficacité énergétique et d'évolutivité. Parmi ces limitations on peut citer le manque de fiabilité, la latence illimitée, une technique de saut de fréquence intégrée et une mauvaise gestion d'énergie.

Ces imperfections rendent le standard IEEE 802.15.4 inapproprié à de nombreuses applications, surtout lorsque ces applications sont très exigeantes en termes de fiabilité et de latence. Pour surmonter ces limitations, un groupe de travail nommé 802.15 *Task Group* 4e a été créé en 2008 pour améliorer et ajouter des fonctionnalités au code de la norme 802.15.4. En 2012, le Conseil d'Association des Standards de l'IEEE a approuvé la norme IEEE 802.15.4e (IEEE802.15.4e (2012)) en tant qu'un amendement à la norme IEEE 802.15.4 afin de mieux soutenir les divers domaines d'application industriels. La technologie de base est inspirée des technologies de réseau industrielles antérieures telles que WirelessHart (HART) et ISA 100.11.a (ISA). La norme améliorée fournit de nouvelles fonctionnalités telles qu'une faible consommation d'énergie, des éléments d'information, des balises améliorées, des métriques de performance MAC et une association rapide.

1.3.1 Améliorations fonctionnelles générales

La norme 802.15.4e étend la norme 802.15.4 précédente en introduisant des modes de comportement MAC, c'est-à-dire des nouveaux protocoles MAC conçus pour prendre en charge des domaines d'application spécifiques. Elle apporte aussi des améliorations MAC et des mécanismes fonctionnelles générales non liées à un domaine d'application en particulier.

Le standard IEEE 802.15.4e introduit les améliorations fonctionnelles générales suivantes :

- Énergie de bas niveau (LE) : Ce mécanisme est destiné aux applications où l'efficacité énergétique est prioritaire à la latence. Cela permet à un nœud de fonctionner à un très faible cycle (1% ou moins). Ce mécanisme est important dans le contexte de l'Internet des

Objets, car les protocoles ont été conçus en supposant que les nœuds capteurs sont toujours actifs.

- Les éléments d'information (IE) : C'est un mécanisme extensible pour échanger des informations dans la sous-couche MAC.
- Les balises améliorées (EB) : Ce sont une extension des trames de balises 802.15.4 qui permettent d'offrir une plus grande flexibilité pour le protocole MAC. Ils permettent de créer des trames spécifiques à l'application, en incluant les IE pertinents.
- Les trames multi-usage : Ils sont basées sur les IE et fournissent un format de trame flexible pouvant traiter un certain nombre d'opérations MAC.
- La mesure de performance MAC : C'est un mécanisme qui permet de fournir à la couche réseau et aux couches supérieures des informations sur la qualité des canaux afin de prendre des décisions appropriées. Par exemple, le protocole IP peut implémenter une fragmentation dynamique des datagrammes en fonction de l'état du canal.
- Association rapide (FastA) : La procédure d'association 802.15.4 introduit un délai important afin d'économiser de l'énergie. Pour les applications critiques, le temps de latence est prioritaire par rapport à l'efficacité énergétique. Par conséquent, le mécanisme FastA permet à un nœud de s'associer dans un laps de temps réduit.

1.3.2 Modes de comportement de la couche MAC

Le standard IEEE 802.15.4e définit cinq nouveaux modes de comportement MAC. Chacun prend en charge des domaines d'application spécifiques :

- Time Slotted Channel Hopping (TSCH) : Il cible des domaines d'application tels que l'automatisation industrielle et le contrôle des processus et offre une prise en charge des communications multi-sauts et multicanaux via l'approche TDMA.
- Deterministic and Synchronous Multi-channel Extension (DSME) : Son objectif est de prendre en charge les applications industrielles et commerciales avec des exigences strictes en termes de rapidité et de fiabilité. À cette fin, DSME combine un accès au support par

division et par répartition dans le temps et propose deux modes différents de diversité de canaux. Ce mode de comportement est conçu spécifiquement pour les réseaux maillés et multi-sauts.

- Low Latency Deterministic Network (LLDN) : Il est conçu pour les réseaux à un seul canal et à un seul saut. Il est destiné à l'automatisation dans l'industrie qui requiert une latence très faible.
- Asynchronous Multi-Channel Adaptation (AMCA) : Il est destiné aux domaines d'application nécessitant de vastes déploiements, tels que les réseaux intelligents, les réseaux de surveillance et les réseaux de contrôle des processus. Dans ces réseaux, l'utilisation d'un seul canal de communication ne permet pas de connecter tous les nœuds d'un même réseau dans le même PAN. De plus, la variance de la qualité des canaux est en général large. Une asymétrie de liens peut se produire entre deux nœuds voisins, c'est-à-dire un nœud peut seulement être en mesure de transmettre à un noeud voisin (et non recevoir). Le mode AMCA repose sur une adaptation multi-canal asynchrone et ne peut être utilisée que sur des *Non-Beacon-Enabled* PANs. Dans un réseau AMCA, chaque noeud choisit le canal d'écoute avec la meilleure qualité et commence à écouter sur cette fréquence. Dès qu'il y a deux nœuds qui doivent échanger des paquets, le noeud émetteur bascule vers le canal d'écoute désigné du noeud récepteur, de manière totalement asynchrone. Après avoir transmis le paquet de données, le transmetteur revient sur son propre canal et continue à écouter. Les nœuds peuvent échanger des informations sur les canaux d'écoute désignés en exigeant la transmission de balises aux coordinateurs ou en envoyant des paquets *Hello* spéciaux.
- Radio Frequency Identification Blink (BLINK) : Ce mode est destiné aux domaines d'application tels que l'identification des articles ou de personnes, la localisation et le suivi. Plus précisément, il permet à un nœud de communiquer son ID à d'autres nœuds sans association préalable ni accusé de réception. Les paquets BLINK sont généralement envoyés par des périphériques "*transmit only*" via le protocole Aloha.

1.4 Time Slotted Channel Hopping TSCH

Le mode TSCH est principalement conçu pour ordonnancer la communication de données aux nœuds du réseau ainsi que leurs liens respectifs. La communication entre les noeuds se déroule en suivant un ordonnancement (*schedule*), ainsi les noeuds voisins, dont les transmissions peuvent s'interférer, ne seront pas planifiés à transmettre sur le même slot de temps (*slot offset*) et décalage de canal (*Channel Offset*). L'ordonnancement est projeté sous forme de matrice composée du *slot offset* et du *channel offset*, où chaque cellule représente un lien spécifique et peut être réservée pour un seul lien ou être partagée entre de multiples liens. En cas de collision de ces dernières, un système appelé *backoff* est défini par le protocole MAC IEEE 802.15.4e dans le but de résoudre ce conflit. De plus, pour les slots partagés, l'algorithme CSMA/CA slotté est utilisé par les noeuds du PAN-TSCH (Accettura *et al.* (2012), IEEE802.15.4e (2012)). Le standard IEEE 802.15.4e à mode TSCH décrit seulement le mode d'exécution de la couche MAC, mais ne spécifie pas la façon de maintenir, construire et mettre à jour l'ordonnancement ni de s'adapter aux containtes de trafic du réseau (Chang *et al.* (2015)).

1.4.1 Aperçu

Le comportement TSCH de la couche MAC permet de garantir une grande fiabilité de fonctionnement et permet aussi de gérer les applications critiques dans le temps. Le mode TSCH est idéal pour la mise en œuvre des réseaux de capteurs-actionneurs dans les industries du pétrole et du gaz, où la sécurité est d'une importance critique (Petersen *et al.* (2007)), étant donné qu'une défaillance de sécurité peut mettre en risque le public et l'environnement. Ces types d'industries sont vulnérables aux interférences qui affectent le fonctionnement des périphériques sans fil. Le mode TSCH supporte le mécanisme de saut de fréquence, ce qui améliore considérablement la fiabilité du réseau en atténuant les effets des interférences et du brouillage multi-chemin sur une grande échelle. Les liens de communication ordonnancés dans le temps réduisent considérablement les collisions indésirables, qui causent parfois des défaillances catastrophiques.

Les centres de traitement de données sont également sujets aux collisions, car le RCSF doit prendre en charge des capteurs denses qui sont étroitement associés à un trafic de réseau élevé (Pereira *et al.* (2012)). Par conséquent, les risques de collision dans le réseau augmentent considérablement. Le mode TSCH maintient des contraintes de temps strictes qui implémentent des intervalles de temps à longueur fixe et un accès multi-canal. Ceci le rend efficace dans les réseaux denses. En outre, TSCH utilise des trames basées sur TDMA, ce qui permet ainsi des transmissions sans collision.

1.4.2 Structure de la *slotframe*

En mode TSCH, les noeuds se synchronisent sur une trame périodique (appelé *slotframe*) composée d'un certain nombre d'intervalles de temps (appelées *timeslots*). Une *slotframe* est un enchaînement d'un ensemble de *timeslots*. Les communications dans chaque *timeslot* peuvent être basées sur la contention (c'est-à-dire en utilisant CSMA-CA) ou la non-contention. La durée de chaque *timeslot* permet de transmettre un paquet et de recevoir un accusé de réception. La taille d'une *slotframe* est définie par le nombre de *timeslots* dans la *slotframe*. Chaque *slotframe* se répète de façon cyclique, formant ainsi un ordonnancement de communication.

Chaque nœud obtient des informations sur la synchronisation, les sauts de canal, le *timeslot* et la *slotframe* à partir des *Enhanced Beacons* (EB) qui sont envoyées périodiquement par d'autres nœuds afin d'introduire le réseau. Lorsqu'un nœud reçoit un EB valide, il se synchronise sur le réseau, initialise la *slotframe* et envoie ses propres balises. À partir de ce moment, la *slotframe* se répète automatiquement selon la notion du temps partagée par les nœuds et ne requiert pas de balises pour initier les communications.

La figure 1.2 montre une *slotframe* composée de 3 *timeslots*. Dans cet exemple, la *slotframe* se répète à tous les trois *timeslots*. Prenons 3 noeuds, qu'on dénote A, B, C. Dans le *timeslot* 0, le noeud A transmet ses données au noeud B. Pendant le *timeslot* 1, B transmet à C. Finalement durant le *timeslot* 2, les noeuds demeurent à l'état inactif. Cela se répète dans le même ordre pour tous les trois *timeslots*. Chaque *timeslot* dans un TSCH-PAN possède un numéro appelé

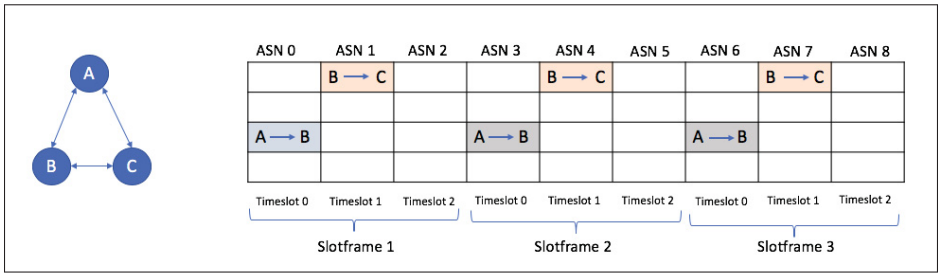


Figure 1.2 Structure d'une *slotframe* TSCH.

ASN (*Absolute Slot Number*), qui s'incrémente globalement et qui sert à calculer la fréquence dans laquelle une paire de nœuds se communiquent.

1.4.3 Saut de canal

La communication multi-canal de mode TSCH repose entièrement sur les sauts de canal. Le mode TSCH peut utiliser jusqu'à 16 canaux de communication définis par un décalage de canal (appelé *channel offset*). Dans le mode TSCH, le lien de communication entre une paire de nœuds est défini par le couple [n, *channel offset*]. Il s'agit d'une affectation de la communication par paire des *timeslots* «n» et de *channel offset* consécutifs. La fréquence utilisée pour la communication peut être définie par la fonction f.

$$f = F\{(ch_{\text{offset}} + \text{ASN}) \bmod N_{\text{ch}}\}. \tag{1.1}$$

Dans l'équation (1.1), N_{ch} est défini comme étant le nombre de canaux utilisés pour le réseau actuel, vu qu'utiliser tous les 16 canaux n'est pas obligatoire. Certains canaux peuvent ne pas être utilisés si on cherche à améliorer l'efficacité énergétique ou si leur qualité s'est détériorée. Comme mentionné précédemment, l'ASN aide à déterminer le nombre de *timeslots* qui se sont écoulées depuis que le noeud a rejoint le réseau. La fonction F peut être définie comme une table de correspondance. À partir de l'équation (1.1), il est à noter qu'un canal différent (*Ncanal*) peut être implémenté avec le même *offset* pour un ASN incrémenté, c'est-à-dire le mécanisme de saut de canal peut être utilisé avec une fréquence différente sur le même lien.

1.4.4 Formation du réseau

Lors de la formation d'un réseau, le coordinateur PAN commence à diffuser un *Enhanced Beacon* (EB) comme réponse à une demande de MLME-BEACON provenant d'une couche supérieure. Cette action est appelée *Advertising*. Les périphériques souhaitant se connecter au coordinateur PAN doivent se trouver dans sa portée de diffusion. Le *Enhanced Beacon* contient des informations sur le temps, le saut de canal, le *timeslot* et les liens initiaux. Les informations sur le temps fournissent la période spécifique pendant laquelle les nœuds doivent se synchroniser au réseau. Les informations sur les *timeslots* décrivent le temps durant lequel les données sont transmises. Enfin, les informations sur les liens initiaux donnent le temps pendant lequel il faut écouter ou transmettre au noeud publicitaire (*Advertising*).

Après avoir reçu une demande MLME-SCAN à partir d'une couche supérieure, un périphérique souhaitant se rejoindre au réseau effectue une analyse active ou passive. Suite à la réception d'une notification MLME-BEACONNOTIFY, la couche supérieure initialise la *slotframe* et les informations des liens initiaux disponibles dans le *Enhanced Beacon* (EB). Une fois le noeud synchronisé au réseau, la couche supérieure programme le noeud en mode TSCH en émettant une demande TSCH MODE.request.Association.

1.4.5 Synchronisation de l'horloge et des nœuds

Dans un réseau basé sur le mode TSCH, le temps se propage à partir du coordinateur PAN. Un nœud capteur en communication doit synchroniser son horloge à des intervalles de temps réguliers avec un autre nœud se trouvant à sa proximité. En utilisant le périphérique voisin comme référence temporelle, tous les nœuds synchronisés doivent avoir une idée préalable du début et de la fin de la *slotframe*.

Dans un réseau basé sur les *slotframes*, la synchronisation entre les noeuds permet de garantir la connexion avec les nœuds voisins, qui gardent toujours une trace de ces derniers. S'ils ne reçoivent pas d'interaction à partir d'un noeud au moins une fois par période de *keep-alive*,

ils envoient une trame vide d'accusé de réception pour effectuer une synchronisation basée sur l'accusé de réception.

La synchronisation s'effectue lorsqu'un périphérique échange une trame avec un noeud voisin. Il existe deux méthodes pour prendre en charge cette opération : soit à travers les informations sur le temps, qui sont transportés dans un accusé de réception provenant du destinataire, soit par l'information du temps d'arrivée d'une trame venant du noeud voisin.

Dans la méthode basée sur l'accusé de réception, la synchronisation s'effectue via un échange de trames de données et d'accusés de réception. Le récepteur calcule la différence entre le temps d'arrivée prévu et le temps d'arrivée réel. Ensuite, une modification du temps est envoyée au noeud émetteur via un accusé de réception.

Dans la synchronisation par trame, les nœuds se synchronisent à l'horloge du réseau à chaque fois qu'ils reçoivent une trame de données à partir d'un noeud voisin. Le récepteur calcule la différence de temps entre l'arrivée prévue et l'arrivée réelle et utilise ces informations pour régler l'horloge du réseau.

1.5 Ordonnancement dans les réseaux 802.15.4e TSCH

Le problème d'ordonnancement a déjà suscité l'intérêt pour la recherche sur les réseaux TDMA comme indiqué dans la littérature (Ergen & Varaiya (2010)). Malgré cela, la plupart des systèmes d'ordonnancement multi-canaux existants ne conviennent pas aux réseaux TSCH. Ils n'ont pas été conçus pour des nœuds aux capacitées limitées, n'autorisent pas le saut de canal par paquet et ne sont pas efficaces en termes d'utilisation de canal. Les chercheurs ont donc conçu de nouveaux algorithmes d'ordonnancement conçus pour les réseaux TSCH. Il existe des différentes approches, qui pourraient être utilisées pour établir l'ordonnancement.

Le standard IEEE 802.15.4-2015 en mode TSCH autorise les couches supérieures à créer des délais que, tous les noeuds doivent respecter. Cela permet aux noeuds de se communiquer en multi-saut, ainsi rendant le flux d'information du point initial jusqu'au point de collecte

plus facile et rapide. Un noeud implémente son ordonnancement globalement et localement en allouant des cellules à chaque flux respectif pour aider au partage des cellules contenant l'information entre les clients. Les ordonnancements peuvent être classifiés en tant que centraux, distribués et autonomes.

1.5.1 Ordonnancement centralisé

Dans une approche centralisée, un seul nœud coordinnateur est responsable de la planification et de la construction de toute la communication, ainsi que du maintien de l'ordonnancement du réseau. Le planificateur peut être un ordinateur centralisé appelé *Path Computation Element* (PCE) (Farrel (2006)). Nous allons décrire ici des solutions d'ordonnancement centralisées les plus importantes proposées pour les réseaux TSCH.

Palattella et al. ont proposé l'algorithme TASA (*Traffic Aware Scheduling Algorithm*), un algorithme d'ordonnancement centralisé pour les réseaux IEEE 802.15.4e à mode TSCH (Palattella *et al.* (2012)). Cette solution considère une topologie de réseau en arborescence et se concentre sur un scénario de *convergecast*, où différentes quantités de données doivent être fournies au noeud coordinateur. L'objectif principal de TASA est de créer le meilleur ordonnancement, en minimisant le nombre de slots requises pour transmettre toutes les données au coordinateur. Cet ordonnancement peut être obtenu par le processus de *matching and coloring*. À chaque itération, la méthode TASA implémente l'algorithme *matching* pour sélectionner un ensemble de liens éligibles pour ordonnancer dans le même *timeslot*. Ensuite, un algorithme de coloration de vertex assigne les différents *channel offsets* pour chaque lien sélectionné dans le processus précédent. En outre, les auteurs ont constaté qu'utiliser davantage de canaux pouvait améliorer le débit du réseau, réduire le temps de latence et réduire considérablement la consommation d'énergie.

Les auteurs de (Soua *et al.* (2012)) ont conçu la méthode MODESA (*Multi-channel Optimized Delay Time Slot Assignment*). À l'inverse de TASA, MODESA cible des conditions de trafic homogènes où tous les nœuds génèrent le même nombre de paquets. Dans TASA, un ordonnan-

cement sans conflit est construit à l'aide d'une procédure itérative. À chaque itération, TASA sélectionne un certain nombre de liens et organise leurs transmissions dans le même *timeslot*, en utilisant plusieurs *channel offsets*, si nécessaire. La méthode MODESA sélectionne un seul nœud et choisit un lien unique pour prendre en charge l'une de ses transmissions obligatoires. De plus, MODESA réduit la congestion des files d'attentes en ordonnançant d'abord les nœuds qui ont plus de paquets dans leur *buffer*, alors que TASA ne prend pas en compte la congestion dans les files d'attente. Dans (Soua *et al.* (2013)), Soua R et al. ont amélioré et élaboré cette approche pour prendre en charge le trafic hétérogène, ainsi que les coordinateurs multiples.

Jin et al. ont proposé une méthode d'ordonnancement adaptative, centralisée et multi-saut (AMUS) qui est basée sur le mode TSCH (Jin *et al.* (2016)). Les auteurs ont implémenté leur solution au niveau de l'unité PCE située sur le réseau principal et ont utilisé un protocole de couche d'application légère (le protocole CoAP) pour collecter les informations nécessaires au calcul de l'ordonnancement. La méthode AMUS permet à une séquence de planification multi-saut (MSS) de fournir une faible latence et alloue des ressources supplémentaires aux liens vulnérables afin de réduire considérablement le délai causé par des interférences ou des collisions. Cette solution surpasse TASA, car elle permet d'améliorer la fiabilité de la communication et aussi d'obtenir une latence extrêmement faible.

Dans (Ojo & Giordano (2016)), Ojo et Giordano ont formulé le problème d'ordonnancement en un problème de maximisation de débit et le délai en un problème de minimisation. Ils ont proposé d'utiliser la théorie des graphes basée sur la théorie des correspondances pour résoudre le problème de maximisation du débit de manière centralisée. Les résultats montrent que la solution proposée atteint un très haut débit. Le même problème a été formulé en un problème de maximisation de l'efficacité énergétique dans (Ojo *et al.* (2017)). Dans cet article, les auteurs ont proposé un planificateur à efficacité énergétique réduite (*Energy Efficient Scheduler* EES) qui performe mieux que *Round Robin Scheduler* (RRS) en termes d'efficacité énergétique, tout en garantissant un débit de transfert de données amélioré.

1.5.2 Ordonnancement distribué

À l'inverse de l'ordonnancement centralisé, les solutions distribuées ont tendance à être plus robustes face aux changements, sans prendre aucune hypothèse a priori sur la topologie de réseau ou sur le volume de trafic à transmettre. Lors de l'implémentation d'une approche distribuée, chaque nœud négocie avec les noeuds voisins et décide localement des liens à planifier avec ces derniers.

L'ordonnancement basée sur le trafic décentralisé (*Decentralized Traffic Aware Scheduling* DeTAS) est la version distribuée de la méthode TASA (Accettura *et al.* (2013)). Cette méthode s'adresse aux réseaux multi-coordinateurs. Par conséquent, elle fait recours au micro-ordonnancement combiné pour créer l'ordonnancement global. Pour éviter les interférences, chaque micro-ordonnancement utilise un ensemble de canaux dédiés. La méthode DeTAS a été comparé à TASA et les résultats obtenus montrent que le premier offre une meilleure gestion des files d'attente. En outre, DeTAS garantit des performances élevées en termes de rapport cyclique, de délai de bout en bout et de rapport de perte de paquets (*Packet Loss Ratio*) (Accettura *et al.* (2015)).

