THÈSE

Pour l'obtention du grade de
DOCTEUR DE L'UNIVERSITÉ DE POITIERS
UFR des sciences fondamentales et appliquées
XLIM
(Diplôme National - Arrêté du 25 mai 2016)

École doctorale : Sciences et Ingénierie des Systèmes, Mathématiques, Informatique (Limoges)
Secteur de recherche : Traitement du signal et des images

Présentée par :
Samir Dawaliby

Machine-to-Machine (M2M) Communications in Next Generation Networks: Spectrum management and energy efficiency
.

Directeur(s) de Thèse :
Yannis Pousset, Abbas Bradai

Soutenue le 11 octobre 2019 devant le jury

Jury :

Président	**Jean-François Diouris**	Professeur, IETR, Polytech, Université de Nantes
Rapporteur	**Laurent Clavier**	Professeur, IRCICA, IMT, Université de Lille
Rapporteur	**Toufik Ahmed**	Professeur, LaBRI, Université de Bordeaux
Membre	**Yannis Pousset**	Professeur, XLIM, Université de Poitiers
Membre	**Abbas Bradai**	Maître de conférences, XLIM, Université de Poitiers
Membre	**Jean-Pierre Cances**	Professeur, XLIM, Université de Limoges

THÈSE

Pour l'obtention du grade de

DOCTEUR DE L'UNIVERSITÉ DE POITIERS

Faculté des Sciences Fondamentales et Appliquées

Ecole Doctorale : Sciences et Ingénierie des Systèmes, Mathématiques, Informatique

Domaine de recherche : Traitement du Signal et des Images

Présentée par

Samir DAWALIBY

Communications machine à machine (M2M) dans les réseaux de nouvelles générations : Gestion de spectre et efficacité énergétique.

Machine-to-Machine (M2M) Communications in Next Generation Networks: Spectrum management and energy efficiency.

Directeur de thèse : **Yannis POUSSET**
Co-directeur de thèse : **Abbas BRADAI**
Soutenue le 11 Octobre 2019 devant la Commission d'Examen composée de :

M. Toufik AHMED	Professeur, LaBRI	Rapporteur
M. Laurent CLAVIER	Professeur, IEMN	Rapporteur
M. Jean-Pierre CANCES	Professeur, XLIM	Examinateur
M. Jean-François DIOURIS	Professeur, IETR	Examinateur
M. Abbas BRADAI	Maître de conférences, XLIM	Co-directeur de thèse
M. Yannis POUSSET	Professeur, XLIM	Directeur de thèse

Acknowledgments

When first arriving to Poitiers, i was about to start my final internship project before getting my engineering diploma. I never thought that this internship would have led to a long fruitful PhD journey full of ups more than downs. What's for sure is that i will never forget this opportunity which was wonderful, overwhelming and full of lessons and responsibilities. Thankfully, my PhD was completed successfully in which we delivered many journals and international conferences. This wouldn't have happened without the aid and support of countless people over the past three years.

First of all, I am deeply grateful to my advisors Professor Yannis POUSSET and Dr. Abbas BRADAI who gave me this opportunity and believed in me from the very beginning. Knowing that this PhD experience wasn't always easy, it was always a pleasure, with a lot of excitement, to be up to the challenge and to beat papers deadlines. Working with you improved me a lot as a student, as a researcher and as a person. It wasn't straightforward for me to understand how to think as a researcher and not as an engineer. You have taught me, both consciously and unconsciously, how good work is done. I appreciate all your contributions of time and ideas to make my PhD experience productive. You have always listened to my ideas and discussions with you frequently led to progress. Your ability to approach research problems and your high scientific standards set an example. I admire your ability to balance research interests and personal pursuits. I am thankful for the excellent example you have provided me as a successful and an ambitious researcher.

I would like to thank all the members of the jury of my thesis. I thank Professors Toufik AHMED and Laurent CLAVIER for finding the time to read my thesis and for their valuable feedback, and Professors Jean-Francois DIOURIS and Jean-Pierre CANCES for being examinators of this jury.

