Optimisation de la performance d'un réseau de capteurs sans fil pour les applications de collection de données à faible énergie et courte latence

par

Mohamed OSMAN ABDI

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 17 DÉCEMBRE 2020

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

PRÉSENTATION DU JURY

CE MÉMOIRE A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

M. Frédéric Nabki, directeur de mémoire
Département de génie électrique à l'École de technologie supérieure

M. Ricardo Izquierdo, président du jury
Département de génie électrique à l'École de technologie supérieure

M. Dominic Deslandes, membre du jury
Département de génie électrique à l'École de technologie supérieure

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

LE 10 DÉCEMBRE 2020

À L'ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

REMERCIEMENTS

Avant tout développement, il apparaît opportun de commencer ce mémoire par des remerciements, à ceux qui m'ont beaucoup appris durant cette maitrise.

Je tiens tout d'abord à remercier mon professeur et encadrant, M. Frédéric Nabki pour son énorme soutien, ses conseils importants, sa volonté de transmettre ses compétences et sa convivialité tout au long de cette maitrise. Je tiens à lui exprimer mon plus grand respect et ma reconnaissance pour toute la motivation et son aide précieuse qui m'ont amené à présenter ce travail aujourd'hui.

Je voudrais également profiter de cette occasion pour remercier ma famille et mes amis pour leur incroyable soutien et leurs précieux conseils.

Un merci spécial à tous les membres du laboratoire LACIME pour leur gentillesse et leurs aides qui m'ont permis de passer un moment très profitable tout au long de cette période.

Enfin, je tiens à remercier chaque membre du comité d'avoir accepté de consacrer du temps à l'examen de ce travail et pour leurs précieux commentaires et suggestions.

Optimisation de la performance d'un réseau de capteurs sans fil pour les applications de collection de données à faible énergie et courte latence

Mohamed OSMAN ABDI

RÉSUMÉ

Les techniques émergentes d'apprentissage automatique et d'intelligence artificielle s'attendent à recevoir de plus en plus de données des capteurs IoT et plus souvent. Ces nœuds doivent donc pouvoir gérer la multitude de données provenant d'innombrables sources. TSCH (Time Slotted Channel Hopping), le protocole MAC défini dans les normes de communication sans fil IEEE 802.15.4e, joue un rôle crucial dans la réduction de la latence et la minimisation de la consommation énergétique. Dans TSCH, le temps est divisé en intervalles de temps et un calendrier détermine l'action qu'un nœud doit entreprendre dans chaque intervalle de temps (émission, réception ou mise en veille). Dans le cas du trafic convergeant, les nœuds proches de la racine sont constamment en demande et ont un trafic important car tous les nœuds acheminent des paquets vers cette destination. Dans ce contexte, les algorithmes de planification des cellules, qui sont responsables du comportement de chaque nœud à chaque intervalle de temps, doivent encore évoluer pour prendre en charge le trafic convergeant et fournir des solutions à la demande croissante de données dans le monde d'aujourd'hui.

Dans ce mémoire, nous proposons OSCAR, un nouvel algorithme d'allocation de cellules TSCH d'ordonnancement autonome basé sur l'algorithme d'Orchestra. Cette nouvelle conception diffère d'Orchestra par le fait qu'OSCAR attribue les slots en fonction de l'emplacement du nœud par rapport à la racine. Le but de cet algorithme est d'allouer des slots aux nœuds en fonction de leurs besoins. Cet algorithme gère le nombre de créneaux horaires alloués à chaque nœud en utilisant la valeur du rang décrit par le protocole de routage RPL. Le but étant que plus le nœud est proche de la racine, plus il obtient de slots afin de maximiser les opportunités de transmission. Pour éviter la surconsommation, OSCAR met en place un mécanisme permettant d'ajuster le cycle de service radio de chaque nœud en réduisant les slots alloués aux nœuds inactifs quelle que soit leur position sur le réseau.

Nous implémentons OSCAR sur Contiki-ng et évaluons ses performances sur un réseau composé de dix nœuds. L'analyse des performances d'OSCAR montre qu'il surpasse Orchestra en termes de latence et de fiabilité moyennes, en particulier lorsque le trafic est dense et encombré. Enfin par rapport au gain précédent la consommation d'énergie n'augmente pas beaucoup.

Mots-clés: réseau de capteurs sans fil, TSCH, planification autonome, RPL, IEEE 802.15.4e, faible latence, optimisation énergétique, réseaux convergeant

Optimizing the performance of a wireless sensor network for low-energy, short-latency data collection applications

Mohamed OSMAN ABDI

ABSTRACT

Emerging machine learning and artificial intelligence techniques expect to receive more and more data from IoT sensors and more often. These nodes must therefore be able to handle the multitude of data coming from countless sources. TSCH (Time Slotted Channel Hopping), the MAC protocol defined in the wireless communication standards IEEE 802.15.4e, plays a crucial role in reducing latency and minimizing power. In TSCH, time is divided into timeslots and a schedule determines what action a node needs to take in each timeslot (transmit, receive, or sleep). In the case of convergecast traffic, nodes close to the root are constantly in demand and have heavy traffic because all nodes are forwarding packets to that destination. In this context, cell scheduling algorithm, which are responsible of each node's behavior at each timeslot, has to evolve further to support convergecast traffic and to provide solutions to the growing demand for data in today's world.

In this thesis, we propose OSCAR, a novel autonomous scheduling TSCH cell allocation algorithm based on Orchestra algorithm. This new design differs from orchestra by the fact that OSCAR allocates slots according to the location of the node relative to the root. The goal of this algorithm is to allocate slots to nodes according to their needs. This algorithm manages the number of timeslot allocated to each node using the value of the rank described by the RPL routing protocol. The goal being that the closer the node is to the root, the more slots it gets in order to maximize the transmission opportunities. To avoid overconsumption, OSCAR sets up a mechanism to adjust the radio duty cycle of each node by reducing the slots allocated to inactive nodes regardless of their position on the network.

We implement OSCAR on Contiki-ng and evaluate its performance on a network made up of ten nodes. The performance analysis of OSCAR shows that it outperforms Orchestra on the average latency and reliability especially when the traffic is heavy and congested. Finally compared to the previous gain the energy consumption does not increase much.

Keywords: wireless sensor network, TSCH, autonomous scheduling, RPL, IEEE 802.15.4e, low latency, energy optimization, convergecast traffic

TABLE DES MATIÈRES

Page

LISTE DES TABLEAUX

Page

LISTE DES FIGURES

Page

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

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
BLINK	Radio Frequency Identification Blink
CDMA	Code Division Multiple Access
CSMA	Carrier Sense Multiple Access
DAO	Destination Advertisement Object
DODAG	Destination Oriented Directed Acyclic Graph
DIO	DODAG Information Object
DSME	Deterministic and Synchronous Multi-channel Extension
EB	Enhanced Beacon
FDMA	Frequency-Division Multiple Access
FFD	Full Function Devices
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
LLDN	Low Latency Deterministic Network
LLN	Low power and Lossy Networks

LR-WPAN	Low-Rate Wireless Personal Area Network
MAC	Medium Access Control
MSF	Minimal Scheduling Function
OSCAR	Optimized Scheduling Cell Allocation algorithm for Convergecast
PDR	Packet Delivery Ratio
QoS	Quality of Service
RFD	Reduced Function Devices
RFID	Radio Frequency Identification
ROLL	Routing Over Low-power and Lossy networks
RPL	Routing Protocol for Low-Power and Lossy Networks
TDMA	Time Division Multiple Access
TSCH	Time Slotted Channel Hopping
WSN	Wireless Sensor Network

CHAPITRE 1

INTRODUCTION

1.1 Généralités sur les réseaux de capteurs sans fil

Un réseau de capteurs sans fil (WSN) est un réseau composé d'un grand nombre de nœuds qui communiquent des informations éventuellement via plusieurs sauts à un ou plusieurs points de collecte appelés sink. La transmission de données est effectuée via des liaisons sans fil comme illustré sur la figure 1.1, où un exemple de WSN est représenté. Chaque nœud de capteur est composé d'une ou plusieurs unités de capteurs, d'une unité de traitement, d'un émetteur-récepteur sans fil et d'une source d'alimentation comme illustré par la figure 1.1.

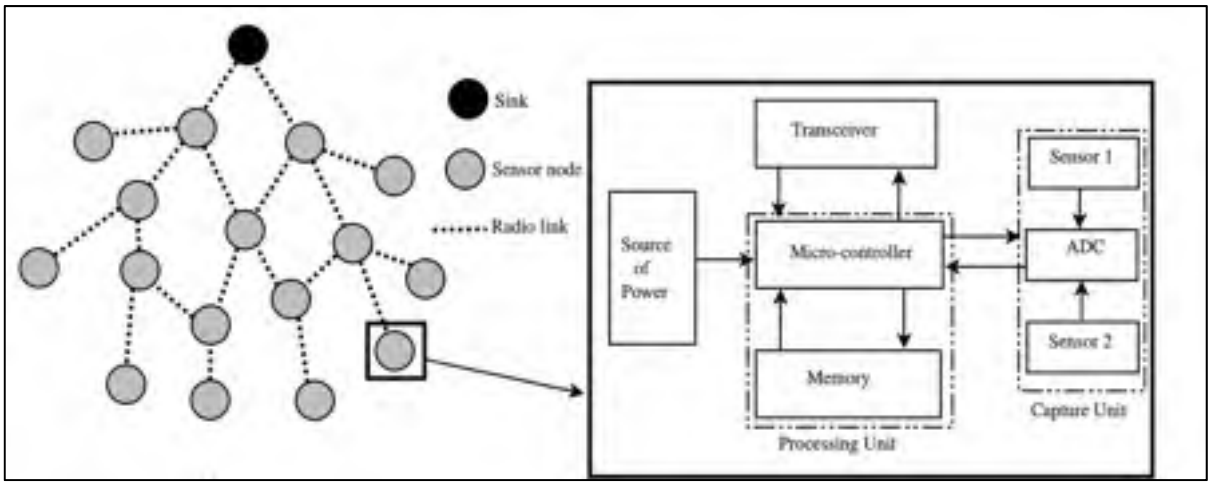

Figure 1.1 Architecture d'un réseau de capteurs sans fil et composition typique d'un nœud capteur (Tall, 2018)

Un réseau de capteurs sans fil doit avoir une consommation d'énergie limitée, car les nœuds de capteurs sont déployés à des endroits où ils n'ont pas besoin d'être entretenus fréquemment. Pour cette raison, le nœud de capteur utilise des radios à faible portée pour économiser de l'énergie dans la transmission de données. Le délai, quant à lui, dépend du nombre de nœuds intermédiaires ou de nœuds de transfert entre l'expéditeur et le récepteur, c'est-à-dire que le multi-saut augmente le délai tout en réduisant la consommation d'énergie. Le délai peut être

réduit si les nœuds se réveillent plus souvent, au détriment d'une consommation d'énergie plus élevée due à une écoute inactive.

1.1.1 L’internet des objets et les opportunités de marché

De nos jours, nous sommes à un tournant pour les réseaux de capteurs sans fil et l'internet des objets (IoT). Au cours des dix dernières années, on assiste à une croissance explosive des appareils connectés entre eux. De la voiture au mobile, en passant par les montres et les télévisions, l'IoT sous toutes ses formes et déclinaisons est omniprésent dans la vie de tous les jours, et continue d'étendre son champ d'action (Ibarra-Esquer et al., 2017). L’IoT a beaucoup de définitions, mais on peut essentiellement désigner l’IoT comme des appareils ou mini-ordinateurs connectés à Internet.
Le terme IoT a été initialement proposé pour désigner des objets connectés interopérables identifiables de manière unique avec la technologie d'identification par radiofréquence (RFID). Plus tard, les chercheurs ont associé l'IoT à d'autres technologies telles que les capteurs, les actionneurs, les appareils GPS et les appareils mobiles.

L'IoT intervient dans un certain nombre de réseaux hétérogènes tels que les réseaux de capteurs sans fil (WSN). Ces réseaux aident les « choses » dans l'IoT à échanger des informations. Dans les topologies maillées, les nœuds de réseau sont connectés directement et dynamiquement de manière non hiérarchique, permettant ainsi aux communications plusieurs à plusieurs (entre les nœuds coopérant entre eux) de router efficacement les données d'une source générique vers une destination générique. En fait, dans un WSN, chaque nœud composant le réseau peut fonctionner à la fois comme hôte et comme routeur, relayant les paquets envoyés par d'autres nœuds lorsque la destination n'est pas dans la plage de visibilité de la source.
Les réseaux de capteurs sans fil sont une solution prometteuse pour prendre en charge des débits élevés. Les domaines d'applications des WSNs sont nombreux (militaires, jeux vidéo, service hospitalier, etc.). Ces réseaux constituent un domaine de recherche en plein boom notamment avec la 5G et le monde actuel de l’Internet des objets (Cheng et al., 2018 ; Li et al., 2018). L’IoT est l'un des sujets les plus en vogue dans le monde de l'ingénierie de la

communication, tant dans l'industrie que dans le monde universitaire. Ces réseaux sont déployés dans les entreprises, les campus universitaires, les maisons, aéroports, etc. . Ci-dessous un large éventail d'applications qui offrent la possibilité de déploiement de manière dense à un coût très bas au WSN et à l'IoT (Ibarra-Esquer et al., 2017 ; Kandris et al., 2020) :

- équipement de santé : les réseaux maillés sans fil peuvent aider à surveiller et à localiser rapidement les dispositifs médicaux. Ils peuvent également servir de secours pour les équipements médicaux qui doivent toujours rester en ligne. Si un noeud perd la connectivité, un autre noeud peut intervenir pour maintenir la connexion active;
- villes intelligentes : la mise en réseau maillée sans fil est idéale pour étendre les signaux radio dans les garages de stationnement, les terrains de campus, les parcs d'activités et autres installations extérieures. Il est également possible d'établir une surveillance périodique des fuites d'eau, de gaz afin de garantir une certaine sécurité aux citoyens. Aussi, on peut envoyer des alertes de pollution et de smog en temps réel aux résidents d'une zone spécifique via des SMS ou des indicateurs visuels en cas d'urgence;
- automatisation de la maison : dans ces applications, le système peut par exemple contrôler l'éclairage ou les appareils afin d'optimiser la consommation ou le confort d'une maison. Pour ce faire, le système est équipé de capteurs et d'actionneurs pouvant communiquer entre eux en formant des réseaux. Ces réseaux nécessitent une faible latence (par exemple, allumer la lumière, changer le volume de la musique) et tolèrent la perte. Les réseaux maillés sans fil peuvent également aider à suivre et à gérer les températures dans une maison. Il est possible de configurer une passerelle alimentée, utiliser des capteurs de température et des noeuds dans chaque salle pour capturer des données en temps réel et ajuster les paramètres automatiquement;
- agriculture de précision : les réseaux maillés sans fil sont également parfaits pour surveiller l'exposition au soleil et les niveaux d'eau de vos cultures. Ils peuvent évoluer à moindre coût avec des nœuds sur une superficie afin de créer un réseau connecté par exemple à un cellulaire où des données climatologiques et environnementales en temps réel sont détectées et transmises.

1.1.2 Impact dans les applications industrielles

En tant que technologie émergente, l'Internet des objets (IoT) devrait offrir des solutions prometteuses pour transformer le fonctionnement et le rôle de nombreux systèmes industriels existants tels que les systèmes de transport et les systèmes de fabrication. Par exemple, lorsque l'IoT est utilisé pour créer des systèmes de transport intelligents, l'autorité responsable des transports sera en mesure de suivre l'emplacement actuel de chaque véhicule, de surveiller ses mouvements et de prévoir son emplacement futur et le trafic routier éventuel.

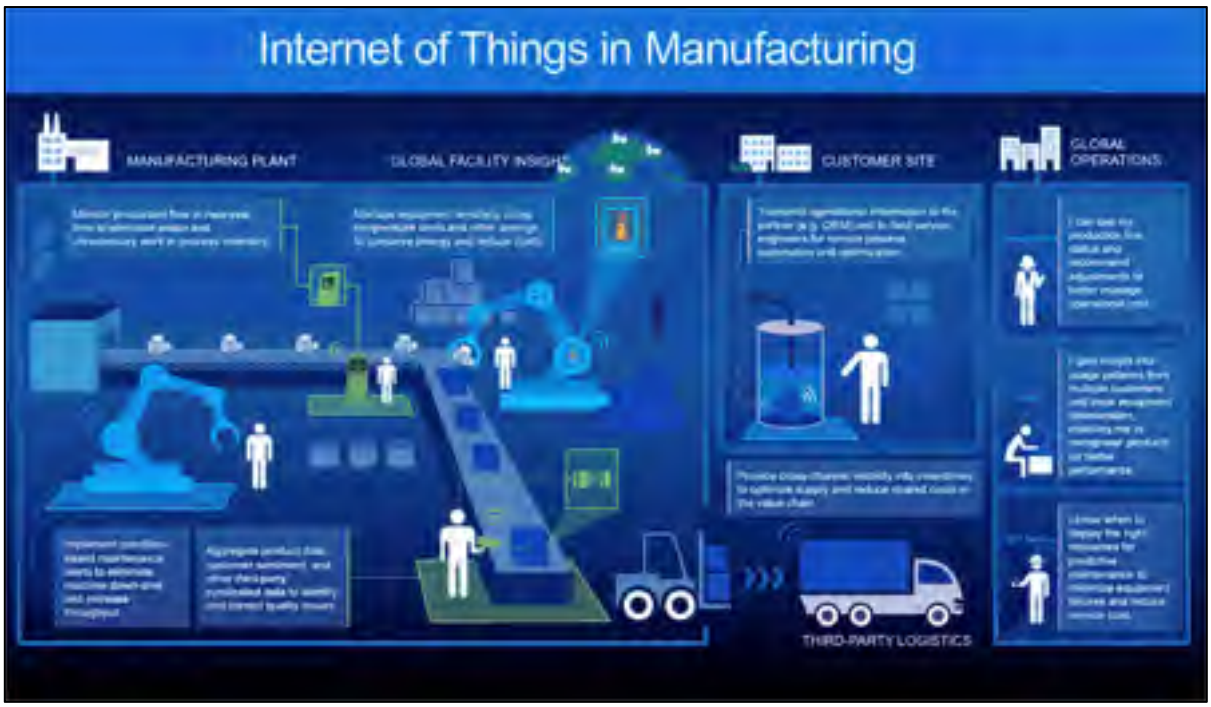

Figure 1.2 Utilisation de l'IoT dans les applications industrielles (« Industrial IoT Applications », 2018)

Dans ces applications IIoT (Industrial Internet of Things), l'objectif est de fournir aux machines industrielles la capacité de détecter leur environnement et de réagir en fonction de celui-ci. Des centaines ou des milliers d'appareils signalent des valeurs détectées telles que la température, la pression, les flux de trafic ou le niveau de remplissage du réservoir, utilisées à la fois pour contrôler l'environnement à distance et pour coordonner les étapes de production. Ces

applications ont des exigences de latence et fiabilité très strictes. La perte de données peut avoir un impact direct sur le processus d'automatisation (interruption de la chaîne d'approvisionnement) et entraîner une perte de production. Les appareils sont généralement densément situés et entourés d'objets métalliques, ainsi les trajets multiples et les interférences externes sont courants dans ces environnements.

Contrairement aux applications présentées précédemment, l'IIoT nécessite la combinaison d'une haute fiabilité, d'une latence minime et d'une durée de vie de la batterie importante. Ces applications doivent être capables de fonctionner même en cas de déploiement dense dans un environnement radio où des trajets multiples et des interférences externes sont présents.
Les exigences pour les applications IIoT peuvent être résumées de la façon suivante :

- latence (Qin et al., 2018) : les applications industrielles nécessitent une latence très faible et une variation de latence aussi faible que possible. La latence peut entraîner une dégradation des performances du système de contrôle et entraîner des pertes économiques. Cela peut faire en sorte que les données ne soient pas précises ou soient d'une utilisation limitée;
- fiabilité : la fiabilité est souvent définie comme la quantité de données reçues avec succès à l'extrémité de réception avec un délai minimum. Elle est souvent mesurée dans des mesures telles que la perte de paquets ou le taux de livraison de paquets (PDR). Dans les applications industrielles, cette métrique est primordiale, car cela peut avoir un impact direct sur le fonctionnement de l'industrie. Chaque bit d'information transmis sur le réseau ne doit pas être perdu. Les retransmissions sont souvent utilisées pour augmenter la fiabilité, mais cela augmente la latence et le débit global, car la bande passante est utilisée pour retransmettre les packages;
- la consommation énergétique (Harb & Makhoul, 2018): les capteurs sans fil utilisent généralement une batterie comme source d'alimentation qui devrait durer plusieurs mois, voire des années avant d'être remplacée ou chargée. La taille du réseau, la charge de trafic, le nombre de retransmissions, la mobilité, tout cela a un impact sur la consommation d'énergie. En raison de la limitation de la taille des nœuds, les nœuds ne

devraient pas traiter de gros protocoles exigeants pour la communication. En raison de la consommation d'énergie limitée, le noeud doit utiliser une couche MAC économe en énergie qui a un faible rapport cyclique et une faible utilisation du processeur.

1.1.3 Avantages des réseaux de capteurs sans fil

Les avantages des réseaux sont :

- auto-organisation et auto-configuration : les réseaux maillés sans fil ont une architecture réseau flexible. L'auto-organisation et l'auto-configuration réduisent le temps d'installation et les coûts de maintenance. Ainsi, en cas de changement de topologie, la connectivité est maintenue sans intervention humaine. En dehors de cela, cela améliore les performances du réseau (Seyedzadegan et al., 2011). Grâce à ces fonctionnalités, les fournisseurs de services réseau sont en mesure de modifier, d'élargir et d'adapter le réseau selon les besoins pour répondre aux demandes des utilisateurs finaux;
- faibles coûts de déploiement : les routeurs maillés sont sans fil et peuvent être utilisés dans des environnements multi-sauts. Ainsi, l'utilisation de routeurs sans fil dans de grandes zones est moins chère que celle des routeurs / points d'accès à sauts uniques qui disposent de connexions câblées. En règle générale, en raison de connexions filaires plus coûteuses à installer et à entretenir, le déploiement des réseaux maillés, avec une installation et une maintenance plus simple et plus rapide, permet de réduire les coûts d'exploitation;
- fiabilité accrue : dans un réseau maillé sans fil, il existe plusieurs chemins allant des noeuds sources aux destinataires. Cela fournit des chemins alternatifs en cas d'échec et permet également d'équilibrer les charges de trafic sur le réseau. L'équilibrage de la charge et la réduction des goulets d'étranglement via un autre routage peuvent augmenter considérablement la fiabilité du réseau dans les réseaux WMN;

- interopérabilité : un réseau maillé sans fil a une architecture compatible avec les normes existantes telles que : WiMAX, Cellulaire, Wi-Fi, Zigbee, Bluetooth, etc. Différentes technologies sans fil peuvent être retrouvées dans un même réseau maillé.

1.2 Motivation

De nos jours, les réseaux de capteurs sans fil s'attendent à recevoir de plus en plus de données et ça plus souvent (Bughin, 2016). Ces objets connectés génèrent des données qui peuvent être analysées pour identifier des tendances et des informations à des fins diverses. Au fur et à mesure que le nombre d'objets connectés augmente, le volume de données générées par l'Internet des objets explose. Ces nœuds doivent donc pouvoir traiter la multitude de données provenant d'innombrables sources. Dans ce contexte, la plus grande problématique au sein d'une communication sans fil est sans aucun doute la minimisation de l'énergie tout en garantissant une fiabilité et une faible latence, en particulier pour les applications industrielles. En effet, les conséquences négatives d'une charge de trafic élevée sont la création de congestion et éventuellement un débordement de file d'attente de paquets, en particulier dans les nœuds proches du sink. Un WSN est un réseau composé de petits dispositifs à ressources limitées qui communiquent sans fil dans un réseau. Chaque périphérique, appelé noeud de capteur, collabore avec d'autres périphériques du réseau pour effectuer certaines opérations, par exemple la surveillance de l'environnement (température, son, pression, etc.).

La consommation d'énergie est l'une des plus grandes préoccupations puisque cela doit pouvoir fonctionner sans intervention ni remplacement de batterie pendant des années. En effet, les utilisateurs souhaitent généralement déployer ces noeuds de capteurs de manière aléatoire dans une zone ciblée, souvent difficile d'accès, en grand nombre (des centaines voire des milliers de noeuds capteurs), leur remplacement est donc pratiquement impossible ou très coûteux. La latence dans les WSNs est également un fléau dans de nombreuses applications telles que l'automatisation industrielle, les systèmes de santé et les applications critiques pour la sécurité dans les bâtiments, telles que la détection des alarmes incendie.

C'est pourquoi, il est nécessaire de répondre à la fois aux exigences de consommation d'énergie et de latence par la conception d'un protocole de communication efficace et robuste pour les

applications qui souffrent de congestion et de débordement de file d'attente. La motivation de ce travail réside donc dans le fait de respecter ces exigences de performance afin de fournir des solutions à la demande croissante de données dans le monde d'aujourd'hui.

1.3 Contributions

TSCH (Time Slotted Channel Hopping), le protocole MAC défini dans IEEE 802.15.4e (IEEE802.15.4e., 2012), orchestre l'accès au support pour les WSNs selon un ordonnancement de communication basé sur le temps. La couche TSCH MAC joue un rôle crucial pour répondre aux exigences de consommation d'énergie et de latence. Plus précisément, les algorithmes de planification des cellules qui déterminent quelle action un nœud doit entreprendre dans chaque intervalle de temps (transmission, réception ou mise en veille). Un certain nombre d'études, qui visent le développement de ces algorithmes, ont vu le jour pour tenter de répondre à ces différents enjeux.

Dans ce mémoire, nous proposons OSCAR, un algorithme d'allocation de cellules TSCH d'ordonnancement autonome basé sur Orchestra (Duquennoy et al., 2015) pour résoudre ces problèmes. Cette nouvelle conception diffère d'Orchestra par le fait qu'OSCAR attribue les slots en fonction de l'emplacement du nœud par rapport à la racine. Le but de cet algorithme est d'allouer des slots aux nœuds en fonction de leurs besoins. En effet, Orchestra détermine l'ordonnancement TSCH pour chaque nœud quelle que soit sa charge de trafic, ce qui peut affecter de manière significative le délai de communication et la consommation d'énergie. Il y a donc une accumulation de paquets dans les files d'attente des nœuds proches de la racine en raison de leur forte charge, ce qui peut conduire à une perte de paquets induite par un débordement de la mémoire tampon. OSCAR se caractérise par la classification des nœuds en classes en fonction de la distance à la racine. Chaque nœud se voit attribuer un ID de classe, représentant la couche qu'il occupe en fonction de la distance géographique de la racine. De plus, OSCAR fournit une méthode pour ajuster le cycle de service radio de chaque nœud et améliorer l'efficacité énergétique en éteignant la radio en l'absence de données à recevoir.