La solution DiSCA (*Distributed Scheduling for Convergecast in Multi-channel Wireless Sensor Networks*) considère deux cas de transmission : sans un accusé de réception et avec un accusé de réception immédiat (Soua *et al.* (2015)). À chaque itération dans cet algorithme, un noeud planifie une transmission en suivant un ensemble de règles. Chaque itération fournit un micro-ordonnancement. Cet algorithme est susceptible de s'imbriquer dans le but de réduire le nombre total de slots. Les auteurs ont comparé DiSCA à (Soua *et al.* (2012)) et à (Soua *et al.* (2016)) et les résultats obtenus montrent que DiSCA est très proche d'un ordonnancement optimal avec un nombre de messages de contrôle réduit.

Wave est un algorithme d'ordonnancement distribué pour les réseaux IEEE 802.15.4e *convergecast* (Soua *et al.* (2016)). Chaque nœud du réseau connaît ses nœuds conflictuels et son noeud parent. Une série d'ondes est inondée dans le réseau, dont la première est déclenchée par un message START envoyée par le coordinateur. Lorsqu'un nœud est prioritaire parmi ses nœuds

conflictuels (c'est-à-dire qu'il dépend du nombre de paquets dans sa file d'attente) et reçoit ce message, il s'assigne une cellule dans la vague et notifie ses nœuds conflictuels par l'envoi d'un message ASSIGN. Ce processus se répète jusqu'à ce que tous les nœuds sélectionnent des cellules. Une fois que la première vague est propagée, le coordinateur transmet un message REPEAT pour appeler une seconde vague qui améliore la première. Ce processus se répète jusqu'à ce que tous les nœuds puissent ordonnancer tous leurs paquets qui ce trouvent dans leurs files d'attente. Les résultats des simulations faites par les auteurs démontrent que, par rapport à DiSCA, la méthode Wave réduit la longueur de l'ordonnancement (Soua *et al.* (2015)) et se démarque en tant qu'un algorithme d'ordonnancement distribué efficace.

Dmir et Bilgili ont suggéré un algorithme d'ordonnancement distribué *divergecast* appelé DIVA (Demir & Bilgili (2017)). À l'inverse de trafic *convergecast*, le trafic *divergecast* circule dans toutes les directions et non pas seulement à partir des nœuds racines. Chaque nœud commence par diffuser une demande de connexion CON-REQ. S'il reçoit un paquet d'accusé de réception de connexion CON-ACK en réponse, un lien sera établi entre ces deux nœuds. Ce processus est exécuté par tous les nœuds du réseau jusqu'à ce que la longueur maximale de la *slotframe* soit atteinte. Cette approche a été comparée à (Tinka *et al.* (2010)). Malheureusement, DIVA n'améliore pas Aloha, mais supporte toutes les tailles de *slotframes*.

1.5.3 Ordonnancement autonome

Chaque type d'ordonnancement mentionné ci-dessus a des avantages et des inconvénients. Pour une classification globale dans la revue de littérature, les auteurs placent l'approche autonome en tête de liste, suivie de l'approche distribuée puis de l'approche centralisée. De façon générale, l'approche centralisée révèle plusieurs faiblesses dont l'initialisation de l'ordonnancement, qui prend beaucoup de temps, ce qui augmente la consommation d'énergie des noeuds capteurs. Ce type de solution ne réagit pas aux changements subites du réseau. De plus, la surcharge des messages de signalisation, utilisés pour définir et maintenir l'ordonnancement, réduit la capacité du réseau et entraîne une consommation d'énergie supplémentaire. Néanmoins, l'ordonnancement centralisé est certainement sans conflit, ce qui augmente la fiabilité

du réseau. Toutefois, il n'est toujours pas adapté aux réseaux dynamiques à grande échelle avec des contraintes énergétiques importantes. Dans cet objectif, l'approche distribuée s'avère comme une solution très prometteuse.

Les méthodes d'ordonnancement autonomes ont démontré leur efficacité. Chaque nœud peut choisir de manière autonome son propre ordonnancement. De ce fait, il n'y a pas de surcharge de signalisation, même localement entre les nœuds voisins. Cependant, cette méthode ne permet pas aux nœuds d'avoir l'information par rapport à la charge de trafic dans le réseau, ne laissant aucune possibilité d'optimiser et d'adapter l'ordonnancement selon les différentes charges de trafic.

1.6 Groupe de travail IETF 6TiSCH

Dans la perspective de l'Internet des Objets (IoT) futur, il est important d'intégrer TSCH dans la pile de protocoles de l'IoT. À cette fin, le groupe de travail 6TiSCH a été créé par l'organisme de normalisation IETF dans le but d'intégrer IPv6 avec le mode TSCH ((Palattella *et al.* (2013)), (Thubert *et al.* (2013)), (Accettura & Piro (2014))). Il vise à lier les capacités IEEE 802.15.4e à mode TSCH aux efforts et aux recommandations de normalisation antérieurs IETF 6LoWPAN et ROLL. Plus précisément, l'objectif de 6TiSCH est de fournir des mécanismes permettant de combiner la haute fiabilité et la faible consommation d'énergie de TSCH à la facilité d'interopérabilité et d'intégration offertes par le protocole IP.

Dans 6TiSCH, le mode TSCH de la couche MAC est placé sous une pile de protocoles compatible avec IPv6. Cette pile exécute également IPv6 au-dessus d'un réseau sans fil personnel à faible puissance (Low-Power Wireless Personal Area Network 6LoWPAN), de protocole de routage IPv6 pour les réseaux à faible consommation (*Routing Protocol for Low-Power and Lossy Networks* RPL) et de protocole des applications contraintes (*Constrained Application Protocol* CoAP).

Pour intégrer correctement le mode TSCH aux protocoles des couches supérieures, le groupe de travail 6TiSCH définit une nouvelle entité fonctionnelle chargée d'ordonnancer des *timelsots* TSCH pour l'envoi des trames sur le réseau.

En fait, bien que la norme IEEE 802.15.4e définisse les mécanismes permettant à un nœud TSCH de communiquer, elle ne définit pas les règles qui permettent de :

- Créer et de gérer l'ordonnancement de la communication
- Correspondre l'ordonnancement aux chemins à sauts multiples gérés par le protocole RPL
- Adapter les ressources allouées entre les nœuds voisins au flux de trafic de données
- Appliquer un traitement différencié pour les données générées au niveau de la couche application et des messages de signalisation dont 6LoWPAN et RPL ont besoin pour détecter les nœuds voisins et pour réagir aux changements topologiques.

1.6.1 Architecture

Le groupe de travail 6TiSCH définit un réseau LLN (*Low Power Lossy Network*) qui est composé de centaines à des milliers de noeuds déployés dans un environnement physique donné et utilisant TSCH au niveau de la couche MAC. Tous les nœuds du réseau appartiennent au même sous-réseau IPv6 et communiquent à travers le protocole IPv6, ainsi ils utilisent 6LoWPAN *Header Compression* (6LoWPAN HC) pour transmettre les paquets.

Pour permettre au réseau de s'adapter aux milliers de nœuds et d'être considéré comme un seul sous-réseau IPv6, 6TiSCH recommande la présence d'un *backbone* de haute vitesse (par exemple, un réseau maillé dit *mesh* sans fil utilisant 802.11) qui couvre l'environnement physique au complet. Le *backbone* fournit une infrastructure rapide pour interconnecter et synchroniser tous les nœuds. Les nœuds sont attachés au réseau principal via un ou plusieurs routeurs *backbone* (*Backbone Routers* BBR), tandis que tout le réseau principal est connecté à l'Internet via une passerelle. La Figure 1.3 montre l'architecture globale d'un réseau 6TiSCH (Dujovne *et al.* (2014)).

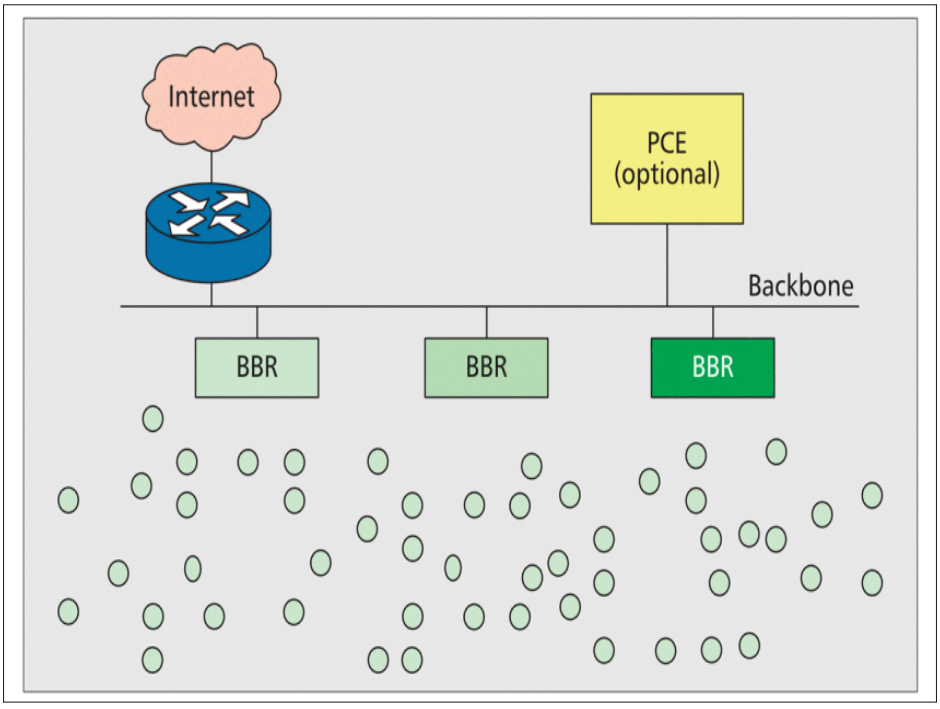


Figure 1.3 Architecture globale d'un réseau 6TiSCH. (Dujovne *et al.* (2014)).

1.6.2 Couche protocolaire

Le groupe de travail 6TiSCH a comme objectif de combiner les couches physique de IEEE 802.15.4 et MAC à mode TSCH de IEEE 802.15.4e avec des couches supérieures IETF (c'est-à-dire 6LoWPAN, RPL, CoAP, etc.) afin de créer une pile de protocoles à standard ouvert (*open-standard*) pour les réseaux sans fil IPv6 maillés (Thubert (2018)).

La pile de protocole 6TiSCH est représentée à la Figure 1.4 (Dujovne *et al.* (2014)). Le protocole CoAP permet des interactions RESTful avec les nœuds du réseau, tandis que RPL construit et maintient une topologie de routage, alors que 6LoWPAN compacte les en-têtes de IPv6 afin de réduire la taille des paquets à transmettre sur le support sans fil (Sudhaakar & Zand (2015)).

Pour permettre aux protocoles de IETF de fonctionner de manière optimale au-dessus du mode de fonctionnement TSCH, certaines lacunes doivent être corrigées. Plus précisément, une nouvelle sous-couche, appelée 6top, est en cours de définition par le groupe de travail 6TiSCH. La

sous-couche 6top fonctionne au-dessus de TSCH et permet à une entité de gestion (*Management Entity* ME) de contrôler l'ordonnancement TSCH pour ajouter ou supprimer des liens (en particulier des cellules, selon la terminologie 6TiSCH). En outre, 6top collecte des informations sur la connectivité, qui peut être utile pour les couches supérieures (par exemple, RPL) et surveille les performances des cellules afin de leur permettre de les ré-ordonnancer si elles ne se comportent pas comme prévu. La sous-couche 6top a été conçue pour être utilisée avec des approches d'ordonnancement centralisés et distribués.

À cette fin, 6top classe chaque cellule en tant que *hard* ou *soft*. Une cellule *hard* ne peut pas être réallouée dynamiquement par 6top, car elle est généralement affectée par l'entité d'ordonnancement centrale. Inversement, les cellules *soft* peuvent être réaffectées de manière dynamique par 6top si elles affichent une mauvaise performance. Les cellules *soft* sont généralement affectées par des algorithmes distribués qui fonctionnent au-dessus de la couche 6top. Cependant, les algorithmes d'ordonnancement indiquent uniquement à la sous-couche 6top combien de cellules *soft* qui doivent être planifiées avec un voisin particulier. Ensuite, 6top affecte chaque cellule à un certain couple (*slot, channel offset*) dans l'ordonnancement TSCH (6top utilisera la procédure de négociation de cellule *soft* décrite dans (Wang *et al.* (2018)) pour effectuer cette tâche).

De plus, puisqu'un réseau 6TiSCH peut transporter différents types de trafic (avec des exigences de qualité de service QoS différentes), 6top peut marquer des cellules avec des étiquettes différentes afin d'identifier les différents flux de trafic, permettant ainsi une isolation des flux. En détail, lorsqu'un paquet entre dans le réseau 6TiSCH, la sous-couche 6top du nœud identifie et marque la classe de service à laquelle appartient le paquet. Ensuite, en se basant sur l'étiquette attribuée, chaque nœud du réseau 6TiSCH peut choisir la cellule à laquelle le paquet doit être transmis.

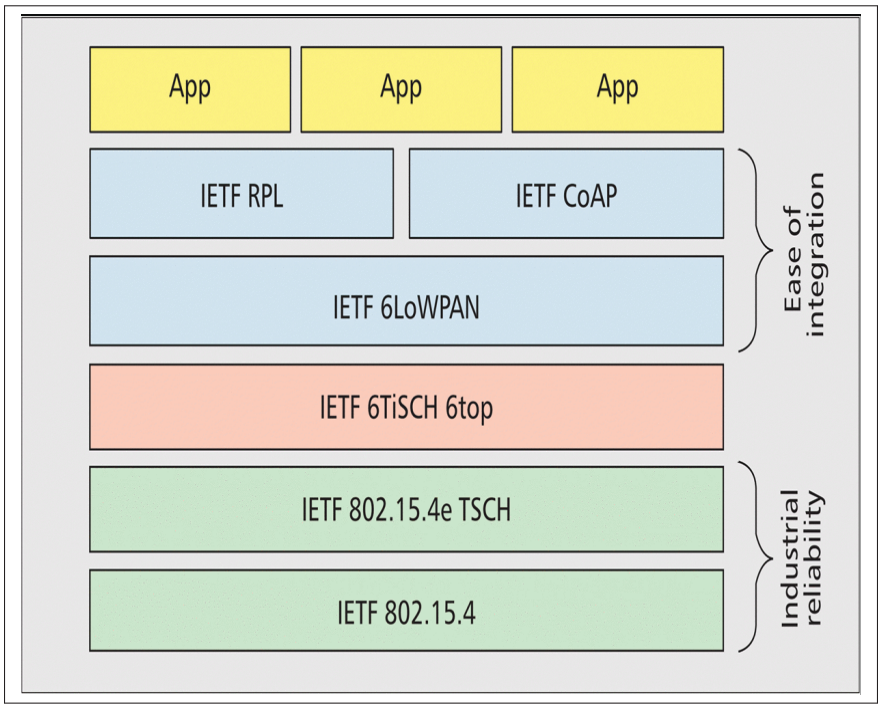


Figure 1.4 Couche protocolaire de 6TiSCH. (Dujovne *et al.* (2014)).

1.7 Conclusion

Ce chapitre a présenté les fondements de base, ainsi que les limitations des deux couches physique et MAC définis dans le standard 802.15.4-2006. Il a présenté aussi l'amendement apporté par le nouveau standard 802.15.4e-2012 à mode TSCH afin de répondre aux besoins des domaines d'application.

Ensuite, une classification des algorithmes d'ordonnancement en 3 catégories a été présentée en se basant sur les objectifs et les contraintes du mode TSCH. Dernièrement, les efforts de normalisation et d'intégration, réalisés par le groupe de travail 6TiSCH afin d'intégrer le mode TSCH à une pile protocolaire IpV6, ont été détaillés et décrits.

CHAPITRE 2

DÉVELOPPEMENT D'UNE MÉTHODE D'ORDONNANCMENT OPTIMISÉE POUR LES RÉSEAUX TSCH

2.1 Introduction

Ce chapitre est composé de trois sections. Dans la première section, nous introduisons quelques définitions en relation avec le modèle énoncé ainsi que quelques hypothèses que nous prendrons en considération lors de la mise en œuvre de la solution proposée. Dans la deuxième section, nous formulerons le problème d'envoi et de réception des paquets dans les réseaux 802.15.4e TSCH en tant qu'un processus de Poisson qui satisfait certaines conditions et qui permettra de déduire une prédiction des évènements futurs dans les prochaines *slotframes*. Ces prédictions permettront de prendre des décisions plus rapides entre les paires de nœuds avec un taux de paquets de contrôle échangés considérablement inférieure à celui implémenté dans la *minimal scheduling function* proposée par défaut dans le standard. Nous développerons, dans la troisième section, un algorithme qui permet à chaque paire de nœuds de prendre des décisions d'ajout ou de suppression de cellules, selon la prédiction calculée à l'aide de processus de Poisson, d'une manière distribuée.

2.2 Solution proposée

Dans le but d'améliorer la *minimal scheduling function*, proposée par le mode de fonctionnement TSCH, introduit dans le standard 802.15.4e, nous proposons un algorithme d'ordonnancement distribué qui répond aux objectifs suivants :

- Un ordonnancement dynamique : L'algorithme proposé alloue et libère les cellules sans tenir compte du mécanisme de calcul de seuil proposé par le standard dans la *minimal scheduling function*.
- Une transmission de données de contrôle minimisée lors de la phase de négociation de l'ordonnancement : L'EMSF est conçu pour atteindre cet objectif en introduisant un modèle

de prédiction qui anticipe les données à transmettre pour chaque paire de nœuds dans la prochaine *slotframe*.

- Un temps de latence réduit : Cette conception est introduite en minimisant l'échange de paquets de contrôle généraux, afin de diminuer le délai de transmission de données de bout en bout. Ainsi, le nombre des paquets dans la file d'attente de chaque nœud sera réduit.

2.2.1 Exigences

Le protocole 6top (6p) permet aux nœuds voisins d'un réseau 6TiSCH d'ajouter ou de supprimer des cellules dans leurs ordonnancements. Il fait partie de la sous-couche opérations de 6TiSCH IEEE 802.15.4e, qui fournit des mécanismes permettant d'effectuer l'orchestration distribuée dans ce type de réseau. C'est la fonction d'ordonnancement qui décide quand ajouter ou supprimer des cellules, ainsi elle utilise 6p pour exécuter efficacement l'allocation de ressources. Lorsque des nouvelles cellules doivent être ajoutées ou supprimées, le protocole 6p exécute une transaction dite 6p, qui forme la négociation d'ajout ou de suppression des cellules entre une paire de nœuds).

Dans le cas de réseaux avec un ordonnancement distribué, les nœuds conservent leurs propres orchestrations. Ceci permet d'assurer une meilleure efficacité en termes de temps de signalisation entre les nœuds. Cependant, moins de signalisation signifie également que les nœuds sont moins informés sur le réseau, ce qui rend plus difficile la création d'un ordonnancement efficace.

2.2.2 Contraintes de conception

Pendant la phase de conception de l'algorithme d'ordonnancement, nous avons défini quelques contraintes de fonctionnement, qui sont indispensables lors de l'exécution de notre solution. Chaque nœud adapte de manière dynamique la quantité de ressources qui a été allouée avec ses nœuds voisins, en fonction de son allocation de ressources actuelle ainsi que de ses propres besoins en ressources. La *minimal scheduling function* ne prend pas en considération la charge

de trafic récurrent, ce qui signifie que chaque cellule réservée se répète à chaque *slotframe* et gaspille par la suite les ressources dans le cas où la génération des paquets dans le réseau n'est pas aussi fréquente. De plus, pendant la phase de détermination de la bande passante, nécessaire pour la communication entre une paire de nœuds, le nombre de messages de contrôle échangés doit être réduit. Cela va réduire le temps d'attente lors d'une transmission de bout en bout et minimiser aussi le nombre des paquets dans la file d'attente de la mémoire de chaque nœud.

2.2.3 Modèle de réseau et notations

Dans les réseaux à mode TSCH, défini par 802.15.4e, lorsqu'un nœud détecte une fluctuation brusque d'un évènement physique, un flux massif de paquets de données est généré et mis en file d'attente dans la mémoire du nœud capteur. Ce nœud vérifie s'il possède suffisamment de bande passante pour envoyer ces paquets à son nœud parent. Autrement, le nœud vérifie, dans la prochaine *slotframe*, s'il a assez de cellules réservées avec son parent et compare s'il est capable de transmettre ces paquets durant ces cellules. La vérification de la bande passante avec le nœud parent déclenche un nombre élevé de transactions par paquet, ainsi générant une augmentation considérable de l'utilisation de ressources énergétiques. Lorsque le nombre limité de transmissions et de retransmissions est dépassé, il y aura une perte de paquets qui entraine un épuisement de ressources énergétiques du nœud.

Dans l'objectif de réduire les erreurs de négociation, le nombre des paquets abandonnés et le temps de latence de bout en bout, nous proposons un nouvel algorithme d'ordonnancement basé sur la *minimal scheduling function* présentée dans le brouillon de l'IETF (Chang *et al.* (2018)). L'algorithme proposé comprend les deux opérations principales suivantes : le calcul de la moyenne du nombre de paquets générées par chaque nœud et la prédiction du nombre de cellules requises pour chaque paire de nœud dans la prochaine *slotframe*. Nous introduisons les définitions suivantes :

- **Définition 1** : Nous nous concentrons seulement sur les RCSFs appelés *event-driven* dans lesquels les nœuds mesurent des évènements physiques et les envoient aux récepteurs dans un mode de transfert de données en amont.
- **Définition 2** : Nous définissons la topologie du réseau sous forme de graphe $G = (V, E)$, où V représente l'ensemble de nœuds du réseau et E désigne l'ensemble des arêtes séparant les nœuds et affichant des liaisons de communication symétriques.
- **Définition 3** : Un arbre de routage d'un nœud n (sauf le nœud *sink*) contient $Parent(n)$, $Subtree(n)$ et $nchild(n)$. Où $Parent(n)$ est le nœud parent du noeud n, $Subtree(n)$ est le sous arbre de l'arbre de routage imbriqué au nœud n et finalement $nchild(n)$ représente les nœuds fils d'un parent donné n.
- **Définition 4** : Dans une trame de collecte de données de tout nœud $n \in G$, on note par $G(n)$ le nombre de paquets de données envoyés par le nœud n. On note aussi par $T(n)$ la somme de tous les paquets envoyés par un nœud n, y compris ceux envoyés par $G(n)$, et le nombre de paquets reçus par le parent $Parent(n)$ de la part de ses fils $Subtree(n)$. Par conséquent, nous définissons $T(n)$ en utilisant l'équation suivante :

$$T(n) = \sum_{v \in \mathrm{subTree(n)}} G(V). \tag{2.1}$$

- **Définition 5** : Nous dénotons $\mathrm{Q_C^P(n)}$ comme étant le nombre de paquets dans la file d'attente d'un nœud n qui doivent être envoyés au parent $Parent(n)$. On dénote aussi $\mathrm{C_C^P(n)}$ comme étant le nombre de cellules qui ont déjà été allouées entre un $Parent(n)$ et un fils $nchild(n)$.
- **Définition 6** :Après l'exécution de l'algorithme d'ordonnancement, $\forall n \in G$, une transaction d'ajout, de suppression ou de maintien de cellules est déclenchée dans le prochain slotframe S_{i+1}.