I would like to thank all XLIM laboratory, but one special thanks goes to Dr. Rita ZROUR, for always being there for me during hard times and for your nonstop motivation. Above all, you made me feel like a member of your family and a little brother, which I appreciate from my heart. In addition, I have been very privileged to get the support from many other great people who became friends over the last several years. Devendra VYAS, it has been a pleasure to work with you. Your excellent knowledge in the world of computer programming and mathematics had a great impact on my work and on the fast progress that I have achieved. Elissa EL RASSY, we started our PhD together side by side and we motivated each other through ups and downs, thank you for always being here for me and i'm very proud that i gained a trustful friend like you. I'm very glad that we successfully finished this journey and looking forward to attend your PhD defense and to call you Dr. El Rassy.

One special thanks goes to Mr. Roberto DUAILIBI, your continuous help and support were the main reason for starting my higher education journey in France. Without you this wouldn't have been possible and i hope i made you proud with all these degrees achieved in those five years.

Another special thanks goes to Dr. Bachir HABIB, you have been a true inspiration and a role model to follow. Your continuous support and your exceptional success inspired me and gave me countless words of encouragement and motivation to prove myself and to arrive that far in my educational and my professional career.

I would like to thank my parents: Fouad and Katia. You gave me my name, you gave me my life, and everything else in between. I pride myself in having words for everything, but you truly shut me up when it comes down to describing how much I love you and appreciate the efforts you have put into giving me the life I have now. You are the reason I did this; you are the reason I thrive to be better. The only thing I aspire for is that you will always think: "I am proud of my son". Thank you, thank you, thank you.

I thank my dear sister Sabine and my dear brother Samer for their illimitable love. Your encouragement and your faith in me gave me strength to achieve my goal. And finally, I thank my uncle Elie Haddad for all his motivation. With every skype call we made together, you encouraged me and I feel that your huge faith in me exceeds in rounds what I am today.

‘ “““““““‘

Abstract

The massive growth of the Internet of Things (IoT) highly impacts current wireless networks due to its heterogeneous quality of service (QoS) requirements. This study deals with the problem of guaranteeing QoS for IoT devices that require urgent and reliable communications. To achieve this objective in next generation IoT networks, various solutions are proposed in this thesis towards improving spectrum management and energy efficiency using emerging paradigms such as network slicing and software defined networking (SDN). First, network slicing is implemented over LoRa Wide Area Networks (LoRaWAN) and its assets are evaluated using various static and dynamic slicing strategies. Simulations performed over NS3 simulator have shown the efficiency of guaranteeing QoS for urgent communications which will always find isolated End-to-End (E2E) physical resources for its slice and cannot be impacted by the interference with less critical communications. Motivated by these results, a slice-based optimization is proposed next to improve the dynamic slicing strategy by investigating more LoRa parameters at the physical layer. The proposed method finds for each device the best parameters configuration that potentially improves the performance of its slice in terms of QoS and energy efficiency. Moreover, we also looked towards meeting upcoming challenges in future IoT networks that comes from the increasing number of IoT devices. Even with network slicing, LoRa scalability remained as a big challenge that should be carefully considered specially due to the lack of flexibility in managing current wireless networks. Therefore, to meet the global objective in guaranteeing QoS in large scale IoT deployments, SDN and network slicing are adopted as backbones for a novel distributed architecture and slicing strategy. The latter proposition is based on game theory and adapts faster to the changes in a massive IoT environment by leveraging slicing decision closer to the edge.