Dans ce mémoire, nous implémentons OSCAR sur Contiki-ng et évaluons ses performances. L'analyse des performances de l'algorithme OSCAR se fera dans un premier temps à l'aide

d’une simulation puis une validation du comportement par mesures expérimentales sera effectuée sur un banc d’essai composé de 11 nœuds CC2650 Launchpad. Le système OSCAR est validé à la fois par des simulations et des expériences réelles.
Ainsi, les contributions de cette étude peuvent être répertoriées comme suit :

- une architecture de classification des nœuds permettant une meilleure répartition des cellules allouées en fonction de leur position;
- un système pour réduire les cellules allouées aux nœuds inactifs afin de limiter les pertes énergétiques.

1.4 Publication

Le travail présenté dans ce mémoire va être soumis au journal MDPI Sensors à l’automne 2020 sous le nom de « OSCAR : An Optimized Scheduling Cell Allocation Algorithm for Convergecast in IEEE 802.15.4eTSCH Networks ».

1.5 Organisation du mémoire

Le mémoire est organisé en 4 chapitres. Le chapitre 2 présente le contexte général et les informations de base nécessaires sur le protocole TSCH et RPL.

Le chapitre 3 donne un aperçu des travaux connexes récents sur les algorithmes d’ordonnancement. Ce chapitre résume la revue de la littérature et présente les différentes approches de conception. Le problème de congestion dans les trafics convergeant ainsi que les limitations d’Orchestra, qui forme la base d’OSCAR, sont également abordés.

Le chapitre 4 décrit la conception détaillée et la mise en œuvre d'OSCAR, la solution optimisée d'allocation de cellules proposée dans ce mémoire. Ce chapitre présente les performances d'OSCAR en le comparant à Orchestra à l'aide d'un banc de test. Pour finir, une discussion

résume les résultats obtenus avec les avantages et inconvénients et compare OSCAR avec d'autres approches.

Enfin, une conclusion avec les futures orientations de recherche ponctue ce mémoire.

CHAPITRE 2

CONTEXTE GÉNÉRAL

2.1 Modèle OSI

Pour commencer, il est important de présenter le modèle OSI ci-dessous, qui est un modèle de référence pour la façon dont les systèmes communiquent sur un réseau.

Figure 2.1 Pile de protocoles OSI

Dans un réseau, tous les périphériques, également appelés nœuds, utilisent ces couches pour communiquer entre eux. Chaque couche sert la couche au-dessus de celle-ci, qui à son tour sert celle au-dessus d'elle-même.

Par exemple, lorsque des informations sont échangées entre des nœuds, le processus fonctionnera comme suit : un flux de données circule dans les couches de la machine qui envoie le message, puis se propage dans le réseau et finalement dans les couches du destinataire.

Dans le contexte des réseaux de capteurs sans fil, les couches les plus importantes sur lesquelles seront accentués les efforts sont la couche MAC et la couche réseau où a lieu le routage.

2.1.1 Couche MAC

La conception du protocole MAC dans la couche liaison de données constitue un défi important. La zone sans fil est une ressource partagée et les nœuds dans la même portée de communication peuvent interférer chaque fois qu'ils essaient d'accéder au support en même temps. Ainsi, le protocole de contrôle d'accès au support (MAC) de la couche liaison est en charge de la régulation de l'accès au support. Il décide quel nœud peut utiliser le support, quand et sur quel canal.

Les protocoles MAC peuvent être basés sur le temps reposant sur le mécanisme d'accès multiple par répartition dans le temps (TDMA), ou basés sur la fréquence en utilisant l'approche d'accès multiple par répartition en fréquence (FDMA), ou basés sur le code suivant l'accès multiple par division de code (CDMA). Il peut également être basé sur un accès aléatoire à l'aide de l'algorithme d'écoute d'un Support à accès multiple (CSMA) ou même hybride combinant certains de ces schémas d'accès au support.

2.1.2 Couche réseau

La couche réseau a pour objectif principal de transférer les paquets correctement et efficacement de l'expéditeur au nœud récepteur. Elle joue un rôle important dans la transmission des données et a la responsabilité de construire des chemins entre les nœuds du réseau. Dans les réseaux de capteurs sans fil, la couche réseau est représentée par des protocoles de routage qui doivent construire le réseau et guider les nœuds pour acheminer leurs paquets de données vers les nœuds de destination et en particulier vers le sink en cas de profil de trafic de données convergeant. Selon les besoins de l'application, le protocole de routage utilisé peut être différent.

2.2 Trafic convergeant dans les réseaux de capteurs sans fil

Convergeant fait référence à un modèle de communication dans lequel le flux de données d'un ensemble de nœuds est transmis à un seul nœud du réseau. De nombreuses applications de réseau de capteurs nécessitent un comportement convergeant pour la collecte de données où les nœuds doivent constamment percevoir ce qui se passe dans leur environnement (surveillance des incendies, surveillance de l'activité sismique, mesure des températures, surveillance de la pollution). Dans le trafic convergeant, les données collectées dans l'ensemble du réseau sont transmises vers le sink par les nœuds. Dans cette transmission, plusieurs nœuds doivent utiliser la communication multi-sauts pour transmettre leurs données, car ils sont hors de portée du sink. Par conséquent, ces réseaux souffrent de graves problèmes de congestion du trafic surtout au niveau du sink. La figure 2.2 présente un exemple de transmission de données convergeant.

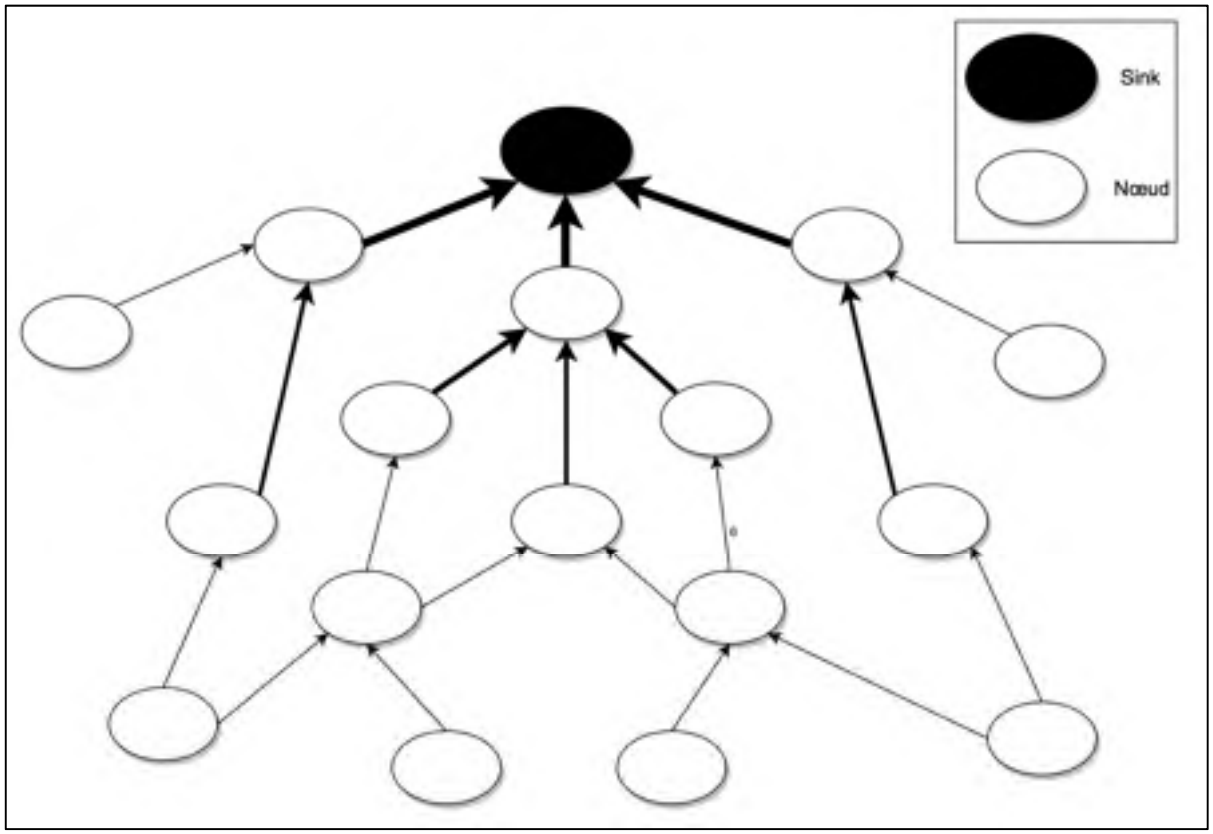

Figure 2.2 Un exemple de réseau de capteurs sans fil avec un trafic convergeant

2.3 Standard IEEE 802.15.4 et IEEE 802.15.4e-TSCH

Dans le but de faire fonctionner et interagir les systèmes fabriqués et développés par divers fournisseurs les uns avec les autres, certaines normes doivent être suivies ou réalisées. IEEE 802.15.4 est un exemple d'une telle norme pour la couche physique et MAC. Cette norme est conçue pour le réseau personnel sans fil à faible débit (LR-WPAN) dans lequel les applications nécessitent un faible débit de données et les appareils disposent de faibles ressources de calcul. Cette norme est un choix populaire pour les applications à faible puissance. Dans cette section, nous expliquerons les principaux concepts du protocole standard IEEE 802.15.4 et discuterons ensuite des caractéristiques importantes introduites dans l'amendement IEEE 802.15.4e.

2.3.1 Standard IEEE 802.15.4

La norme IEEE 802.15.4 est un protocole de communication défini par l'IEEE destiné aux réseaux sans fil à bas débit et faible consommation qui ont pour nom LR-WPAN (Low Rate Wireless Personal Area Network). Il spécifie et standardise la couche physique et la sous-couche MAC des périphériques réseau. La couche physique (PHY) contient l'émetteur/récepteur radio (RF).

L'IEEE 802.15.4 a été introduite en 2003 et révisée par la suite en 2006, 2011, 2015 et tout récemment en 2020. Les versions les plus récentes intègrent le saut de canal synchronisé (TSCH) dans la couche MAC sous la norme 802.15.4e. La topologie de la norme IEEE 802.15.4 est sous forme d'un réseau maillé, en étoile. Elle est gérée par le groupe de travail IEEE 802.15 et constitue la base des technologies Zigbee, ISA100.11a, WirelessHART et Thread. Chacune d'elles élargissant encore la norme en développant les couches supérieures qui ne sont pas définies dans IEEE 802.15.4.

En termes d'appareils, la norme définit deux types de noeuds de réseau :

- le premier est le dispositif ayant toutes les fonctions possibles (FFD). Un dispositif FFD peut agir en tant que coordinateur ou routeur et peut donc être responsable de la gestion du réseau ou de son extension en recherchant des itinéraires et en transférant des paquets de données. Ces dispositifs FFD disposeraient donc davantage de ressources

en processeur et en mémoires et se mettraient en veille pendant des périodes relativement courtes ou écouteraient en permanence le canal, ce qui entraînerait une plus grande consommation d'énergie;

- le second est le dispositif ayant des fonctions limitées (RFD). Contrairement au FFD, ce dispositif consomme moins d'énergie, car il ne transfère que les paquets d'application sans capacité de routage et peut rester en mode basse consommation pendant des durées relativement longues. Le RFD est prévu pour des appareils extrêmement simples avec des besoins en ressources et de communication très modeste. Ces derniers ne sont souvent pas essentiels au réseau et sont considérés comme des dispositifs d'extrémité (end device). Le rôle de coordinateur ne peut être joué que par un périphérique FFD, tandis que les périphériques RFD, en raison de leurs limitations, ne peuvent communiquer et se connecter qu'au coordinateur.

Les appareils conformes à la norme IEEE 802.15.4 peuvent utiliser l'une des trois bandes de fréquences possibles pour le fonctionnement :

- 868.0–868.6 MHz,
- 902–928 MHz,
- 2400–2483.5 MHz.

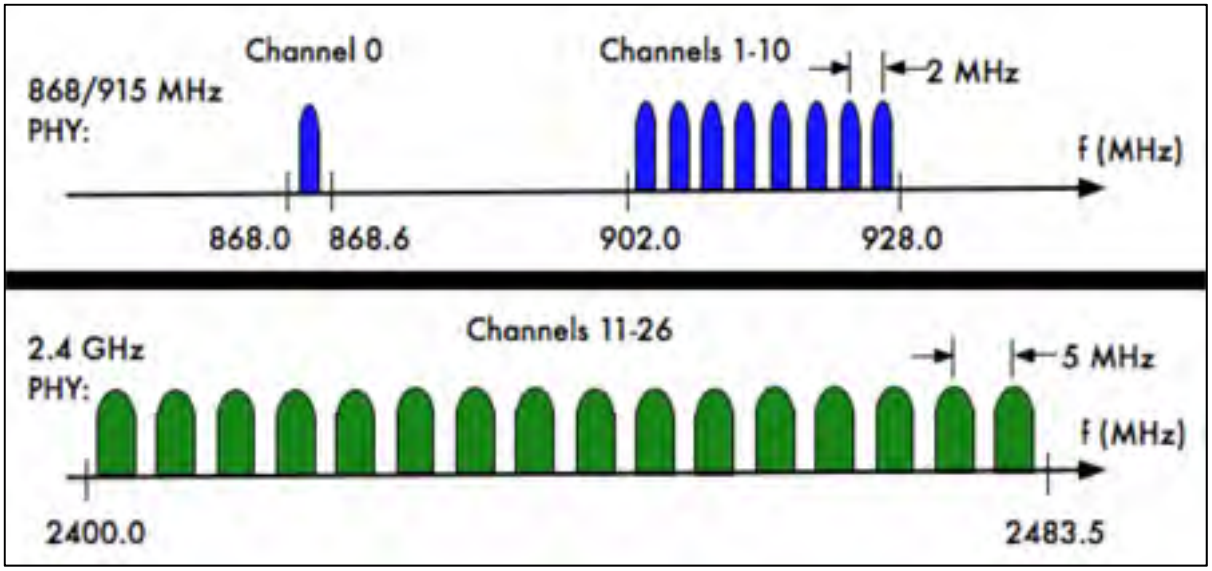

Figure 2.3 Bandes de fréquences du standard 802.15.4. (Marques, 2017)

De plus, en théorie, le débit de données maximal réalisable est de 250 kb/s lorsqu'il est utilisé à une fréquence de 2,4 GHz. Cependant, en pratique, en raison de liaisons asymétriques, les paquets peuvent être supprimés et des retransmissions peuvent se produire.

Tableau 2.1 Débit de données et nombre de canaux des bandes de fréquences

Bande de fréquence	Débit de données (kbps)	Nombre de canaux
868.0–868.6 MHz	20	1
902–928 MHz	40	10
2400–2483.5 MHz	250	16

2.3.1.1 Limitations

Les performances du protocole MAC 802.15.4 montre un certain nombre de limitations et lacunes qui le rendent impropre aux applications critiques qui ont des exigences strictes en termes de fiabilité, latence, efficacité énergétique, évolutivité et fonctionnent généralement dans des environnements difficiles. Ces limites sont les suivantes :

- manque de fiabilité MAC,
- latence accrue,
- aucune technique de saut de fréquence intégrée,
- mauvaise gestion de l'énergie.

2.3.2 Amélioration MAC du standard IEEE 802.15.4e

La norme IEEE 802.15.4e a été proposée en tant qu'amendement de la norme IEEE 802.15.4-2011 héritée, afin de satisfaire aux exigences des applications IoT émergentes, en particulier dans le domaine industriel.

En général dans un réseau sans fil, une part importante de la gestion de l'énergie est réalisée par la sous-couche mac, c'est pourquoi un bon nombre d'optimisations sont possibles à ce niveau.

IEEE802.15.4-2011 (la version antérieure de IEEE 802.15.4e) exigeait que les radios des noeuds de relais soient toujours allumées. En effet, sans synchronisation horaire, un noeud ne peut pas savoir quand son voisin va envoyer une trame. La norme IEEE 802.15.4e a été proposée en tant qu'amendement de la norme IEEE 802.15.4-2011 existante, afin de satisfaire aux besoins de plus de performances dans les réseaux denses et les applications IoT émergentes, en particulier dans le domaine industriel. La norme IEEE 802.15.4e n'apporte que des améliorations à la sous-couche MAC, sans toucher aux couches physiques et de sécurité.

2.3.2.1 Améliorations fonctionnelles

IEEE 802.15.4e introduit des améliorations fonctionnelles par rapport au standard IEEE 802.15.4. Ci-dessous quelques améliorations énumérées:

- accès multicanal : l'un des principaux inconvénients de l'IEEE 802.15.4-2011 d'origine était le manque d'accès multicanal. L'accès multicanal contribue principalement à atténuer la dégradation des performances due aux interférences dans le réseau. La norme précédente ne prenait en charge qu'un seul canal de communication, ce qui restreignait la capacité de prendre en charge un grand nombre de nœuds sans conflit, et donc de détériorer les performances de son réseau, son délai global et son débit. Avec

cette nouvelle norme, certains modes MAC d'IEEE 802.15.4e (TSCH et DMSE), surmontent ces limitations en prenant en charge le fonctionnement multicanal au niveau de la couche physique. Avec TSCH par exemple, via le mécanisme de saut de canal, la séquence de saut de canal est prédéterminée statiquement à l'avance;

- faible latence et faible énergie : le protocole IEEE 802.15.4e offre une meilleure prise en charge des communications à faible latence, plus adaptées aux applications de contrôle industriel, tout en offrant un compromis entre la latence et l'efficacité énergétique. IEEE 802.15.4e permet aux appareils de fonctionner à un cycle d'utilisation très faible et fournit également une latence déterministe, qui est une exigence principale pour les applications à temps critique;
- éléments d'information (IE): les éléments d'information sont un concept déjà défini dans la norme IEEE 802.15.4 d'origine; cependant, il a été étendu dans l'IEEE 802.15.4e avec des fonctionnalités supplémentaires. Outre les éléments d'information basés sur l'en-tête et la couche de gestion utilisés dans IEEE 802.15.4, un IE unique a été introduit pour prendre en charge les divers comportements MAC. Par exemple, l'IE de TSCH contient des informations pertinentes telles que la longueur et l'ID de l'intervalle de temps (timeslot) ou la séquence de saut de canal;
- balises améliorées: la balise améliorée (EB) est une révision du format de balise standard utilisé dans les réseaux IEEE 802.15.4-2011. Il offre une plus grande flexibilité et il est utilisé pour fournir un contenu de balise spécifique à l'application. Les EBs contiennent les IE pertinents tels que les séquences de saut de canal par exemple;
- association rapide: La procédure d'association 802.15.4 introduit un délai important pour économiser de l'énergie. Cela est principalement dû au fait que dans IEEE 802.15.4, le noeud doit attendre la fin du temps d'attente de réponse MAC avant de demander les données d'association au coordinateur. Pour résoudre ce problème, l'IEEE 802.15.4e introduit un mécanisme d'association rapide, en vertu duquel le périphérique demande l'association au coordinateur PAN (Personal Area Networks). Le mécanisme d'association rapide permet à un nœud de s'associer en un temps réduit.

2.3.2.2 Cinq modes de comportement MAC

Cinq nouveaux modes de comportement MAC, qui visent chacun un domaine d'application spécifique, ont été introduit dans le standard IEEE 802.15.4e :

- Time Slotted Channel Hopping (TSCH) : le mode MAC TSCH offre une fiabilité très élevée et des garanties de temps critique. Ce mode est un candidat approprié pour la mise en œuvre d'un réseau de capteurs dans un milieu industriel où les interférences affectent la fonctionnalité des appareils sans fil. TSCH prend en charge les communications multi-sauts et multicanaux via une approche TDMA, ce qui améliore considérablement la fiabilité du réseau en atténuant efficacement les effets des interférences;
- Low Latency Deterministic Network (LLDN) : le mode MAC des réseaux déterministes à faible latence (LLDN) est conçu pour les réseaux à un seul canal et à un seul saut. Il cible les applications qui exigent généralement une latence très faible et une robustesse en raison de la nature critique des données;
- Radio Frequency Identification Blink (BLINK) : le mode BLINK permet à l'appareil de communiquer son ID, un identifiant unique, et des données de charge utile supplémentaires aux appareils avec lesquels ils communiquent. Les appareils peuvent se connecter entre eux sans aucune association ou reconnaissance préalable. La trame BLINK peut être utilisée uniquement à des fins de transmission et coexiste avec d'autres appareils du réseau. Il est destiné aux applications tels que la localisation et le suivi de personnes par exemple;
- Deterministic and Synchronous Multi-channel Extension (DSME) : le comportement MAC DSME (Deterministic Synchronous Multichannel Extension) cible les applications ayant des exigences de QoS (Qualité du service) telles que la latence déterministe, la haute fiabilité et l'évolutivité. Il combine un accès au support basé sur la contention et une répartition dans le temps et est spécialement conçu pour les réseaux multi-sauts et maillés. DSME et TSCH sont les seuls modes qui prennent en charge les topologies d'égal à égal (peer-to-peer) et à sauts multiples;

- Asynchronous multi-channel adaptation (AMCA) : Le mode AMCA (Asynchronous Multi Channel Adaptation) peut être utilisé dans les applications de réseaux de capteurs denses. Il peut être activé pour le mode Non-Beacon-Enabled des réseaux IEEE 802.15.4e.

Le tableau ci-dessous résume les différents modes de comportement MAC du standard IEEE 802.15.4e.

Tableau 2.2 Modes de comportement MAC du standard IEEE 802.15.4e

Modes MAC	**Évolutivité**	**Fiabilité**	**Efficacité énergétique**	**Accès multicanal**	**Applications possibles**
TSCH	Oui	Très élevé	Plus élevé que DSME	Oui	Applications à temps critique avec des exigences d'efficacité énergétique
LLDN	Non	Très élevé	Plus élevé que DSME	Non	Automatisation industrielle - nombre fixe de nœuds
BLINK	-	Moyen	Élevé	Non	Suivi et identification
DMSE	Oui	Élevé	Dépend du nombre de nœuds	Oui	Applications à temps critique à haute évolutivité
AMCA	Oui	Moyen	Moyen	Oui	Applications non sensibles au temps avec des exigences d'évolutivité élevées

2.4 Time Slotted Channel Hopping

2.4.1 Description

Le mode TSCH a été introduit en 2012 en tant qu'amendement (IEEE 802.15.4e) à la partie MAC (Medium Access Control) de la norme IEEE 802.15.4. Pour améliorer les protocoles MAC, les groupes de travail ont introduit le système Time Slotted Channel Hopping (TSCH). Il a été adopté et normalisé dans l'amendement IEEE 802.15.4e publié en 2012 (IEEE802.15.4e., 2012). Le mode TSCH a été conçu pour des applications avec une fiabilité rigoureuse et des contraintes de consommation d'énergie. Avec TSCH, tous les noeuds du réseau sont synchronisés sur une base de temps fractionnée, ce qui fait que la mesure et la transmission de données sont effectuées périodiquement. Une planification indique à un noeud quelle action effectuer dans chaque emplacement de la trame : transmettre, recevoir ou dormir. La diversité de fréquence est utilisée pour atténuer les effets des évanouissements par trajets multiples et pour améliorer la résistance aux interférences externes. En effet, TSCH effectue la division des fréquences, c'est-à-dire que la bande de fréquences disponible est divisée en canaux par la couche physique (De Guglielmo et al., 2016). Chaque canal est une ressource disponible que la couche MAC peut utiliser. Plusieurs canaux peuvent être utilisés en même temps, ce qui signifie que dans chaque créneau horaire, tous les canaux disponibles peuvent être utilisés en parallèle. Il en résulte la notion de cellule, où une cellule fait partie de la planification qui peut être identifiée avec un décalage de créneau et un décalage de canal.

Le saut de canal est réalisé en envoyant des paquets successifs sur différentes fréquences. La séquence de saut de canal est fixe et connue de tous les noeuds. Dans un intervalle de temps à une fréquence particulière, un noeud peut transmettre, écouter ou dormir. Le planificateur établit le calendrier de communication afin de satisfaire les exigences de bande passante, de latence et de fiabilité des applications. Le mode TSCH joue un rôle important pour un fonctionnement à faible consommation et une fiabilité élevée pour les réseaux sans fil industriels.

À noter que le TSCH n'apporte aucune modification à la couche physique spécifiée dans la norme IEEE 802.15.4.

TSCH définit également deux types de cellules :

- cellules partagées : si une cellule est marquée comme partagé, cela signifie que plusieurs nœuds sont autorisés à émettre en même temps et sur le canal de fréquence, mais cela peut provoquer une collision qui est résolue en utilisant un algorithme de retransmission exponentielle de secours appelé TSCH CSMA-CA;
- cellule dédiée : si une cellule est dédiée à une paire de nœuds, cela signifie que c'est une cellule sans contention. C'est à dire que dans cette cellule fixée, il n'y a pas de concurrence avec les autres nœuds entre la source et la destination. Lors de l'utilisation d'une cellule dédiée, un nœud source peut supposer qu'aucun autre émetteur n'utilisera cette même cellule dédiée avec le nœud de destination.

2.4.2 Structure de la slotframe

Dans TSCH, le temps est divisé en créneaux horaires (timeslot) qui durent généralement environ 10 ms. Les timeslots sont regroupés dans une slotframe qui se répète au fil du temps. La façon dont les timeslots sont organisés en slotframe est appelée un ordonnancement. À chaque créneau horaire est associée une action: dormir, transmettre ou recevoir pour que chaque nœud sache à l'avance quoi faire ensuite. Chaque emplacement est affecté à une paire de nœuds source et de destination. Un nœud source transmet à son emplacement attribué et le nœud de destination se réveille en mode de réception dans cet emplacement. Si un paquet est perdu, la retransmission a lieu dans le timeslot de la slotframe suivante. Une durée d'intervalle de temps est suffisamment longue pour transmettre un paquet de taille de charge utile maximale et pour recevoir l'accusé de réception correspondant.
Si l'action est de dormir, le nœud éteint sa radio et attend la durée d'un timeslot. En veille, le nœud consomme très peu d'énergie. Si l'action est de transmettre, le nœud allume sa radio et transmet une trame. Si l'action est de recevoir, le nœud allume sa radio et écoute une trame.