Une fois que les notions de notre modèle de système sont définis, nous adoptons les hypothèses suivantes :

- **Hypothèse 1** : Nous supposons que la topologie de collecte de données ainsi que l'arborescence de routage sont fournies au préalable.
- **Hypothèse 2** : Nous considérons également que les liens de l'arbre de routage sont symétriques, car les données du réseau sont collectées en amont, alors que l'ordonnancement est négocié d'une manière distribuée entre les nœuds. Chaque paire de nœuds se décide un ordonnancement indépendamment du nœud central (*sink*).
- **Hypothèse 3** : Les liens symétriques dans le réseau sont obligatoires pour la stratégie d'accusé de réception instantané.

2.3 Prédiction de la quantité de données

EMSF est un protocole d'ordonnancement distribué basé sur le protocole MSF qui a pour objectif de décider quand augmenter ou diminuer la bande passante entre deux nœuds voisins (ajout/suppression de cellules) en interagissant avec la sous-couche 6top.

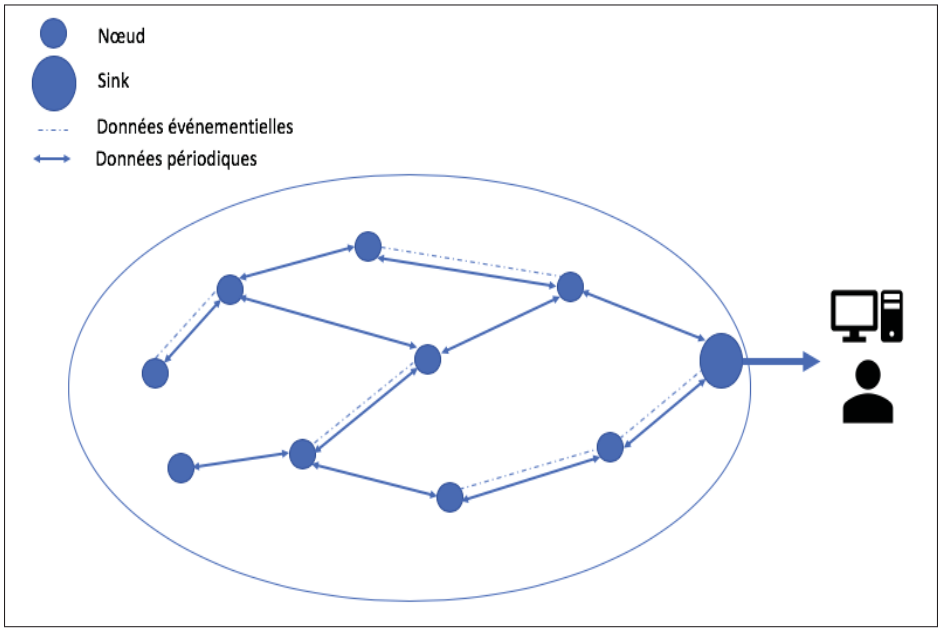


Figure 2.1 Type de données dans un réseau 802.15.4e TSCH.

À l'opposé du protocole MSF, qui récupère des statistiques de la sous-couche 6top pour enfin prendre les décisions d'ajout ou de suppression des cellules, EMSF se base sur un calcul statistique qui se fait au niveau de la couche MAC. Ceci réduit la quantité de paquets de contrôle échangées entre un nœud parent et son nœud fils.

Dans les réseaux 802.15.4e à mode TSCH, un nœud n peut transmettre deux types de données : des paquets périodiques notés par $A_i^w(n)$ et des paquets évènementiels notés par $A_i^v(n)$, comme affiché dans la figure. 2.1. Les paquets périodiques sont constitués par des balises appelées *enhanced beacons* qui contiennent des informations à propos de l'ASN de la *slotframe* actuelle, sa longueur en secondes, ainsi que d'autres informations sur le réseau. Ces données sont utilisées lorsqu'un nœud veut rejoindre le réseau (le nombre de nœuds dans son voisinage et le nœud à choisir en tant que proxy de jointure) et aussi pour le maintien de la synchronisation entre les nœuds du réseau. Les données événementielles sont envoyées lorsqu'un événement physique est détecté par le nœud. Le débit total généré par un nœud n noté par $A_i^T(n)$ est illustré comme suit :

$$A_i^T(n) = A_i^w(n) + A_i^v(n). \tag{2.2}$$

Ce qui peut être développé comme suit :

$$A_i^T(n) = \sum_{i \neq sink} A_{i-sink} + \sum_{k=1}^{i-1} A_{k-sink}, \mathrm{i} < 1 < \mathrm{sink}. \tag{2.3}$$

Ceci peut être traduit à l'équation suivante :

$$A_i^T(n) = \sum_{i-1}^{i} \mathrm{subTree(n)} + \sum_{\mathrm{i}-1}^{\mathrm{i}} G(n), \mathrm{i} < 1 < \mathrm{sink}. \tag{2.4}$$

2.3.1 Modèle de génération de paquets à base de processus de Poisson

Dans notre modèle, nous considérons les réseaux de capteurs 802.15.4e à mode TSCH, qui transmettent des paquets seulement lorsqu'ils détectent un changement ou une déviation d'un événement physique (un événement se produisant dans un champ de détection). En prenant en considération la nature du mode d'envoi et de réception des paquets dans le réseau, nous adoptons un processus de Poisson qui satisfait aux lemmes suivants :

- **Lemme 1** : Les événements sont considérés comme indépendants dans le temps et dans l'espace et ils se produisent avec une probabilité égale sur toute la zone de couverture de réseau. Dans le réseau, nous considérons les paquets générés par les différents nœuds comme des événements indépendants qui n'ont aucune relation avec l'événement qui précède leur génération.
- **Lemme 2** : La durée de la *slotframe* est inchangeable durant tout l'ordonnancement. Ainsi, le temps T entre chaque *slotframe* est fini.

En se basant sur les lemmes (1) et (2), nous pouvons adopter un modèle de processus de Poisson pour formuler la génération des paquets de données dans le réseau.

2.3.2 Formulation mathématique

Premièrement, nous représentons par $N(t)$ le nombre d'évènements qui se produisent dans un intervalle de temps (0, t], et nous admettons que $N(0) = 0$. Le processus $\{N(t); t \geq 0\}$ est appelé processus de comptage. Il vérifie les conditions suivantes :

- $\forall\, t \geq 0, N(t) \in N$
- $t \rightarrow N(t)$ est croissante
- $\forall\, 0 < a < b$, $N(b)$ - $N(a)$ désigne le nombre d'évènements qui ont eu lieu dans l'intervalle de temps]a, b].

Dans notre modèle, le processus de comptage est considéré à croissement indépendant parce que les évènements (transmission/réception de paquets de données) qui se produisent dans un intervalle de temps disjoint (la durée d'une *slotframe*) sont indépendants (figure 2.2). De plus, nous adoptons les conditions suivantes :

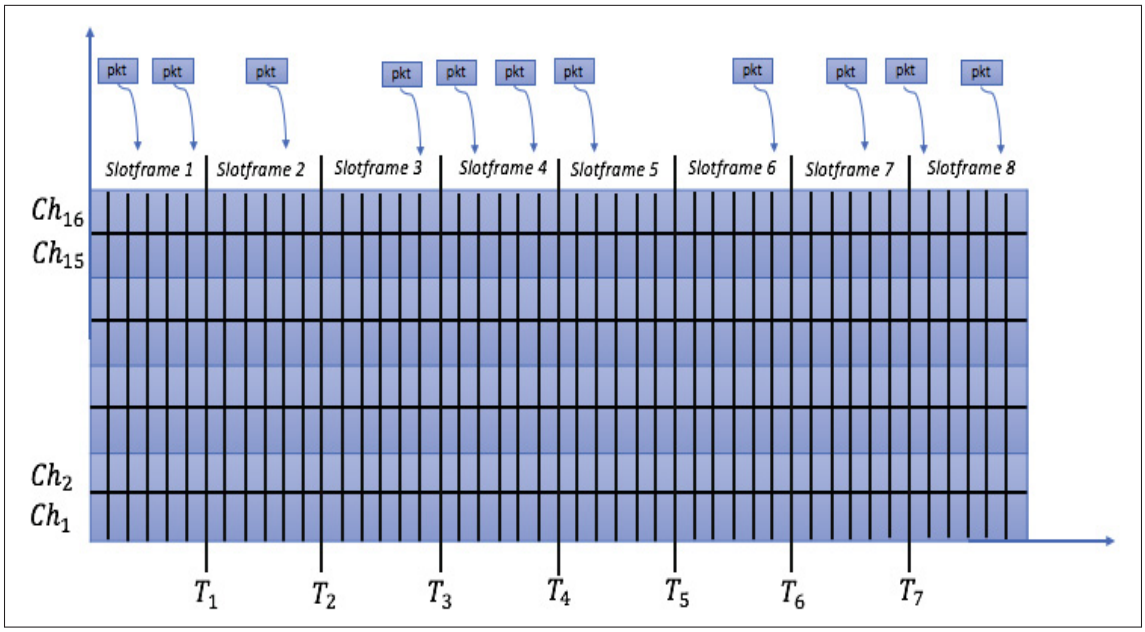


Figure 2.2 Génération des paquets au cours des *slotframes*.

- **Condition 1** : Les événements qui se répètent dans des intervalles de temps disjoints sont indépendants ; un couple de nœuds commence l'échange de paquets au cours des *slotframes* si et seulement si un événement est détecté. Par conséquent, pour tout choix de nombre réels $0 \leq t_1 < t_2 < ... < t_n$, les variables aléatoires $N(t_2) - N(t_1), N(t_3) - N(t_2), ..., N(t_n) - N(t_{n-1})$ sont mutuellement indépendantes.
- **Condition 2** : Une paire de nœuds transmet des paquets indépendamment de leur état pendant la *slotframe* précédente. On peut alors conclure que pour tout nombre réel positif t et h, le nombre $N(t+h) - N(t)$ d'évènements qui se réalisent au cours d'une période de temps $(t, t+h]$ est indépendant à la valeur de t et ne dépend que de la longueur de l'intervalle h.
- **Condition 3** : Dans le standard 802.15.4e à mode TSCH, la longueur maximale d'une *slotframe* ne doit pas dépasser 101 *timeslots* de durée de 10 millisecondes chacune. Cela signifie

que la longueur h de l'intervalle du temps sur lequel nous dénombrons les évènements est réduite. Par conséquent, on peut déduire que la probabilité d'observer plus qu'un évènement est quasiment nulle. $g(h)$ ici est une fonction de petit ordre de h.

$$P\left[N(t+h)-N(t)\geq 2\right]=g(h) \quad et \tag{2.5}$$

$$P\left[N(t+h)-N(t)\geq 1\right]=\lambda h+g(h). \tag{2.6}$$

La distribution du nombre de paquets de données, $N(t)$, générée par chaque nœud du réseau depuis le début de la *slotframe* jusqu'à sa fin est calculée comme suit :

$$P\{N(t)=n\}=\frac{(\lambda t)^n}{n!}e^{\lambda t}. \tag{2.7}$$

où λ représente la valeur moyenne du nombre de paquets générés et transmis entre une paire de nœuds depuis qu'ils se sont synchronisés à l'ordonnancement du réseau. Une fois les nœuds synchronisés, le standard 802.15.4e implémente la *minimal scheduling function* qui assure un fonctionnement minimal du réseau. Mathématiquement, λ est définie comme suit :

$$\lambda(n)=\frac{\sum_{i=T-\beta}^{T}\mathrm{nbPacket_i(n)}}{\beta}. \tag{2.8}$$

où T représente le temps actuel, $nbPacket_i(n)$ est le nombre de paquets générés par le nœud n à l'instant $T=i$ et β est la somme de nombres précédant des *slotframes* passées. Dans l'objectif d'avoir une valeur précise de λ , le standard 802.15.4e implémente le *minimal scheduling function* jusqu'à la *slotframe* numéro 10.

2.3.3 Calcul de la probabilité

En observant la distribution d'un processus de Poisson, lorsque le taux d'occurrence de certains évènements (que nous avons appelé λ) est faible, la gamme des possibilités probables se situera près de l'axe de 0 (Poulsen *et al.* (2011)). Ce qui signifie que lorsque λ est faible, il sera très probable d'obtenir la valeur 0 comme résultat. Au fur et à mesure que le taux augmente (à mesure que l'occurrence de ce que nous regardons devient plus courante), le centre de la courbe se déplace vers la droite. Ainsi, les occurrences nulles deviennent réellement improbables, comme indiqué dans la Figure 2.3.

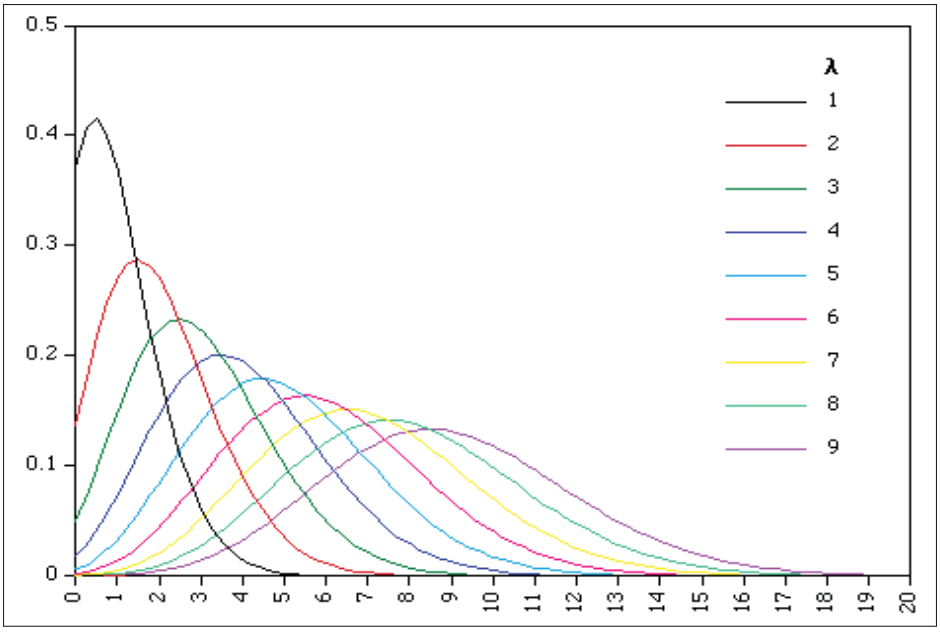


Figure 2.3 Distribution d'un processus de Poisson. (Poulsen *et al.* (2011)).

Conséquemment, dans l'objectif d'avoir une valeur précise dans le temps, nous fixons $\beta = 10$ *slotframes*. À partir de la *slotframe* numéro 10, chaque nœud calcule la moyenne du nombre de paquets transmis λ à partir du moment qu'il s'est synchronisé avec un autre nœud à l'ordonnancement du réseau. Ensuite, chaque nœud calcule la probabilité de transmettre un nombre donné de paquets en commençant par la probabilité de transmettre 1 paquet jusqu'à avoir une probabilité inférieure à une autre calculée précédemment. Toujours en se basant sur l'allure de

la courbe de distribution d'un processus de Poisson présentée dans la figure 2.3, la plus grande valeur de la probabilité calculée sera retenue et considérée comme le nombre de paquets qu'un nœud pourra transmettre pendant la prochaine *slotframe*.

2.3.4 Ajout/Suppression des cellules (6top)

La sous-couche 6TiSCH Operation (6top) est la couche immédiatement supérieure à la couche de contrôle d'accès au support de communication IEEE Std 802.15.4 TSCH (Figure 2.4). Les rôles de la sous-couche 6top sont les suivants :

- Permettre aux nœuds voisins de communiquer pour ajouter / supprimer des cellules les unes aux autres.
- Exécuter une ou plusieurs fonctions d'ordonnancement 6top (*Scheduling Function*), qui définissent les règles déterminant à quel moment ajouter / supprimer des cellules.

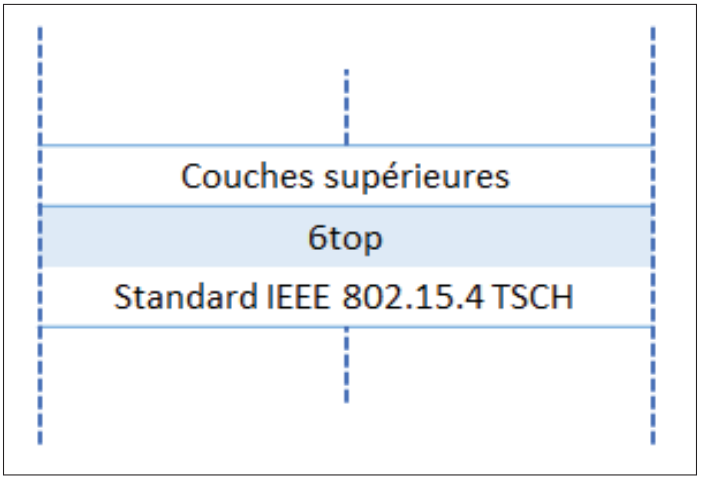


Figure 2.4 Couche protocolaire dans le standard 802.15.4e.

Une fois un nœud rejoint un réseau 6TiSCH, il pourra ajouter, supprimer ou déplacer des cellules avec son nœud parent préféré pour les trois raisons suivantes :

- Adapter les ressources de la couche liaison au trafic entre le nœud et son parent préféré.

- Gérer le changement de parent préféré (déclenché par le protocole RPL).
- Gérer une collision dans l'ordonnancement.

Nous nous concentrons seulement dans le cas où il y aura un changement de l'ordonnancement pour s'adapter au trafic du réseau. L'algorithme proposé fait recours à la *minimal scheduling function* dans le cas d'une collision ou de changement de parent. Notre solution fournit des statistiques sur l'utilisation de la bande passante à la sous-couche 6top pour ensuite pouvoir prendre des décisions d'ajout ou de suppression des cellules. À partir de la *slotframe* $\beta = 10$, chaque nœud (parent ou fils) exécute l'algorithme décrit dans le prochain paragraphe. L'objectif est de prédire le nombre de paquets qui seront générés par chaque nœud dans la prochaine *slotframe*.

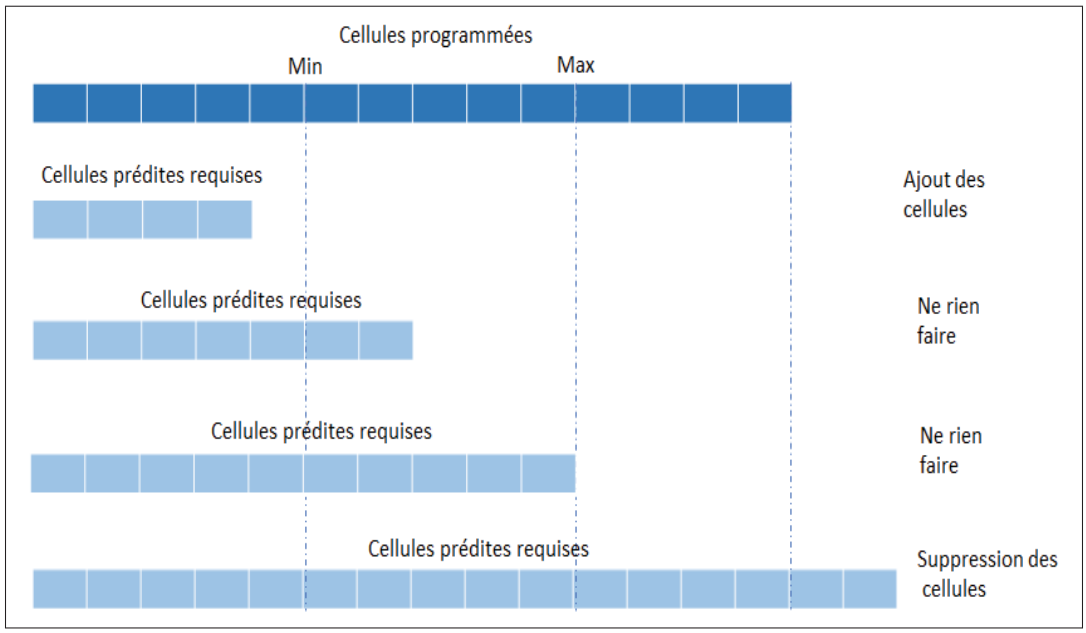


Figure 2.5 Ajout/suppression des cellules.

Dans le but de réduire les ressources de calcul faites par chaque nœud, l'algorithme arrête son exécution lorsqu'une probabilité maximale de générer λ paquets est atteinte. En connaissant le nombre de paquets qui seront générés dans la *slotframe* suivante, un nœud peut prédire le nombre de cellules nécessaires pour échanger des données avec son parent préféré. En outre,

en fonction de la sortie de l'algorithme, un nœud peut déclencher une transaction 6p avec son parent préféré pour ajouter ou supprimer des cellules à l'ordonnancement TSCH entre les deux nœuds (Figure 2.5).

2.4 Algorithme d'ordonnancement

Dans l'algorithme ci-dessous, chaque nœud va exécuter certains calculs en se basant sur des statistiques recueillies avec son parent préféré dans l'objectif de prédire combien de cellules il aura besoin pendant la prochaine *slotframe*.

Algorithme 2.1 Algorithme d'ordonnancement

```
1  Input : Set G(n) to be a random value chosen from a Poisson distribution with mean
     λ = rt (r in unit of 1/time)
2  for S = S_11 to {Network lifetime}; S_i++ do
3      Determine λ using Eq.(2.8) /* Average generated number of
           packets in S_0, S_1, S_2, ..., S_i-1                          */
4      while p < max do
5          p ← Determine (PN(t), λ) /* probability of generating λ
               packets                                                   */
6          if (p ⩾ pmax) then
7              max ← p
8          end
9      end
10     return (p) /* the maximum value of the probability               */
11     if (p < C_c^p(n)) then /* Compare p with the actual reserved cells
        for node n                                                       */
12         6P_DELETE_command (p) /* 6P delete request of C_c^p(n) - p
               slots                                                     */
13     else if (p > C_c^p(n)) then
14         6P_ADD_command (p) /* 6P add request of C_c^p(n) + p slots
               */
15     else
16         Do_nothing() /* keep the same number of cells                 */
17     end
18 end
```

Cette prédiction remplace la méthode utilisée dans le *minimal scheduling function* qui se base sur un échange de paquets de contrôle pour calculer la bande passante requise pour la communication entre une paire de nœuds. Comme mentionné dans le paragraphe précédent, cette prédiction se déroule au niveau d'un seul nœud et évitera par conséquence un échange de paquets.