Chaque slot dans la slotframe est marqué d'un numéro de slot unique appelé numéro de slot absolu (ASN). L'ASN est utilisé pour avoir une vue globale du timeslot actuel dans le réseau et aide à la synchronisation pour les nœuds qui souhaitent rejoindre un réseau. Elle permet également de calculer la fréquence dans laquelle une paire de nœuds évolue. L'ASN est continu et ne se réinitialise pas après la fin d'une slotframe. Il peut être considéré comme le nombre total de timeslot écoulés depuis le démarrage du réseau. En effet, lorsque les nœuds qui ont rejoint reçoivent l'ASN, ils utilisent la fonction ci-dessous pour calculer sur quel canal communiquer.

$$frequency = (ASN + channel\_offset)\%number\_channels \qquad (2.1)$$

Où $channel\_offset$ est le décalage de canal de cette cellule et $number\_channels$ est le nombre total de canaux disponibles. Cette équation implémente le mécanisme de saut de canal en renvoyant une fréquence différente pour la même liaison à différents intervalles de temps. Cela garantit que pendant le temps tous les canaux disponibles sont utilisés pour les communications dans une liaison et, par conséquent, permet d'atténuer l'effet négatif des interférences externes. La figure ci-dessous montre un exemple d'ordonnancement TSCH.

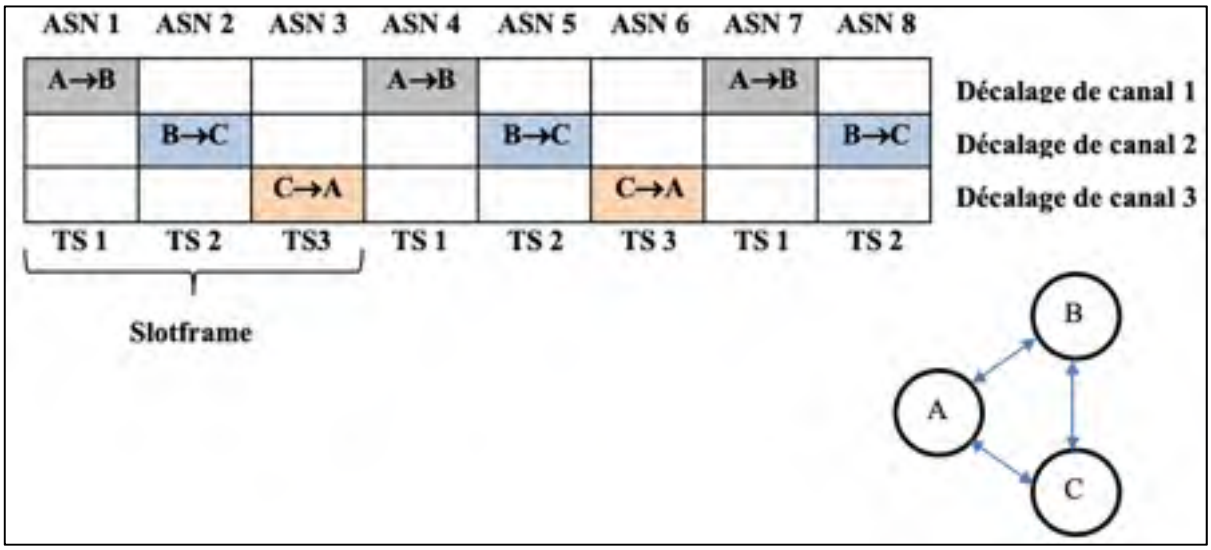

Figure 2.4 Exemple d'ordonnancement TSCH

2.4.3 Saut de canal

L'une des principales caractéristiques de TSCH est la communication multicanal, basée sur le saut de canal. TSCH peut utiliser 16 canaux de communication qui sont définis par un décalage de canal sur la bande de fréquence 2,4 GHz (la bande 2400-2483,5 MHz a 16 canaux, 11-26). En fonction de la législation nationale, tous les canaux peuvent ne pas être disponibles à utiliser. Chaque canal de fréquence est identifié par un décalage de canal. Chaque transmission a lieu sur un canal de fréquence différent, qui est extrait d'une liste de canaux appelée liste de séquences de sauts.
Dans TSCH, la liaison entre deux nœuds est définie par [n, décalage de canal] avec n le timeslot où les deux nœuds communiquent et leur décalage respectif. Cela signifie que si le nœud A a un créneau d'émission vers le nœud B sur le décalage de canal 1, le nœud B aura un créneau de réception pour le nœud A sur le même décalage de canal.

2.5 RPL

Le protocole de routage RPL (Routing Protocol for Low Power and Lossy Networks) pour les réseaux à faible consommation et à perte (LLN) a été développé et publié sous la référence (RFC6550) par le groupe de travail ROLL de l'IETF (T. Winter et al., 2012). RPL est essentiellement un protocole de routage vectoriel à distance proactif pour les réseaux LLN basés sur IPv6. La topologie RPL est organisée en une structure DODAG (Destination Oriented Direct Acyclic Graph). Outre la mémoire des périphériques, le nombre de noeuds dans un DODAG n’est pas limité. Typiquement, la racine DODAG (DODAGroot) correspond au routeur de bordure IPv6 qui relie les nœuds au monde extérieur (Internet ou tout autre réseau) et à partir duquel il peut recevoir des commandes pour la gestion des données collectées. Chaque nœud est associé à un rang, c'est-à-dire à sa distance à la racine en utilisant une fonction de coût (plus le nœud est proche de la racine, plus il a un rang plus petit). Un nœud envoie un paquet vers la racine en le transmettant à un nœud voisin de rang inférieur.

L'établissement et la maintenance du DODAG sont assurés par des messages de contrôle DIO (DODAG Information Object) diffusés à l'aide d'un minuteur de maintien. Les paquets DIO contiennent des informations telles que les métriques utilisées pour le calcul du coût du chemin. Au départ, seule la racine DODAG fait partie de la topologie active RPL. Par la suite, les messages DIO diffusent périodiquement les paramètres de configuration comme le DODAG-ID et son rang au sein de son voisinage. Dès qu'un nœud reçoit plusieurs DIO provenant de différentes sources, il sélectionne l'un d'entre eux comme son parent préféré (selon le rang) qui agit également comme le prochain bond pour atteindre la racine DODAG. Par la suite, lorsqu'un nouveau nœud entre dans le réseau à partir du nœud de départ spécifié en tant que racine, il acquiert des informations de voisinage via les messages DIO. Parmi les nœuds déjà présent, un nœud est sélectionné comme nœud parent et la valeur RANK du nouveau nœud est calculée en fonction de la valeur RANK de ce nœud parent.

RPL définit également le DAO (Destination Advertisement Object) qui est un message ascendant permettant à un nœud d'annoncer son préfixe pour l'établissement de routes descendantes. Ainsi par exemple, lors de la modification du parent préféré en raison de la variation de lien, un nœud envoie un DAO par la voie ascendante au nouveau parent préféré pour configurer une nouvelle route descendante. Pour s'assurer que le message a bien été reçu, un DAO-Ack est envoyé par le parent au nœud enfant.

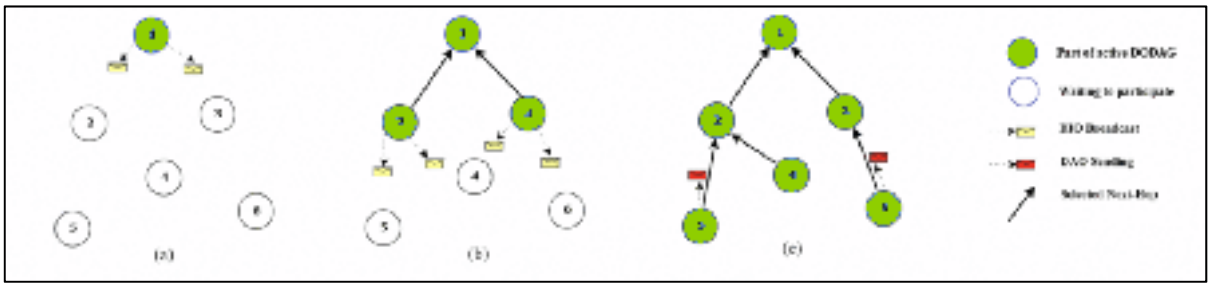

Figure 2.5 Construction DODAG : (a) Démarrage de RPL: paramètres de configuration de diffusion du nœud racine via DIO. (b) Les nœuds environnants commencent à participer au DODAG en construction. (c) Lorsque tous les nœuds sont engagés, le DODAG est construit. (Kamgueu et al., 2018)

RPL prend en charge plusieurs mesures de routage et modèles de trafic afin de répondre aux critères de consommation d'énergie et de fiabilité, et est conçu pour fonctionner avec des contraintes de mémoire strictes.

2.6 Ordonnancement d'Orchestra

Le travail présenté dans ce mémoire est basé sur Orchestra. Orchestra est une solution d'ordonnancement récente pour TSCH qui apporte des avantages importants tels que l'utilisation de règles d'ordonnancement simples avec un taux de livraison élevé. Orchestra présente plusieurs atouts qui en font une solution d'ordonnancement prometteuse pour les réseaux TSCH. L'ordonnancement Orchestra n'est ni centralisé ni distribué et est calculé de manière autonome, où chaque nœud du réseau maintient localement son propre planning basé sur les informations de la couche de routage.

2.6.1 Timeslot d'orchestra

Quatre principaux types de timeslots sont identifiés dans Orchestra :

- timeslots partagés communs (CS) : Les timeslots CS Orchestra consistent en un timeslot partagé utilisé par tous les nœuds du réseau pour Rx (réception) et Tx (transmission). Il s'agit d'un seul slot partagé pour tous les nœuds;
- timeslots partagés basés sur le récepteur (RBS) : En règle générale, les RBS sont utilisés pour la communication enfant-parent. À chaque nœud, la configuration RBS correspond à un timeslot de réception (Rx) et un timeslot de transmission (Tx) par voisin. C'est à dire que si le nœud parent à X voisins enfants, les X nœuds se disputent la cellule pour envoyer le paquet au nœud parent, ce qui peut provoquer une collision de nœuds cachés;
- timeslots partagés basés sur l'expéditeur (SBS) : SBS est similaire à RBS, sauf que le slot SBS Orchestra présente un slot Rx par voisin et un seul slot Tx. Cela entraîne une consommation d'énergie plus élevée que RBS car les slots Rx nécessitent toujours

d’être à l’ecoute, alors que les slots Tx ne coûtent rien en l'absence de trafic. L’avantage par rapport aux slots RBS est que les slots SBS peuvent diminuer les conflits car il n’y a pas qu’un seul slot Rx pour tous les nœuds enfants;

- timeslots dédiés basés sur l'expéditeur (SBD) : Orchestra parvient à une communication sans contention avec SBD, qui sont similaires à SBS sauf qu'ils utilisent des slots TSCH dédiés au lieu de partagés. Pour parvenir à cette communication sans contention, cela exige que la longueur de la slotframe soit plus longue que le nombre de nœuds du réseau pour placer des slots Tx uniques sur chaque nœud.

2.6.2 Slotframe d’orchestra

L’ordonnancement pour chaque nœud se fait avec différentes slotframe ayant des longueurs différentes. Les slotframes se répètent à des périodes qui sont mutuellement amorcées, assurant ainsi un cycle indépendant. Chaque slotframe a un identifiant unique, tel que plus le chiffre est petit, plus la priorité de la slotframe est élevée. En cas de chevauchement de timeslots de différentes slotframes, le timeslot de la slotframe de priorité la plus élevée est prioritaire. Les timeslots sont ignorés chaque fois qu'ils sont en concurrence avec un timeslot qui présente une priorité plus élevée.
La configuration d’orchestra se compose de 3 types de slotframes, décrits comme suit :

- slotframe TSCH EB: cette slotframe, qui a la priorité la plus élevée, est dédié aux transmissions EB, utilisées pour l'association TSCH et la synchronisation enfant-parent. Elle est composée d'un emplacement SBD, de sorte que les transmissions de balises TSCH soit sans conflit. Ce type de slotframe est composé de 397 timeslots. Ainsi, chaque nœud a un emplacement Tx et un emplacement Rx pour écouter les EB à partir de sa source de temps;
- slotframe de diffusion RPL: ce type de slotframe a une priorité inférieure à celle de la slotframe EB et est destiné à la diffusion de messages de contrôle, tels que les DIOs, et à la transmission de tout paquet lorsqu'une cellule de monodiffusion n'est pas planifiée.

Les transmissions de messages de diffusion RPL ont lieu dans 1 emplacement CS. En effet dans une slotframe de diffusion, orchestra active une seule cellule fixe où tous les nœuds se réveillent pour prendre en charge la diffusion. La slotframe est composé par défaut de 31 timelsots. Elle se répète donc plus fréquemment que la slotframe EB en raison de la longueur de la slotframe qui est plus petite;

- slotframe d'application (unicast) : cette slotframe a la priorité la plus basse et est destiné au trafic unicast entre le parent RPL et tous ces enfants. Elle assure les échanges de paquets entre deux nœuds dédiés. À cette fin, Orchestra propose deux types de planification: basée sur le récepteur (RB) et basée sur l'expéditeur (SB). Dans RB, chaque nœud n'a qu'une seule cellule fixe dans une slotframe unicast pour recevoir des paquets de n'importe quel nœud. Alors que dans SB, chaque nœud n'a qu'une seule cellule fixe dans une slotframe unicast pour envoyer des paquets à n'importe quel nœud. Une slotframe est composé uniquement de 11 timeslots.

La figure ci-dessous montre un exemple de l'ajustement automatique des slots alloués avec le degré de priorité sachant que les longueurs des slotframes correspondant à l'application, à la diffusion RPL et EBs sont respectivement de 2, 3 et 5.

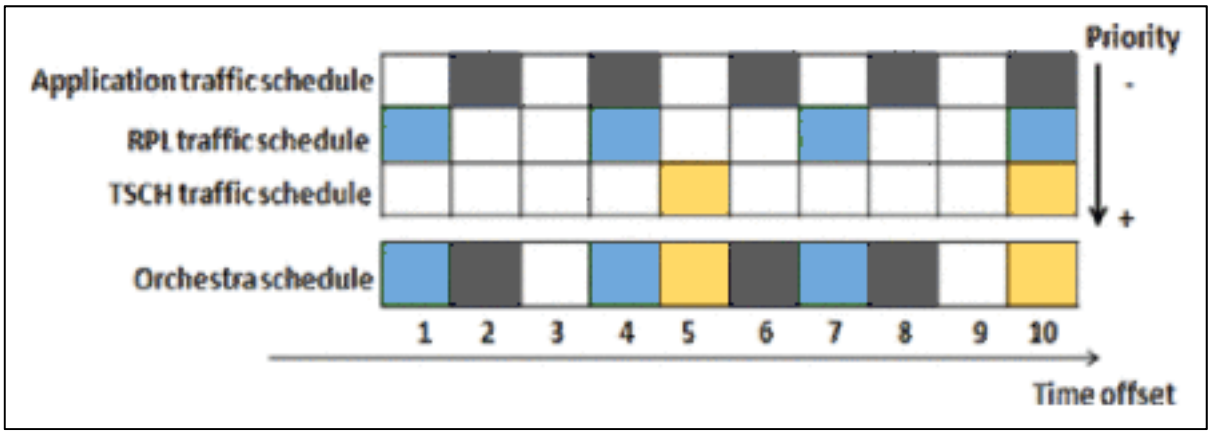

Figure 2.6 Illustration des slotframes Orchestra(Duquennoy et al., 2015)

2.7 Résumé

Pour permettre aux appareils limités en énergie et en puissance de calcul de fonctionner dans des environnements industriels, les bonnes technologies doivent être sélectionnées. Ce chapitre présente un examen approfondi du standard IEEE 802.15.4. Le mode TSCH ainsi que le protocole de routage RPL, qui répondent aux exigences de latence et énergie très strictes de ces capteurs sans fil, sont également introduits.

Enfin l'algorithme Orchestra qui forme la structure et la base du système développé dans ce mémoire est décrits. Dans le chapitre suivant, les limites et les compromis d'Orchestra sont expliqués.

CHAPITRE 3

REVUE DE LITTÉRATURE ET DESCRIPTION DU PROBLÈME

3.1 Revue de littérature

Un aspect non spécifié dans la norme IEEE 802.15.4 est la construction d'un ordonnancement TSCH. Dès lors, cette tâche a attiré une grande attention de la part de la communauté des chercheurs. Les approches d'ordonnancement TSCH peuvent être classées en trois groupes principaux: planification centralisée, distribuée et autonome.

Dans les schémas centralisés, une entité centrale recueille toutes les informations du réseau, puis effectue la planification des cellules pour chaque nœud. Les schémas d'ordonnancement centralisés visent à générer un horaire optimal en utilisant les informations telles que l'état de la liaison et les exigences de trafic des nœuds collectées par l'entité centrale du réseau. TASA (Palattella et al., 2012) est une approche centralisée où le planning est construit par un seul nœud. Cette planification crée des modèles de temps / fréquence basés sur le graphique de topologie du réseau et la charge de trafic à chaque nœud, tout en réduisant la latence et le cycle de service radio en même temps. L'algorithme CLS (Choi & Chung, 2016) construit des programmes multi-sauts efficaces en utilisant un nombre minimal de messages de contrôle centralisés, car il alloue et désalloue des créneaux sans replanifier le programme entier à chaque fois. CLS se concentre sur la réduction de l'écoute inactive, tout en répondant aux besoins en bande passante du réseau. Dans (Soua et al., 2012), les auteurs ont proposé une autre planification centralisée des données brutes convergecast, appelée MODESA (Multichannel Optimized Delay Time Slot Assignment), qui prend en compte la disponibilité de plusieurs canaux pour réduire la durée du cycle TDMA tout en garantissant un accès moyen équitable. Il trie les nœuds en fonction de leurs priorités, la priorité la plus élevée correspondant au plus grand nombre de paquets restants dans la mémoire tampon d'un nœud.
Bien que ces approches présentent une planification théorique qui semble efficace, en fait, elles subissent une surcharge de contrôle importante pour mettre à jour les informations réseau pour le nœud racine et un ajustement de planification lent lorsque la topologie est modifiée. C'est

pourquoi ce type d'approche centralisée est recommandé pour un environnement statique où la topologie de routage change rarement.

Vient ensuite l'approche distribuée qui tente d'allouer des cellules à travers une procédure d'établissement de connexion entre voisins. Les nœuds échangent des informations telles que l'état de la liaison et les exigences de trafic et utilisent ces informations pour déterminer l'ordonnancement. Les schémas distribués sont plus flexibles face à des changements brusques de réseau que les schémas centralisés, qui collectent des informations auprès de l'entité centrale et déterminent l'ordonnancement. DeTAS (Decentralized Traffic Aware Scheduling) (Accettura et al., 2013) , l'un d'entre eux, est une version distribuée de TASA qui cible les réseaux RPL avec plusieurs nœuds récepteurs. Cette planification distribuée est un algorithme prenant en charge le trafic qui construit la planification en fonction du trafic généré par chaque nœud source où le réseau est modélisé sous forme de plusieurs graphiques de routage, chacun enraciné dans un sink différent. Wave, un autre algorithme d'ordonnancement distribué qui planifie les nœuds par vagues successives, est proposé dans (Soua et al., 2016). Dans chaque onde, chaque nœud ayant un paquet à transmettre se voit attribuer une tranche de temps et un canal. Le nœud récepteur envoie un message à ses enfants pour déclencher le calcul de la première vague. Chaque slotframe est divisé avec une unité appelée wave et génère des ondes à plusieurs reprises pour former un programme pour que tous les nœuds du réseau transmettent des paquets générés à intervalles réguliers au sinks. Dans (Aijaz & Raza, 2017), DeAMON, qui est un protocole de planification multi-saut adaptatif décentralisé pour les réseaux sans fil 6TiSCH, est introduit. Cette planification distribuée pour les applications de surveillance et de contrôle industrielles offre des solutions pour le trafic ascendant uniquement et ne prête pas attention au trafic descendant. En allouant des emplacements de sauvegarde de manière dynamique et robuste, DeAMON atteint une fiabilité élevée. DIVA, un algorithme de planification de diffusion divergente distribuée est présenté dans (Demir & Bilgili, 2019). Il s'agit d'un algorithme d'ordonnancement entièrement distribué pour le trafic divergeant où les nœuds quittent le trafic vers tous les nœuds voisins par opposition à convergeant où les nœuds de sortie concentrent le trafic vers le nœud racine. Les connexions aux nœuds voisins sont

établies en fonction d'une opportunité aléatoire indépendante du débit de trafic réseau à chaque nœud.
L'approche distribuée présente des faiblesses telles que la nécessité de négocier entre voisins, ce qui peut entraîner un retard de planification en raison d'une messagerie supplémentaire.

Dans l'ordonnancement autonome, chaque nœud du réseau détermine l'ordonnancement de communication comme l'approche distribué. Cependant, contrairement à l'ordonnancement distribué, chaque nœud peut choisir de manière autonome son propre horaire sans aucune signalisation pour la création ou la modification d'horaire. L'ordonnancement autonome présente l'avantage de réduire la surcharge du réseau pour l'ordonnancement que les approches centralisées et distribuées. Le domaine de la planification autonome pour TSCH a été lancé par la publication d'Orchestra (Duquennoy et al., 2015).
Orchestra est une solution d'ordonnancement récente pour TSCH qui apporte des avantages importants tels que l'utilisation de règles d'ordonnancement simples avec un taux de livraison élevé. Orchestra possède plusieurs atouts qui en font une solution de planification prometteuse pour les réseaux TSCH. La planification d'Orchestra n'est ni centralisée ni distribuée et est calculée de manière autonome, où chaque nœud du réseau maintient localement son propre calendrier en fonction des informations de la couche de routage.
Escalator (Oh et al., 2018), un autre schéma d'ordonnancement autonome basé sur orchestra, génère un planning d'intervalle de temps consécutif le long du chemin de transmission des paquets pour minimiser le délai de transmission des paquets. Cela signifie que dans un intervalle de temps, un nœud reçoit un paquet généré à partir d'un nœud descendant, et dans l'intervalle de temps suivant, le nœud envoie immédiatement le paquet reçu à son parent.
D'autre part, ALICE (Kim et al., 2019), qui se base également sur Orchestra, alloue une cellule unique pour chaque lien de direction du trafic au lieu d'allouer une cellule pour chaque nœud.
e-TSCH-Orch (Rekik et al., 2018) a également essayé de réduire la latence d'Orchestra en se concentrant sur le scénario de collecte (en amont). En fonction de la planification d'Orchestra, lorsqu'un nœud envoie un paquet à un nœud voisin, il indique le nombre de paquets dans sa file d'attente de transmission. Son nœud voisin planifie cette quantité de cellules Rx consécutives et lorsque la slotframe a suffisamment de plages horaires libres, le nœud peut les

exploiter pour vider sa file d'attente. Cependant, cette stratégie simple aggrave les problèmes de conflit et de collision.

BOOST (Jin et al., 2018) est un protocole qui combine TSCH avec un routage opportuniste et une planification autonome. BOOST complète la diversité spatiale et de canal de TSCH avec la diversité des récepteurs du routage opportuniste, ce qui conduit à une fiabilité accrue. Il prend en charge plusieurs priorités de trafic. Techniquement, BOOST regroupe les nœuds du réseau en couches; la n-ième couche est constituée de nœuds à n sauts de la racine. L'ordonnancement des créneaux d'émission / réception pair-impair est utilisé de façon à ce que selon la couche, les nœuds émettent ou reçoivent dans le premier créneau de la slotframe et font l'action inverse dans le deuxième créneau, et ainsi de suite. La fonction de planification minimale (MSF) (Hamza & Kaddoum, 2019) décrit un mécanisme de planification distribué pour TSCH au-dessus du protocole 6top. Dans les cellules autonomes utilisées dans le MSF, chaque nœud qui exécute le MSF alloue certaines cellules de manière autonome. Une cellule Rx est allouée au nœud lui-même. Le reste des cellules sert potentiellement de cellules Tx. Contrairement à Orchestra, les cellules Tx dans MSF sont ajoutées au besoin et supprimées lorsqu'elles ne sont plus utilisées et qu'il n'y a pas de trafic à envoyer à un voisin particulier. Dans le cas où plusieurs cellules Tx sont planifiées dans le même intervalle de temps, un mécanisme est appliqué: la cellule avec le plus de paquets à envoyer est sélectionnée.

L'approche autonome est absolument le type de solutions actuellement proposées par les chercheurs, notamment en raison de ses avantages par rapport aux deux autres approches. C'est ce qui est utilisé dans ce mémoire.

3.2 Problèmes de charge de trafic élevée

3.2.1 Congestion et débordement de file d'attente dans les trafics convergeant

Les réseaux avec un trafic convergeant, tels que ceux utilisés pour les applications de collecte de données, souffrent de graves problèmes de congestion (Muravyov et al., 2015). Ces applications ont souvent un comportement convergeant où toutes les données collectées par tous les capteurs du réseau sont acheminées vers le sink. Avec une charge de trafic élevée, le

nombre de paquets de données voyageant des nœuds feuilles vers le sink devient plus élevé, ce qui entraîne une congestion principalement dans les nœuds situés à proximité du sink. La congestion a des conséquences négatives sur les performances du réseau telles que le retard, le débordement de file d'attente et la perte de paquets. Lorsque le réseau est encombré, il n'est pas simple pour les nœuds d'accéder au support et de transmettre leurs données vers le sink en raison de conflits et de collisions élevés. Afin d'atténuer l'impact d'une telle congestion, une meilleure répartition des cellules allouées aux nœuds est nécessaire en fonction de la position sur le réseau.

Dans les réseaux 802.15.4e en mode TSCH, un nœud n peut transmettre deux types de données: les paquets périodiques notés $A_l^W(t)$ et les paquets d'événements notés $A_l^v(t)$.

Le flux total généré dans un intervalle de temps t par un nœud n noté $A_l^T(t)$ est illustré comme suit (Hamza & Kaddoum, 2019):

$$A_l^T(t) = A_l^W(t) + A_l^v(t) \tag{3.1}$$

La distribution du nombre de paquets de données, $N(t)$, générés par chaque nœud du réseau depuis le début de la slotframe jusqu'à sa fin est calculée comme suit (Venkataramanan & Lin, 2011):

$$P\{N(t) = n\} = \frac{(\lambda t)^n}{n!} e^{\lambda t} \tag{3.2}$$

Supposons que la file d'attente associée au lien l soit notée $Q_l(t)$ et $E_l(t)$, la quantité de données transmises sur un lien l dans un intervalle de temps t. Avec $Q_l(t) \leq E_l(t)$ comme contrainte, l'évolution de la file d'attente de trafic pour un nœud spécifique est:

$$Q_l(t+1) = Q_l(t) + A_l^T(t) - E_l(t) \tag{3.3}$$

La racine ne gère aucune file d'attente car elle est le nœud de destination de tous les flux. L'idée serait donc de s'assurer que les nœuds proches de la racine présentent le moins de congestion possible.

3.2.2 Limitations d'Orchestra

L'ordonnancement TSCH actuel ne prend pas en charge l'allocation dynamique de cellules en fonction du trafic et des besoins du nœud. Malgré ses caractéristiques uniques, Orchestra détermine le calendrier TSCH pour chaque nœud quelle que soit sa charge de trafic, ce qui peut considérablement affecter le délai de communication. Dans la planification Orchestra, chaque nœud réserve des slots par slotframe, indépendamment de la quantité de trafic de données qu'il doit livrer au DODAGroot. C'est un problème car la demande de chaque nœud n'est pas la même et varie en fonction de l'emplacement du nœud dans le réseau. Les nœuds proches de la racine sont constamment en demande et ont un trafic important car tous les nœuds transmettent des paquets vers cette destination. Ces nœuds proches de la racine, en plus d'envoyer eux-mêmes des paquets, transmettent également des paquets des nœuds feuilles au nœud racine. Il en résulte une accumulation de paquets dans les files d'attente des nœuds proches de la racine du fait de leur forte charge, ce qui peut conduire à une perte de paquets induite par un débordement de la mémoire tampon (Venkataramanan & Lin, 2011). En revanche, le les nœuds situés à l'extrémité du réseau ne présentent pas beaucoup de trafic. C'est pourquoi une meilleure répartition des cellules est nécessaire. Cette limitation rend Orchestra peu pratique pour de nombreuses applications sensibles aux délais. La limitation est que lorsque le trafic n'est pas réparti uniformément sur le réseau, Orchestra n'adapte pas son mécanisme d'allocation de cellules. Cette limitation est due au fait qu'orchestra alloue le même nombre de créneaux horaires à tous les nœuds, ce qui entraîne des retards importants, en particulier au niveau du nœud central où le trafic est le plus dense en raison de la mise en file d'attente des paquets dans les tampons des nœuds encombrés (Rekik et al., 2018).

En effet, les nœuds proches de la racine ont une charge de trafic plus élevée que les nœuds situés à l'extrémité du réseau. En conséquence, chacun de ces nœuds proches de la racine

nécessite, par rapport aux nœuds feuilles, un plus grand nombre d'emplacements disponibles pour pouvoir évacuer cette congestion du trafic.

Il faut donc essayer de minimiser la latence à ce niveau et leur donner plus de ressources pour une meilleure évacuation du trafic afin d'améliorer les performances du convergecast. En effet, plusieurs applications critiques telles que les applications de surveillance et d'alerte peuvent souffrir de ces retards.

CHAPITRE 4

OSCAR : ALGORITHME OPTIMISÉ D'ALLOCATION DE CELLULES

4.1 Conception

Pour surmonter les limitations ci-dessus, nous proposons d'optimiser le protocole TSCH à l'aide d'Orchestra et de concevoir un ordonnancement MAC plus efficace en termes de consommation énergétique et de latence. Dans cette section, la conception détaillée et la mise en œuvre de l'algorithme d'allocation de cellule d'ordonnancement optimisé OSCAR est présenté.

Pour concevoir et valider un protocole de communication, vous devez tester ses capacités dans différents scénarios. Idéalement, un protocole devrait être testé dans autant de scénarios et d'environnements que possible. Il existe trois façons de procéder: la simulation, le banc d'essai et le monde réel. Malheureusement, chaque solution a ses inconvénients. La simulation est rapide, flexible et ne nécessite pas de matériel, mais reste très souvent moins précis. Les bancs d'essai, quant à eux, bien qu'ils soient statiques, ont l'avantage de mieux refléter la réalité, notamment en termes de propagation radio et d'interférences. Enfin, le déploiement proprement dit présente les mêmes avantages et inconvénients que les bancs de test avec un avantage supplémentaire, celui d'être lié à une application spécifique. Cette spécificité peut être intéressante si l'étude apporte des solutions pour un domaine d'application spécifique.

Dans ce mémoire, l'analyse des performances de l'algorithme OSCAR se fera dans un premier temps à l'aide d'une simulation puis une validation du comportement par mesures expérimentales sera effectuée sur un banc d'essai avec des nœuds réels.

Deux contributions principales, à savoir la conception d'une architecture de classification des nœuds en fonction de leur position permettant une meilleure répartition des cellules allouées et un système pour réduire les cellules allouées aux nœuds inactifs, sont détaillées dans cette partie.

4.1.1 Algorithme 1 : Répartition optimisée de l'allocation de cellules

Le but de cet algorithme d'accorder des slots aux nœuds qui en ont le plus besoin (selon leur besoin). Nous savons que les nœuds proches de la racine sont ceux qui présentent le plus de trafic. Dans cette vision, cet algorithme s'inscrit dans la planification de ces nœuds et propose de gérer le nombre de timeslots alloués à chaque nœud en utilisant la valeur du rang. Cette technique d'allocation de timeslot, basée sur le rang, proposée ici est un mécanisme dans lequel le nœud se voit allouer un nombre d'opportunités de transmission en fonction de la valeur de son rang.

Comme mentionné précédemment, RPL utilise une stratégie en introduisant le concept de rang pour définir la position individuelle d'un nœud par rapport à tous les autres voisins et par rapport à la racine. Chaque nœud calcule la valeur RANK en utilisant la fonction objective (OF). Le OF est une équation comprenant des métriques pour atteindre l'objectif. Un avantage du RPL est que chaque appareil peut déterminer de manière flexible le chemin de liaison montante avec la valeur RANK grâce à une simple opération OF. Le rang RPL est utilisé comme base de l'algorithme pour déterminer la distance par rapport à la racine.

OSCAR se caractérise par la classification des nœuds en 6 classes selon leur valeur de rang sachant que le rang de la racine est égal à 0. Chaque nœud se voit attribuer une classe ID, représentant la couche qu'il occupe. L'ID de la classe, qui va de 0 à 5, est défini en fonction de la distance géographique par rapport à la racine. En effet, plus l'ID de la classe augmente, plus la distance augmente.
Pour expliquer la démarche d'allocation de slots, nous discutons d'abord de la stratégie de partitionnement. Il s'agit d'un mécanisme de distribution des intervalles de temps sur les six prochaines slotframes. Si le nœud appartient à la classe 0 il aura 6 slots sur les six prochaines slotframes. La classe 1 aura cinq slots sur les six prochaines slotframes et ainsi de suite jusqu'à la dernière classe qui ne disposera que d'un seul slot sur les six prochaines slotframes.
Quand le rang change, OSCAR alloue ou enlève un slot selon la nouvelle valeur de ce rang. Donc en éloignant le nœud de la racine, moins de slots seront alloués. En approchant on alloue

plus. Ainsi, tout nœud qui change de position dans la topologie du réseau voit son numéro de classe changé et par conséquent le nombre de slots qui lui est attribué. L'algorithme d'ordonnancement se produit tous les cinq paquets, c'est-à-dire qu'il y a un rafraîchissement de l'état actuel du réseau et de l'allocation des cellules tous les cinq paquets.

La figure suivante montre la classification des nœuds selon leur distance par rapport à la racine.

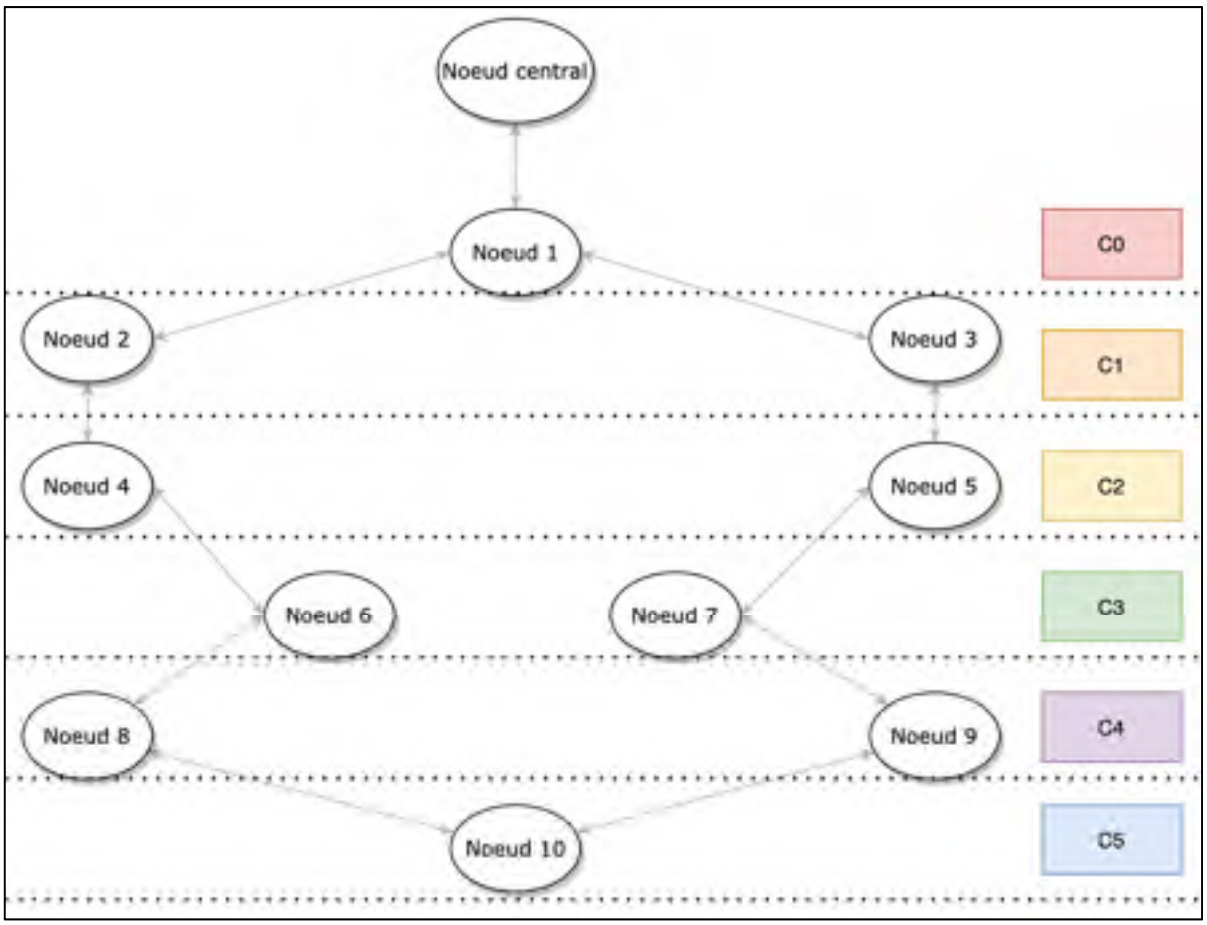

Figure 4.1 Un exemple de classification de la topologie du réseau

Pendant chaque intervalle de temps, les nœuds se réveillent tous pour transmettre ou recevoir des paquets en fonction de leur classe ID. En outre, différents canaux sont utilisés par deux paires de couches adjacentes afin d'atténuer les interférences. L'allocation du nombre de slots

se fait comme décrite dans la figure 4.2 qui illustre le réseau de la figure 4.1 avec une slotframe constitué de six slots.

Par exemple, dans le fonctionnement normal d'orchestra, le noeud X aura les slots numéros 1, 7, 13, 19, 25, 31, 37.

Avec la nouvelle implémentation le nœud X s'il est situé en classe 0 garde ses slots 1, 7, 13, 19, 25, 31, 37.

Cependant s'il est situé en classe 1, il n'aura que les slots 1, 7, 13, 19, 25,…,37. On délaisse le slot 31 parce que c'est un nœud de classe 1 donc il n'appartient pas au nœud les plus proche de la racine. Si le noeud X était classe 2 il se verrait retirer deux slots sur les six, soit les slots 25 et 31.

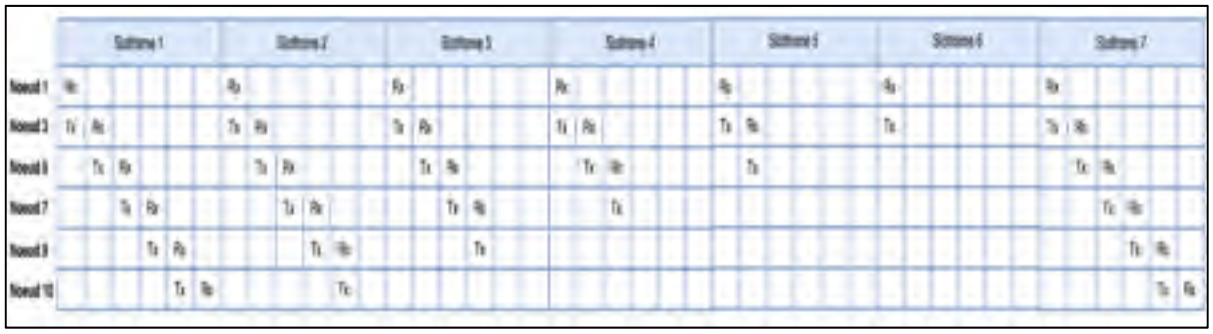

Figure 4.2 Un exemple d'allocation des timeslots de la solution proposée

Pour les noeuds loin de la racine où le rang est important, l'énergie sera moins consommée car moins de slots seront attribués. Pour les plus proches il y aura plus de consommation mais c'est minime. L'autre avantage de ce mécanisme est qu'il maintient la décongestion du réseau car les nœuds présentant le plus de trafic ne sont pas affectés.

```
Algorithm 1: Reschedule cell based on class
Classes definition and initialization of the current
 class;
Input : New Class
if New_Class = Current_Class then
end
if New_Class < Current_Class then
  | Allocate_more_slots();
else
  | Reduce_allocated_slots();
end
```

Algorithme 4.1 Algorithme de répartition optimisé des cellules allouées aux nœuds

4.1.2 Algorithme 2 : Mécanismes de réduction de slots alloués aux nœuds inactifs

Chaque nœud du système fonctionne selon un cycle de service. Chaque intervalle se compose de plages horaires de sommeil et de réveil. Dans les créneaux horaires de sommeil, le nœud éteint sa radio et dans les créneaux horaires de réveil, le nœud allume sa radio pour transmettre ou recevoir des paquets. Avec Orchestra, des créneaux horaires réguliers de réception et de transmission sont clairement définis et inclus dans la slotframe unicast. Tous les nœuds du réseau présentent ainsi un cycle de service de 100% dans les créneaux horaires de réveil qui leur sont attribués pour transmettre ou recevoir. Ceci n'est pas particulièrement efficace en termes d'efficacité énergétique. L'idée ici est qu'en l'absence de données à recevoir ou transmettre, un nœud n'allume pas sa radio.

L'algorithme 2 propose une méthode pour ajuster le cycle de service radio de chaque nœud et améliorer l'efficacité énergétique. Celui-ci instaure un mécanisme complémentaire permettant que même lorsqu'on a un bon rang et si le nœud est inactif, de diminuer les slots alloués en déclassant le nœud petit à petit (rang par rang). Par exemple, si le nœud X qui se trouve en classe 1 ne présente pas d'activité durant un laps temps, il est basculé en classe 2 puis en classe

3 si l'échange de paquets ne reprend pas et ainsi de suite jusqu'à la classe 5. Par conséquent, il bénéficie de moins de slots alloués progressivement. Par la suite, une fois que l'activité reprend sur ce nœud, c'est-à-dire que l'on détecte qu'un paquet a été reçu ou envoyé, le nombre de slot initial calculé sur la base du rang est rétabli.

Ce processus se fait à l'aide d'un timer qui se déclenche toutes les 10 secondes et qui supervise l'activité de chaque nœud à savoir les paquets reçus et envoyés. Au fond, lorsque le minuteur d'un nœud expire et si on n'a pas d'activités, on alloue moins de slots progressivement
L'avantage principal de ce mécanisme est qu'il réduit considérablement la consommation énergétique du réseau. L'objectif étant de minimiser la consommation d'énergie de l'ensemble du réseau avec la contrainte que le délai de bout en bout attendu de chaque nœud soit également le plus faible possible.

Algorithm 2: Duty cycle adjustment

```
initialization;
while (1) do
    Wait_10_seconds;
    if no_packet_selected then
        New_Class = Current_Class + 1;
        Reschedule(New_Class);
    end
end
```

Algorithme 4.2 de réduction de slots alloués aux nœuds inactifs

Le diagramme de flux présenté à la figure 4.3 illustre le processus de l'algorithme 2 afin de répondre aux exigences d'énergie du réseau.

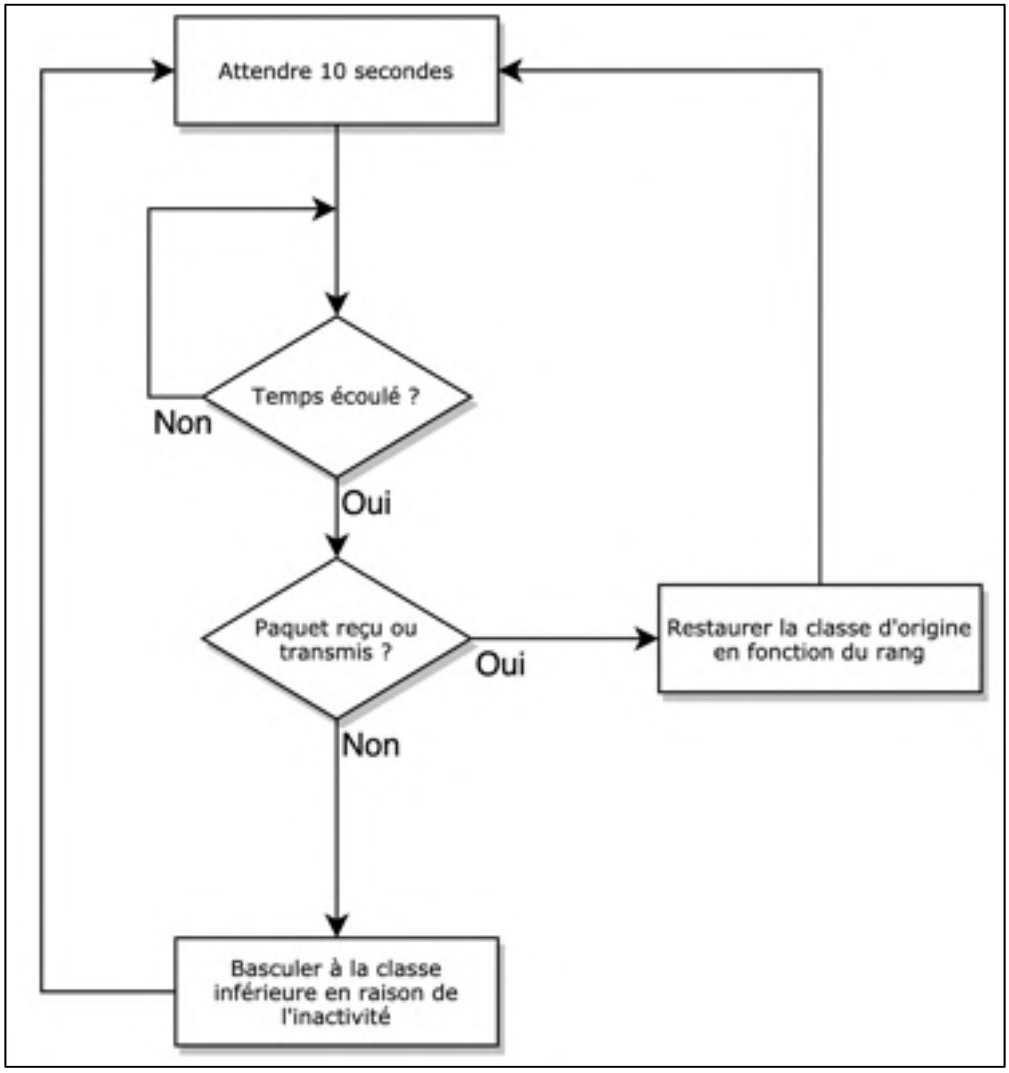

Figure 4.3 Graphique de flux de conception de l'algorithme 2

4.2 Exemple d'ordonnancement

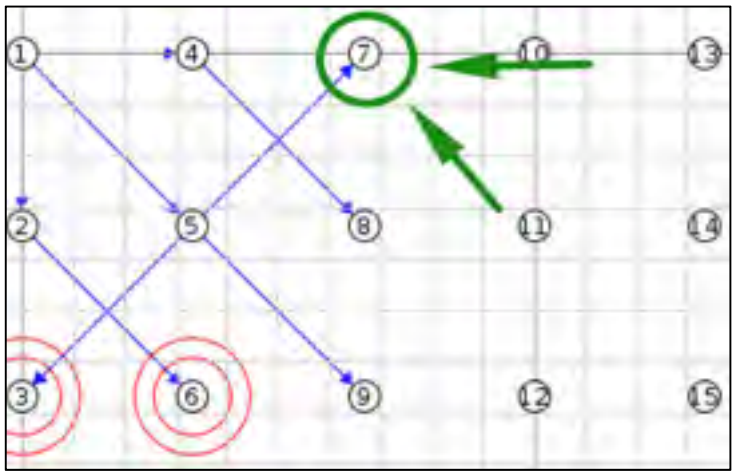

Figure 4.4 Topologie du réseau de simulation sous Cooja

Pour illustrer le système introduit par les deux algorithmes vus précédemment, un réseau composé de 15 nœuds a été mis en place sur l'outil de simulation cooja proposé par contiki-ng. Une fréquence de transmission de 1 paquet / min avec une slotframe unicast d'une taille de 5 slots est générée. Dans cette démarche, un focus sur le nœud 7 a été opéré comme on peut le voir sur la figure 4.4. Les logs du nœud 7 sont présentés sur la figure 4.5.

Encadré en rouge, on peut apercevoir que le mécanisme de l'algorithme 1 détermine les slots alloués au nœud 7 en fonction de sa position. Cette étape se fait après que le nœud a déterminé son parent préféré et sa classe en fonction du rang,

En vert, l'ajustement du cycle de service de l'algorithme 2 est mis à l'œuvre. Ainsi, au bout de 10 secondes d'inactivité, le nœud se voit automatiquement déclassé. C'est-à-dire qu'un slot est enlevé à chaque fois pour pouvoir éviter au nœud d'allumer sa radio et donc économiser l'énergie consommée.

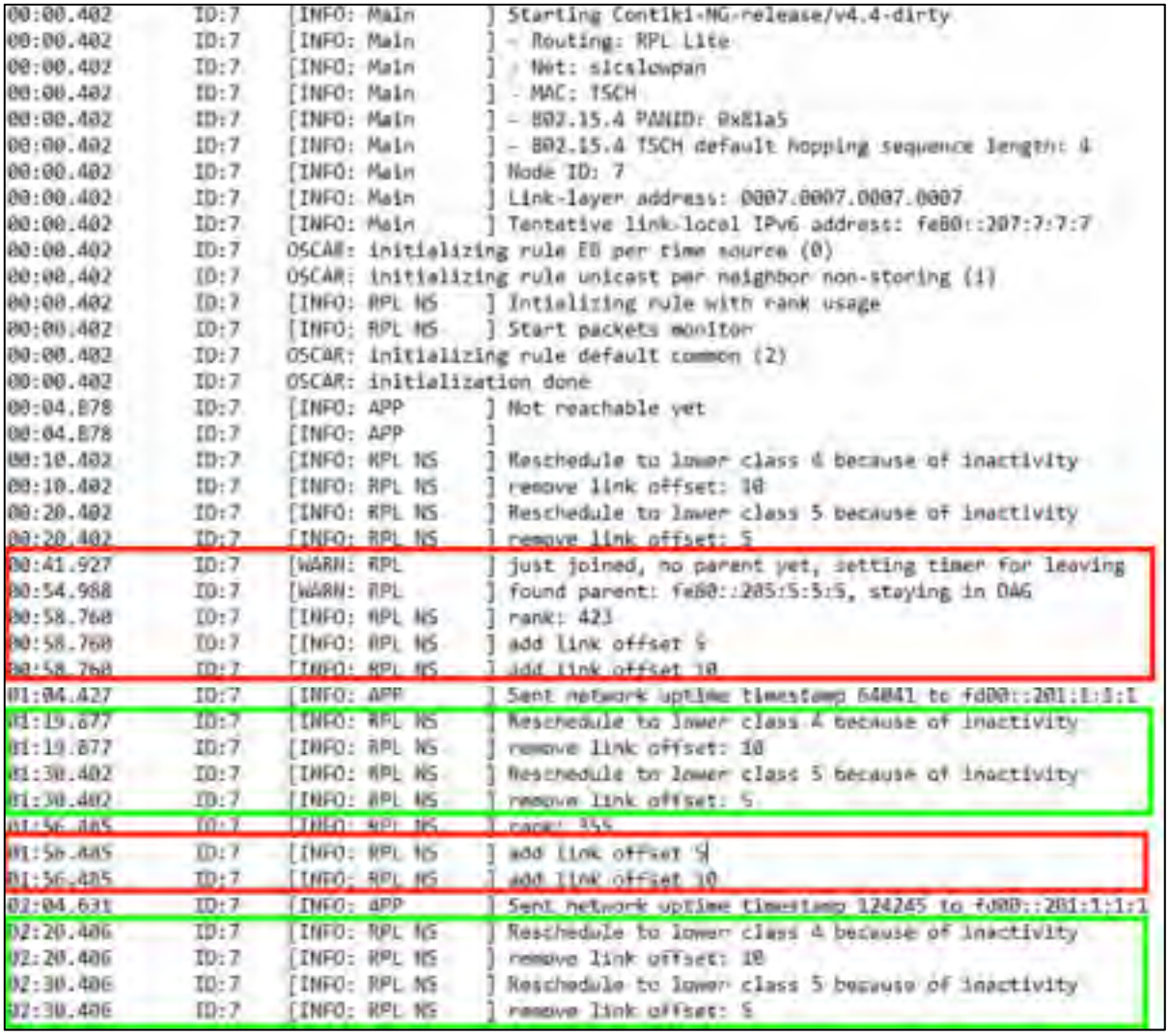

Figure 4.5 Logs du nœud 7

4.3 Simulations

4.3.1 Métriques de performance

Nous comparons les performances d'OSCAR avec Orchestra à l'aide de métriques que nous décrirons dans cette section. Les trois métriques observées dans cette étude sont :

- latence : la latence des communications sans fil est un indicateur de performance clé et doit être prise en compte. Il est important de déterminer le temps nécessaire pour véhiculer un paquet au travers d'un réseau. Le délai point-à-point d'un paquet représente le temps total que met le paquet pour aller de la source vers la destination. Après avoir mesuré le délai de chaque paquet, nous faisons la moyenne pour en déduire le délai moyen point-à-point. La latence dépend de la façon dont la planification est construite. Sachant que la planification est généralement effectuée en fonction de la quantité de données à transporter et de la qualité du lien;
- taux de livraison des paquets : la communication sans fil dans les réseaux IoT à faible puissance et avec perte est très difficile, où le lien entre les périphériques sans fil contraints est généralement instable avec des taux de livraison de paquets (PDR) relativement bas. Les réseaux industriels nécessitent une fiabilité élevée. C'est pourquoi un accusé de réception et plusieurs retransmissions sont parfois nécessaires pour offrir une fiabilité de communication élevée avec des exigences PDR élevées. Si l'expéditeur ne reçoit pas d'accusé de réception dans un délai, il suppose que le récepteur n'a pas reçu le paquet et transmet à nouveau le paquet, ce qui augmente la latence. C'est le rapport entre le nombre de paquets livrés et le nombre de paquets envoyés par la source. Le résultat est exprimé en pourcentage;
- cycle de service : une exigence essentielle des systèmes d'exploitation est de fournir des options d'économie d'énergie aux nombreux petits appareils fonctionnent sur des batteries. L'une des options offertes est le mécanisme de cycle de service où les nœuds alternent entre périodes actives et sommeil en fonction de l'activité du réseau. Un cycle de service est défini comme étant la fraction de temps où les noeuds sont actifs. Le moyen le plus efficace pour conserver l'énergie est de mettre la radio de l'émetteur en

mode veille à chaque fois que la communication n'est pas nécessaire. Le modèle duty cycle est largement utilisé pour analyser les appareils IoT et est calculé en pourcentage comme suit.

$$Cycle\ de\ service = \frac{T1}{T2} \times 100\% \quad (4.1)$$

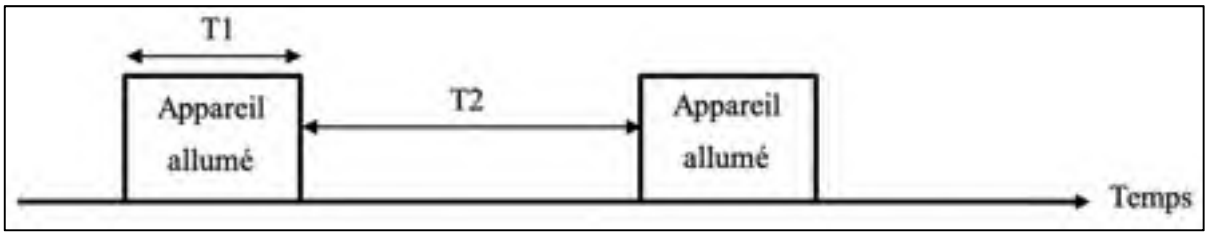

Figure 4.6 Cycle de service radio d'un nœud

Tous les tests que nous avons effectués comparent les performances entre notre architecture et Orchestra à la fois en termes de fiabilité, de consommation d'énergie et de latence. L'objectif est de tester brièvement le comportement de notre architecture et sa mise en œuvre dans un réseau de capteurs sans fil.