2.4.1 Calcul de la moyenne

Dans l'objectif d'avoir une valeur précise de la moyenne de génération de paquets entre une paire de nœuds, l'algorithme utilise l'équation 2.8 à partir de la *slotframe* $S = \beta = 11$. Cette manipulation se répète à chaque début d'une *slotframe* jusqu'à ce que le nœud se désynchronise du réseau (Figure 2.6).

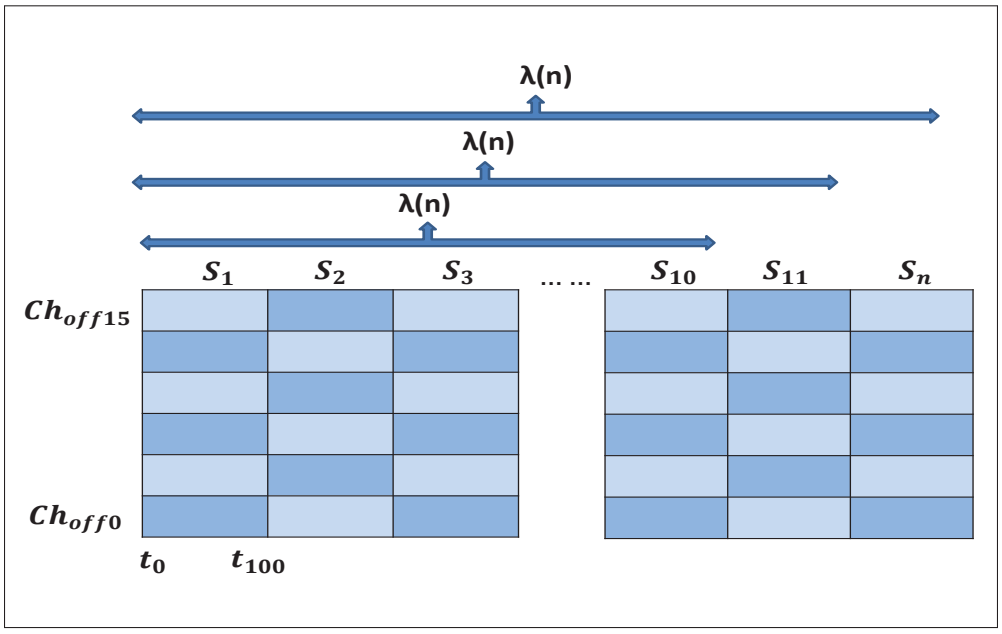


Figure 2.6 Calcul de la moyenne de paquets générés.

2.4.2 Prédiction du nombre de paquets

Prenons comme exemple un réseau composé de 5 nœuds. Après la mise en place du réseau et la synchronisation de ses nœuds, on considère que le nœud B a comme parent préféré le

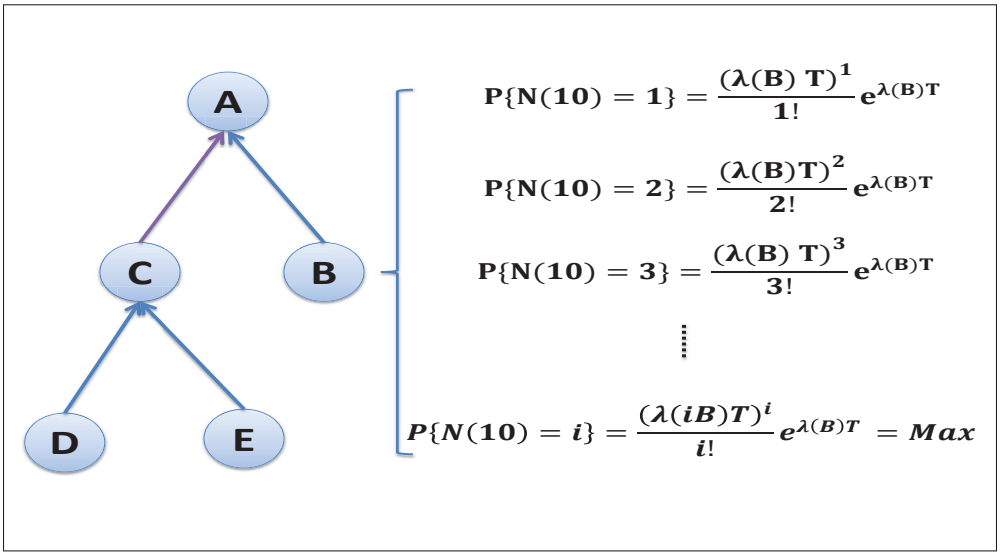


Figure 2.7 Prédiction du nombre de paquets.

nœud A. Ce dernier commence à calculer la probabilité d'avoir 1 seul paquet en se basant sur le nombre moyen des paquets générés entre la paire de nœuds durant les 10 dernières *slotframes* jusqu'à trouver une valeur de probabilité maximale (Figure 2.7). Dans l'objectif d'économiser les ressources de calcul et en se basant sur la fonction de masse de probabilité d'un processus de Poisson, l'algorithme arrête le calcul lorsqu'il atteint cette probabilité maximale.

2.4.3 Ajout/Suppression des cellules

Une fois la prédiction du prochain nombre de paquets, qui sera généré entre le nœud fils et son parent préféré, est calculée et en se basant sur le nombre de cellules déjà alloués, un nœud peut ajouter ou supprimer ou même garder le même nombre de cellules allouées. Ces résultats sont envoyés à la sous-couche 6top qui sera responsable d'envoyer les requêtes suivantes, comme indiqué à la Figure 2.8 :

- 6p_addCell (Max) : Si le nombre de cellules déjà alloué pour la paire de nœuds est inférieur à la prédiction calculée par l'algorithme, la requête prend comme paramètre le nombre de cellules (Max) qui seront ajoutées pendant la prochaine *slotframe*.

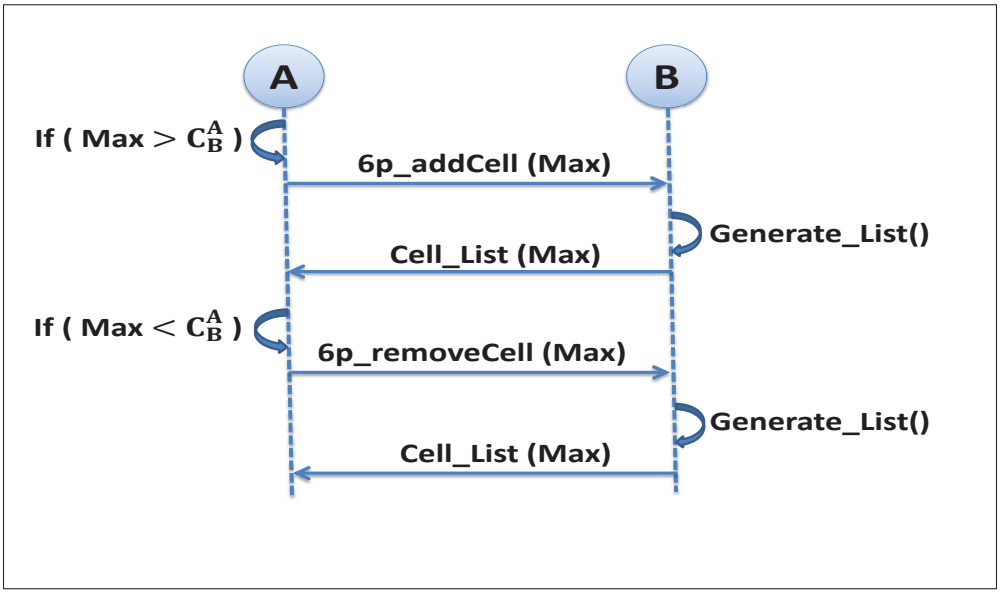


Figure 2.8 Transaction 6p pour ajout et suppression de cellules.

- 6p_addCell (Max) : Si le nombre de cellules déjà alloué pour la paire de nœuds est supérieur à la prédiction calculée par l'algorithme, la requête prend comme paramètre le nombre de cellules (Max) qui seront supprimées pendant la prochaine *slotframe*.
- Ne rien envoyer si Max est égal au nombre de cellules déjà allouées.

2.4.4 Exemple d'exécution

Pour expliquer l'algorithme utilisé dans le protocole EMSF, nous considérons le réseau illustré précedamment dans la Figure 2.9, qui est constitué par 5 nœuds.

Nous assumons que le protocole *minimal scheduling function* est utilisé durant les 10 premières *slotframes*. Dans la Figure 2.9, nous désignons les notations suivantes : p est le nombre de paquets générés par un nœud, λ est le nombre moyen des paquets générés après un certain nombre de *slotframes*, c est le nombre de cellules allouées pour un nœud et q est le nombre de paquets dans la file d'attente de la mémoire d'un nœud. Nous considérons que les nœuds du réseau collectent des données et les transmettent vers le nœud passerelle A pour le traitement et l'envoi au serveur. Nous montrons par la suite que, à partir de la *slotframe* 11, dans le cas

	Nœud A			Nœud B				Nœud C				Nœud D				Nœud E			
Slotframes	parent		fils	parent		fils		parent		fils		parent		fils		parent		fils	
	-------	B	C	A		-------		A		D	E	C		-------		C		--------	
				p	λ	c	q	p	λ	c	q	p	λ	c	q	p	λ	c	q
0	...	...	...	2	...	1	1	3	...	1	2	3	...	2	1	5	...	4	1
1	...	...	...	3	...	2	2	4	...	2	4	2	...	2	1	5	...	3	3
2	...	...	...	2	...	3	1	2	...	3	3	1	...	4	0	4	...	2	5
3	...	...	...	1	...	2	0	3	...	3	3	2	...	4	0	3	...	4	4
4	...	...	...	4	...	4	0	3	...	2	4	1	...	3	0	5	...	5	4
5	...	...	...	5	...	3	2	3	...	2	5	4	...	2	2	3	...	3	4
6	...	...	...	3	...	2	3	2	...	4	3	3	...	3	2	2	...	4	2
7	...	...	...	2	...	2	3	5	...	3	5	2	...	4	0	5	...	5	2
8	...	...		5	...	4	4	4	...	2	7	2	...	1	1	4	...	2	4
9	...	...	...	1	...	3	2	3	...	1	9	1	...	2	0	0	...	1	3
10	...	...	...	3	...	2	3	3	...	3	9	1	...	1	0	4	...	2	5
λ	...	...	...	3.1				3.5				2.2				4			
11	...	...	...	5	3.2	3	5	7	3.8	4	12	2	2.1	2	0	2	3.8	4	3
12	...	...	...	4	3.3	3	6	3	3.7	4	11	1	2	2	0	1	3.6	4	0
13	...	...	...	1	3.1	3	4	2	3.6	4	9	1	2	2	0	0	3.3	3	0
14	...	...	...	2	3.2	3	3	1	3.4	3	7	0	1.8	2	0	1	3.1	3	0
15	...	...	...	3	3	3	3	2	3.3	3	6	1	1.8	2	0	0	2.9	3	0
16	...	...	...	2	3	3	2	3	3.3	3	6	1	1.7	2	0	1	2.8	3	0
17	...	...	...	3	3	3	2	3	3.2	3	6	2	1.7	2	0	0	2.6	3	0
18	...	...	...	2	2.9	3	1	2	3.2	3	5	0	1.6	2	0	1	2.5	2	0
19	...	...	...	2	2.8	3	0	1	3.1	3	3	0	1.5	1	0	1	2.4	2	0
20	...	...	...	1	2.7	3	0	1	3	3	1	1	1.5	1	0	0	2.3	2	0

Figure 2.9 Exemple d'exécution d'EMSF.

de détection d'un évènement brusque (génération des paquets intense, ex. nœud C), les nœuds exécutants le protocole EMSF gardent une file d'attente de mémoire modérée au cours des *slotframes* qui suivent la détection brusque tout en évitant les échanges de paquets de contrôle.

2.5 Conclusion

Dans ce chapitre, nous avons présenté des définitions et des notations par rapport à notre modèle de système. Nous avons formulé le problème de génération de paquets en tant qu'un processus de Poisson afin de prédire combien de paquets un nœud pourra envoyer dans la prochaine *slotframe*. Nous avons ensuite présenté l'algorithme exécuté au niveau de tous les nœuds à partir de la *slotframe* 10. Cet algorithme a pour objectif de calculer le nombre de cellules qu'une paire de nœuds aura besoin dans la prochaine *slotframe*. Finalement, nous avons

présenté un exemple d'exécution de l'algorithme en montrant son efficacité à réduire la taille de la file d'attente sans avoir recours aux échanges de paquets de contrôle.

CHAPITRE 3

SIMULATIONS ET RÉSULTATS

3.1 Introduction

Dans ce chapitre, nous allons évaluer la performance de l'algorithme proposé pour améliorer la *minimal scheduling function* dans les réseaux 802.15.4e à mode TSCH. Nous allons ainsi considérer plusieurs éléments de performance dans des différents scenarios de simulation, afin de mieux étudier les avantages de EMSF. Les réseaux 802.15.4e sont parmi les plus couramment implémentés dans l'industrie, où le temps de latence représente la contrainte la plus importante vu la nature des évènements à détecter, qui seront ensuite acheminés pour le traitement. Nous allons aussi évaluer d'autres critères de performance, qui peuvent influencer le temps de latence (critique), tels que le taux d'erreur des messages 6p, la charge de trafic supplémentaire et finalement la taille de la mémoire de la file d'attente.

3.2 Environnement de la simulation

Nous effectuons nos simulations sur OpenWSN (Vasiljević & Gardašević (2016)). Il s'agit d'un simulateur *open source* pour les RCSFs, qui supporte la pile protocolaire basée sur l'IoT avec prise en charge de IEEE802.15.4-2015 TSCH, 6LoWPAN, RPL et CoAP. Ce simulateur est organisé en deux sous-parties, une partie logicielle (appelée OpenVisualizer) et une partie *firmware*. La partie logicielle est le code qui s'exécute sur l'ordinateur et qui sert de passerelle vers Internet. Elle est entièrement écrite en Python et elle est responsable de la compression 6LoWPAN / IPv6, des routes en aval RPL, de la topologie et des statistiques du réseau (RSSI, LQI, ETX) ainsi que de la topologie de routage. La partie *firmware* est le code qui s'exécute sur les nœuds capteurs. Elle utilise GCC comme compilateur par défaut et elle supporte 3 architectures de processeurs (AVR, MSP430 et ARM Cortex) ainsi que 11 plateformes de capteurs (Watteyne *et al.* (2012)).

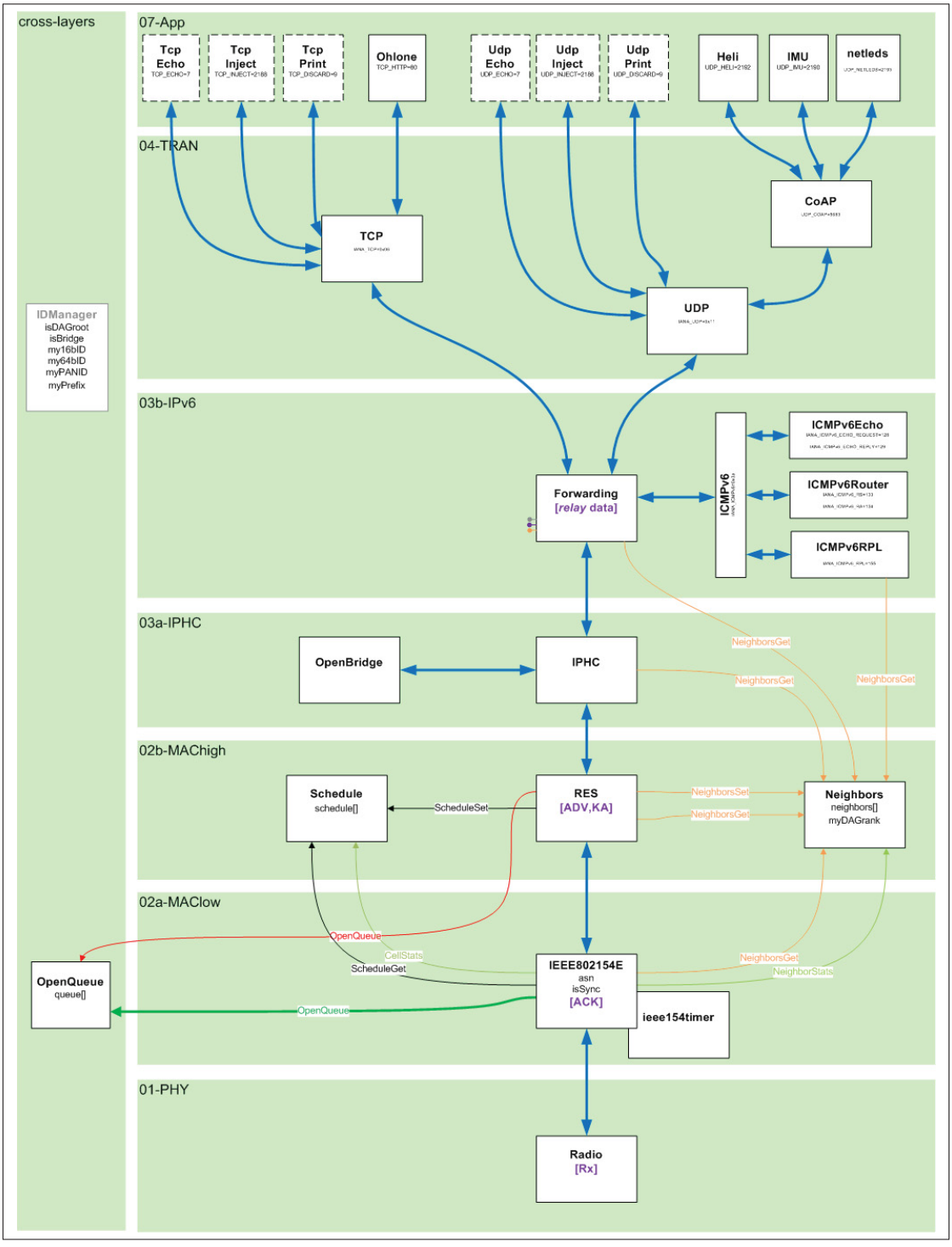


Figure 3.1 Architecture de OpenWSN. (Vasiljević & Gardašević (2016)).

Pour évaluer notre solution proposée, nous avons développé plusieurs scénarios de simulation sous différentes conditions. Nous avons simulé un nombre de nœuds qui varie entre 2 et 100, comme affiché dans la Figure 3.2. Le taux de délivrance de paquets entre chaque paire de nœuds (PDR) a été fixé à 100%. Chaque nœud génère un nombre aléatoire de paquets pendant chaque *slotframe*. Dans la configuration de la couche physique 802.15.4-2015, nous avons considéré que tous les canaux de communication sont disponibles et ont les mêmes caractéristiques physiques.

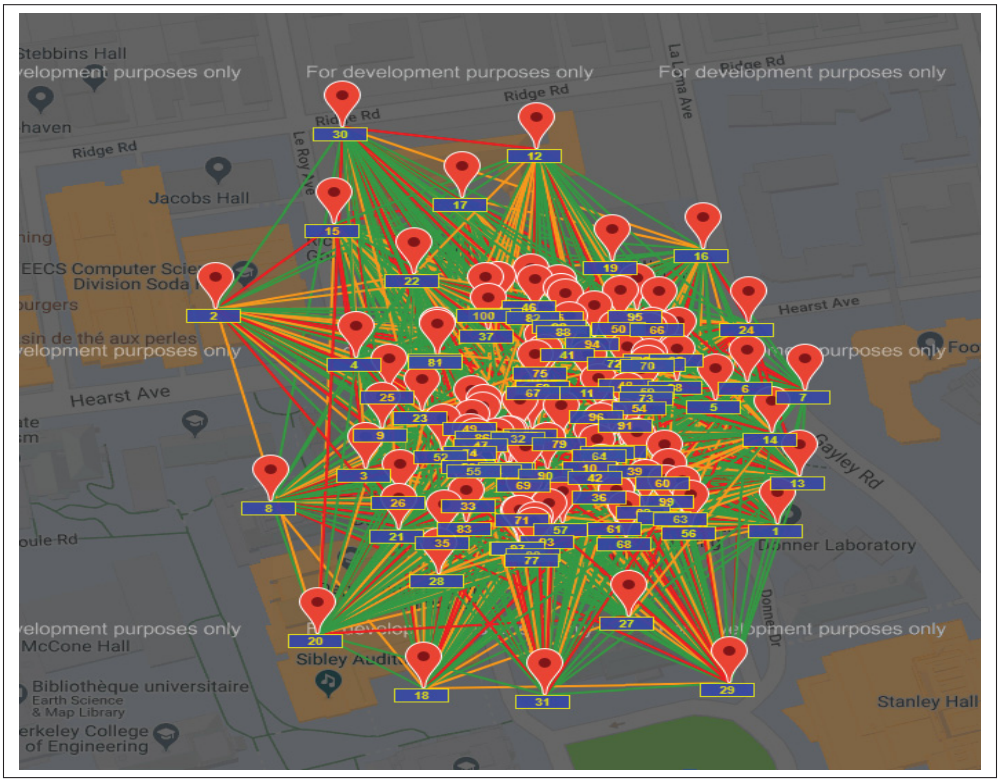


Figure 3.2 Topologie de réseau.

En fonction du nombre de nœuds dans le réseau et dans l'objectif d'avoir une moyenne précise de paquets générés λ, nous avons considéré qu'une *slotframe* est composée de 101 *timeslots* (longueur maximale). La durée de chaque *timeslot* est fixée à 10 millisecondes comme indiqué dans le standard 802.15.4e. Les autres paramètres essentiels sont indiqués dans le Tableau 3.1.

Tableau 3.1 Paramètres de simulation.