4.3.2 Environnement de simulation

Dans ce travail, nous utilisons Contiki-NG, un système d'exploitation open source pour l'Internet des objets qui prend en charge de minuscules microcontrôleurs à faible coût et à faible consommation connectés à Internet. Le simulateur cooja disponible dans Contiki-NG est utilisé pour simuler le scénario de réseau et exécuter la méthode proposée sur les nœuds. La plateforme Zolertia Z1 est utilisé comme nœud pour la simulation de réseau. Cooja utilise le module logiciel Energest pour estimer la consommation énérgetique des nœuds. Avec ce module, le temps passé pour chaque nœud de capteur dans différents états est mesuré. La latence dans cooja peut être calculée comme une différence entre le moment à laquelle le paquet a été envoyé depuis la source et le moment à laquelle il a été reçu à la destination. Enfin la valeur PDR peut être obtenue en divisant le nombre total de paquets reçus et le nombre total de paquets envoyés. Cette valeur prédit la fiabilité du réseau. Plus la valeur de PDR est élevée,

plus la fiabilité du réseau est élevée. La racine DODAG est située aux coordonnées (0,0). Enfin, le débit de données est égal à 1 paquets par minute.

4.3.3 Impact de la position des nœuds

Nous mesurons la performance d'OSCAR en la comparant à Orchestra. La taille de la slotframe EB et de diffusion reste inchangée avec respectivement 397 et 31 slots. La slotframe unicast, quant à elle, a une taille de 6 slots pour OSCAR et 16 slots pour Orchestra. Pour OSCAR, cela signifie que si l'on prend une portion composée de 36 slots de taille 6 pour la slotframe unicast, la classe 0 bénéficiera de 6 slots sur les 36 slots, soit les 6 slots suivants, donc 1 slot tous les 6 slots. Les classes 1, 2, 3 et 4 obtiendront respectivement en moyenne 1 emplacement tous les 7, 9, 12 et 18 emplacements. Enfin, la dernière classe 5 aura 1 emplacement tous les 36 emplacements.
Comme le comportement d'un nœud avec OSCAR varie en fonction de sa classe, le but de cette simulation sera de faire varier la disposition du réseau. Pour cela, un réseau composé de 60 nœuds est déployé sous cooja. Deux topologies sont considérées :

Tableau 4.1 Nombre de nœuds par classe

	Première topologie	Deuxième topologie
Nombre de nœuds en classe 0	5	14
Nombre de nœuds en classe 1	11	15
Nombre de nœuds en classe 2	11	15
Nombre de nœuds en classe 3	11	15
Nombre de nœuds en classe 4	11	0
Nombre de nœuds en classe 5	10	0

Dans les deux cas, les nœuds sont répartis sur différentes couches en fonction de leur proximité par rapport à la racine. Ainsi, la deuxième forme de réseau est plus condensée et proche de la racine tandis que l'autre est un peu plus espacé et distant de la racine.

4.3.3.1 Latence

La figure 4.7 ci-dessous compare les performances sur le délai moyen de bout en bout du réseau entre OSCAR et Orchestra en fonction de la position des nœuds. Ce que l'on aperçoit, c'est que pour la première topologie, de meilleurs résultats sont présents pour OSCAR. Clairement, une latence plus courte de 21% est remarquée. Ce n'est pas surprenant lorsqu'on le sait que le nombre de slots alloués aux nœuds proche de la racine, pour évacuer le trafic, est supérieur avec OSCAR.

Pour ce qui est de la deuxième disposition, OSCAR accentue l'écart avec Orchestra sur la latence moyenne et cela est principalement dû au fait que la disposition des nœuds est plus proche de la racine. Étant donné que les nœuds sont exclusivement répartis entre les classes 0 et 3, ces nœuds ont par conséquent plus d'opportunités de transmission comparées à la première topologie.

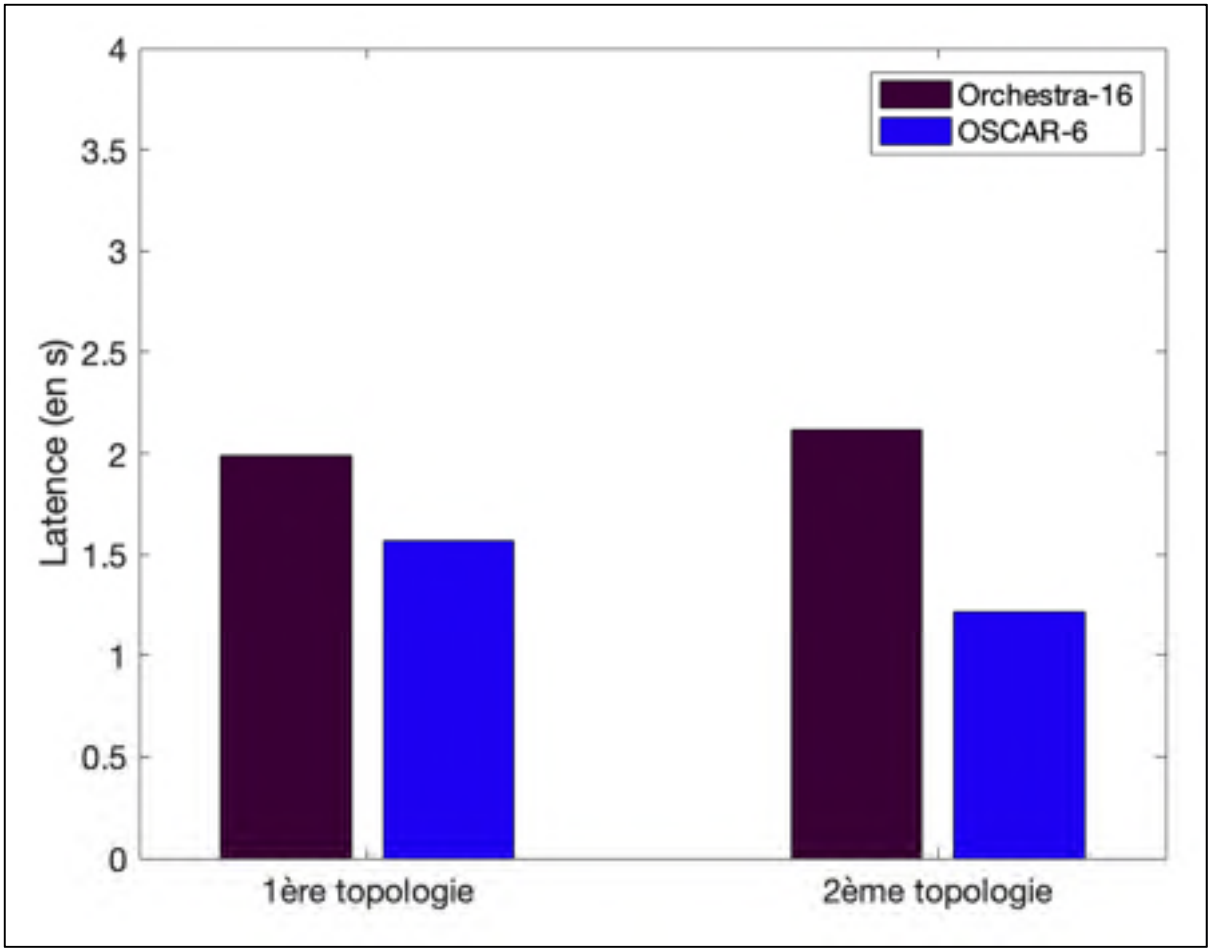

Figure 4.7 Délai moyen entre OSCAR et Orchestra en fonction de la position des nœuds

4.3.3.2 Cycle de service

La figure 4.8 montre les performances de cycle de service entre OSCAR et Orchestra sur les deux topologies décrites précédemment. Comme on a vu précédemment, le cycle de service indique la période pendant laquelle le nœud est actif et sa radio allumée. On constate pour la première topologie que le cycle de service moyen d'OSCAR est inférieur à celui d'Orchestra. En effet, des gains d'énergie peuvent être observés dans les nœuds moins actifs, car l'ajustement du cycle de service de l'algorithme 2 permet de sauver de l'énergie avec une

réduction de slots alloués aux nœuds inactifs, équilibrant la consommation d'énergie accrue des nœuds les plus encombrés.

Cependant, les résultats pour la deuxième disposition montrent une augmentation du cycle de service plus importante pour OSCAR. Cela peut s’expliquer par le fait que le réseau présente une disposition plus rapprochée et donc les nœuds ont plus souvent leur radio allumée que pour la première topologie.

Avec OSCAR, les nœuds bénéficient d’un mécanisme dynamique d’allocation de slots qui joue sur la position. La classe où se situe le nœud est un paramètre central, ce qui explique cette variation. Orchestra, en revanche, a un cycle d'utilisation radio relativement similaire pour les deux dispositions, car le système ne module pas l'allocation des slots et alloue le même nombre de créneaux horaires à tous les nœuds.

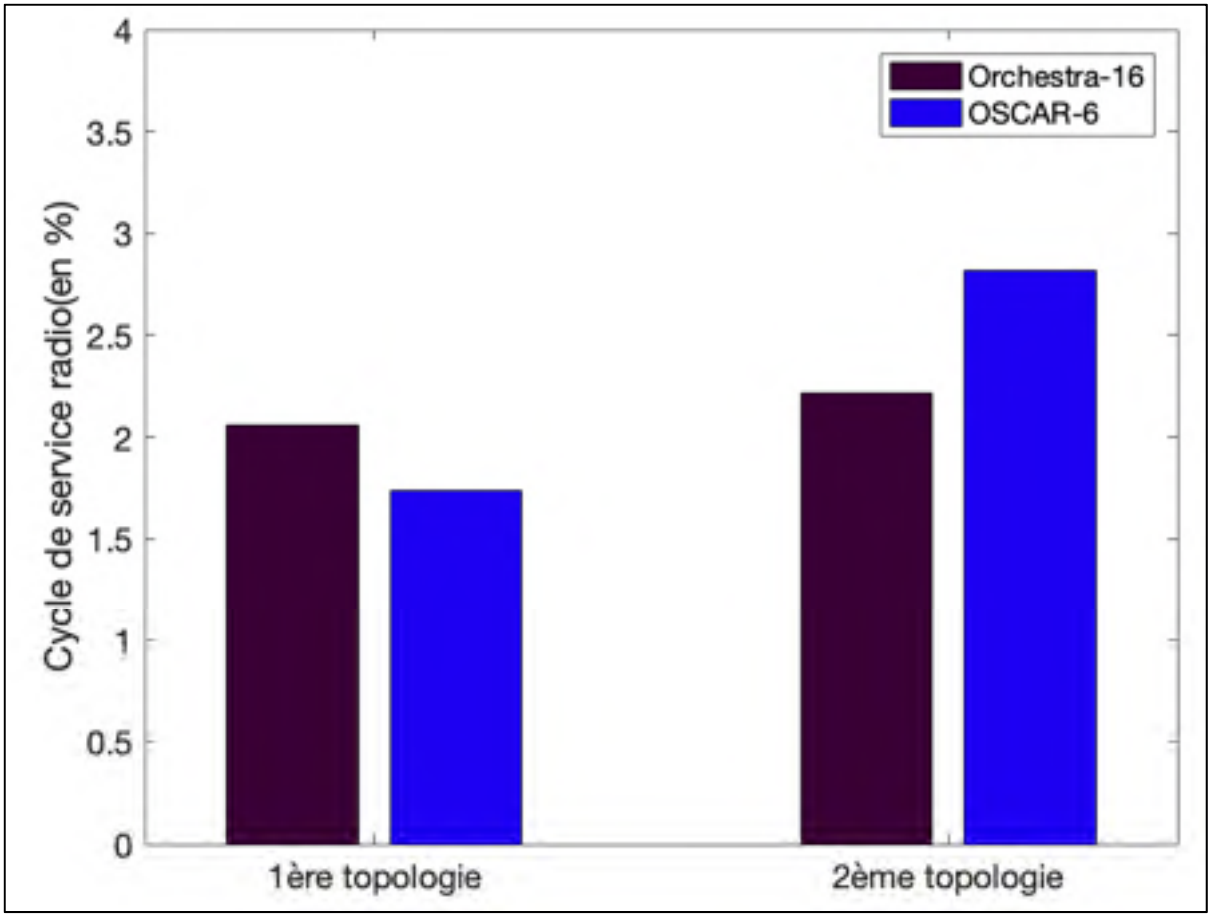

Figure 4.8 Cycle de service moyen entre OSCAR et Orchestra en fonction de la position des nœuds

4.3.3.3 Taux de livraison des paquets

La figure 4.9 représente le taux moyen de livraison de paquets d'Orchestra et d'OSCAR en fonction du nombre de nœuds. Durant cette simulation, chaque nœud a envoyé 2000 paquets jusqu'à la racine, soit un total de 118000 envoyés par l'ensemble du réseau au nœud central. La première topologie indique des résultats pour OSCAR et Orchestra de 91,8% et 95,3% respectivement. Concrètement, cela veut dire qu'environ 11700 paquets ont été perdus sous Orchestra contre 7500 environ pour OSCAR. Le système OSCAR présente donc de meilleurs résultats et la cause est que les nœuds ont plus de slots pour évacuer le trafic.

Pour la deuxième topologie, les chiffres se dégradent pour Orchestra qui passe à 86,6% de taux moyen de livraison de paquets. Orchestra se révèle être sensible à l'encombrement des nœuds

de la deuxième topologie et de la congestion occasionnée. Cette dégradation montre que l'accumulation de paquets dans la file d'attente des nœuds congestionnés entraine un débordement et donc un abandon de ces paquets. En revanche, OSCAR affiche 94,7% et préserve une certaine stabilité notamment grâce à la décongestion du réseau maintenue par les nœuds situés exclusivement entre les classes 0 et 3 de la deuxième topologie.

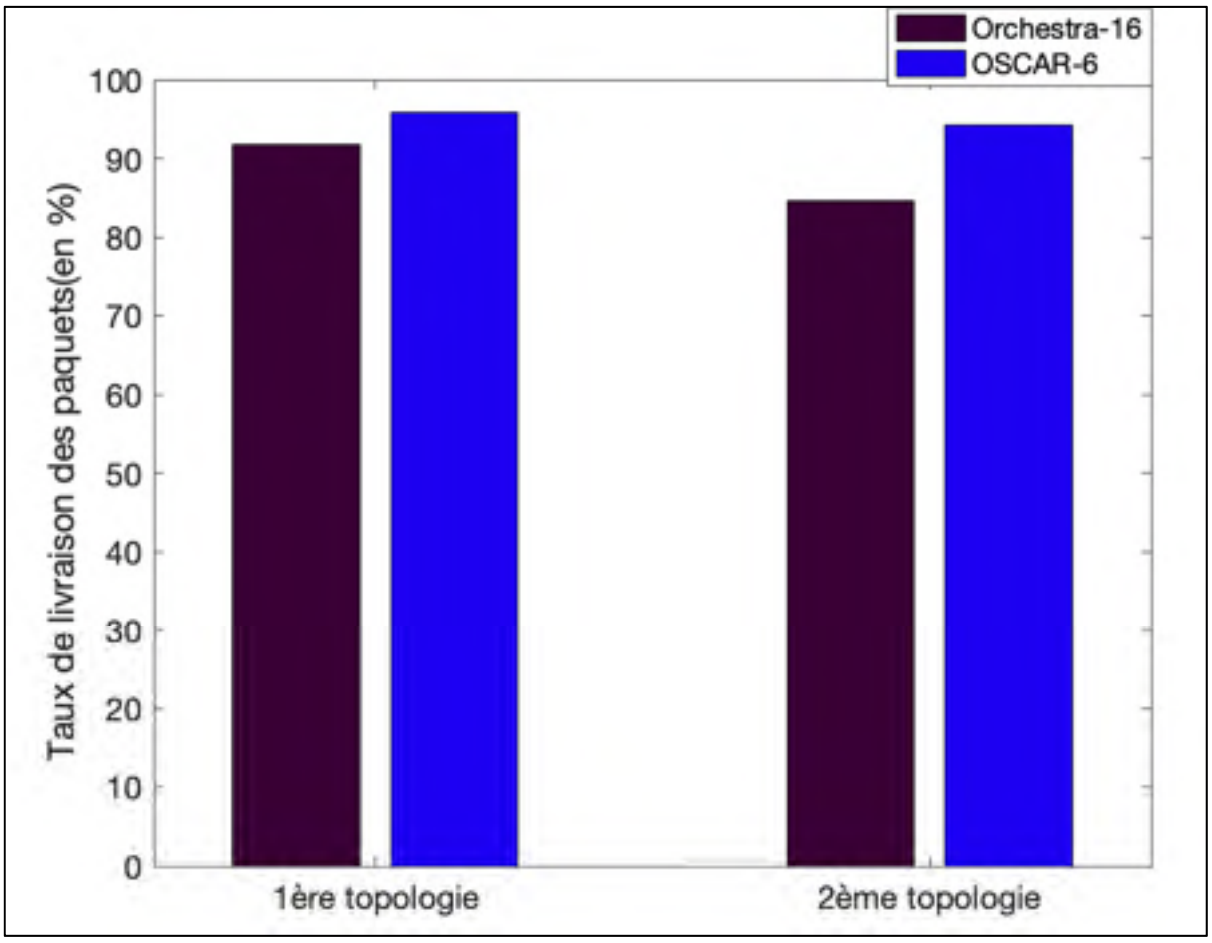

Figure 4.9 Taux moyen de livraison des paquets en fonction de la position des nœuds

4.3.4 Impact du nombre de nœuds

Pour montrer la performance d'OSCAR par rapport à Orchestra à la charge de trafic, nous étudions la performance en faisant varier cette fois-ci le nombre de nœuds dans le réseau. En effet, comme il s'agit d'une simulation, on peut se permettre de faire varier ce paramètre sans surcoût pour voir comment le système OSCAR réagit face à un réseau à grande échelle. À cette fin, nous avons considéré un réseau où les nœuds sont déployés dans une topologie de grille uniforme. Nous avons fait varier le nombre de nœuds de 40 à 100.

4.3.4.1 Latence

La figure 4.10 ci-dessous compare les performances sur le délai moyen de bout en bout du réseau entre OSCAR et Orchestra en fonction du nombre de nœuds. De cette figure, on peut clairement retenir qu'OSCAR surpasse Orchestra sur la latence moyenne du réseau. Ainsi, plus le nombre de nœuds dans le réseau augmente, plus la différence se fait ressentir.
OSCAR améliore le délai de communication et cette observation peut se justifier par le fait que les nœuds les plus congestionnés ont plus de slots pour évacuer le trafic. En effet la particularité d'OSCAR est que le système accorde plus d'opportunités de transmission aux nœuds qui en ont le plus besoin, à savoir les nœuds proches de la racine dans le cas de trafic convergeant.

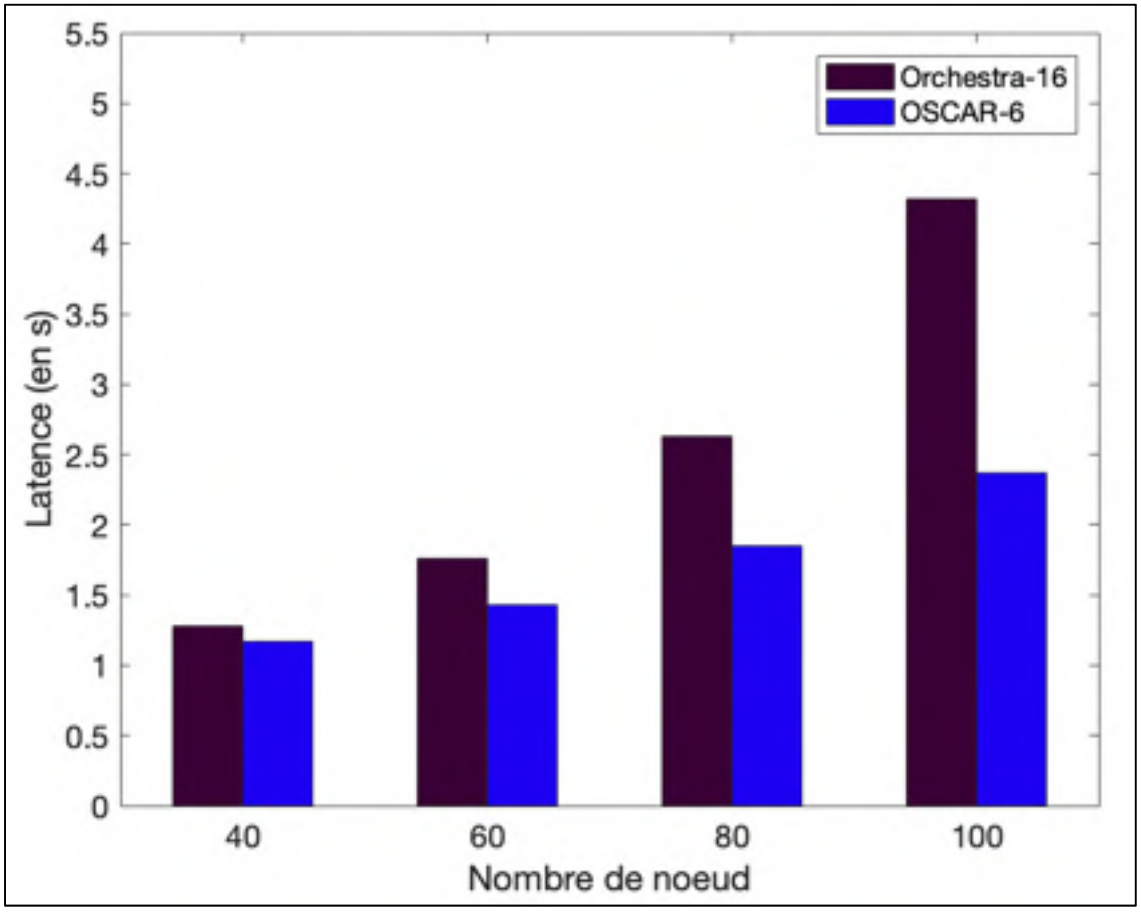

Figure 4.10 Délai moyen entre OSCAR et Orchestra en fonction du nombre de nœuds

4.3.4.2 Cycle de service

La figure 4.11 montre les performances de cycle de service entre OSCAR et Orchestra. Ce qui est à remarquer est qu'entre 40 et 60 nœuds, le cycle de service moyen du réseau est en faveur d'OSCAR. Cela peut s'expliquer par le fait que certains nœuds sont moins sollicités que d'autres et bénéficient du mécanisme de réduction de slots alloués aux nœuds inactifs introduit par OSCAR. En revanche, les bonnes performances d'OSCAR en termes de délais entraînent une consommation d'énergie plus élevée lorsque le nombre nœuds dépassent les 60 nœuds et que la charge de trafic devient importante. La balance énergie/latence est donc bien réelle car les gains observés précédemment se font au prix d'une plus grande consommation énergétique.