Paramàtères	**Valeurs**
Nombre de nœuds	2-100
Canaux disponibles	11-26
Longueur d'une *slotframe*	101 *timeslot*
Durée d'un timeslot	10 ms
Charge utile	127 octets
Maximum de tentatives MAC	4
Longueur maximale de la file d'attente	5
Période de transmission	200 ms

3.3 Résultats

Dans l'objectif de simuler les métriques souhaitées du réseau, le simulateur OpenWSN fournit un outil de configuration basé sur Python, qui permet de modifier en tout temps les paramètres de réseau. Le simulateur permet d'enregistrer et de simuler l'horodatage de chaque paquet envoyé ou reçu, ainsi que le comportement du réseau, tel que le PDR dans chaque canal et la priorité de la mise en file d'attente dans la mémoire de chaque nœud. Pendant chaque simulation, nous avons comparé les résultats des métriques calculées avec celles obtenues en implémentant la *minimal scheduling function*.

3.3.1 Taux d'erreur des messages 6p

Le protocole 6p fait partie de la sous-couche Opérations de 6TiSCH, qui est la couche immédiatement supérieure à la couche contrôle d'accès au support de communication 802.15.4e à mode TSCH. Il permet à une paire de nœuds voisins d'ajouter ou de supprimer des cellules dans l'ordonnancement TSCH tout en permettant une gestion d'ordonnancement distribué. Cette couche peut implémenter une ou plusieurs fonctions d'ordonnancement et permet aussi d'exécuter les transactions 6p permettant l'ajout ou la suppression des cellules.

Le taux d'erreur de négociation des transactions 6p est la moyenne de rapport entre le nombre d'erreurs de transaction 6p et le nombre total des transactions 6p tout au long d'un cycle d'une *slotframe*. La Figure 3.3 montre comment le taux d'erreur de négociation 6p est affecté par la

densité du réseau. À chaque simulation, nous avons augmenté par 10 le nombre de nœuds. À chaque augmentation, nous avons imbriqué les réponses des transactions 6p, qui contiennent un sous-registre de 8 bits contenant un code d'erreur (défini dans le protocole 6top (Wang *et al.* (2018))). Nous avons pris en compte toutes les types d'erreurs qu'une transaction 6p peut retourner. Ensuite, nous avons calculé la moyenne de ces transactions d'erreur en fonction du nombre de nœuds.

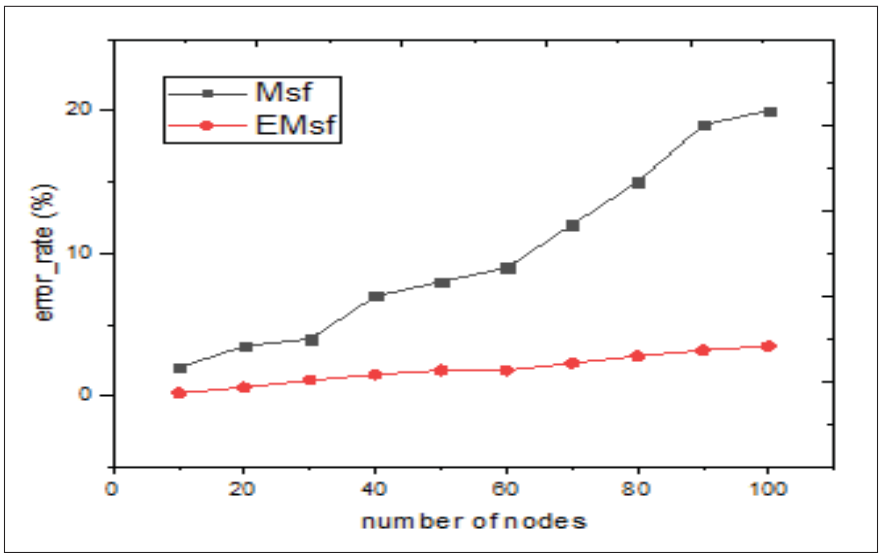


Figure 3.3 Taux d'erreur des messages 6p.

Le taux d'erreur de négociation augmente avec l'augmentation de la densité du réseau. Dans le cas où nous avons implémenté MSF, le taux d'erreur augmente considérablement de 2.1% jusqu'à 19.8%, alors qu'en implémentant EMSF le taux augmente de 0.2% jusqu'à 3.5%. Le mécanisme proposé, EMSF, surpasse largement MSF et maintient un taux d'erreur de négociation moins de 3,5% pour toutes les densités du réseau. Cela est dû à la substitution du mécanisme de mesure de la bande passante, implémenté dans MSF, nécessaire à la communication entre une paire de nœuds par le calcul de prédiction effectué indépendamment dans chaque nœud et qui est implémenté dans EMSF. Autrement, EMSF permet de réduire le nombre de paquets de contrôles échangés dans le réseaum, ce qui permet au nœud d'envoyer des transactions précises d'ajout ou de suppression de cellules.

3.3.2 Charge de trafic supplémentaire

Dans les réseaux 802.15.4e à mode TSCH, la planification des cellules qui forme l'ordonnancement d'un réseau peut être soit statique ou dynamique. La réservation statique alloue les cellules une fois au début et conserve cette planification jusqu'à ce que le nœud quitte le réseau. Au contraire, la réservation dynamique des cellules se fait uniquement au besoin, comme par exemple en réponse aux fluctuations du trafic d'un nœud. Cependant, la négociation pour l'ajout ou la suppression des cellules génère un nombre important de charge de trafic supplémentaire.

Dans la simulation suivante, nous avons calculé le nombre de paquets de contrôle et de mesure qu'une paire de nœuds transmet afin de décider le nombre de cellules obligatoires dans la prochaine *slotframe*. Nous avons augmenté progressivement le nombre de nœuds dans le réseau de 2 jusqu'à 100, par des intervalles de 10. Le protocole MSF implémente l'algorithme d'estimation de la bande passante pour traduire les besoins des nœuds en un nombre de cellules.

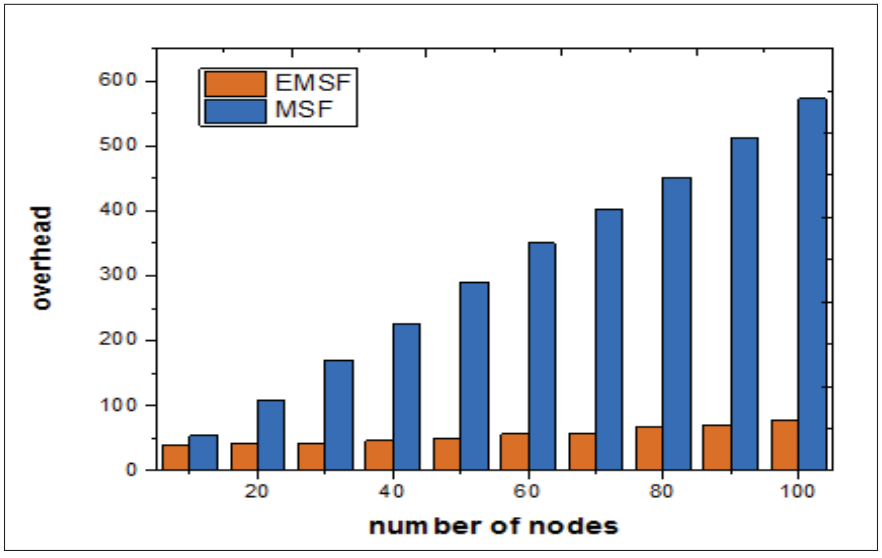


Figure 3.4 Charge de trafic supplémentaire.

Ce dernier surveille la quantité de données envoyées entre chaque paire de nœuds. Quand cette quantité devienne grande ou petite (atteint un seuil déterminé) par rapport au nombre

de cellules déjà allouées, MSF demande à 6top d'ajouter ou de supprimer des cellules avec le nœud désigné. Ce processus génère la transmission de certains paquets de contrôle, qui entraine une charge de trafic supplémentaire.

La Figure 3.4 représente la charge de trafic supplémentaire (mesurée en octets), utilisée par les nœuds pour échanger des informations 6p dans le réseau. Nous remarquons que le nombre de messages échangés augmente linéairement avec le nombre de nœuds déployés. Cela est dû aux échanges de négociation faites entre les paires de nœuds pour déterminer la transaction 6p à déployer comme réponse à un changement de l'ordonnancement. Nous constatons aussi que le protocole EMSF maintient un nombre presque constant de paquets échangés dans le réseau. Cela est dû à l'algorithme de prédiction qui prévoit le nombre de cellules requises pour chaque paire de nœuds pendant la prochaine *slotframe*. Il élimine la transmission de transactions échangés entre une paire de nœuds dans le protocole MSF pour déterminer la bande passante nécessaire. Le protocole EMSF évite d'envoyer des surcharges et conserve une moyenne constante des paquets de contrôle pendant la durée de vie du réseau. À noter également que MSF ne réserve pas de cellules supplémentaires pour gérer des charges de trafic imprévues et, comme elles ne prévoient pas le trafic récurrent, les nœuds doivent envoyer plus de charges de trafic 6p.

3.3.3 Temps de latence

Une des métriques les plus importantes pour détecter et signaler l'état d'urgence (détection d'un évènement brusque) dans les réseaux 802.15.4e à mode TSCH est le temps de latence. Dans les méthodes d'ordonnancement conventionnelles, si un paquet est rejeté en raison d'une mauvaise qualité de lien radio ou en raison d'une collision, une retransmission au niveau de la couche MAC se produit. Cela mène à réserver plus de cellules pour une paire de nœuds dans une *slotframe* tout en empêchant les autres nœuds à réserver des cellules pour envoyer ou recevoir des paquets durant le reste de l'ordonnancement.

Dans l'objectif d'étudier le temps de latence de bout en bout de notre proposition EMSF, nous avons simulé un réseau composé de 20 nœuds. Chaque nœud dans le réseau génère, pendant les 50 premières *slotframes*, une charge de trafic fixe égale à 2 paquets par *slotframe*. Ensuite, chaque nœud génère une charge de trafic sporadique allant de 2 à jusqu'à 7 paquets par *slotframe*. Nous avons simulé le même réseau sous les mêmes conditions en implémentant le protocole MSF dans le but d'approuver EMSF en termes de temps de latence. La Figure 3.5 présente une comparaison entre MSF et EMSF en terme de temps de latence.

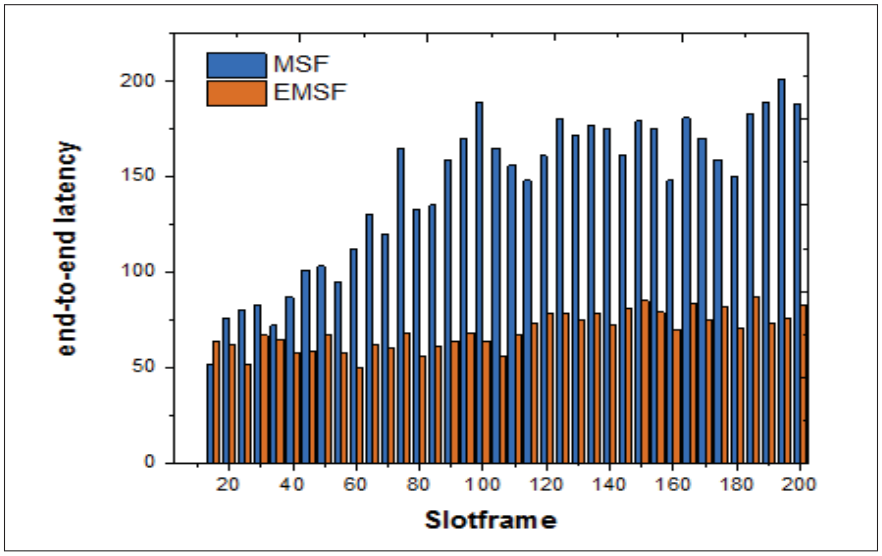


Figure 3.5 Temps de latence de bout en bout.

Nous avons calculé le temps de latence de la façon suivante : chaque paquet est horodaté depuis le temps qu'il a été généré au niveau de la couche application d'un nœud source jusqu'à ce qu'il atteigne la couche application du nœud racine DAGroot. Les retransmissions au niveau de la couche MAC ne sont pas prises en considération. Lorsqu'un paquet atteint 4 tentatives de retransmission et la mémoire de la file d'attente du nœud devienne pleine, le paquet sera considérablement rejeté. Cependant, si un paquet est transmis à plusieurs reprises, une augmentation du temps de latence peut survenir. Chaque tentative de retransmission d'un paquet génère une augmentation du temps de latence, puisque chaque paquet prend plus de temps que la durée qui lui a été allouée pour atteindre sa destination. Le temps de latence était presque constant durant les 50 premières *slotframes* en raison de flux de trafic de données stable qui a été générés par

les nœuds du réseau. Par la suite, la latence varie entre les cycles en raison des transmissions stochastiques défaillantes, basées sur la valeur de PDR entre les nœuds. Nous remarquons que le protocole EMSF maintient un temps de latence de bout en bout inférieur à 75 millisecondes pendant toutes les *slotframes*. Cela est dû à la réduction des erreurs de négociation ainsi que des collisions, que le système d'ordonnancement du protocole EMSF garantit en se basant sur le calcul de prédiction. Par conséquent, la charge de trafic supplémentaire est réduite, ce qui résulte à une valeur élevée de PDR lorsque des flux de données irréguliers sont générés.

3.3.4 Taille de la file d'attente

Le flux de trafic dans un réseau 802.15.4e à mode TSCH varie en fonction de l'emplacement du nœud dans le DODAG. Les nœuds proches de la racine ont une charge de trafic supérieure à celle des nœuds terminaux. Suite à la détection d'un évènement brusque par un groupe de nœuds, une charge de trafic élevée sera transportée dans le réseau. Lorsque ce trafic atteint les nœuds proches de la racine, il y'aura une accumulation de paquets dans les files d'attente de leurs mémoires. En conséquence, chacun de ces nœuds aura besoin de plusieurs cellules dans une *slotframe* pour vider les paquets de sa file d'attente. Cela va entrainer des retards indésirables, affectant négativement le délai de la communication globale. Ce délai est particulièrement inacceptable lorsque l'application doit fournir des données critiques telles que des alarmes.

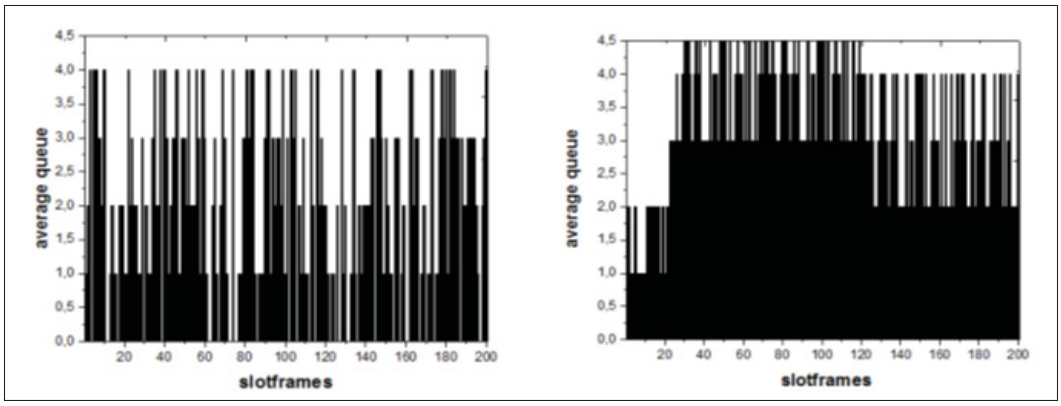


Figure 3.6 Taille de la file d'attente.

Afin d'étudier l'impact de l'algorithme de prédiction proposé dans le protocole EMSF sur la taille de la file d'attente de la mémoire, nous avons simulé un réseau de 100 nœuds et nous avons comparé le résultat obtenu en implémentant le protocole MSF. La taille de la mémoire de chaque nœud peut contenir dans la file d'attente au maximum 5 paquets. Une fois cette mémoire est pleine, le premier paquet arrivé dans la mémoire sera rejeté en premier. Pendant chaque *slotframe*, nous avons calculé la moyenne du nombre de paquets dans les files d'attente de tous les nœuds dans le réseau.

La Figure 3.6 représente le nombre moyen de paquets dans les files d'attente des nœuds du réseau. Dans cette figure, on peut voir que, dans le cas du protocole MSF, les files d'attente des mémoires des nœuds sont presque pleines et atteignent leurs maximums (5 paquets) dans certaines *slotframes*, ce qui entraîne davantage de paquets perdus. Par ailleurs, le protocole EMSF présente des meilleures performances en gardant une file d'attente pratiquement modérée, affichant une moyenne allant de 0 à 4 paquets dans la file d'attente durant les 200 *slotframes*. En particulier, les nœuds qui se localisent à coté de nœud DAGroot, surallouent un nombre de cellules supérieur à ceux qui lui sont distants. Le flux de paquets élevé traversant ces nœuds permet à l'algorithme de prédiction, proposé dans le protocole EMSF, de réserver plus de cellules. Par conséquent, ceci permet de garantir une bande passante suffisante, qui satisfait aux besoins fluctuantes des paires de nœuds.

3.4 Discussions

En se basant sur les résultats de simulation obtenus ci-dessus, nous pouvons conclure les avantages et les inconvénients de l'algorithme proposé dans le protocole EMSF :

- En termes de transactions 6p échangés : le protocole EMSF permet de réduire largement le taux d'erreurs des transactions 6p par rapport au protocole MSF.
- En termes de charge de trafic : grâce au mécanisme de prédiction proposé dans le protocole EMSF, chaque paire de nœuds détermine ses besoins en bande passante d'une façon auto-

ritaire, c'est-à-dire sans avoir recours aux échanges de paquets de contrôle. Ceci diminue considérablement la charge de trafic supplémentaire.

- En termes de temps de latence : en réduisant le taux d'erreur des transmissions des paquets, le nombre de tentatives de retransmission des paquets diminue considérablement. Ceci donne un temps de latence de bout en bout réduit par rapport au protocole MSF.
- En termes de la taille de la file d'attente : grâce au mécanisme de prédiction, dans le cas où un flux de trafic massif est généré, les nœuds se retrouvent déjà alloués des cellules supplémentaires qui peuvent faire face à ce flux de trafic. En contrepartie, certaines paires de nœuds se trouvent avec des cellules réservées dont ils n'en ont pas besoin. Ce qui gaspille l'énergie des nœuds en mettant leur antenne en état actif alors qu'ils ne vont rien envoyer ni recevoir.

3.5 Conclusion

À partir des résultats de simulations trouvés, nous déduisons que le protocole proposé EMSF surpasse le protocole MSF, qui est implémenté par défaut dans les réseaux 6TiSCH. Le protocole EMSF permet de diminuer la charge de trafic qui circule dans le réseau et le taux d'erreurs de transactions 6p, réduire le temps de latence de bout en bout et ainsi optimiser la taille de la mémoire de file d'attente.

D'autre part, dans le protocole EMSF chaque nœud exécute un nombre de calcul supplémentaire (calcul de prédiction) par rapport de celui fait dans le protocole MSF. Cela cause une consommation plus élevée d'énergie. Néanmoins, les réseaux 6TiCH sont déployés dans l'industrie, ce qui signifie que le temps de latence constitue la contrainte la plus importante.

CONCLUSION ET RECOMMANDATIONS

Dans ce mémoire, nous avons abordé l'un des principaux défis auquel sont confrontés les réseaux de capteurs sans fil implémentés dans l'industrie. Ce défi consiste à transmettre un évènement physique, détecté par un noeud capteur, vers l'entité principale du réseau, pour être traité dans un laps de temps minimum.

Nous avons d'abord formulé le mode de transmission des paquets dans un réseau 802.15.4e à mode TSCH en un processus de Poisson. En se basant sur un calcul de probabilité faite entre chaque noeud parent et son fils, nous avons pu prédire le nombre de paquets qui sera échangé entre cette paire de noeuds. Cette prédiction nous a permis de déduire le nombre de cellules nécessaires dans la prochaine slotframe. Par conséquent, la charge de trafic circulée dans le réseau, utilisée pour déterminer l'ordonnancement, a été réduite considérablement.

Nous avons ensuite effectué nos simulations sur OpenWSN afin d'approuver notre solution. Les résultats ont démontré que l'algorithme que nous avons proposé a réalisé des meilleures performance en permettant de : 1) Réduire le taux d'erreur des messages 6p, 2) Réduire la charge de trafic supplémentaire, 3) Réduire la taille de la mémoire de la file d'attente et 4) minimiser le temps de latence de bout en bout.

L'importance de notre travail consiste à définir un ordonnancement global du réseau en partant d'une simple négociation et d'un calcul de prédiction entre une paire de noeuds. La nature distribuée de notre modèle a permis de réduire considérablement le nombre de messages échangés et le temps de latence, qui représente l'un des facteurs les plus importants dans un réseau de capteurs sans fil industriel.

Dans une perspective future, on compte intégrer à notre modèle une méthode de sélection de cellules, qui donne priorité aux noeuds les plus proches du noeud racine. Cela va offrir plus de bande passante aux noeuds qui ont plus de trafic, ainsi le temps de latence sera encore réduit.

De plus, vu la nature distribuée de notre modèle et la quantité de calcul considérée au niveau de chaque noeud, nous avons l'intention d'introduire une méthode, qui se base sur la sélection des liens entre les noeuds, pour déterminer l'ordonnancement afin d'alléger le calcul et par conséquent diminuer la consommation d'énergie.

ANNEXE I

ARTICLES PUBLIÉS

1. Article 1

Cet article, intitulé "A Survey on Intelligent MAC Layer JammingAttacks and Countermeasures in WSNs", a été publié à IEEE 84th Vehicular Technology Conference (VTC-Fall 2016, Montréal).

A Survey on Intelligent MAC Layer Jamming Attacks and Countermeasures in WSNs

Taieb Hamza , Georges Kaddoum , Aref Meddeb , Georges Matar
Department of Electrical Engineering, LACIME Laboratory, University of Quebec, ETS, Montreal, Canada.
Emails: taieb.hamza.1, georges.matar.1 @ens.etsmtl.ca, georges.kaddoum@etsmtl.ca
National Engineering School Of Sousse, NOCCS Laboratory, University Of Sousse, Sousse, Tunisia.
Email: aref.meddeb@infcom.rnu.tn

***Abstract*—Security abides a tremendous key requirement in the context of Internet of Things (IoT). IoT connects multiple objects together through wired and wireless connections in the aim of enabling ubiquitous interaction where any components can communicate with each other without any constraint. One of the most important elements in the IoT concept is Wireless Sensor Network (WSN). Due to their unattended and shared nature of radio for communication, security becomes an important issue. Wireless sensor nodes are vulnerable to radio jamming. When the jammer has the ability to interpret data link layer protocols, it becomes as energy-efficient as legitimate nodes. This paper presents a comprehensive survey on different sophisticated jamming attacks based on MAC layer. Techniques used to defeat each one of the intelligent jammers are classified based on the knowledge capacity of MAC protocols rules. The concepts behind existing protocols, that are dedicated by design to defeat such type of jammers, are presented. We conclude by a recapitulative table summarizing jamming attacks and proposed MAC-based solutions, and highlight open research directions.**

***Index Terms*—Wireless sensor networks, MAC layer security, energy-efficient jamming attack, intelligent jammer.**

I. Introduction

The Internet of Things (IoT) is a set of electronic devices interconnected and linked to the existing internet infrastructure [1]. Examples include cars with integrated sensors, smart thermostats, smart homes, remote monitoring systems and many other areas. According to several studies and predictions done by Gartner Inc [2], the IoT will include 26 billion units installed by 2020. Due to the multiple aspects involved, IoT security will be a key concern that should be addressed to ensure the reliability of these components and their service despite the existing risks.