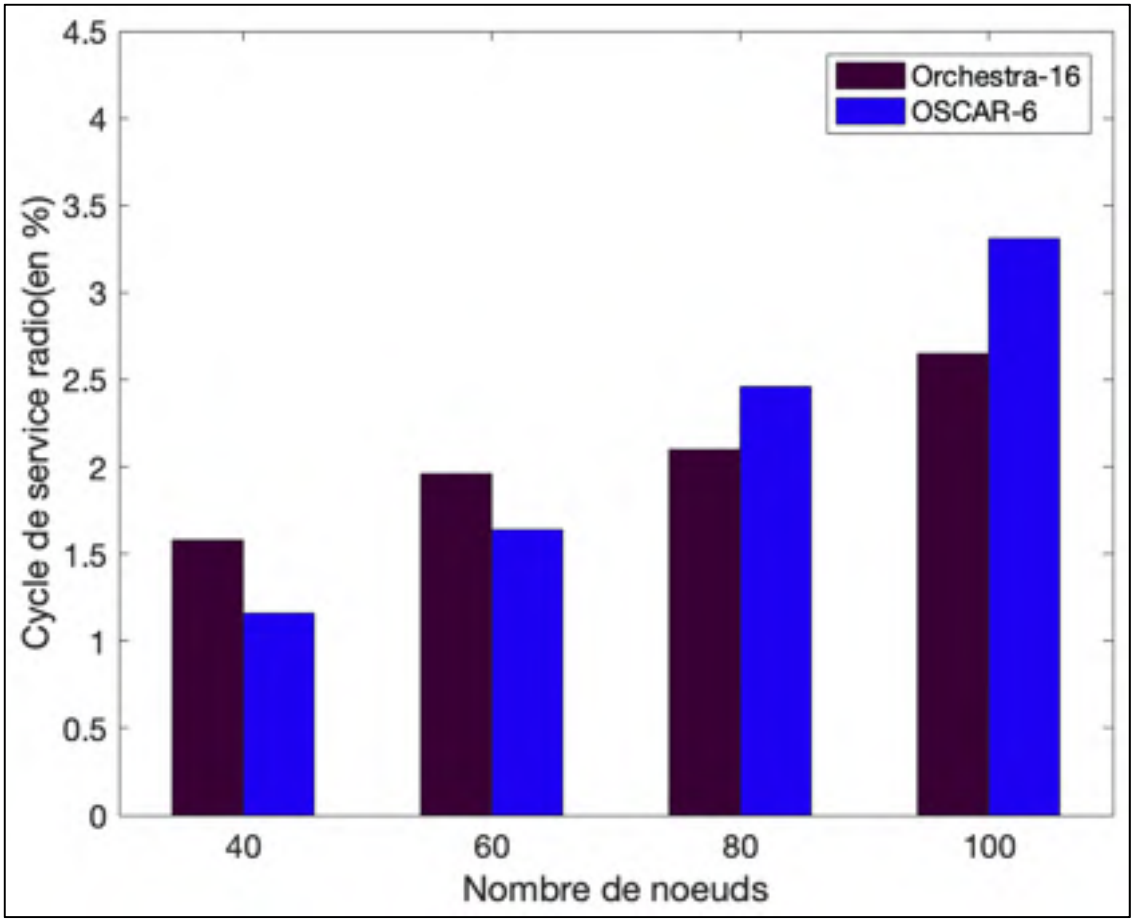

Figure 4.11 Cycle de service moyen entre OSCAR et Orchestra en fonction du nombre de nœuds

4.3.4.3 Taux de livraison des paquets

Enfin, la figure 4.12 représente le taux moyen de livraison de paquets d'Orchestra et d'OSCAR en fonction du nombre de nœuds. Comme pour la simulation précédente, 2000 paquets ont été envoyés par chaque nœud jusqu'à la racine Pour un réseau composé de moins de 60 nœuds, les deux systèmes ont quasiment les mêmes performances, ce qui démontre qu'OSCAR préserve les bonnes performances d'Orchestra en termes de livraison. Seulement, lorsque le nombre de nœuds est supérieur à 60, l'écart s'accentue et de de meilleurs résultats sont observés pour OSCAR par rapport à Orchestra. Cette observation peut être expliquée par le fait que lorsque le nombre de nœuds augmente, la charge de trafic augmente également, entraînant des pertes de paquets au niveau des nœuds congestionnés. Cependant, comme vu précédemment, OSCAR

minimise cette congestion en allouant plus de slots à ces nœuds, ce qui se traduit par un meilleur taux de livraison.

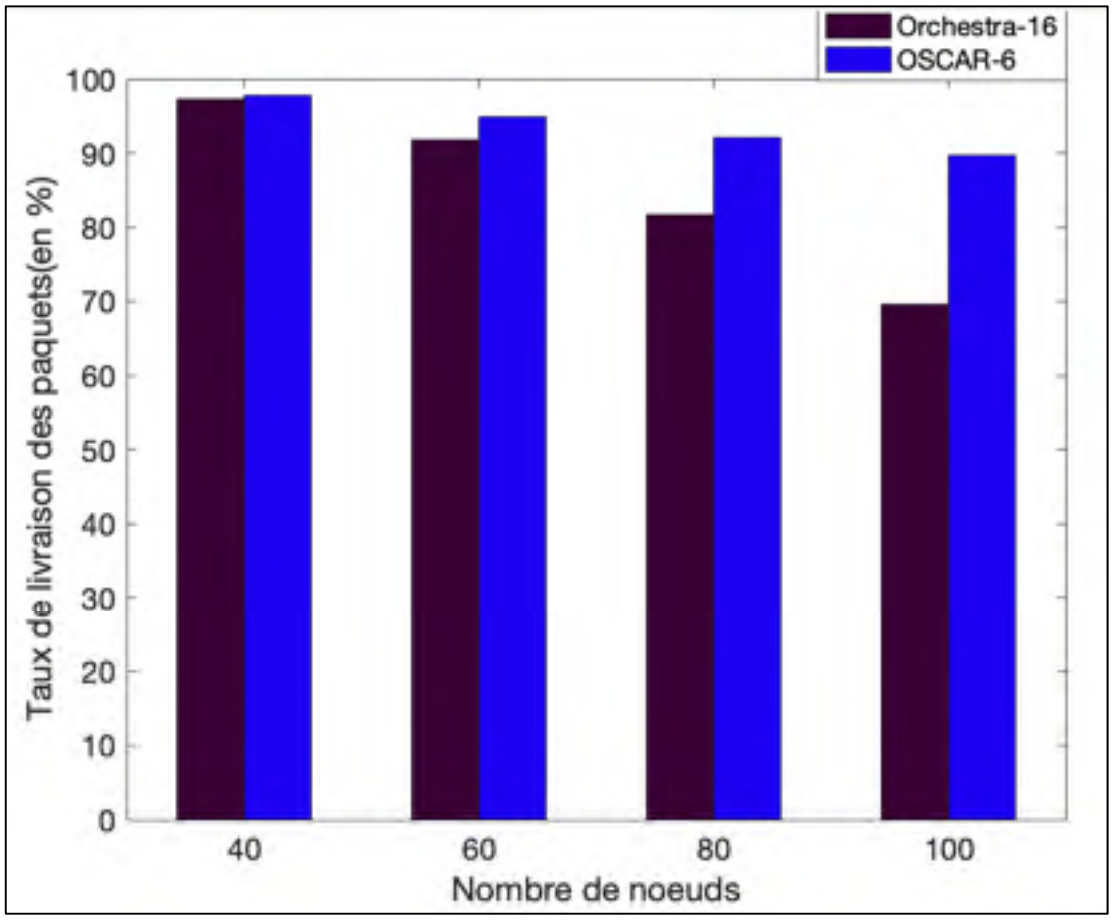

Figure 4.12 Taux moyen de livraison des paquets en fonction du nombre de nœuds

4.4 Étude expérimentale

Dans cette partie, nous présentons les performances d'OSCAR de façon expérimentale avec des nœuds réels. Comme indiqué précédemment, la demande de trafic pour LLN a été beaucoup plus accrue ces jours-ci, comme les applications IoT exigeantes en données et les déploiements réseau à grande échelle (Ang et al., 2018 ; Boubiche et al., 2018 ; Cohen et al., 2020 ; Liu et al., 2017).

Dans ce contexte et dans le but d'évaluer la solution proposée, nous avons développé des scénarios d'expérimentation dans différentes conditions. Deux scénarios différents sont donc envisagés :

- le premier scénario sera de modifier la position des nœuds pour voir l'impact que l'emplacement a sur les performances;
- dans le contexte actuel de l'explosion des échanges de données, le second scénario visera à mesurer l'impact de la charge de trafic sur les performances d'OSCAR.

Les résultats suivants ont été réalisés sur un banc d'essai. Les performances des différentes métriques, dans un réseau composé de 10 nœuds et d'un DODAGroot, sont indiquées ci-dessous.

4.4.1 Configuration du banc d'essai

Pour valider les résultats de simulation obtenus précédemment, un banc d'essai a été mis en place dans le cadre d'une étude expérimentale. Comme mentionné précédemment, le nouveau système d'exploitation Contiki-NG (Next Generation), une plate-forme open source pour l'IoT qui est un fork de contiki (Dunkels et al., 2004), a été choisi.
Dans nos expérimentations, nous avons déployé un réseau composé de 11 CC2650 Launchpad qui comprend une racine et 10 nœuds répartis dans une topologie arborescente. Le DODAGroot est situé avec un rang égal à 0. La puissance de transmission est fixée à 5 dBm.
Nous générons un trafic ascendant de 1 pkts / min de manière convergeant, ce qui signifie que tous les nœuds transmettent les paquets au sink. Comme pour la simulation effectuée précédemment, nous avons considéré qu'une slotframe unicast est composée de 6 slots pour OSCAR et de 16 slots pour Orchestra. La durée de chaque intervalle de temps est fixée à 10 ms comme indiqué dans la norme TSCH 802.15.4e.

Tableau 4.2 Paramètres de l'expérience

Paramètres	**Valeurs**
Nombre de nœuds	11
Plateforme	Launchpad CC2650
Protocole MAC	IEEE 802.15.4e TSCH
Taille de la file d'attente des paquets	8
Taille du paquet de données	128 Bytes
Taille de la slotframe unicast (classe 0)	6
Taille de la slotframe unicast (classe 1)	7
Taille de la slotframe unicast (classe 2)	9
Taille de la slotframe unicast (classe 3)	12
Taille de la slotframe unicast (classe 4)	18
Taille de la slotframe unicast (classe 5)	36

4.4.2 Impact de la position des nœuds

Dans cette partie, deux topologies de réseaux ont été mise en place, comme pour la simulation, pour tester l'impact de la position des nœuds sur la performance du réseau. La première et la deuxième topologie sont ceux illustrées respectivement sur la figure 4.13 et 4.14. Comme pour la simulation, la deuxième forme de réseau est plus condensée et proche de la racine tandis que l'autre est un peu plus espacé et distant de la racine.

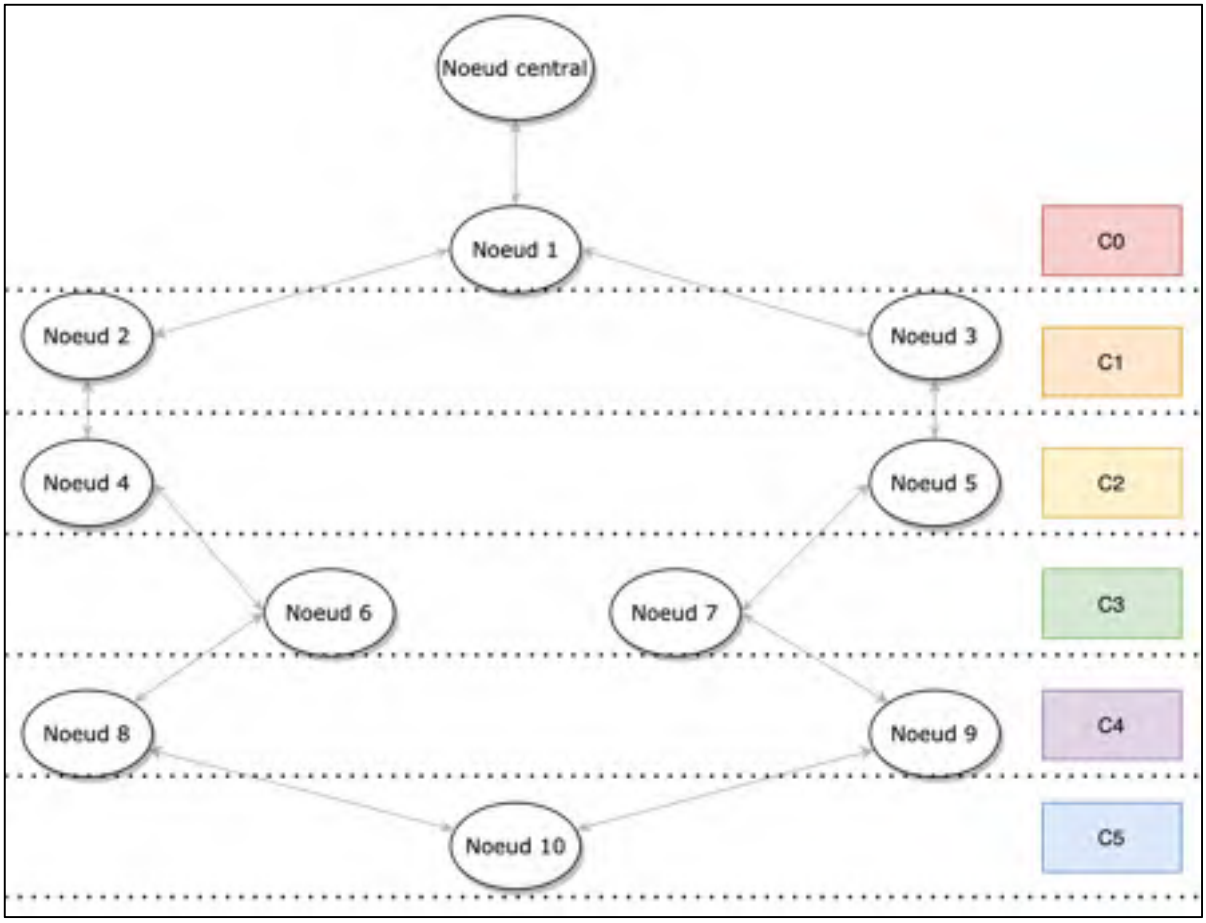

Figure 4.13 Première topologie de réseau

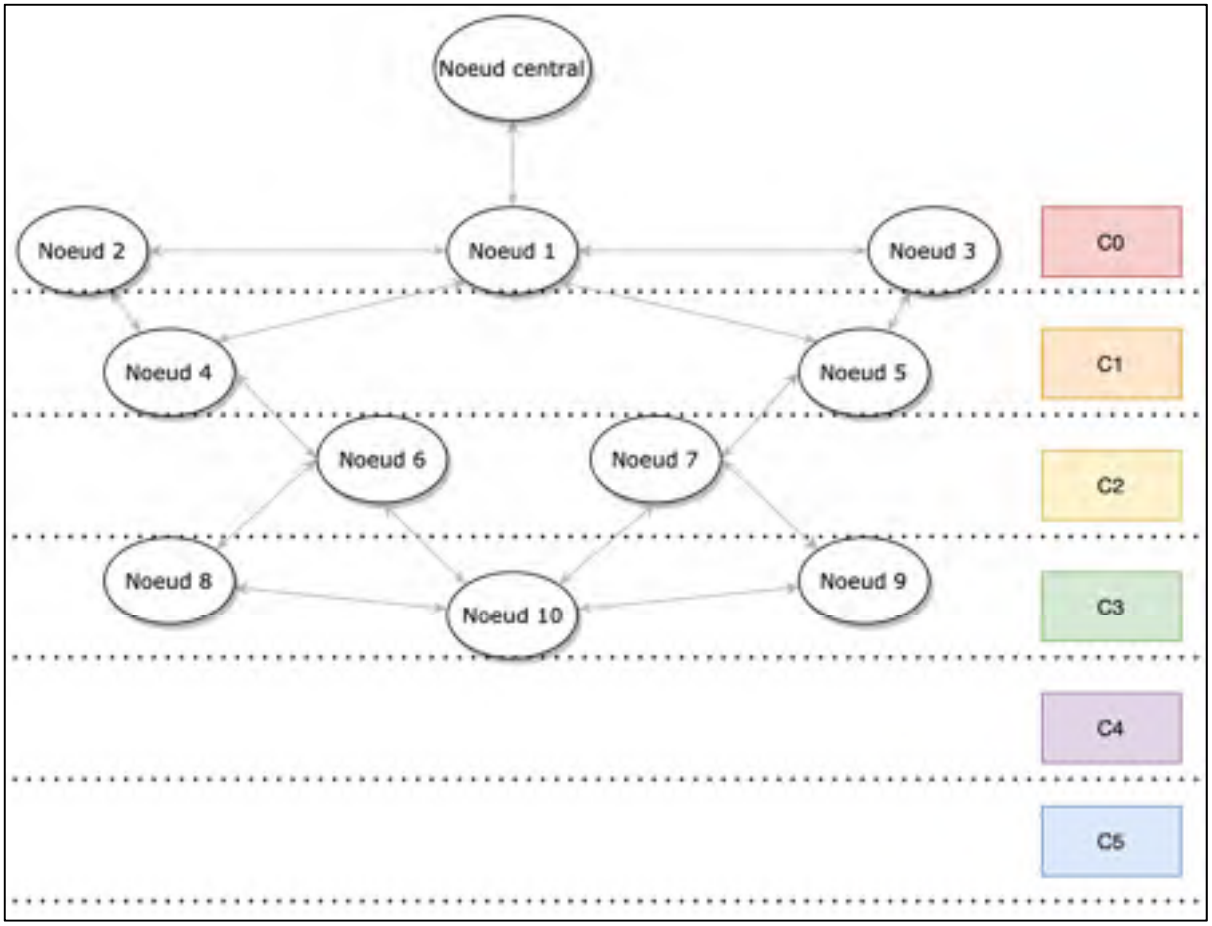

Figure 4.14 Deuxième topologie de réseau

4.4.2.1 Latence

Les figures 4.15 et 4.16 suivantes expliquent les résultats obtenus en termes de latence pour la première et deuxième topologie respectivement. On aperçoit que la latence moyenne d'OSCAR est plus petite que celle d'Orchestra pour la figure 4.15. Si on va dans les détails, on remarque que la plupart des nœuds montrent une latence moindre mais qu'à partir de la classe 4 et 5, soit les nœuds 8, 9 et 10, un plus grand délai point à point est présent.

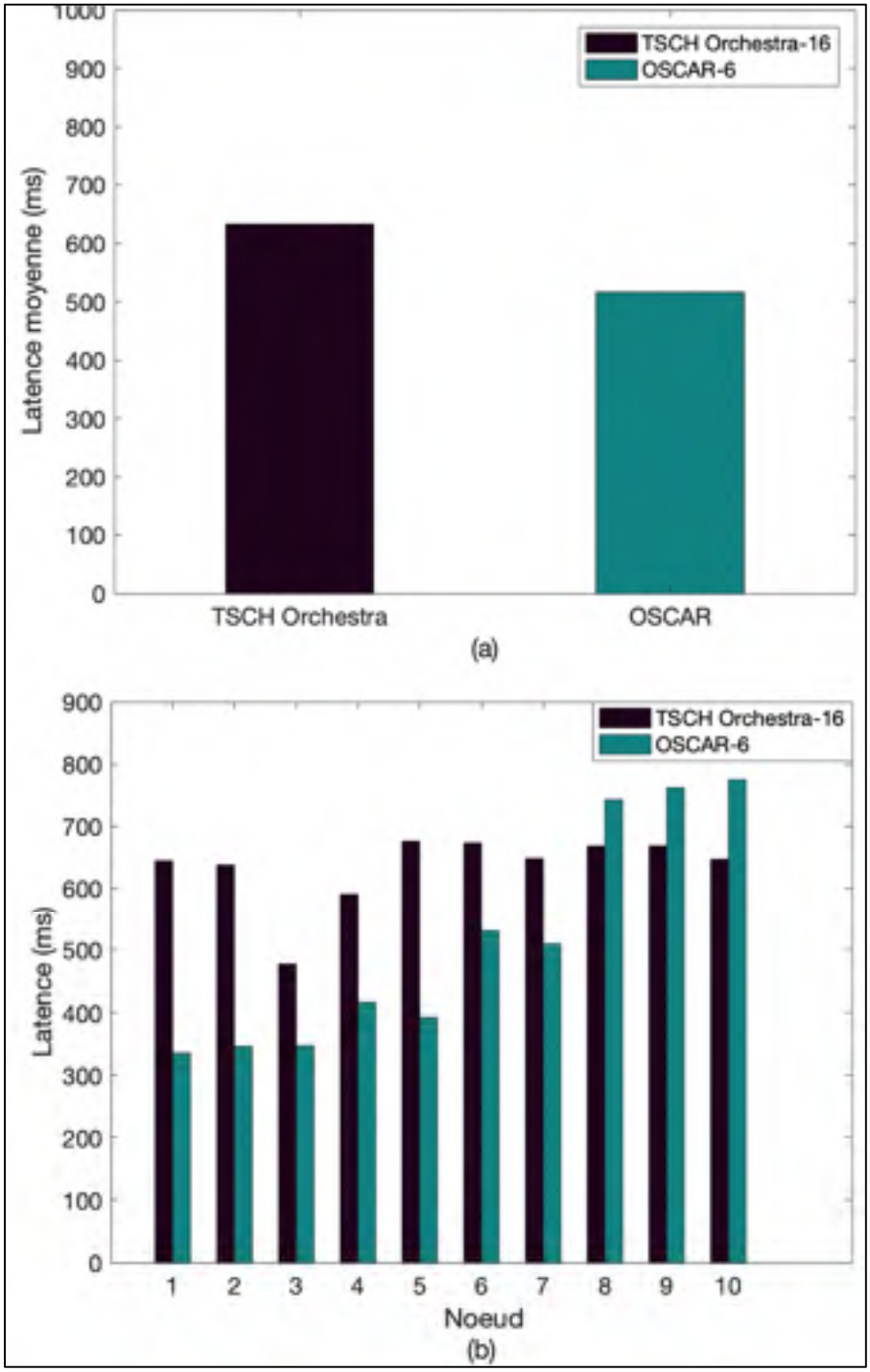

Figure 4.15 Première topologie : Délai moyen de bout en bout d'OSCAR et d'Orchestra : (a) Moyenne de l'ensemble du réseau (b) Moyenne noeud par noeud

Pour ce qui est de la deuxième disposition, OSCAR surpasse Orchestra sur la latence moyenne et le gain de performance devient plus important pour les premiers nœuds qui se situe dans la classe 0. Cela s'explique par le fait que ces nœuds ont plus d'opportunités pour évacuer le trafic, ce qui permet de résoudre le problème de conflit et de collision.

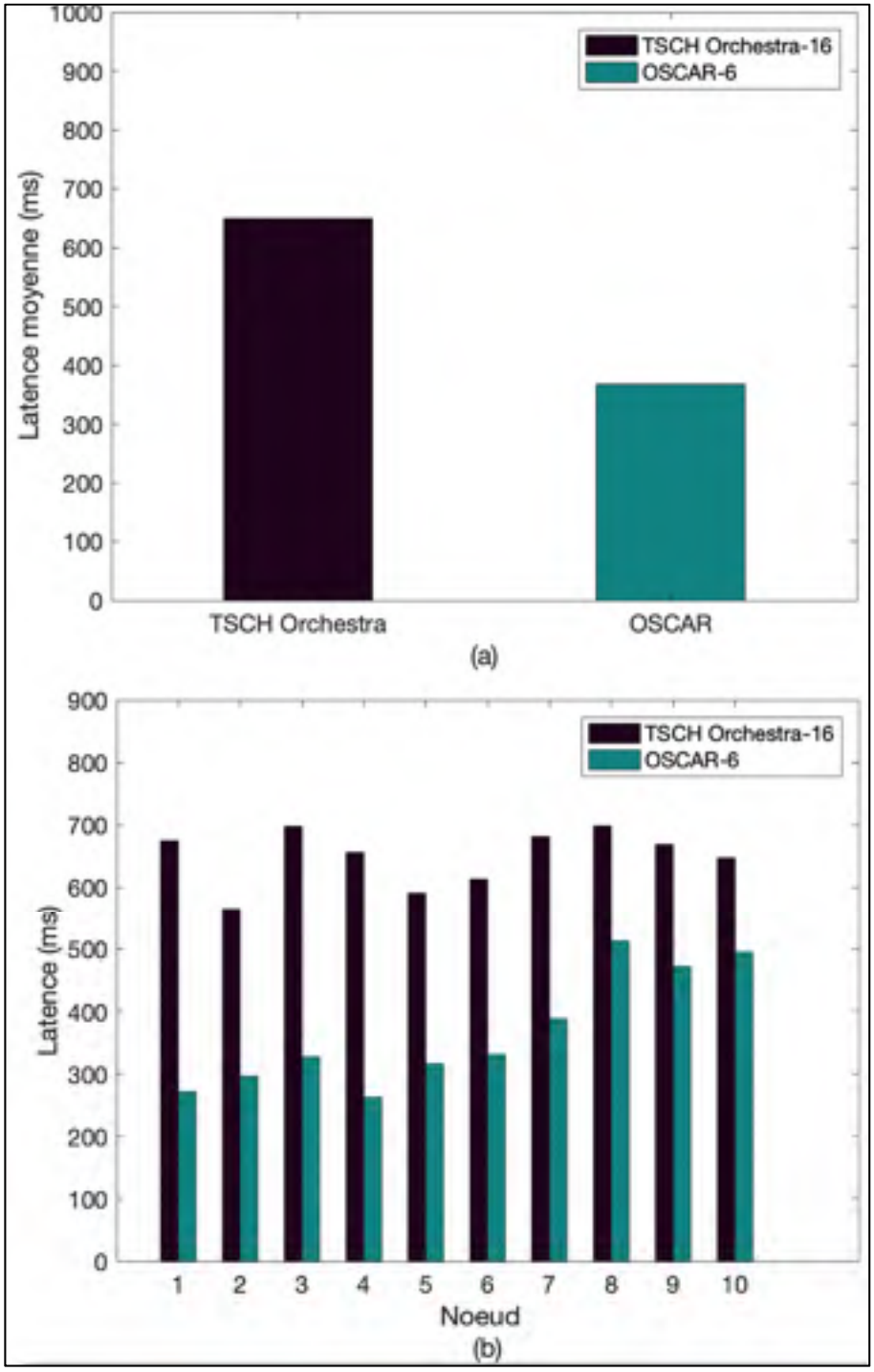

Figure 4.16 Deuxième topologie : Délai moyen de bout en bout d'OSCAR et d'Orchestra : (a) Moyenne de l'ensemble du réseau (b) Moyenne nœud par nœud

4.4.2.2 Cycle de service

La deuxième métrique testée est le cycle de service radio. La figure 4.17 montre la période de temps pendant laquelle le nœud est active pour la première topologie. On constate sur cette figure que le cycle de service moyen d'OSCAR est inférieur à celui de Orchestra. Pour OSCAR, les nœuds 1, 2 et 3 correspondent à ceux les plus sollicités car ils bénéficient d'un plus grand nombre de slots alloué et sont donc plus active. Tandis que les nœuds 8, 9 et 10 sont les moins demandants et actifs. Les mécanismes de planification des cellules utilisent efficacement la répartition des créneaux horaires pour enlever les slots des nœuds inactive.