Wireless sensor networks (WSNs) form the major part of the IoT [3]. They are used in many fields such as health, agriculture, environment and detection of natural disasters. The fundamental characteristics of WSN make it vulnerable to attacks due to the operating nature of its components (wireless broadcasting). This will exhibit them to passive and active attacks, which vary by nature and goals. Wireless communications are exposed to eavesdropping and signal interception [4]. One such attack that can occur almost in all environments is radio jamming [5] [6] [7] [8].

MAC protocols are vulnerable to intelligent jamming attacks. Some of the existing protocols fail to deliver any packet in the case that they are being exposed to a jammer. In case of large scale networks with high-energy constraints, nodes are scheduled to communicate and sleep simultaneously as in TDMA-based protocols [9]. The aim is to avoid overhearing and idle listening that represent the major sources of energy loss. However, such a temporal ordering can introduce a pattern of communication that allows the attacker to predict the next communication cycle. This can be exploited by a malicious node to launch an energy-efficient attack [10]. Accordingly, jammed signals coincide with the packets sent from legitimate network nodes. The malicious node will be able to exploit the temporal pattern of communication and block sending legitimate packets with a small set of jamming pulses. Therefore, it achieves the same energy-efficiency as network nodes and becomes more difficult to spot.

Based on the communication pattern, a mobile jammer can choose the right attack area [11]. First, it chooses the regions with the highest communication flow and then it begins to attack. Thus, the neighboring nodes will suffer the most. However, nodes such as the sink or the Personal Area Network (PAN) in WSNs are very important due to the fact that they require more energy resources and high channel utilization. This leads to high cost of operation with a low message delivery rate. Furthermore, by listening to the traffic in multi-channel networks [12], an intelligent jammer could elicit the control channel. It sends continuous jammed signals in order to block channel negotiation. Moreover, it can extract the sequence of next control channels from legitimate nodes, which will damage the whole network.

In this paper we present a detailed study of jamming attacks with emphasis on sophisticated energy-efficient link-layer jammers. Our objective is to provide a general overview of the critical issues when facing an intelligent jammer and cover the works that aim to defeat these attacks. Thus, our contribution can be outlined as follows: Providing a brief analysis of basic and intelligent jammers, developing a literature review of different link-layer jamming methods and existing countermeasures.

The remainder of this paper is outlined as follow: a categorization of existing jammers in the context of WSN is depicted in Section II. Link-layer intelligent jammers and propsed solutions to defeat their leverage are presented in Section III. Discussions and conclusion are presented in section IV.

II. Jamming attacks

A. Basic jammers

In [6], authors introduced empirical methods based on signal strength indicators and packet delivery ratio measurements to identify jamming attacks. They studied four types of jammers: constant, random, deceptive and reactive. The first one is based on the physical layer while the latter ones are based on the MAC layer. Constant jammers send random bits continuously in the channel without following any MAC label. Deceptive jammers continuously inject regular packets to the channel without any space between subsequent packet transmissions. Therefore, a node will be fooled into believing that it is a legitimate packet and would remain in the receiving state. Instead of sending a constant radio signal, random jammers alternate between jamming and sleeping. The times of attack and sleep can vary, which allows a malicious node to achieve different levels of compromise between energy-efficiency and the effectiveness of jamming, while depending on the application. Reactive jammers [13] [6] do not necessary block the channel when there is no communication. They begin to emit a radio signal when they detect activity on the channel.

B. Intelligent jammers

Some jammers target directly the physical layer. They were designed to destroy the signal, congest the network and require the nodes to consume more energy. Other jammers target the network privacy by targeting the MAC layer. They aim to determine which MAC protocol is used by victim nodes in order to launch an energy-efficient attack. MAC layer jammers have a better energy efficiency than physical layer jammers. They target directly the data packets, while other jammers focus on any type of packet. The MAC layer jammer is more likely to jam, but harder to implement than the physical layer jammer [14]. An intelligent jammer called protocol-aware that knows the MAC layer operating rules can deprive legitimate nodes access to the channel. Such an attack is more difficult to detect, because the jammer knows the current MAC protocol rules. A statistical jammer observes the packet inter-arrival distribution and, based on this estimation, emits pulses of jammed signals in order to disrupt the communication in an energy-efficient manner. Fig 1 depicts the stealth and the energy-efficiency of different types of jammers.

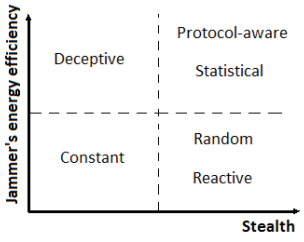


Fig. 1. Jammer's energy efficiency vs Stealth.

Some of these attacks may target the control signal such as acknowledgment (ACK) packets. These typical jammers are called collision-makers [15]. For instance, if the ACK is required to send a data packet or a beacon frame, an interfering node may simply inhibit the transmission, which forces a PAN coordinator to resume sending and therefore consume more energy and bandwidth. Several solutions in the MAC layer level have been proposed to defeat intelligent jammers such as frame masking, frequency hopping, packet fragmentation and redundant coding to reduce the impact of damage caused by a jammer [10]. In the case of SMAC protocol [16], the authors proposed a high duty cycle [10]. In the case of LMAC protocol [17], they proposed encrypting data and using short size packets before transmission.

III. Energy-Efficient Link-Layer Jamming Attacks

A. Protocol-aware jammers

Table I cites intelligent MAC jammers and discusses proposed protocols to defeat each attack while highlighting the strengths and weaknesses of all the categories proposed in this section. Wood et al. [10] propose DEEJAM, a protocol that helps to defeat energy-efficient jamming based on IEEE 802.15.4-compatible hardware [18]. The main objective of the protocol is to hide messages from attacker, evade its search and reduce the impact of corrupted messages. DEEJAM is able to maintain a 88% packet delivery ratio under any circumstance. This results in a novel protocol, allowing network nodes to function effectively even in the presence of a jammer. This work contributed to define, implement and evaluate four classes of jamming attacks namely scan, pulse, activity and interrupt jamming. Four solutions reducing the possibility of an effective jamming were proposed: frame masking, channel hopping, packet fragmentation, and redundant encoding.

The work in [19] presents two modifications to the LMAC protocol: Data Packet Separation (DS-SSR) and Round Robin (RR). DS-SSR separates a data packet into parts to make the network traffic seem faster. This would mislead the jammer, who is unaware that the slot size is much smaller than it is in reality, therefore making him jam more and lose power earlier than normal. RR slot assignment technique improves the network throughput by guaranteeing that all nodes transmit for the same quantity of time. Authors proved that the advantage of reducing by 8% the jammer's lifetime was achieved by applying a data separation technique.

EM-MAC [20] is an asynchronous predictive duty-cycling MAC protocol that uses multiple orthogonal radio channels for communication. It lies to the category of receiver-initiated MAC [21]. A sender would wake up before the receiver, transmit a packet and then go into a sleeping state in order to reduce both idle listening and overhearing. Any pseudo-random function can be used by EM-MAC to generate the channel and node wake-up schedules. In order to optimize a set of channels and avoid choosing jammed or congested channels, each node in EM-MAC uses a dynamic mechanism based on channel conditions.

Moreover, energy link-layer jamming attack using game theory is proposed in [22]. The authors propose Jam-Buster, a jam-resistant protocol that aims to stamp out the differentiation

between packets by using equal sizes, randomize the wake-up times to defeat schedule prediction and implements multi-bloc payloads to enhance network resilience. These three techniques are combined to cope with an intelligent jammer and force it to spend more energy to be effective. Authors evaluated energy consumption only on jammer's side whereas the lifetime of legitimate nodes was not considered. Since Jumbuster behaves as a proactive defense aginst a jammer, other MAC constraints such as overhearing, idle listening and end-to-end delay communication should be also approved.

In [23], authors present SAD-SJ, a Self-Adaptive and Decentralized MAC-layer solution against Selective Jamming in TDMA-based WSNs. The protocol permits to neutralize selective jamming making the attack random. SAD-SJ is based on a random slot reallocation where each node achieves a random permutation of slots. The permutation process can be done after generating a random number. The protocol was proved to be self-adaptive in that it allowed nodes to freely join and leave yet keeping security of other nodes intact. It does not reduce performance and the additional energy consumed is insignificant.

B. Statistical jammers

The individual characteristics of the proposed protocols to defeat statistical MAC jammers, are reviewed in table I. Law et al. [24] show that an attacker is able to learn about S-MAC parameters by listening to the channel for some period of time. As shown in Fig. 2, a period jam-sleep schedule can be adopted once a jamming mote is able to estimate and . A jamming mote will be asleep when the neighbors are asleep and jamming when the neighbours are listening.

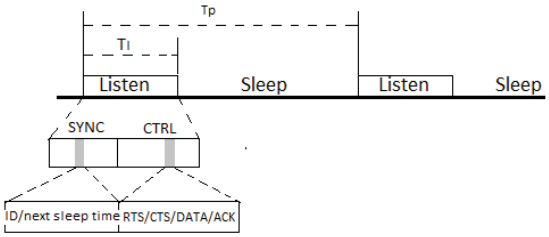


Fig. 2. SMAC Schedule [16].

CTRL jamming attack forces nodes to consume more energy than other attacks which makes it a much more successful type of attack. Thus, it allows the victim nodes to consume energy by listening for the entire CTRL and sleep interval, forcing them to send SYNC packets which are sent in the SYNC interval. Data packet jamming in censorship rate has been shown to be better than other attacks by a small margin, while CTRL interval jamming was shown to have the best advantage in terms of its lifetime. Finally, as a countermeasure to data packet jamming, authors proposed a new technique based on schedule switching and data blurting.

In [25], authors proposed a Wispernet protocol in order to defeat statistical jammers. The aim is to diminish or remove spatio-temporal communication patterns while at the same time managing to maintain energy-efficient, synchronized and collision-free operations in WSNs. In order to prevent the epoch and length of next channel activity from being predicted by the jammer, the Wispernet protocol adds temporal randomization for slot durations and schedules, which coordinates between every node and its k-hop neighbor. This will result in the statistical jammer being of same degree of efficiency as a random pulse jammer and prevents preemptive attacks from occurring, since the jammer cannot determine packet inter-arrival times. The censorship is positively correlated with channel utilization and jamming activity. Furthermore, adaptive routing is employed in order to avoid random jammers that are co-located close to nodes with active routes. This is achieved by selecting routes that are most likely to have the highest end-to-end packet delivery ratio. The protocol aims to guarantee non predictable schedules, non predictable slot sizes, coordinated and scheduled transmissions, coordinated changes in slot sizes and collision-free transmissions. Some issues may seldom occur in slot conflict resolution, in particular in low duty cycle networks, which would result in lower end-to-end bandwidth and delay the message. WisperNet Spacial routing is not very scalable to large networks of over 500 nodes, and needs additional memory resources.

In [15], the effectiveness of link layer jamming attacks was shown to be the same as constant, deceptive and reactive jamming while being more energy efficient. The fundamental idea is that data packets need to be found and jammed, but their arrival times are difficult to predict in order to be attacked, due to their random nature. The probability distribution of inter-arrival times of packets is the proposed solution to jam the network. Authors studied 3 protocols, S-MAC [16], B-MAC [27] and L-MAC [17]. Two separated clusters are represented by the S-MAC probability distribution of packet inter-arrival times and are kept in a way that even if mobile nodes or data packets of varying lengths are used. If the jammer can define the packet inter-arrival model prediction, then the attack becomes feasible. L-MAC protocol is a bit different, since the probability distribution of packets inter-arrival times is different, but the same cluster prediction could be applied. The clusters idea in case of B-MAC protocol is impossible to use since it uses periodic listening cycles only, which also tends to save energy, not periodic sent cycles. To do a jamming attack on B-MAC, the victim nodes preamble check interval needs to be determined. Link layer jamming attacks can be prevented in a few different ways. One way would be to prevent clustering based analysis on S-MAC, by reducing the distance between the clusters. Another way would be to make L-MACs clusters harder to estimate. This needs the packet transmission slot sizes to be changed pseudo-randomly as a function of time (e.g. the sensor node changes its packet slot size each second by choosing a random value). In case of B-MAC, the preamble needs to be shortened up to a minimum of 10 ms so that it is harder to detect. The authors didn't suggest any type of spectrum technique as a countermeasure and the use of spread spectrum hardware wasn't tested nor simulated.

TABLE I
COMPARISON BETWEEN DIFFERENT INTELLIGENT JAMMERS AND PROPOSED COUNTERMEASURES

Protocols	Proposed Attacks	Countermeasures	Strengths	Weaknesses	Performance metrics	Category
DEEJAM [10]	-Interrupt jamming -Activity jamming -Scan jamming -Pulse jamming	-Hide messages from the jammer -Evade its research -Redundant encoding	-Significant reduction in jamming impact -High PDR despite jamming	-High overhead -Expanded latency	Hardware	Protocol-aware
EM-MAC [20]	-Continuous Jamming	-Avoiding jammed channels selection -Randomization of wake up schedules	-Enhancement of multi-channel utilization -High energy-efficiency -Low duty cycle -High delivery ratio	-Need for high accurate synchronization	Hardware	Protocol-aware
Jam-Buster [22]	-Schedule prediction	-Multi-bloc payloads -Randomization of wake up times -Using equal size packets	-Forcing the jammer to ruin its battery -Better utilization of power resources	-No energy model support	Hardware	Protocol-aware
SAD-SJ [23]	-Transmitting malicious signals during slots of the super-frame	-Random permutation of slot times -Network dynamicity management	-Limited overhead -Self-adaptive -Decentralized -Significant reduction in collisions between nodes transmission	-Additional energy consumption	Simulation	Protocol-aware
SMAC [24]	-Data packet jamming -CTRL interval jamming -Listen interval jamming	-Schedule switching -Data blurting	-Reducing jammer's lifetime -Low censorship rate	-Low throughput -Synchronization issues	Simulation	Statistical
Wispernet [25]	-Predict the following communication schedule	-Adaptive routing -Schedule randomization -Slot size randomization	-Reducing jammer's lifetime -Maintaining coordination -Contention-free communication	-High Overhead -Need for additional memory -High end-to-end delay	Simulation Hardware	Statistical
-SMAC [15] -BMAC [15] -LMAC [15]	-Estimate the probability distribution of packets inter-arrival times	-Reduce the time between the clusters (SMAC) -Randomize the packet transmission (LMAC) -Shortening the preamble size (BMAC)	-Low censorship rate -Waning jammers lifetime (high attrition rate) -High effort ratio.	-No spread spectrum techniques proposed -High processing delay	Simulation Hardware	Statistical
IEEE 802.15.4 MAC [26]	-Passive listening -Man In The Middle	-Message authentication	-Manipulates 802.15.4 capabilities with minimum modifications	-Not suitable for ad hoc architecture	Hardware	Statistical

Authors in [28] define a sniper attack, which exploits the GTS allocation decision so that it jams the longest slot. The Jammer will produce a collision during specific slots leading to DoS attack. Furthermore, the work in [26] targets DoS attack in IEEE 802.15.4 nodes since encryption covers only MAC payload. A sophisticated jammer could intercept ACK messages of specific nodes and block only these messages. A Jammer sends a modified message and could start to begin a *Man In The Middle* attack [29]. Authors propose message authentication as a preemptive defense. Link layer jamming study included slot-based MAC protocols (T-MAC [30], D-MAC [31]), frame-based protocols (BMA [32]), and random access-based protocols (PCM [33], WiseMAC [34]).

IV. DISCUSSIONS AND CONCLUSION

WSN constitutes the most important element in IoT since they connect industrial entities (new standards are under development such as 802.15.4e [35]), persons (e.g BAN), vehicles (e.g VAN) and other networks to each other through the internet. Jamming attacks are considered a dangerous threat since they may cause a total DoS, especially when the attacker is intelligent. After evaluating various jamming and anti-jamming techniques, we conclude that, at present, there is no common anti-jamming technique which can be applied to all kinds of jammers. Power saving was the primary focus of research. However, since jammers became more sophisticated and started to act like legitimate nodes, studying link-layer defenses strategies is becoming a crucial task to face intelligent jammers. The purpose is to transform a selective jammer into a random one. The proposed solutions based on MAC layer aim to create a high entry barrier for jammers by encrypting link-layer packets, using random forward shifting, adapting a TDMA protocol and using randomized intervals. Regardless of which MAC protocol the nodes are adapting, a less intelligent jamming can rattle a sensor node using a single channel for communication. Thereby returning in needs to implement multi-channel MAC protocol in order to make the network more resilient against smart jammers. In addition, there are increasingly more new wireless network technologies (e.g. 5G networks, vehicular networks), making anti-jamming a more challenging issue, especially when compromising between network's requirement constraints and security purpose. The upcoming IEEE 802.15.4e standard aims to support industrial markets that are time-critical based applications. It introduces channel hopping and secured acknowledgments as an IEEE standard. Despite the fact that these applications require higher security, aspects and policies against smart jammers are not

well considered which makes these networks vulnerable to smart jammers. On the other hand, network latency and security should be improved without negatively affecting energy consumption.

In this paper we have surveyed the challenges relevant to link-layer security in the context of wireless sensor networks. A literature review on intelligent jammers, based on MAC knowledge degree and the capacity to learn statistics from observed traffic flow, was detailed. The article classifies intelligent jammers in two main categories: protocol-aware and statistical, highlighting the methods to defend against each type of attacks. Moreover, it features open research issues in the field of industrial WSN and new network technologies to emphasize the extensive importance of security against intelligent attackers.

REFERENCES

[1] R. Minerva, A. Biru, and D. Rotondi, "Towards a definition of the internet of things (iot)," *IEEE Internet Initiative, Torino, Italy*, 2015.

[2] C. STAMFORD. (2016) Gartner says by 2020, more than half of major new business processes and systems will incorporate some element of the internet of things. [Online]. Available: http://www.gartner.com/newsroom/id/3185623

[3] L. Mainetti, L. Patrono, and A. Vilei, "Evolution of wireless sensor networks towards the internet of things: A survey," in *19th International Conference on Software, Telecommunications and Computer Networks (SoftCOM),*, Sept 2011, pp. 1–6.

[4] Y.-W. P. Hong, P.-C. Lan, and C.-C. J. Kuo, *Signal processing approaches to secure physical layer communications in multi-antenna wireless systems*. Springer Science & Business Media, 2013.

[5] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in wsns," *IEEE Comun. Surveys Tutorials*, vol. 11, no. 4, pp. 42–56, Fourth 2009.

[6] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2005, pp. 46–57.

[7] G. Noubir and G. Lin, "Low-power dos attacks in data wireless lans and countermeasures," *ACM SIGMOBILE Mobile Computing and Comun. Review*, vol. 7, no. 3, pp. 29–30, 2003.

[8] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *Network, iEEE*, vol. 20, no. 3, pp. 41–47, 2006.

[9] A. S. Althobaiti and M. Abdullah, "Medium access control protocols for wireless sensor networks classifications and cross-layering," *Procedia Computer Science*, vol. 65, pp. 4–16, 2015.

[10] A. D. Wood, J. A. Stankovic, and G. Zhou, "Deejam: Defeating energy-efficient jamming in ieee 802.15. 4-based wireless networks," in *Society Conference on Sensor, Mesh and Ad Hoc Comun. and Networks. SECON'07. 4th Annual IEEE Comun.* IEEE, 2007, pp. 60–69.

[11] T. Cheng and P. Li, "An algorithm for mobile jammer localization in wireless sensor networks," in *Proceedings of the 2012 Second International Conference on Electric Information and Control Engineering - Volume 03*, ser. ICEICE '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 90–94. [Online]. Available: http://dx.doi.org/10.1109/ICEICE.2012.915

[12] L. Lazos, S. Liu, and M. Krunz, "Mitigating control-channel jamming attacks in multi-channel ad hoc networks," in *Proceedings of the second ACM conference on Wireless network security*. ACM, 2009, pp. 169–180.

[13] M. Wilhelm, I. Martinovic, J. B. Schmitt, and V. Lenders, "Short paper: reactive jamming in wireless networks: how realistic is the threat?" in *Proceedings of the fourth ACM conference on Wireless network security*. ACM, 2011, pp. 47–52.

[14] A. Tayebi, S. Berber, and A. Swain, "Wireless sensor network attacks: An overview and critical analysis," in *Seventh International Conference on Sensing Technology (ICST)*. IEEE, 2013, pp. 97–102.

[15] Y. W. Law, M. Palaniswami, L. V. Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-efficient link-layer jamming attacks against wsn mac protocols," *ACM Trans on Sensors Networks*, vol. 5, no. 1, pp. 1–38, 2009.

[16] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Comun. Societies. Proceedings. IEEE*, vol. 3. IEEE, 2002, pp. 1567–1576.

[17] L. F. Van Hoesel and P. J. Havinga, "A lightweight medium access protocol (lmac) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches," 2004.

[18] "Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpans)," *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pp. 1–320, Sept 2006.

[19] A. R. Mahmood, H. H. Aly, and M. N. El-Derini, "Defending against energy efficient link layer jamming denial of service attack in wireless sensor networks," in *9th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2011, pp. 38–45.

[20] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "Em-mac: a dynamic multichannel energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2011, p. 23.

[21] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, "The evolution of mac protocols in wireless sensor networks: A survey," *Comun. Surveys & Tutorials, IEEE*, vol. 15, no. 1, pp. 101–120, 2013.

[22] F. Ashraf, Y.-C. Hu, and R. H. Kravets, "Bankrupting the jammer in wsn," in *9th International Conference on Mobile Adhoc and Sensor Systems (MASS)*. IEEE, 2012, pp. 317–325.

[23] M. Tiloca, D. De Guglielmo, G. Dini, and G. Anastasi, "Sad-sj: A self-adaptive decentralized solution against selective jamming attack in wireless sensor networks," in *18th Conference on Emerging Technologies & Factory Automation (ETFA)*. IEEE, 2013, pp. 1–8.

[24] Y. W. Law, P. Hartel, J. D. Hartog, and P. Havinga, "Link-layer jamming attacks on s-mac," in *Proceeedings of the Second European Workshop on Wireless Sensor Networks, 2005.* IEEE, 2005, pp. 217–225.

[25] M. Pajic and R. Mangharam, "Wispernet: Anti-jamming for wireless sensor networks," *Departmental Papers (ESE)*, p. 526, 2008.

[26] C. P. O'Flynn, "Message denial and alteration on ieee 802.15. 4 low-power radio networks," in *4th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2011, pp. 1–5.

[27] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*. ACM, 2004, pp. 95–107.

[28] R. Daidone, G. Dini, and M. Tiloca, "A solution to the gts-based selective jamming attack on ieee 802.15. 4 networks," *Wireless networks*, vol. 20, no. 5, pp. 1223–1235, 2014.

[29] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Comun. Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2016.

[30] T. Van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*. ACM, 2003, pp. 171–180.

[31] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks," in *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*. IEEE, 2004, p. 224.

[32] J. Li and G. Y. Lazarou, "A bit-map-assisted energy-efficient mac scheme for wireless sensor networks," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 55–60.

[33] E.-S. Jung and N. H. Vaidya, "A power control mac protocol for ad hoc networks," in *Proceedings of the 8th annual international conference on Mobile computing and networking*. ACM, 2002, pp. 36–47.

[34] A. El-Hoiydi and J.-D. Decotignie, "Wisemac: an ultra low power mac protocol for the downlink of infrastructure wireless sensor networks," in *Ninth International Symposium on Computers and Comun, 2004. Proceedings. ISCC 2004.*, vol. 1. IEEE, 2004, pp. 244–251.

[35] L. S. Committee *et al.*, "Ieee std 802.15. 4e-2012. ieee standards association," *IEEE Computer Society*, 2012.

2. Article 2

Cet article, intitulé "Enhanced Minimal Scheduling Function for IEEE802.15.4e TSCH Networks", a été publié à " IEEE Wireless Communications and Networking Conference (WCNC 2019, Marrakech) ".

Enhanced Minimal Scheduling Function for IEEE 802.15.4e TSCH Networks

Taieb Hamza and Georges Kaddoum
Department of Electrical Engineering
LACIME Laboratory
University of Quebec, ETS, Montreal, Canada.
taieb.hamza.1@ens.etsmtl.ca, georges.kaddoum@etsmtl.ca

***Abstract*—MAC layer protocol design in a WSN is crucial due to the limitations on processing capacities and power of wireless sensors. The latest version of the IEEE 802.15.4, referenced to as IEEE 802.15.4e, was released by IEEE and outlines the mechanism of the Time Slotted Channel Hopping (TSCH). Hence, 6TiSCH working group has released a distributed algorithm for neighbour nodes to agree on a communication pattern driven by a minimal scheduling function. A slotframe contains a specific number of time slots, which are scheduled based on the application requirements and the routing topology. Sensors nodes use the schedule to determine when to transmit or to receive data. However, IEEE 802.15.4e TSCH does not address the specifics on planning time slot scheduling. In this paper, we propose a distributed Enhanced Minimal Scheduling Function (EMSF) based on the minimal scheduling function, which is compliant with 802.15.4e TSCH. In this vein, we introduce a distributed algorithm based on a Poisson process to predict the following schedule requirements. Consequently, the negotiation operations between pairs of nodes to agree about the schedule will be reduced. As a result, EMSF decreases the exchanged overhead, the end-to-end latency and the packet queue length significantly. Preliminary simulation results have confirmed that EMSF outperforms the 802.15.4e TSCH MSF scheduling algorithm.**

***Keywords*—6TiSCH WG, IEEE 802.15.4e TSCH, Scheduling, Poisson Process, Prediction.**

I. Introduction

The next industrial revolution is announced to be Industry 4.0, which will reduce cost and maximize flexibility with the use of digital automation [1]. The Industrial Internet of Things (IIoT) plans to connect to the Internet a large number of industrial objects. In this regard, it is mandatory for real-time infrastructure to be highly reliable for wireless transmissions. So far, the efforts undertaken in IoT were for best-effort solutions, but industrial applications demand a higher level of control in terms of reliability and delay [2]. For that reason, specific MAC protocols are needed to add strict guarantees. Deterministic approaches, especially, can be used to allocate a fixed bandwidth to every device or flow. In addition, these approaches can isolate flows, where each type of flow gets a specific transmission bandwidth. In IEEE 802.15.4e TSCH mode, channel hopping mechanism is used to reduce external interference and fading. TSCH deploys a deterministic approach to avoid collisions by careful scheduling, which involves allocating a group of cells for interfering transmitters to avoid collisions while reducing contention. A number of centralized and distributed scheduling algorithms have been introduced as of date for TSCH [3]. Over-provisioning is a method used to cope with unreliable links in the schedule by reserving a few cells to allow a packet to be retransmitted. However, extra cells leads to an increase in delays and jitter [4]. Network capacity is also reduced if there is too much traffic and/or lack of reliability due to external interference. In order to handle the issues concerning scheduling, routing and internet integration, the IETF IPv6 through the TSCH mode of IEEE802.15.4e was established for an improved integration of the IPv6-enabled protocols such as RPL, 6LoWPAN and CoAP [5]. In this paper, we propose an Enhanced Minimal Scheduling Function (EMSF) in order to improve reliability and latency when pairs of nodes are dynamically determining their bandwidth requirements. The proposed algorithm is based on the minimal scheduling function (MSF) schemed for the 6TiSCH networks. Our proposal is designed to meet the following goals:

1) A dynamic scheduling: The proposed scheme allocates and deallocates cells without considering the threshold-based mechanism.
2) A reduction of the transmitted data during the scheduling negotiation phase: The EMSF is designed to meet this target by introducing a prediction system model which anticipates the data to be transmitted for each pair of nodes in the next slotframe.
3) A reduced latency: This design is introduced by minimizing the overhead control packets in order to diminish the end-to-end data transmission delay and the queued data.

The remainder of this paper is organized as follows. In section II, we provide an outlook to the 6TiSCH WG over the 802.15.4e TSCH mode. We discuss some background information and related work in section III. The design of the system is described in section IV and V. We illustrate, in section VI, the results of simulations that show the performance of our approach, and we conclude the paper in section VII.

II. IEEE 802.15.4e TSCH

A. Concept of IEEE 802.15.4e TSCH

The nodes in IEEE 802.15.4e TSCH network communicate using a time-slotted mechanism over multiple frequencies, which follows a Time Division Multiple Access (TDMA)

schedule. It partitions the wireless spectrum into time and frequency, which is scheduled over a set period of time. This scheduling is also called a superframe or a slotframe. A node transmits/receives data to/from its neighbours on a predefined timeslot and channel in the schedule. A cell, which is usually 10ms in length, is a basic unit of bandwidth scheduled. The transmitter in a cell sends a data packet to the receiver. Once successfully received, an acknowledgement is sent back by the receiver. Channel-hopping improves communication by making it more reliable through diversification of frequencies. This statistical mitigation reduces narrow-band interference and multi-path fading. Cells can contain multiple communication links as long as they are not conflict links nor links interfering amongst each other. Conflict links are those that have the same receiver and/or transmitter. The communication links bounce over a series of available channels in a quasi-random way between the super-frames. Both the sender and the receiver for each scheduled cell will use Eq.(1) to calculate communication frequency, i.e., f [6]:

$$f = F\{(ch_{\text{offset}} + \text{ASN}) \bmod N_{\text{ch}}\}. \quad (1)$$

where F is the mapping function for channel frequency, ch_{offset} is the channel offset, ASN is the total number of timeslots, and finally mod N_{ch} is the modular division of N_{ch}, which refers to the number of available physical channels. ASN is calculated as follows:

$$\text{ASN} = K \times S + T. \quad (2)$$

where K is the slotframe cycle, S is the size of the slotframe, and T are the allocated timeslots.

B. Scheduling in 6TiSCH networks

Recently, 6TiSCH Wireless Group was working to define a pre-configured or learned minimal schedule of a node that joins a network with static scheduling configuration [7]. Various kinds of frame are transmitted and received through cells within a schedule determined by slotted ALOHA protocol. Note that static scheduling within 6TiSCH is used only for specific situations like during the bootstrapping stage or as a fallback during network failures. For other situations, a scheduling function is used in order to allocate or deallocate cells between neighboring nodes. 6TiSCH uses MSF as a default scheduling function presented in the IETF draft [8]. The allocation policy and the bandwidth estimation algorithm are used by MSF to determine when cells in neighbouring nodes should be added or deleted [9]. In addition, MSF uses a threshold-based mechanism to mitigate against sudden fluctuations and increases in bandwidth by adding or deleting operations. Both add and delete negotiations are similar, however. Despite that, given two nodes A and B in a network, node B contains all slotOffsets of node A candidate cells of the remove CellList requests. Since candidate cells are randomly selected, there is a high likelihood of negotiation errors at node A. This is due to the fact that node B contains fewer cells in CellList with available slotOffsets than NumCells. In addition, other pairs of nodes may use allocated cells that are also used by nodes A and B. This may cause network collisions, which are mitigated by the scheduling function. The minimal scheduling function uses 6top Housekeeping function to track cell usage and performance and to relocate cells that have collided [10].

III. Related works

Constructing a schedule is application specific and several scheduling algorithms have been introduced to schedule TSCH networks. Different proposals could be used to set up the schedule. They can be classified as centralized and distributed. In centralized approaches, the DAG root builds and maintains the schedule for the entire network. Palattella et al. proposed Traffic Aware Scheduling Algorithm (TASA), a centralized scheduling technique that uses a leading-edge matching and colouring method of graph theory to map the distribution of time slots and channel offsets [11]. The motes in the network send the requirements of bandwidth and energy, TASA computes a schedule that satisfies those requests and returns them back to the motes. In distributed approaches, the nodes negotiate with their neighbors to build their own schedule. Accentura et al. proposed Decentralized Traffic Aware Scheduling (DeTAS), which builds optimal collision-free multi-hop schedules [12]. DeTAS employs neighbour-to-neighbour signaling in order to gather network and traffic information. It ensures the smallest queue utilization and the shortest possible end-to-end latency period between when the data was generated to when it was received. Orchestra is another non-graph scheduling technique in which nodes compute their own local schedules, hence the reason its referred to as autonomous scheduling of the TSCH in IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) [13]. It has no central entity nor negotiation and allocates slots such that it can be installed or deleted automatically with the evolution of the RPL topology. Despite that, it does not offer any solution against bursty traffic. Domingo-Prieto et al. introduced a distributed scheduling algorithm as a solution against sudden or bursty traffic, which uses a control paradigm called Proportional Integral and Derivative (PID) [14]. Techniques based on graph theory have been introduced to other networks including peer-to-peer networks and cognitive radio networks. It is worth mentioning that the work of Domingo-Prieto et al. is the first one to use graph-theory based combinatorial properties to address scheduling in IEEE 802.15.4e TSCH networks. Soua et al. proposed the Wave scheduling algorithm [15]. It aims to reduce the slotframe size by dividing it with a unit called wave and achieving waves many times to build a schedule for all the nodes at proper intervals of time to the DAG root. Every node with a transmittable packet is allocated both a timeslot and a channel for every wave. The slot or channel pattern is determined by the first wave. The following waves are modeled after the first wave, however only the transmission-containing slots and the order in which they repeat themselves are copied from the first wave. Determining a schedule is application-specific and it's based on the metrics that need to be improved. The proposed 6TiSCH WG minimal

scheduling function [7] defines how nodes add or remove cells based on a bandwidth estimation algorithm. This method is based on exchanging control packets to measure the required bandwidth. Hence, more resources are appealed to determine the appropriate measurements which leads to an increased probability of dropped packets and a high latency. Note that, unlike the existing scheduling approaches, our method will guarantee a minimum negotiation overhead which will reduce the latency, the packet queue size and the resource utilization. Our proposed model is based on three indispensable steps:

1) Computation of the average generated packets in the previous slotframes
2) Prediction of the amount of data (event/periodic).
3) 6p add/remove transaction according to the predicted model.

IV. Proposed system

In this section, we will introduce some definitions and assumptions of the system model. When a 802.15.4e TSCH node detects a substantial unpredictable physical event, a massive flow of data packets is generated and queued by the node. These nodes check if they have sufficient bandwidth to send these packets to their parents. A limited number of transmissions and re-transmissions are allowed by each node, which, when exceeded, cause packet dropping. This appliance triggers a tremendous number of packet transactions and a spike in resources usage. To reduce negotiation errors, number of dropped packets and end-to-end latency times, we propose a novel scheduling function based on the minimal scheduling function presented in the IETF draft [8]. The proposed mechanism consists of the following two main operations: Computation of the mean of packets generated by each node and the prediction of the required cells in the next slotframe. First, we introduce some definitions:

- **Definition 1:** We focus on event-driven WSN where nodes measure physical events and send it to the sink in an upstream data circulation transfer.
- **Definition 2:** We define the network topology as graph $G = (V, E)$ where V is the set of all nodes and E is the set of edges between the nodes displaying symmetric communications links.
- **Definition 3:** In a data gathering frame of any node n we denote G(*n*), which is the number of data transmitting packets sent by node *n* and T(*n*), which is the sum of all the packets sent by *n* including $G(n)$ and the number of packets received by the parent from its children. Therefore, we define *T(n)* using the following equation:

$$T(n) = \sum_{v \in \mathrm{subTree(n)}} G(V). \tag{3}$$

- **Definition 4:** We define $\mathrm{Q_C^P(n)}$ as the number of packets in the queue of n that has to be sent to parent and $\mathrm{C_C^P(n)}$ the number of cells already allocated between the parent $\mathrm{p(n)}$ and the child $\mathrm{c(n)}$.
- **Definition 5:** After executing the scheduling algorithm and based on its output, nodes in G will either add, delete, or keep cells in the next slotframe S_{i+1}.

In this paper we adopt the following assumptions:

- **Assumption 1:** We assume that the gathered data network topology and the routing tree are provided.
- **Assumption 2:** We consider also that the links of the routing tree are symmetric, since user data is gathered upstream, whereas the schedules are negotiated in a distributed fashion between nodes.
- **Assumption 3:** Symmetric links are a requirement for the instant acknowledgment policy.

V. Enhanced minimal scheduling algorithm

A. Prediction of the amount of data

In 802.15.4e TSCH networks, a node transmits two types of packets: Periodic data $A_i^w(n)$ and Event-driven data $A_i^v(n)$. Periodic data consist of enhanced beacons, which contain information about the actual ASN, the length of the timeslot and other information about the network. Event-driven data are sent upon the detection of a physical event. The total of a throughput a node generating is illustrated by the following formulas:

$$A_i^T(n) = A_i^w(n) + A_i^v(n). \tag{4}$$

Which is equivalent to:

$$A_i^T(n) = \sum_{i-1}^{i} \mathrm{subTree(n)} + \sum_{\mathrm{i-1}}^{\mathrm{i}} \textit{G(n)}, \mathrm{i} < 1 < \mathrm{sink}. \tag{5}$$

B. Poisson-based packet generation model

As mentioned in the last section, we considered the 802.15.4e TSCH network as an event-driven network, where sensor nodes report data to the sink only when they obtain new data (an event occurs in a sensing area). We formulate the scheduling problem as a Poisson process model to describe the data generation.

Process Poisson Definition [16]: A Poisson process $N(t), t \geq 0$ with intensity $\lambda > 0$ is a counting process with the following properties.

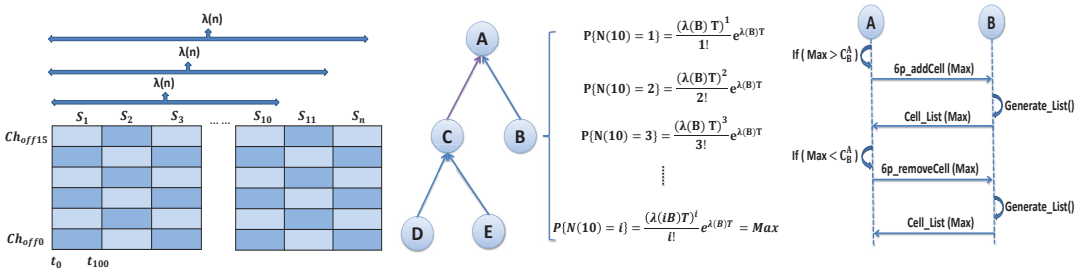
1) **Independent increment**: For all $t_0 = 0 < t_1 < t_2 < < t_n$, then $N(t_1) - N(t_0), N(t_2) - N(t_1), ..., N(t_n) - N(t_{n1})$ are independent random variables.
2) **Stationary increments with Poisson Distribution** : For all $s \geq 0, t > 0, N(s+t) - N(s) \sim Poisson(\lambda t)$.

The system satisfies the previous Poisson lemmas explained as follows:

Lemma 1: Events are considered as independent (temporally and spatially) and occur with equal probability over the area. In the network, we consider the generated packets by different nodes as an independent event with no relation with the preceding event.

Lemma 2: The Time *T* between each slotframe *S* is finite (slotframe durations stay the same through all the scheduling).

Based on lemma (1) and (2), we can adopt a Poisson process



(a) Average generated packets (b) Poisson prediction model (c) 6p operations

Fig. 1: System model

model to describe data generation packets in the network. The distribution of the number of data packets, $N(T)$, generated by each node in the network from the beginning to the end of the slotframe is equal to the following formula:

$$P\{N(t) = n\} = \frac{(\lambda t)^n}{n!} e^{\lambda t}. \quad (6)$$

where λ is the average number of packets generated after a certain number of slotframes have passed since the two nodes are synchronized to the schedule. Mathematical, λ is defined by the following formula:

$$\lambda(n) = \frac{\sum_{i=T-\beta}^{T} \mathrm{nbPacket_i(n)}}{\beta}. \quad (7)$$

where T is the actual instant, $nbPacket_i(n)$ is the number of packets generated by the node n in the instant $T = i$ and β is the sum of previous number of slotframes which can be validated through simulations. In order to get an accurate value of λ, the protocol will execute it's scheduling algorithm with the minimal scheduling function up to $\beta = 10$ slotframes.

An illustration of the execution of the system model is described in Fig. 1. As an example, the nodes A and B are exchanging packets over the built scheduling by the minimal scheduling function. Starting from slotframe S = 10, using Eq.(7), node B determines the average number of generated packets in the previous slotframes since it joined the network as it is shown in Fig. 1(a). Then, the average number is used in Eq.(6) in order to determine the probability of getting a determined number of packets as depicted in Fig. 1(b). This operation will be repeated until a maximum probability value is reached. Based on the predicted output as observed in Fig. 1(c), the algorithm will execute either an add or a remove 6p operation explained in the following section.

C. Add/remove cells

Starting from a given slotframe cycle, each node in the network will execute the Algorithm 1. The purpose is to predict the number of packets that will be generated by each node in the next slotframe. In the aim of reducing resource computation by each node, the algorithm stops when a maximum value of λ is reached. By knowing the number of packets that will be generated in the next slotframe, a node can predict how many cells are required in order to exchange data with its preferred parent. Furthermore, based on the output of the algorithm, a node can trigger a 6p transaction with its preferred parent either to add, or remove cells to the TSCH schedule of both nodes.

Algorithm 1 Cell prediction algorithm Starting from Slotrame S=β

1: Set *G(n)* to be a random value chosen from a Poisson distribution with mean $= rt$ (r in unit of 1/time)
2: **for** $S = S_{11}$ to {Network lifetime} ; S_{i++} **do**
3: Determine λ using Eq.(7) ▷{Average generated number of packets in S_0, S_1, S_2,..., S_{i-1} }
4: $\lambda \Longleftarrow (n)$
5: **while** $pmax$ **do**
6: $p \Longleftarrow$ Determine $(PN(t), \lambda)$ ▷{probability of generating λ packets}
7: **if** $(p \geqslant pmax)$ **then**
8: $max \Longleftarrow p$
9: **end if**
10: **end while**
11: return (p) ▷ {the maximum value of the probability}
12: **if** $(p < C_c^p(n))$ ▷{Compare p with the actual reserved cells for node n} **then**
13: 6P_DELETE_command (p) ▷{6P delete request of $C_c^p(n)$ - p slots}
14: **elseif** $(p > C_c^p(n))$ **then**
15: 6P_ADD_command (p) ▷{6P add request of $C_c^p(n)$ + p slots}
16: **end if**
17: **end for**

VI. Performance Evaluation

We conducted our simulations on OpenWSN, which is an opensource network simulator for wireless sensor networks. The simulator supports a few major types of protocol based on

IoT standards, such as 6LoWPAN, RPL, ROLL, and CoAP. All the protocols based on OpenWSN 6TiSCH use the latest IEEE 802.15.4e TSCH standard in order to improve their reliability and stability [17]. In order for OpenWSN to remain up to date and have the desired network metrics, it provides a python-based configuration tool that allows its network parameters to be modified. The behavior of the network (such as PDR in every channel), the node queuing priority and the timestamp of each sending and receiving packet is recorded and then simulated. Table I shows the parameters used in the OpenWSN simulation.

TABLE I: OpenWSN Simulation parameters

Simulation Parameters	Values
Available channels (N)	11-26
Number of nodes	2-100
Timeslot duration	10 ms
Slotframe length p	101 timeslot
Payload	127 bytes
MAC max retries	4
Max queue size	5
Period of transmission	200 ms

A. 6p error ratio

6p negotiation error ratio is the average ratio between the number of 6p transaction errors and the total 6p transactions throughout a slotframe cycle. Fig. 2 displays the way that the 6p negotiation error ratio is affected by the network density. The number of nodes passes from 10 to 100. The negotiation error ratio increases with the increase in network density. In the MSF scenario, it goes from 2.1% to 19.8%, whereas in EMSF scenario it goes from 0.2% to 3.5%. This is due to the decrease in the number of control packets and to the substitution of the threshold-measurement mechanism that is needed for the prediction system to add or delete 6p operations. The number of 6p exchanged packets decreases, which leads to a lower negotiation-error ratio. The suggested mechanism, the EMSF, largely surpasses the MSF and keeps the negotiation-error ratio under 3.5% for all of the network densities.