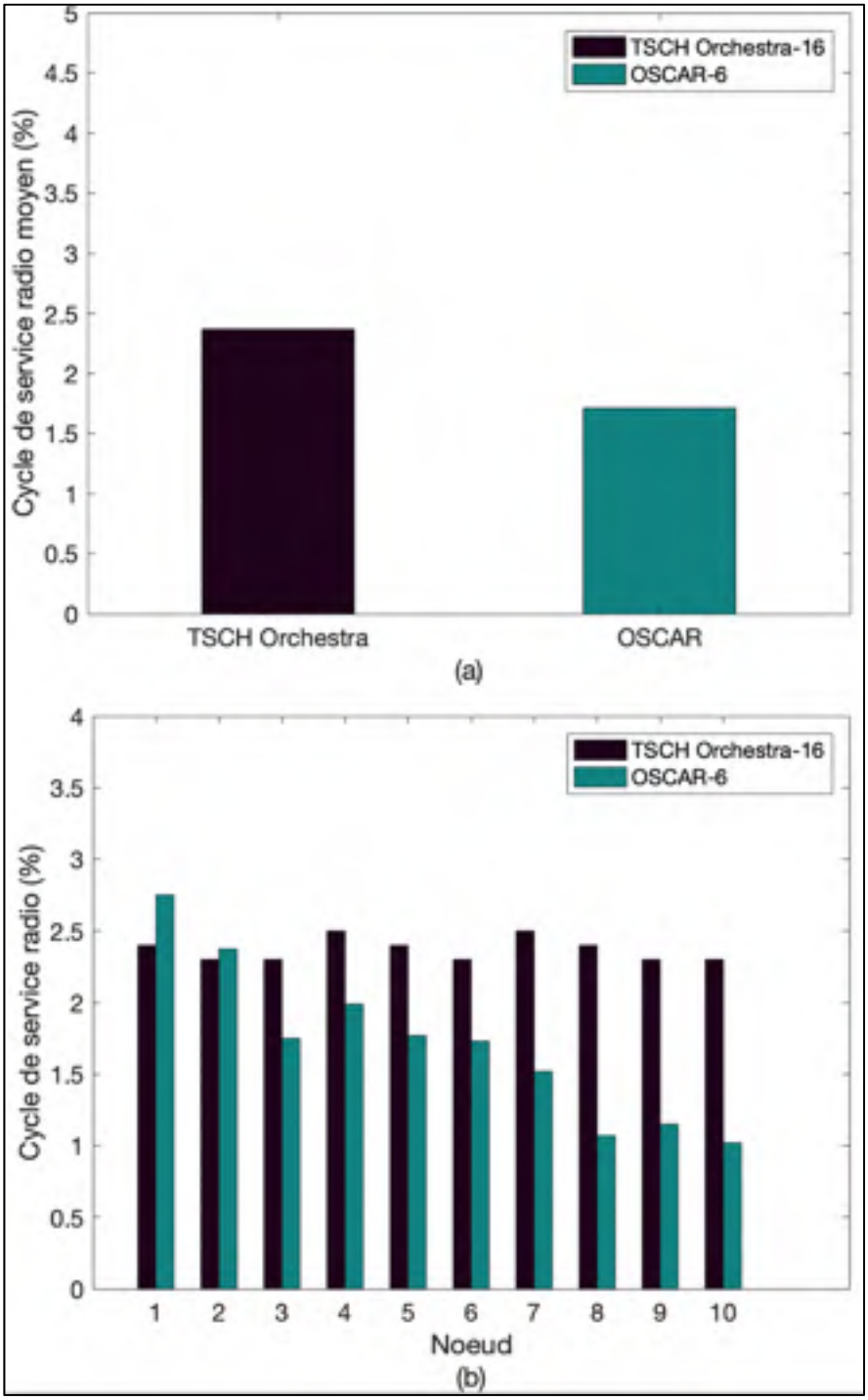

Figure 4.17 Première topologie : Cycle de service moyen d'OSCAR et d'Orchestra : (a) Moyenne de l'ensemble du réseau (b) Moyenne nœud par nœud

La deuxième topologie, quant à elle, qui est représenté sur la figure 4.18 indique un cycle de service moyen légèrement supérieur par rapport à la première disposition. Elle est à peu près équivalente au cycle de service moyen d'Orchestra. Cela s'explique par le fait que les nœuds bénéficient d'un plus grand nombre de slots alloués du fait de leur position rapprochés par rapport au nœud racine. L'ajustement du cycle de service de l'algorithme 2 permet de sauver de l'énergie. Ceci s'explique par le fait que les nœuds bénéficient d'un plus grand nombre de slots alloués du fait de leur position proche du nœud racine.

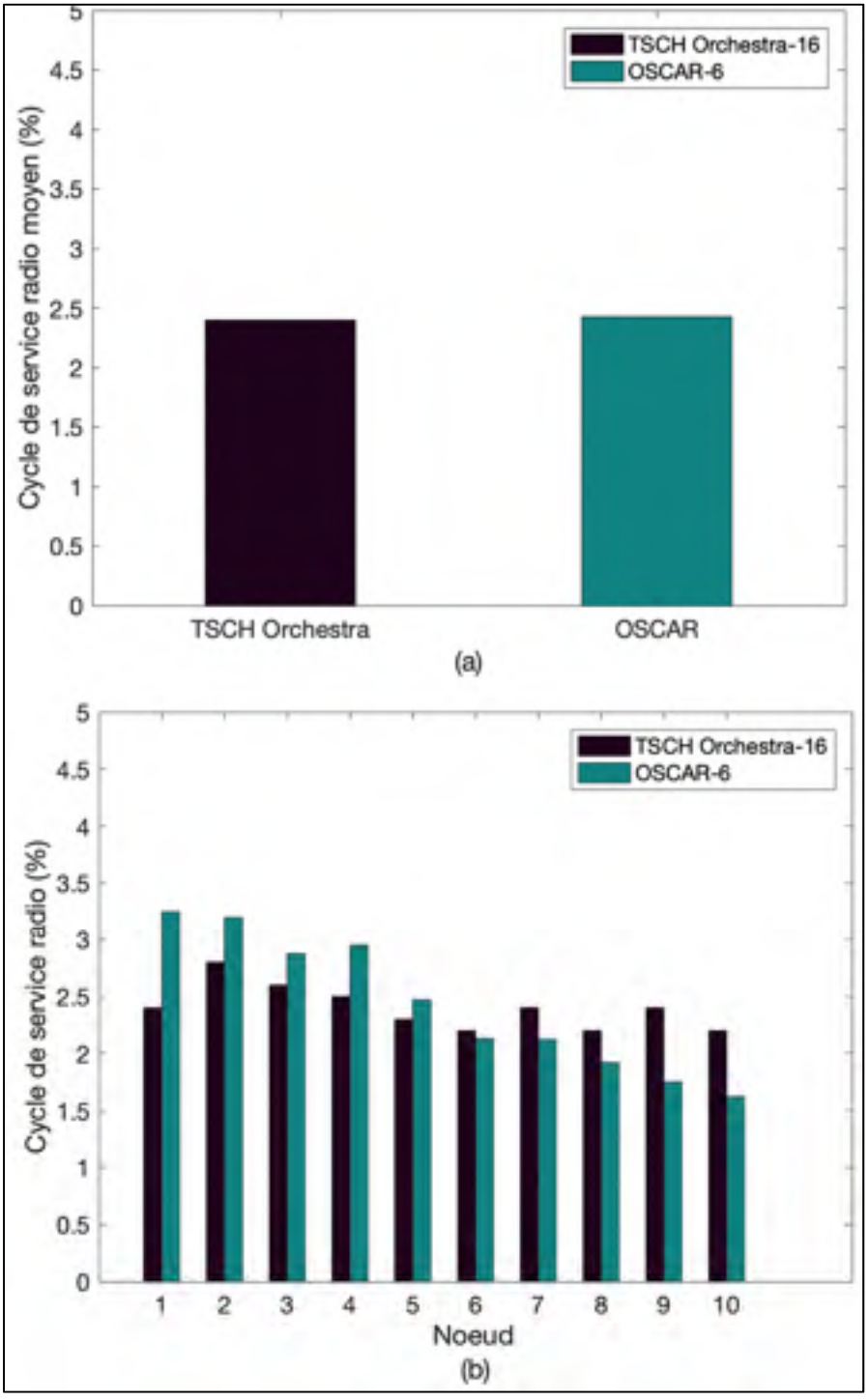

Figure 4.18 Deuxième topologie : Cycle de service moyen d'OSCAR et d'Orchestra : (a) Moyenne de l'ensemble du réseau (b) Moyenne nœud par nœud

4.4.2.3 Taux de livraison des paquets

La figure 4.19 illustre le taux moyen de livraison de paquets pour OSCAR et Orchestra. Dans cette expérimentation, chaque nœud a envoyé 2000 paquets jusqu'à la racine, soit un total de 20000 paquets envoyés par l'ensemble du réseau au nœud central. De plus, le débit de données a été fixé à 5 pkts / min. Pour la première et deuxième topologie, OSCAR présente un meilleur taux de livraison comparé à Orchestra. Si on va dans les détails pour OSCAR, on remarque que les nœuds 8,9 et 10 présentent des taux de livraison de paquets supérieurs lorsque la topologie est condensée. En effet ces nœuds se trouvent dans la classe 3 alors que lorsque la topologie est espacée, ces nœuds sont répartis dans les classes 4 et 5. Par exemple, le nœud 10 affiche un taux de 88.5% pour la première topologie et 95.7% pour la deuxième topologie. Cela démontre que l'allocation de slots en fonction de la classe du nœud joue un rôle important dans les performances du réseau.

Le taux de livraison des paquets pour Orchestra reste à peu près équivalent dans chacune des deux dispositions du réseau. Cela peut s'expliquer par le fait qu'Orchestra alloue le même nombre de slots à tous les nœuds, indépendamment de la position de ce nœud.

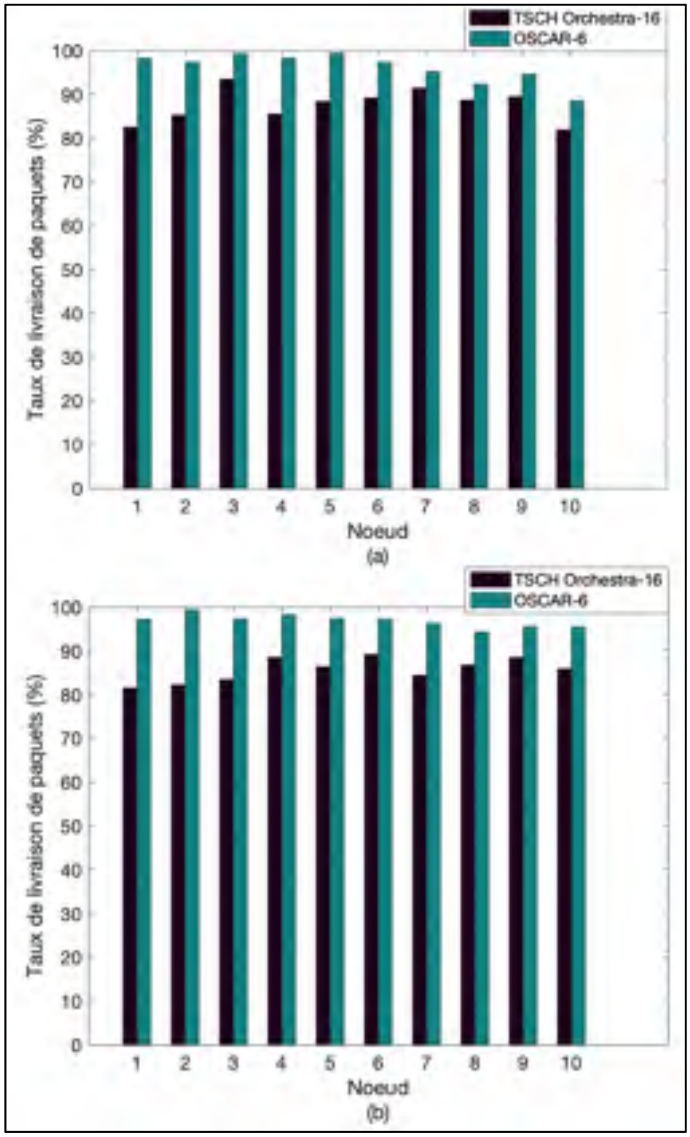

Figure 4.19 Taux moyen de livraison de paquets d'OSCAR et d'Orchestra : (a) Première topologie (b) Deuxième topologie

4.4.3 Impact de la charge du trafic

Le but de ce scénario est de tester le comportement du réseau lorsque le nœud a plus de trafic entrant, plus de nœuds enfants ou une planification qui se chevauche. Pour cela, nous utilisons la disposition du réseau de la figure 4.13.

4.4.3.1 Latence

Le délai moyen de bout en bout de OSCAR et Orchestra en fonction du débit de données est illustré à la figure 4.20. Trois configurations pour le débit de données sont utilisées à savoir 1, 2 et 6 pkts / min. Les résultats montrent que les performances de latence sont dégradées pour Orchestra à mesure que la charge de trafic augmente. Ceci est principalement due au manque d'opportunités de transmission. OSCAR améliore considérablement le délai de communication par rapport à Orchestra, qui s'est révélé très sensible aux variations de débit de données. L'installation de slots de communication supplémentaires dans la slotframe courant en fonction de la demande de trafic permet de maintenir le nombre de paquets mis en file d'attente aussi petit que possible et évite d'attendre les slotframes pour nettoyer la file. OSCAR réagit mieux aux trafics denses et présente ainsi une meilleure latence dans l'ensemble.

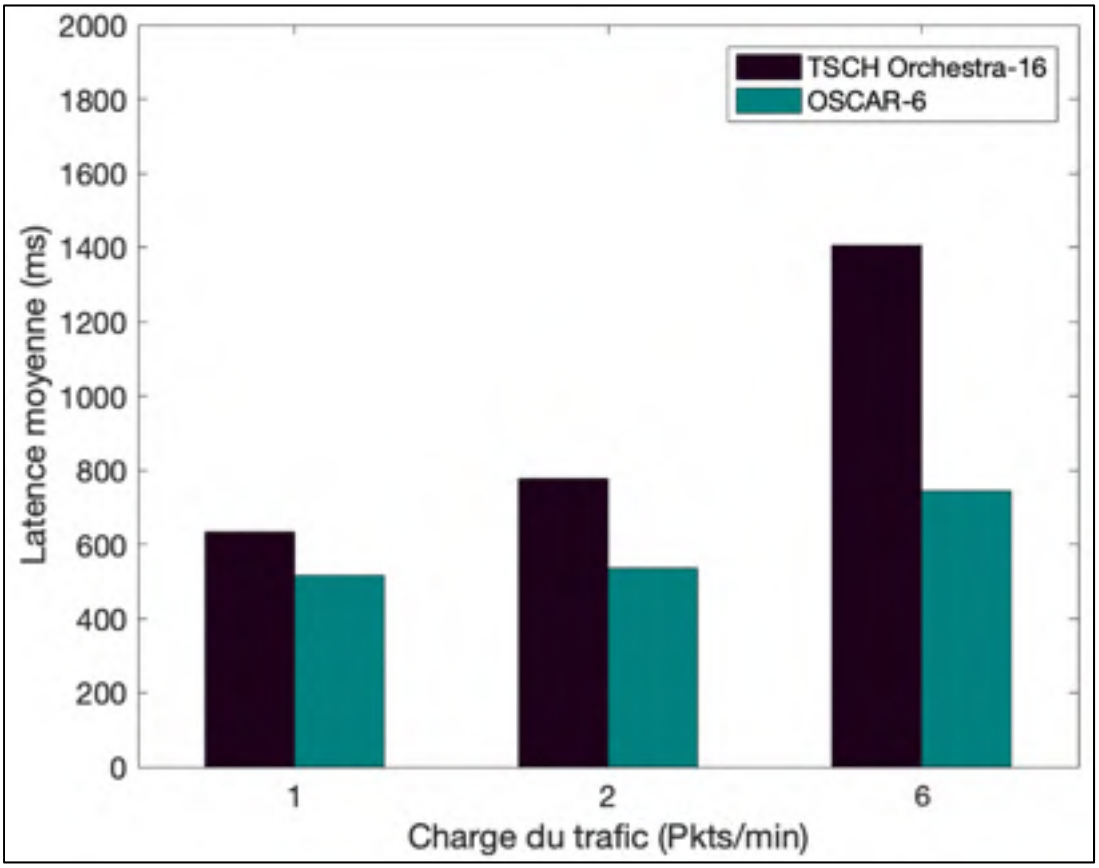

Figure 4.20 Délai moyen de bout en bout d'OSCAR et d'Orchestra en fonction de la charge de trafic

4.4.3.2 Cycle de service

Le gain de performance précédent est obtenu, comme on peut le voir sur la figure 4.21, en sacrifiant la consommation d'énergie étant donné que OSCAR accorde plus de cellules aux nœuds. Comme pour l'expérience précédente, trois configurations pour le débit de données sont utilisées à savoir 1, 2 et 6 pkts / min. Cette figure montre que les améliorations en termes de latence et de livraison de paquets d'OSCAR indiqués ci-dessus se font au détriment d'un cycle de service légèrement plus élevé. Plus précisément, l'existence de plus de créneaux de transmission et de réception dans l'ordonnancement de communication augmente la part du temps passé avec la radio allumée et augmente ainsi la consommation d'énergie, en particulier pour les nœuds proches du DODAGroot. Néanmoins, cette surconsommation intervient lorsque le réseau est surchargé et OSCAR n'augmente pas significativement le cycle de service moyen par rapport au gain sur la latence.

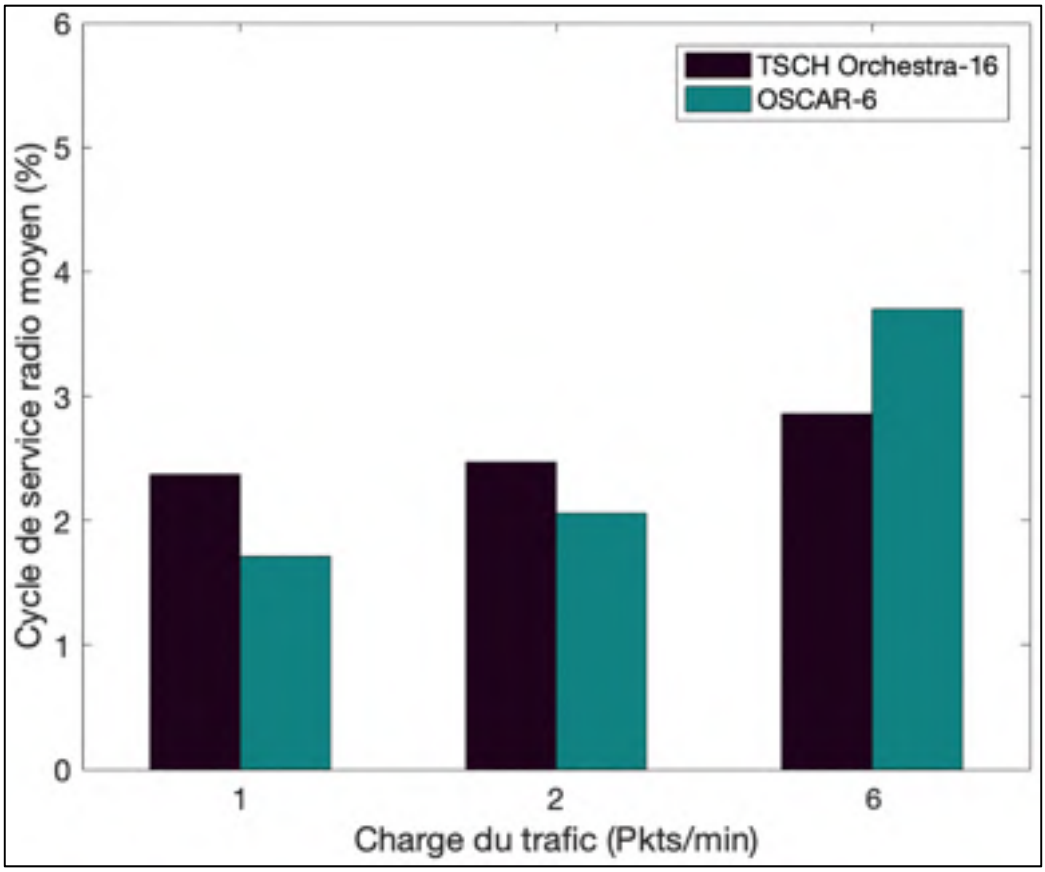

Figure 4.21 Cycle de service moyen d'OSCAR et d'Orchestra en fonction de la charge de trafic

4.4.3.3 Taux de livraison des paquets

Sur la figure 4.22, le taux moyen de livraison de paquets d'OSCAR et d'Orchestra en fonction du débit de données est tracé. Dans cette expérimentation, chaque nœud a envoyé 2000 paquets jusqu'à la racine, soit un total de 20000 paquets envoyés par l'ensemble du réseau au nœud central. De plus, nous utilisons un intervalle de transmission de paquets de 12 s, 6 s et 5 s soit 5, 10 et 15 pkts / min respectivement. Cette figure indique que les performances de livraison de paquets d'Orchestra sont dégradées à mesure que la charge de trafic augmente alors qu'en parallèle, OSCAR se trouve être moins affecté. Avec un débit de données de 5 pkts/min, OSCAR affiche un résultat de 96,1% contre 87,5% pour Orchestra. En d'autres mots, OSCAR et Orchestra ont perdu environ 800 et 2500 paquets respectivement. Lorsque l'intervalle de transmission des paquets passe à 15 pkts/min, le taux de livraison de paquets d'Orchestra s'aggrave considérablement et est réduit à 76,4%. En raison de l'augmentation de la retransmission et du manque de cellules pour la transmission de paquets, le temps pour que le paquet reste dans le tampon du nœud devient plus long et le paquet fini par être abandonné dans les nœuds congestionnés. Ainsi avec l'augmentation du débit de trafic, le niveau de congestion du réseau (notamment des nœuds proches de la racine) augmente, ce qui conduit à une perte de paquets. Avec un débit de 15 pkts/min, OSCAR affiche 90,7% de taux de livraison, soit une perte d'environ 1900 paquets contre 4700 pour Orchestra. OSCAR empêche (ou réduit) l'accumulation de paquets dans les files d'attente (et donc le débordement de la mémoire tampon), il préserve de bons ratios de livraison même lorsque le débit de trafic augmente à 15 pkts / min.

Une meilleure fiabilité est ainsi constatée pour ce système principalement due au fait qu'il dispose de plus d'unicast slots à mesure que l'on s'approche de la racine, résultant ainsi en une minimisation du nombre de paquets mis en file d'attente pour éviter le débordement des tampons pour les nœuds qui présentent un plus grand nombre de paquets dans leur tampon.

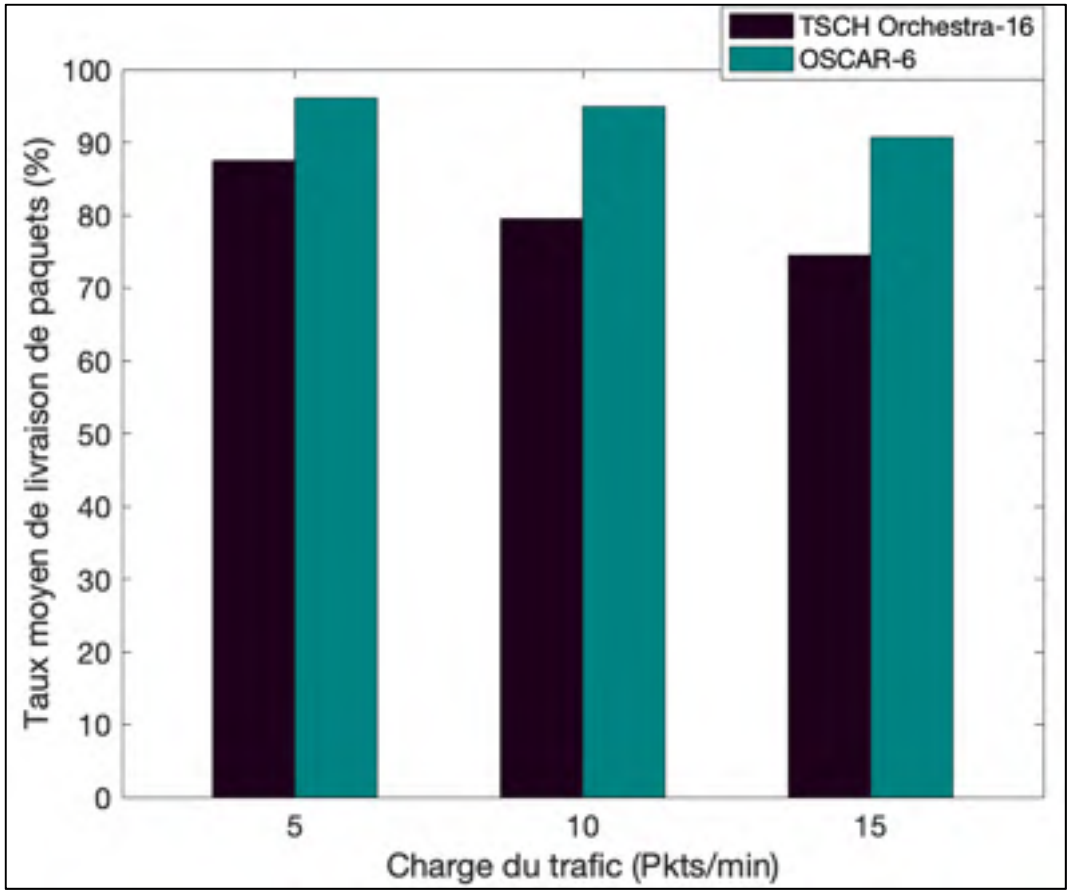

Figure 4.22 Taux moyen de livraison de paquets d'OSCAR et d'Orchestra en fonction de la charge de trafic

4.5 Comparaison des résultats de simulation et expérimentaux

Dans ce chapitre, l'analyse des performances de l'algorithme OSCAR s'est faite dans un premier temps à l'aide d'une simulation puis par mesures expérimentales.