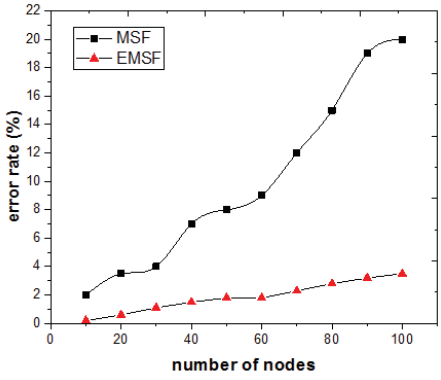


Fig. 2: 6p error ratio

B. Packet overhead

Fig. 3 represents the overhead traffic load in bytes that was employed by nodes in order to exchange 6p information in the network. We notice that the number of exchanged messages increases linearly with the number of deployed nodes. This is due to negotiation exchanges taken between pairs of nodes to determine the 6p operation that will be deployed. We notice that the EMSF maintains an almost constant amount of exchanged packets. This is due to the prediction algorithm that anticipates the number of required cells for each pair of nodes in the following slotframe. EMSF avoids sending overloads and keeps a constant average of control packets through the network lifetime.

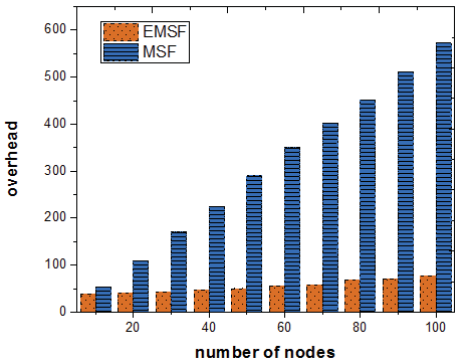


Fig. 3: packet overhead

C. Latency

The latencies are calculated as follows: Every packet is timestamped since it is generated in the application layer of a source node, until it reaches the application layer of the DAG root. As a result of this condition, retransmission on a MAC layer is not taken into consideration. However, if a packet is repeatedly transmitted, a peak in latency may occur. When a packet achieves 4 MAC retransmissions attempts and the queue is full, a packet will be considerably dropped. Each node sends a fixed traffic load of 2 packets per slotframe for the first 50 slotframes. Thereafter, each node sends a sporadic traffic load ranging from 2 to 7 packets per slotframe. 20 nodes were deployed in this simulation. An end-to-end latency comparison between MSF and EMSF is illustrated in Fig.4. The latency was almost constant for the first 50 slotframes because of stable transmitted data flow. Thereafter, latency varies in between cycles as a result of the failing stochastic transmissions based on the PDR between the nodes. EMSF keeps an end-to-end latency below 75 milliseconds in almost all the slotframe cycles. This is due to its scheduling mechanism that minimizes the overhead charges, which increases the PDR even when irregular data flow is generated.

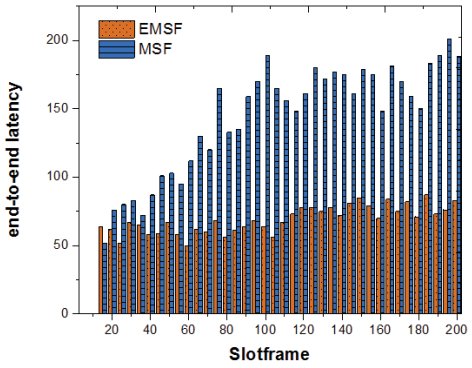
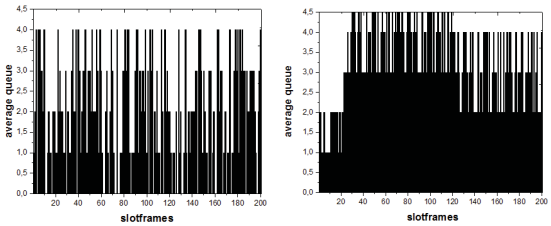


Fig. 4: Latency

D. Node queue size

The following Fig. 5 represents the average number of packets in the queues of the network nodes. From this figure, it can be seen that in the case of MSF, node queues are almost full and reach their maximum (5 packets) in some slotframes causing more dropped packets. On the other hand, EMSF shows better performance in this concern. Throughout 200 slotframes, EMSF shows an average queue size ranging from 0 to 4 packets. In a lot of cases, nodes over-allocate cells, especially those near to the DAG root. This is due to the prediction algorithm in the case where a high packet rate is generated in the previous slotframes. This will allocate more cells between pairs of nodes, which will be used when an unpredictable event occurs. For instance, given an average number of generated packets equal to 10 in the previous slotframes, a node generating 5 packets will allocate cells based on the average number of packets generated since the time it joined the network. This means that, even though a huge number of packets is generated, the node has already reserved more cells than its requirements. Therefore, packets tend to queue up less, but consume more energy and than in the case of MSF.



((a)) EMSF ((b)) MSF

Fig. 5: Node queue size

VII. Conclusion

In this paper, we formulated the scheduling function for IEEE 802.15.4e TSCH networks as a Poisson process model. We demonstrated that, based on the prediction algorithm, scheduling negotiation overhead is decreased noticeably. In addition, simulations approved that EMSF outperforms MSF in terms of the average packet queue length in the network and the end-to-end latency. In a future work, we will improve EMSF by combining it with an enhanced cell selection mechanism in order to reduce energy consumption throughout the network.

References

[1] X. Li, D. Li, J. Wan, V. Vasilakos, C. Lai, S. Wang, *A review of industrial wireless networks in the context of Industry 4.0*, *Wireless Netw.*, vol. 23, no. 1, pp. 23-41, Nov. 2017.

[2] I. Al-Anbagi, M. Erol-Kantarci, T. Mouftah, *A survey on cross-layer quality of-service approaches in WSNs for delay and reliability-aware applications*, *IEEE Commun. Surveys Tuts.* , vol. 18, no. 1, pp. 525-552, Oct. 2016 .

[3] R. Hermeto, A. Gallais, F. Theoleyre, *Scheduling for IEEE802.15.4-TSCH and slow channel hopping MAC in low power industrial wireless networks: a survey*, *Comput. Commu.*, vol. 114, pp. 84-105, Dec. 2017.

[4] Georgios Z. Papadopoulos, T. Matsui, P. Thubert, G. Texier, T. Watteyne, N. Montavont, *Leapfrog collaboration: toward determinism and predictability in industrial- IoT applications*, *in Proc. Int. Conf. Commun. (ICC 2017)*, Paris, pp. 16, Jul. 2017.

[5] 6TiSCH working group, https://datatracker.ietf.org/wg/6tisch/charter/l, (Accessed 18 Sept. 2018).

[6] 6TiSCH working group, https://tools.ietf.org/html/rfc7554 (Accessed 20 Sept. 2018).

[7] X. Vilajosana, K. Pister, *Leapfrog collaboration: toward determinism and predictability in industrial- IoT applications*, *in Proc. Int. Conf. Commun. (ICC 2017)*, Paris, Jul. 2017.

[8] D. Dujovne, *6tisch 6top Scheduling Function Zero (SF0) (work in progress)*, *Ietf draft*, 2017.

[9] M. Palattella, T. Watteyne, Q. Wang, K. Muraoka, N. Accettura, D. Dujovne, L. Grieco, T. Engel, *On-the-fly bandwidth reservation for 6TiSCH wireless industrial networks*, *IEEE Sensors J.*, vol. 16, no. 2, pp. 550-560, Jan. 2016

[10] M. D. Prieto, T. Chang, X. Vilajosana, T. Watteyne, *Distributed pid-based scheduling for 6tisch networks*, *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 1006-1009, Mar. 2016.

[11] M. R. Palattella, N. Accettura, M. Dohler, L. A. Grieco, and G. Boggia, *Traffic aware scheduling algorithm for reliable low-power multi-hop ieee 802.15.4e networks*, *in Proc. Personal Indoor and Mobile Radio Communications (PIMRC)*, Australia, pp. 327-332, Sept. 2012.

[12] N. Accettura, M. R. Palattella, G. Boggia, L. A. Grieco, and M. Dohler, *Decentralized Traffic Aware Scheduling for multi-hop Low power Lossy Networks in the Internet of Things*, *in Proc. Int. Symp on World of Wireless, Mobile and Multimedia Netw. (WoWMoM 2013)*, Spain, pp. 16, Jun. 2013.

[13] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, *Orchestra: Robust mesh networks through autonomously scheduled TSCH*, *in Proc. Conference on Embedded Networked Sensor Syst.*, Seoul, pp. 337350, Nov. 2015.

[14] M. Domingo-Prieto, T. Chang, X. Vilajosana, and T. Watteyne, *Distributed pid-based scheduling for 6tisch networks*, *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 1006-1009, Mar. 2016.

[15] R. Soua, P. Minet, E. Livolant, *A distributed joint channel and slot assignment for convergecast in wireless sensor networks*, *in Proc. Int. Conference on New Technologies, Mobility and Security (NTMS)*, Dubai, May 2014.

[16] W. Paul, J. Baschnagel,*Stochastic processes*, springer, 2013.

[17] T. Watteyne, X. Vilajosana, B. Kerkez, F. Chraim, K. Weekly, Q. Wang, S. Glaser, K. Pister, *Openwsn: a standards-based lowpower wireless development environment*, *IEEE Trans. Emerging Telecommun. Technol.*, vol. 23, no. 5, pp. 480-493, Aug. 2012.

BIBLIOGRAPHIE

Abdeddaim, N., Theoleyre, F., Heusse, M. & Duda, A. (2013). Adaptive IEEE 802.15.4 MAC for Throughput and Energy Optimization. *IEEE International Conference on Distributed Computing in Sensor Systems*, pp. 223-230.

Accettura, N. & Piro, G. (2014, June). Optimal and secure protocols in the IETF 6TiSCH communication stack. *IEEE 23rd International Symposium on Industrial Electronics (ISIE)*, pp. 1469-1474.

Accettura, N., Palattella, M. R., Dohler, M., Grieco, L. A. & Boggia, G. (2012, April). Standardized power-efficient & internet-enabled communication stack for capillary M2M networks. *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 226-231.

Accettura, N., Palattella, M. R., Boggia, G., Grieco, L. A. & Dohler, M. (2013, June). Decentralized Traffic Aware Scheduling for multi-hop Low power Lossy Networks in the Internet of Things. *IEEE 14th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 1-6.

Accettura, N., Vogli, E., Palattella, M. R., Grieco, L. A., Boggia, G. & Dohler, M. (2015). Decentralized Traffic Aware Scheduling in 6TiSCH Networks : Design and Experimental Evaluation. *IEEE Internet of Things Journal*, 2(6), 455-470.

Althobaiti, A. S. & Abdullah, M. (2015). Medium Access Control Protocols for Wireless Sensor Networks Classifications and Cross-Layering. *Procedia Computer Science*, 65, 4-16.

Bluetooth. (2005). *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Personal Area Networks (WPANs), IEEE Standard 802.15.1*.

Callaway, E., Gorday, P., Hester, L., Gutierrez, J. A., Naeve, M., Heile, B. & Bahl, V. (2002). Home networking with IEEE 802.15.4 : a developing standard for low-rate wireless personal area networks. *IEEE Communications Magazine*, 40(8), 70-77.

Chang, T., Watteyne, T., Pister, K. & Wang, Q. (2015). Adaptive synchronization in multi-hop TSCH networks. *Computer Networks*, 76, 165-176.

Chang, T., Vučinić, M., Vilajosana, X., Duquennoy, S. & Dujovne, D. (2018). *6TiSCH Minimal Scheduling Function (MSF)* (Rapport n°draft-ietf-6tisch-msf-01). Internet Engineering Task Force. Repéré à https://datatracker.ietf.org/doc/html/draft-ietf-6tisch-msf-01.

Daidone, R., Dini, G. & Anastasi, G. (2014). On evaluating the performance impact of the IEEE 802.15.4 security sub-layer. *Computer Communications*, 47, 65-76.

Demir, A. K. & Bilgili, S. (2017). DIVA : a distributed divergecast scheduling algorithm for IEEE 802.15.4e TSCH networks. *Wireless Networks*, 1-11.

Dujovne, D., Watteyne, T., Vilajosana, X. & Thubert, P. (2014). 6TiSCH : deterministic IP-enabled industrial internet (of things). *IEEE Communications Magazine*, 52(12), 36-41.

Ergen, S. C. & Varaiya, P. (2010). TDMA scheduling algorithms for wireless sensor networks. *Wireless Networks*, 16(4), 985-997.

ETSI. (2011). *Electromagnetic compatibility and Radio spectrum Matters (ERM), System Reference document (SRdoc), Spectrum Requirements for Short Range Device, Metropolitan Mesh Machine Networks (M3N) and Smart Metering (SM) applications*. Sophia Antipolis, France.

Farrel,Vasseur, A. (2006). *A Path Computation Element (PCE)-Based Architecture* (Rapport n°4655). Network Working Group.

Gaillard, G., Barthel, D., Theoleyre, F. & Valois, F. (2014, May). Service Level Agreements for Wireless Sensor Networks : A WSN operator's point of view. *IEEE Network Operations and Management Symposium (NOMS)*, pp. 1-8.

Guglielmo, D. D., Restuccia, F., Anastasi, G., Conti, M. & Das, S. K. (2016). Accurate and Efficient Modeling of 802.15.4 Unslotted CSMA/CA through Event Chains Computation. *IEEE Transactions on Mobile Computing*, 15(12), 2954-2968.

Görmüş, S. & Yavuz, A. F. (2017, May). A protocol for Internet of Things : IETF 6TiSCH. *Signal Processing and Communications Applications Conference (SIU)*, pp. 1-4.

Hamza, T., Kaddoum, G., Meddeb, A. & Matar, G. (2016, Sep.). A Survey on Intelligent MAC Layer Jamming Attacks and Countermeasures in WSNs. *IEEE Vehicular Technology Conference (VTC-Fall)*, pp. 1-5.

Hamza, T. & Kaddoum, G. (2019). Enhanced Minimal Scheduling Function for IEEE802.15.4e TSCH Networks. *IEEE Wireless Communications and Networking Conference (WCNC)*.

HART. (2011). *Hart Communication Foundation Std. HART Field Communication Protocol Specification*.

Hermann, M., Pentek, T. & Otto, B. (2016, Jan). Design Principles for Industrie 4.0 Scenarios. *Hawaii International Conference on System Sciences (HICSS)*, pp. 3928-3937.

Huawei-Technologies. (2015). *Global Connectivity Index*. Repéré à https://www.digitaleschweiz.ch/wp-content/uploads/2016/05/Huawei-global-connectivity-index-2015-whitepaper-en-0507.pdf.

IEEE. (2003). IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks Specific Requirements Part 15.4 : Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). *IEEE Std 802.15.4-2003*.

IEEE. (2006). IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 15.4 : Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs).

IEEE. (2018). IEEE Standard for Low-Rate Wireless Networks Corrigendum 1. *IEEE Std 802.15.4-2015/Cor 1-2018 (Amendment to IEEE Std 802.15.4-2015 as amended by IEEE Std 802.15.4n-2016, IEEE Std 802.15.4q-2016, IEEE Std 802.15.4u-2016, IEEE Std 802.15.4t-2017 and IEEE Std 802.15.4v-2017)*.

IEEE802.15.4e. (2012). IEEE Standard for Local and Metropolitan Area Networks. Part 15.4 : Low-Rate Wireless Personal Area Networks Amendment 1 : MAC Sublayer.

ISA. (2009). *International Society of Automation. Standard ISA-100.11a, Wireless Systems for Industrial Automation : Process Control and Related Applications*.

Jin, Y., Kulkarni, P., Wilcox, J. & Sooriyabandara, M. (2016, April). A centralized scheduling algorithm for IEEE 802.15.4e TSCH based industrial low power wireless networks. *IEEE Wireless Communications and Networking Conference*, pp. 1-6.

Khajenasiri, I., Estebsari, A., Verhelst, M. & Gielen, G. (2017). A Review on Internet of Things Solutions for Intelligent Energy Control in Buildings for Smart City Applications. *Energy Procedia*, 111, 770 - 779.

Lu, C., Blum, B. M., Abdelzaher, T. F., Stankovic, J. A. & He, T. (2002, Sep.). RAP : a real-time communication architecture for large-scale wireless sensor networks. *Proceedings. Eighth IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 55-66.

Mainetti, L., Patrono, L. & Vilei, A. (2011, Sep.). Evolution of wireless sensor networks towards the Internet of Things : A survey. *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1-6.

Mpitziopoulos, A., Gavalas, D., Konstantopoulos, C. & Pantziou, G. (2009). A survey on jamming attacks and countermeasures in WSNs. *IEEE Communications Surveys Tutorials*, 11(4), 42-56.

Ojo, M. & Giordano, S. (2016, Oct). An efficient centralized scheduling algorithm in IEEE 802.15.4e TSCH networks. *IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 1-6.

Ojo, M., Giordano, S., Portaluri, G., Adami, D. & Pagano, M. (2017, May). An energy efficient centralized scheduling scheme in TSCH networks. *IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 570-575.

Palattella, M. R., Accettura, N., Dohler, M., Grieco, L. A. & Boggia, G. (2012, Sep.). Traffic Aware Scheduling Algorithm for reliable low-power multi-hop IEEE 802.15.4e networks. *IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)*, pp. 327-332.

Palattella, M. R., Accettura, N., Vilajosana, X., Watteyne, T., Grieco, L. A., Boggia, G. & Dohler, M. (2013). Standardized Protocol Stack for the Internet of (Important) Things. *IEEE Communications Surveys Tutorials*, 15(3), 1389-1406.

Pereira, N., Tennina, S. & Tovar, E. (2012). Building a Microscope for the Data Center. *Wireless Algorithms, Systems, and Applications*, pp. 619-630.

Petersen, S., Doyle, P., Vatland, S., Aasland, C. S., Andersen, T. M. & Sjong, D. (2007, Sep.). Requirements, drivers and analysis of wireless sensor network solutions for the Oil amp ; Gas industry. *IEEE Conference on Emerging Technologies and Factory Automation (EFTA)*, pp. 219-226.

Polastre, J., Hill, J. & Culler, D. (2004). Versatile Low Power Media Access for Wireless Sensor Networks. *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, pp. 95-107.

Poulsen, D. R., Spivey, M. Z. & Marks, R. J. (2011, March). The Poisson process and associated probability distributions on time scales. *IEEE 43rd Southeastern Symposium on System Theory*, pp. 49-54.

Soua, R., Minet, P. & Livolant, E. (2012, Dec). MODESA : An optimized multichannel slot assignment for raw data convergecast in wireless sensor networks. *IEEE 31st International Performance Computing and Communications Conference (IPCCC)*, pp. 91-100.

Soua, R., Livolant, E. & Minet, P. (2013, July). MUSIKA : A multichannel multi-sink data gathering algorithm in wireless sensor networks. *International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1370-1375.

Soua, R., Minet, P. & Livolant, E. (2015, May). DiSCA : A distributed scheduling for convergecast in multichannel wireless sensor networks. *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 156-164.

Soua, R., Minet, P. & Livolant, E. (2016). Wave : a distributed scheduling algorithm for convergecast in IEEE 802.15.4e TSCH networks. *Transactions on Emerging Telecommunications Technologies*, 27(4), 557-575.

Sudhaakar, R. S. & Zand, P. (2015). *6TiSCH Resource Management and Interaction using CoAP* (Rapport n°draft-ietf-6tisch-coap-03). Internet Engineering Task Force. Repéré à https://datatracker.ietf.org/doc/html/draft-ietf-6tisch-coap-03.

Thubert, P., Watteyne, T., Palattella, M. R., Vilajosana, X. & Wang, Q. (2013, July). IETF 6TSCH : Combining IPv6 Connectivity with Industrial Performance. *International*

Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 541-546.

Thubert, P. (2018). *An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4* (Rapport n°draft-ietf-6tisch-architecture-19). Internet Engineering Task Force. Repéré à https://datatracker.ietf.org/doc/html/draft-ietf-6tisch-architecture-19.

Tinka, A., Watteyne, T. & Pister, K. (2010). A Decentralized Scheduling Algorithm for Time Synchronized Channel Hopping. *Ad Hoc Networks*, pp. 201-216.

van Dam, T. & Langendoen, K. (2003). An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks. *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 171-180.

Vasiljević, D. & Gardašević, G. (2016, Nov). Performance evaluation of OpenWSN operating system on open mote platform for industrial IoT applications. *International Symposium on Industrial Electronics (INDEL)*, pp. 1-6.

Wang, Q., Vilajosana, X. & Watteyne, T. (2018). 6TiSCH Operation Sublayer (6top) Protocol (6P). RFC Editor. doi : 10.17487/RFC8480.

Watteyne, T., Mehta, A. & Pister, K. (2009). Reliability Through Frequency Diversity : Why Channel Hopping Makes Sense. *Proceedings of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, pp. 116-123.

Watteyne, T., Vilajosana, X., Kerkez, B., Chraim, F., Weekly, K., Wang, Q., Glaser, S. D. & Pister, K. (2012). OpenWSN : a standards-based low-power wireless development environment. *Trans. Emerging Telecommunications Technologies*, 23(5), 480-493.

Winter, T., Alexander, R., Brandt, A., Vasseur, J., Hui, J., Pister, K., Thubert, P., Levis, P., Struik, R. & Kelsey, R. (2012). *RPL : IPv6 Routing Protocol for Low-Power and Lossy Networks* (Rapport n°6550). CA, USA : Internet Engineering Task Force (IETF).

Wood, A. D., Stankovic, J. A. & Zhou, G. (2007, June). DEEJAM : Defeating Energy-Efficient Jamming in IEEE 802.15.4-based Wireless Networks. *IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 60-69.

Yazdi, E. T., Moravejosharieh, A., Willig, A. & Pawlikowski, K. (2014, Nov). Coupling power and frequency adaptation for interference mitigation in IEEE 802.15.4-based mobile body sensor networks : Part II. *Australasian Telecommunication Networks and Applications Conference (ATNAC)*, pp. 105-110.

Ye, J., Chen, B., Liu, Q. & Fang, Y. (2013, June). A precision agriculture management system based on Internet of Things and WebGIS. *International Conference on Geoinformatics*, pp. 1-5.

Ye, W., Heidemann, J. & Estrin, D. (2002, June). An energy-efficient MAC protocol for wireless sensor networks. *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, 3, 1567-1576.

Yeh, L.-W. & Pan, M.-S. (2014). Beacon scheduling for broadcast and convergecast in ZigBee wireless sensor networks. *Computer Communications*, 38, 1-12.

Zanella, A., Bui, N., Castellani, A., Vangelista, L. & Zorzi, M. (2014). Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, 1(1), 22-32.

ZigBee-Alliance. (2005). ZigBee Specification. Repéré à https://www.zigbee.org/.