Le premier scénario d'expérimentation a été de modifier la disposition du réseau pour voir l'impact sur les performances. Si pour l'expérimentation, des résultats ont pu notamment être tirés à l'aide d'un réseau de 10 nœuds, pour ce qui est des simulations, le résultat est tout autre. En effet, il a fallu pas moins de 60 nœuds sous cooja pour commencer à distinguer OSCAR et Orchestra en termes de latence, cycle de service et taux de livraison de paquets. Le constat est que les simulations ont démontré des résultats identiques entre OSCAR et Orchestra pour un

réseau composé de moins de 40 nœuds. L'hypothèse ici qui peut causer cette différence notable de nombre de nœuds serait que la plateforme de simulation nécessite un minimum conséquent de nœud pour qu'Orchestra montre ses limitations. En dessous de 40 nœuds, le trafic du réseau généré par cooja est trop faible pour gêner Orchestra. C'est pourquoi Orchestra et OSCAR arrivent à gérer le trafic de la même manière. En effet, OSCAR tire son épingle du jeu lorsqu'Orchestra montre ses limites. Une autre hypothèse serait que la plateforme Launchpad CC2650 s'adapte mieux à l'ordonnancement d'OSCAR par rapport à la plateforme de simulation Zolertia Z1 et que les différents passages de modes du noeud (Transmission, Reception, Basse consommation) sont davantage pris en compte. Mis à part le nombre de nœuds, les résultats ont révélé une similitude du comportement entre OSCAR et Orchestra lorsqu'on se concentre sur le scénario à savoir la variation entre une topologie de réseau espacée et condensée. Une petite différence est à noter pour le résultat de simulation du cycle de service de la deuxième topologie avec OSCAR qui présente une augmentation plus marquée avec Orchestra que pour le résultat expérimental. Ce résultat est sans aucun doute causé par le nombre de nœud avec leur radio allumée plus important en simulation.

Le second scénario de l'expérimentation qui était de tester le comportement du réseau avec des variations du débit de données n'a pas pu être réalisé avec le simulateur cooja. La cause est que des problèmes de mémoire gênent la simulation au-delà d'un débit de données de 1 paquet/minute. À chaque changement de ce paramètre, le simulateur ne peut le supporter et une erreur interrompt la simulation. Cependant, avec cooja, la variation du nombre de nœuds a montré qu'OSCAR réagit mieux à mesure que la charge de trafic augmente. Avec l'expérimentation, le constat est le même en faisant varier le débit de données.

Dans un aspect plus global, des limitations ont été constatées tout au long de ce travail de la part du simulateur cooja. Les problèmes de mémoire ont longtemps retardé la phase de simulation qui permet de valider le comportement de l'algorithme développé avant le déploiement de nœuds réels. Cooja souffre d'un nombre insuffisant de nœuds de simulation disponible et il est impossible de régler la puissance de transmission. Dans un aspect plus positif, on peut noter la présence de l'option vitesse de simulation qui permet d'exécuter la simulation plus rapidement qu'il ne faudrait dans une exécution en temps réel. Dans l'ensemble, cooja a démontré des résultats relativement précis qui ont été confortés et

confirmés par le banc d'essai. Les bancs d'essai et les simulateurs sont deux outils de conception et de validation importants et complémentaires. C'est ce qui a permis de valider le comportement du système OSCAR dans ce travail.

4.6 Discussions

Les résultats expérimentaux et de simulation ont montré qu'OSCAR présente de meilleures performances dans le taux de livraison de paquets et le délai de bout en bout par rapport à l'ordonnancement d'Orchestra grâce notamment à la meilleure répartition des cellules allouées aux nœuds instaurée par l'algorithme 1. De plus, même avec ces gains, OSCAR n'augmente pas considérablement le cycle de service moyen. En effet avec l'algorithme 2, les nœuds économisent de l'énergie en éteignant les nœuds inactifs. Lorsque les nœuds sont souvent actifs et transmettent des données, le délai peut être réduit car les slots qui sont attribués permettent d'évacuer le trafic. Tandis que si les nœuds deviennent inactifs, OSCAR commence à attribuer moins de slots progressivement et évite ainsi que les nœuds se réveillent plus souvent. Cette spécifie limite la consommation d'énergie due à une écoute au ralenti des nœuds.

Dans ce chapitre, on a démontré qu'OSCAR présente une meilleure performance qu'Orchestra. On constate ainsi un gain sur la latence moyenne et le taux de livraison de paquets, en particulier lorsque le trafic est dense et encombré. Les réseaux de capteurs sans fil ont des exigences contradictoires pour fonctionner en même temps avec une faible consommation d'énergie et une faible latence. Comme les réseaux sont conçus en tenant compte du facteur de faible consommation d'énergie, la latence de l'ensemble du système augmente. Inversement pour les réseaux où la priorité d'optimisation se situe au niveau de la latence, une plus grande consommation énergétique est observée. Ainsi, les deux exigences ont tendance à se contredire et une balance est nécessaire à trouver. L'avantage d'OSCAR ici est que les gains en latence et fiabilité sont bien plus importants que les pertes minimes en énergie engrangées par le réseau.

Comme vu précédemment d'autres solutions qui se base sur un ordonnancement autonome ont été proposées. e-TSCH-Orch fait partie de ces solutions et utilise une stratégie où lorsque la slotframe a suffisamment de plages horaires libres, le nœud peut les exploiter pour vider sa file

d'attente. Cependant, cette méthode aggrave les problèmes de contention et de collision car les nœuds se disputent les cellules libres. Pour Escalator, les nœuds souffrent davantage de la surcharge d'écoute inactive car lorsqu'un nœud reçoit un paquet, il envoie immédiatement le paquet reçu à son parent dans la cellule qui suit de façon consécutive. L'écoute inactive est répertoriée sur tous les nœuds de sous-arbre et amplifie l'énergie gaspillée. ALICE alloue le même nombre de slots Tx / Rx dans une slotframe sous une liaison directionnelle. Néanmoins, alors que le trafic s'intensifie, ALICE souffre également de collisions sur certains nœuds conduisant à des pertes de liaison.

Les résultats expérimentaux d'OSCAR montrent que la consommation d'énergie des nœuds autour de la racine est augmentée. Cependant, OSCAR attribue arbitrairement des slots aux nœuds en fonction de leur position. Une attribution entièrement dynamique qui se base sur la charge trafic du réseau en temps réel serait intéressant à explorer.

Tableau 4.3 Évaluation qualitative des approches d'ordonnancement TSCH autonomes

Approche	**Réallocation de cellules**	**Adaptation au trafic**	**Application visée**	**Commentaires**
ALICE	✓	×	Général	Avantage : approche directionnelle plutôt que basé sur le nœud Inconvénient : nombre plus élevé de collisions
Escalator	×	×	Sensible au délai	Avantage : Faible latence Inconvénient : écoutes inactives des nœuds plus importantes
e-TSCH-Orch	×	✓	Général	Avantage : faible niveau de file d'attente Inconvénient : Ordonnancement instable entraînant une forte probabilité de collision
Orchestra	×	×	Général	Avantage : routage indépendant Inconvénient : nombre plus élevé de collisions
OSCAR	✓	✓	Sensible au délai	Avantage : très faible latence combinée à une faible perte en énergie en présence d'un trafic dense Inconvénient : consommation énergétique

plus élevée des nœuds près du sink

CONCLUSION ET PERSPECTIVES FUTURES

Dans ce mémoire, une nouvelle conception d'ordonnancement autonome est proposée pour les réseaux TSCH convergeant. La technologie sans fil est extrêmement attrayante pour les applications industrielles, car elle réduit considérablement les coûts d'installation qui sont très élevés. Les établissements industriels tels que les raffineries, les industries chimiques et les centrales électriques mettent en œuvre des processus complexes de surveillance et de gestion qui nécessitent une latence et une fiabilité très strictes. Une application courante des réseaux de capteurs sans fil dans ce domaine-là est la collecte vers un nœud sink de valeurs telles que la température, la pression, les flux de trafic ou le niveau de remplissage du réservoir, utilisées à la fois pour contrôler l'environnement à distance et pour coordonner les étapes de production.

Ainsi dans ce sens, un examen approfondi et une revue de littérature ont mis en évidence les progrès récents dans le domaine et les principales forces et limites de ces contributions ont été identifiées dans ce mémoire. Cette étude a permis de mettre en lumière l'enjeu derrière le processus de conception d'algorithmes efficaces de planification des cellules qui déterminent quelle action un nœud doit entreprendre dans chaque intervalle de temps. Principalement, le compromis bien connu entre l'énergie et la latence a été le défi à respecter tout au long de cette conception. De ce fait, les contributions suivantes ont été réalisées :

- une architecture de classification des nœuds permettant une meilleure répartition des cellules allouées en fonction de leur position;
- un système pour réduire les cellules allouées aux nœuds inactifs afin de limiter les pertes énergétiques.

Cet algorithme, nommé OSCAR, gère le nombre de créneaux horaires alloués à chaque nœud en utilisant la valeur du rang décrit par le protocole de routage RPL. Les nœuds sont répartis sous différentes couches où chaque nœud se voit attribuer un numéro de classe basé sur la proximité de DODAGroot. Le but étant que plus le nœud est proche de la racine, plus il obtient de slots afin de maximiser les opportunités de transmission. Pour éviter la surconsommation,

OSCAR met en place un mécanisme permettant d'ajuster le cycle de service radio de chaque nœud en réduisant les slots alloués aux nœuds inactifs quelle que soit leur classe.

L'analyse des performances montre qu'OSCAR surpasse Orchestra sur la latence moyenne, en particulier lorsque le trafic est dense et encombré. Concernant l'énergie consommée, nous avons vu que le cycle de service moyen d'OSCAR dépend, comme pour la latence, de la position du nœud dans le réseau. Dans les deux agencements testés, il est inférieur à Orchestra lorsque la charge de trafic est faible, en raison de l'algorithme d'ajustement du cycle de service. Cependant, cela devient important au fur et à mesure que la charge de trafic réseau augmente car le réseau est constamment sollicité et les nœuds sont plus actifs qu'Orchestra. Enfin, le débit de livraison des paquets reste similaire entre OSCAR et Orchestra, quelle que soit la position du nœud à faible trafic. Cependant, OSCAR affiche de meilleures performances à mesure que la charge de trafic augmente.

Pour les travaux futurs, des recherches continus doivent être menés pour développer des techniques et des méthodologies qui répondent aux défis des applications de collection de données. Il sera intéressant d'explorer la combinaison de ce travail avec l'intelligence artificielle où les nœuds pourraient s'adapter en temps réel à la charge de trafic en décidant d'allouer ou de supprimer des slots dynamiquement de façon autonome. Cela permettrait d'ajouter de l'efficacité en ciblant les nœuds réellement concernés par la congestion ou l'inactivité et de réduire à la fois la latence et la consommation d'énergie.
Tout au long de ce mémoire nous avons envisagé un seul sink car il s'agit du scénario générique de l'opération convergecast. Un autre apport sera d'explorer l'utilisation de plusieurs sinks afin d'alléger la pression exercée sur le réseau à mesure que l'on s'approche du nœud central.

ANNEXE I

OSCAR: CODE

```
static uint16_t slotframe_handle = 0;
static struct tsch_slotframe *sf_unicast;


/* Set to 1 to use OSCAR : rank based slots allocation, 0 for default NS behaviour */
#define USE_RANK  1



#if USE_RANK


#if ROUTING_CONF_RPL_CLASSIC
#include "net/routing/rpl-classic/rpl-private.h"
#else
#include "net/routing/rpl-lite/rpl-dag.h"
#endif


#define PACKET_INTERVAL   5
#define MONITOR_INTERVAL  (10 * CLOCK_SECOND)

typedef enum
{
  RANK_CLASS_MAX = 0,
  RANK_CLASS_5 = RANK_CLASS_MAX,
  RANK_CLASS_4,
  RANK_CLASS_3,
  RANK_CLASS_2,
  RANK_CLASS_1,
  RANK_CLASS_0,
  RANK_CLASS_COUNT //6
} RankClass;

static uint16_t get_node_timeslot(const linkaddr_t *addr);
static uint16_t get_node_channel_offset(const linkaddr_t *addr);

static RankClass current_class = RANK_CLASS_0;
static int packet_selected = 0;
```

```
static uint16_t get_rank()
{
  rpl_dag_t *dag;
  rpl_rank_t root_rank;
  //rpl_rank_t dag_rank;
#if ROUTING_CONF_RPL_LITE
  dag = &curr_instance.dag;
  root_rank = ROOT_RANK;
  //dag_rank = DAG_RANK(dag->rank);
#else
  rpl_instance_t *instance;
  dag = rpl_get_any_dag();
  instance = dag != NULL ? dag->instance : NULL;
  root_rank = ROOT_RANK(instance);
  //dag_rank = DAG_RANK(dag->rank, instance);
#endif

  if(dag != NULL && dag->rank != RPL_INFINITE_RANK) {
  LOG_INFO("rank: %u Current class: %u \r\n", dag->rank - root_rank,current_class);
  return (dag->rank - root_rank);
  }

  return RPL_INFINITE_RANK;

}

static RankClass get_class(uint16_t rank)
{
  if(rank <= 128)
    return RANK_CLASS_0;//5
  if(rank <= 256)
    return RANK_CLASS_1;
  if(rank <= 384)
    return RANK_CLASS_2;
  if(rank <= 512)
    return RANK_CLASS_3;
  if(rank <= 768)
    return RANK_CLASS_4;
  else
    return RANK_CLASS_5;//0

}
```

```
static void add_links(uint16_t offset, uint16_t timeslot,  linkaddr_t *addr)
{
  int i;
  for(i = 0; i < ORCHESTRA_UNICAST_PERIOD; i++) {
    tsch_schedule_add_link(sf_unicast,
        LINK_OPTION_SHARED | LINK_OPTION_TX | ( i == timeslot ? LINK_OPTION_RX : 0 ),
        LINK_TYPE_NORMAL, &tsch_broadcast_address,
        offset + i, get_node_channel_offset(addr));
  }

}

static void remove_links(uint16_t offset,  linkaddr_t *addr)
{
  int i;
  for(i = 0; i < ORCHESTRA_UNICAST_PERIOD; i++) {
  tsch_schedule_remove_link(sf_unicast,
          tsch_schedule_get_link_by_timeslot(sf_unicast, offset + i, get_node_channel_offset(addr)));

  }

}
```

```
static void reschedule(uint16_t c)
{
  if(c == current_class)
    return;

    linkaddr_t *local_addr = &linkaddr_node_addr;
  uint16_t rx_timeslot = get_node_timeslot(local_addr);

  uint16_t i, offset, diff;
  if(current_class < c)
  {
    diff = c - current_class;
    for (i=0; i< diff; i++)
    {
      offset = (current_class + i + 1) * ORCHESTRA_UNICAST_PERIOD;
      LOG_INFO("add link offset: %u\r\n", offset);
      add_links(offset, rx_timeslot, local_addr);
    }

  }
  else
  {
    diff = current_class - c;
    for (i=0; i< diff; i++)
    {
      offset = (c + i + 1) * ORCHESTRA_UNICAST_PERIOD;
      LOG_INFO("remove link offset: %u\r\n", offset);
      remove_links(offset, local_addr);
    }

  }
  current_class = c;
}
```

```
PROCESS(packets_monitor_process, "Packets monitor");
PROCESS_THREAD(packets_monitor_process, ev, data)
{
  static struct etimer periodic_timer;
  PROCESS_BEGIN();
  LOG_INFO("Start packets monitor\r\n");
  while(1)
  {
    packet_selected = 0;

    etimer_set(&periodic_timer, MONITOR_INTERVAL);
    PROCESS_WAIT_EVENT_UNTIL(etimer_expired(&periodic_timer));

    if(!packet_selected)
    {
      if(current_class != RANK_CLASS_MAX && get_rank() > 128)
      {

        uint16_t c = current_class - 1;
        LOG_INFO("Reschedule to lower class %u because of inactivity\r\n", RANK_CLASS_0 - c);
        reschedule(c);
      }
    }
  }
  PROCESS_END();
}
```

```
static int
select_packet(uint16_t *slotframe, uint16_t *timeslot, uint16_t *channel_offset)
{
#if USE_RANK
  static uint16_t counter;
  counter  = (counter + 1) % PACKET_INTERVAL;
  if(!counter)
  {
  uint16_t rank = get_rank();
  RankClass c = get_class(rank);
  reschedule(c);
  }
  packet_selected = 1;
#endif

  /* Select data packets we have a unicast link to */
  const linkaddr_t *dest = packetbuf_addr(PACKETBUF_ADDR_RECEIVER);
  if(packetbuf_attr(PACKETBUF_ATTR_FRAME_TYPE) == FRAME802154_DATAFRAME
     && !linkaddr_cmp(dest, &linkaddr_null)) {
    if(slotframe != NULL) {
      *slotframe = slotframe_handle;
    }
    if(timeslot != NULL) {
      *timeslot = get_node_timeslot(dest);
    }
    /* set per-packet channel offset */
    if(channel_offset != NULL) {
      *channel_offset = get_node_channel_offset(dest);
    }
    return 1;
  }
  return 0;
}
```

BIBLIOGRAPHIE

Accettura, N., Palattella, M. R., Boggia, G., Grieco, L. A., & Dohler, M. (2013). Decentralized Traffic Aware Scheduling for multi-hop Low power Lossy Networks in the Internet of Things. Dans *2013 IEEE 14th International Symposium on « A World of Wireless, Mobile and Multimedia Networks » (WoWMoM)* (pp. 1-6). https://doi.org/10.1109/WoWMoM.2013.6583485

Aijaz, A., & Raza, U. (2017). DeAMON: A Decentralized Adaptive Multi-Hop Scheduling Protocol for 6TiSCH Wireless Networks. *IEEE Sensors Journal*, *17*(20), 6825-6836. https://doi.org/10.1109/JSEN.2017.2746183

Ang, K. L.-M., Seng, J. K. P., & Zungeru, A. M. (2018). Optimizing Energy Consumption for Big Data Collection in Large-Scale Wireless Sensor Networks With Mobile Collectors. *IEEE Systems Journal*, *12*(1), 616-626. https://doi.org/10.1109/JSYST.2016.2630691

Boubiche, S., Boubiche, D. E., Bilami, A., & Toral-Cruz, H. (2018). Big Data Challenges and Data Aggregation Strategies in Wireless Sensor Networks. *IEEE Access*, *6*, 20558-20571. https://doi.org/10.1109/ACCESS.2018.2821445

Bughin, J. (2016). Big data, Big bang? *Journal of Big Data*, *3*(1), 2. https://doi.org/10.1186/s40537-015-0014-3

Cheng, J., Chen, W., Tao, F., & Lin, C.-L. (2018). Industrial IoT in 5G environment towards smart manufacturing. *Journal of Industrial Information Integration*, *10*, 10-19. https://doi.org/10.1016/j.jii.2018.04.001

Choi, K.-H., & Chung, S.-H. (2016). A New Centralized Link Scheduling for 6TiSCH Wireless Industrial Networks. *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*, 360-371. https://doi.org/10.1007/978-3-319-46301-8_30

Cohen, A., Cohen, A., & Gurewitz, O. (2020). Efficient Data Collection Over Multiple Access Wireless Sensors Network. *IEEE/ACM Transactions on Networking*, *28*(2), 491–504. https://doi.org/10.1109/TNET.2020.2964764

De Guglielmo, D., Brienza, S., & Anastasi, G. (2016). IEEE 802.15.4e: A survey. *Computer Communications*, *88*, 1-24. https://doi.org/10.1016/j.comcom.2016.05.004

Demir, A. K., & Bilgili, S. (2019). DIVA: a distributed divergecast scheduling algorithm for IEEE 802.15.4e TSCH networks. *Wireless Networks*, *25*(2), 625-635. https://doi.org/10.1007/s11276-017-1580-4

Dunkels, A., Gronvall, B., & Voigt, T. (2004). Contiki - a lightweight and flexible operating system for tiny networked sensors. Dans *29th Annual IEEE International Conference on Local Computer Networks* (pp. 455-462). https://doi.org/10.1109/LCN.2004.38

Duquennoy, S., Al Nahas, B., Landsiedel, O., & Watteyne, T. (2015). Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH. Dans *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems - SenSys '15* (pp. 337-350). Seoul, South Korea : ACM Press. https://doi.org/10.1145/2809695.2809714

Hamza, T., & Kaddoum, G. (2019). Enhanced Minimal Scheduling Function for IEEE 802.15.4e TSCH Networks. Dans *2019 IEEE Wireless Communications and Networking Conference (WCNC)* (pp. 1–6). Marrakesh, Morocco : IEEE Press. https://doi.org/10.1109/WCNC.2019.8885940

Harb, H., & Makhoul, A. (2018). Energy-Efficient Sensor Data Collection Approach for Industrial Process Monitoring. *IEEE Transactions on Industrial Informatics*, *14*(2), 661-672. https://doi.org/10.1109/TII.2017.2776082

Ibarra-Esquer, J. E., González-Navarro, F. F., Flores-Rios, B. L., Burtseva, L., & Astorga-Vargas, M. A. (2017). Tracking the Evolution of the Internet of Things Concept Across Different Application Domains. *Sensors*, *17*(6), 1379. https://doi.org/10.3390/s17061379

IEEE802.15.4e. (2012, 6 février). IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer. Repéré à https://standards.ieee.org/standard/802_15_4e-2012.html

Industrial IoT Applications. (2018, 1 juin). *DataFlair*. Repéré à https://data-flair.training/blogs/industrial-iot-applications/

Jin, Y., Raza, U., & Sooriyabandara, M. (2018). BOOST: Bringing Opportunistic ROuting and Effortless-Scheduling to TSCH MAC. Dans *2018 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-7). https://doi.org/10.1109/GLOCOM.2018.8647851

Kamgueu, P. O., Nataf, E., & Ndie, T. D. (2018). Survey on RPL enhancements: A focus on topology, security and mobility. *Computer Communications*, *120*, 10-21. https://doi.org/10.1016/j.comcom.2018.02.011

Kandris, D., Nakas, C., Vomvas, D., & Koulouras, G. (2020). Applications of Wireless Sensor Networks: An Up-to-Date Survey. *Applied System Innovation*, *3*(1), 14. https://doi.org/10.3390/asi3010014

Kim, S., Kim, H.-S., & Kim, C. (2019). ALICE: Autonomous Link-based Cell Scheduling for TSCH. Dans *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)* (pp. 121-132). https://doi.org/10.1145/3302506.3310394

Li, S., Xu, L. D., & Zhao, S. (2018). 5G Internet of Things: A survey. *Journal of Industrial Information Integration*, *10*, 1-9. https://doi.org/10.1016/j.jii.2018.01.005

Liu, X., Sheng, Z., Yin, C., Ali, F., & Roggen, D. (2017). Performance Analysis of Routing Protocol for Low Power and Lossy Networks (RPL) in Large Scale Networks. *IEEE InternetofThingsJournal*, *4*(6), 2172-2185.https://doi.org/10.1109/JIOT.2017.2755980

Marques, B. F. (2017). Application-Driven Wireless Sensor Networks. *undefined*. Repéré à /paper/Application-Driven-Wireless-Sensor-Networks-Marques/76cfc696664a270f9e8641def9255c28c0a32322

Muravyov, S. V., Tao, S., Chan, M. C., & Tarakanov, E. V. (2015). Consensus rankings in prioritized converge-cast scheme for wireless sensor network. *Ad Hoc Networks*, *24*, 160-171. https://doi.org/10.1016/j.adhoc.2014.08.015

Oh, S., Hwang, D., Kim, K.-H., & Kim, K. (2018). Escalator: An Autonomous Scheduling Scheme for Convergecast in TSCH. *Sensors*, *18*(4), 1209. https://doi.org/10.3390/s18041209

Palattella, M. R., Accettura, N., Dohler, M., Grieco, L. A., & Boggia, G. (2012). Traffic Aware Scheduling Algorithm for reliable low-power multi-hop IEEE 802.15.4e networks. Dans *2012 IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC)* (pp. 327-332). https://doi.org/10.1109/PIMRC.2012.6362805

Qin, Z., Wu, D., Xiao, Z., Fu, B., & Qin, Z. (2018). Modeling and Analysis of Data Aggregation From Convergecast in Mobile Sensor Networks for Industrial IoT. *IEEE Transactions on Industrial Informatics*, *14*(10), 4457-4467. https://doi.org/10.1109/TII.2018.2846687

Rekik, S., Baccour, N., Jmaiel, M., Drira, K., & Grieco, L. A. (2018). Autonomous and Traffic-aware Scheduling for TSCH Networks. *Computer Networks*, *135*, pp.201-212. https://doi.org/10.1016/j.comnet.2018.02.023

Seyedzadegan, M., Othman, M., Ali, B. M., & Subramaniam, S. (2011). Wireless Mesh Networks: WMN Overview, WMN Architecture. *undefined*. Repéré à /paper/Wireless-Mesh-Networks%3A-WMN-Overview%2C-WMN-Seyedzadegan-Othman/dcbcf74471f2455e18c5ee8584d62a52e224177c

Soua, R., Minet, P., & Livolant, E. (2012). MODESA: An optimized multichannel slot assignment for raw data convergecast in wireless sensor networks. Dans *2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC)* (pp. 91-100). https://doi.org/10.1109/PCCC.2012.6407742

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. https://doi.org/10.1002/ett.2991

T. Winter, P. Thubert, & A. Brandt. (2012, mars). RPL: IPv6 Routing Protocol for Low-Power

and Lossy Networks. Repéré à https://tools.ietf.org/html/rfc6550

Tall, H. (2018). *Load balancing in multichannel data collection wireless sensor networks*.

Venkataramanan, V. J., & Lin, X. (2011). Low-complexity scheduling algorithm for sum-queue minimization in wireless convergecast. Dans *2011 Proceedings IEEE INFOCOM* (pp. 2336-2344). https://doi.org/10.1109/INFCOM.2011.5935052