Résumé

L'ÉVOLUTION massive de l'internet des objets (IoT) a un impact important sur les réseaux sans fil actuels en raison de ses exigences hétérogènes en qualité de service (QdS). Dans cette étude on s'intéresse aux différentes techniques permettant le maintien de la QdS pour les dispositifs IoT qui exigent des communications urgentes et fiables. Pour atteindre cet objectif dans la nouvelle génération des réseaux IoT, différentes solutions sont proposées dans cette thèse pour améliorer la gestion du spectre et l'efficacité énergétique en utilisant des nouvelles approches telles que le découpage en réseaux virtuels (Network Slicing) et l'approche de séparation du plan de données et du plan de contrôle (SDN). Premièrement, le découpage du réseau en tranches virtuelles (slices) est mis en œuvre sur un réseau LoRa étendu à longue portée (LoRaWAN). Ensuite, ses avantages sont évalués à l'aide de diverses stratégies de découpage statique et dynamique. Les simulations effectuées sur le simulateur NS3 ont montré l'efficacité du découpage du réseau en matière de QdS. Les noeuds ayant besoin d'envoyer des informations urgentes ont toujours trouvé des ressources isolées pour leur slice et n'ont pas été affectées par l'encombrement provoqué par des communications moins critiques en délai. Motivés par ces résultats, une optimisation dans chaque slice est ensuite proposée pour améliorer la stratégie de découpage dynamique en recherchant davantage à bien configurer les paramètres LoRa de la couche physique. La méthode proposée trouve pour chaque appareil la meilleure configuration susceptible d'améliorer les performances de ses slices en matière de QdS et d'efficacité énergétique. En outre, nous avons également envisagé de relever les futurs défis liés à la croissance du nombre des dispositifs IoT connectés. Même avec le découpage de réseau, le passage à l'échelle avec Lora demeurait un défi à prendre en compte en raison du manque de flexibilité dans la gestion des réseaux sans fil actuel. Par conséquent, pour atteindre l'objectif global consistant à garantir la QdS dans un réseau IoT à grande échelle, le découpage du réseau en slices virtuels et SDN sont adoptés comme éléments principaux afin d'arriver à implémenter une stratégie de découpage et une optimisation distribuée. Cette dernière proposition est basée sur la théorie des jeux et s'adapte plus rapidement aux changements d'un environnement IoT massif en appliquant l'approche de découpage à la périphérie du réseau.

Introduction Générale

Avec le développement des réseaux sans fil de cinquième génération (5G), on s'attend d'ici 2022 à voir s'établir une connexion efficace entre les humains et les machines afin de garantir la flexibilité nécessaire pour gérer des réseaux avec des besoins en qualité de service hétérogènes. Actuellement, les opérateurs font face à divers défis et challenges lors du déploiement des communications IoT à travers les réseaux existants. Inévitablement, l'augmentation du nombre de dispositifs IoT connectés pose des problèmes de charge et de congestion et auront un impact important sur les systèmes de communication sans fil en usage actuels. Les dispositifs IoT nécessitent principalement une batterie de longue durée de vie, une portée étendue, une capacité plus grande pour prendre en charge des millions de dispositifs avec un coût de déploiement faible. Pour répondre à ces caractéristiques, plusieurs technologies ont été proposées et développées pour assurer la meilleure efficacité des réseaux étendus à faible consommation énergétique (LPWAN).

L'objectif du chapitre I est de présenter les technologies LPWAN existant actuellement sur le marché. Une partie de ces technologies fonctionne dans un spectre de fréquences sous licence (LTE-M et NB-IoT) tandis que l'autre communique via un spectre de fréquences libre (LoRa et Sigfox). Nous avons commencé par évaluer la performance du protocole LTE-M à travers plusieurs simulations sur NS3 [26]. En se basant sur cette étude, nous avons également proposé par la suite une optimisation pour allouer les ressources dans un réseau LTE-M qui considère les contraintes d'énergie et de qualité de service pour chaque dispositif IoT [27][30]. La méthode proposée a beaucoup amélioré la durée de vie des noeuds et le pourcentage des noeuds qui ont respecté leurs contraintes en QdS mais nous avons était limité par le nombre de noeuds servis par une station de base LTE qui n'a pas pu dépassé 250 dispositifs servis du à la limitations des ressources radios LTE [75]. C'est pourquoi après avoir étudié et comparé les caractéristiques de chaque technologie, nous avons choisi de travailler sur LoRaWAN qui fonctionne sous un spectre de fréquences libre et supporte un réseau IoT plus condensé (des milliers de dispositifs IoT peuvent être simulés dans une seule cellule au lieu d'une centaines dans un réseau LTE-M). LoRa commence à être de plus en plus répandu vu son accessibilité basée sur un code source ouvert contrairement à Sigfox qui est plutôt une technologie propriétaire. Ensuite, nous mettons l'accent sur les derniers travaux de recherche visant à optimiser les communications IoT et la gestion des ressources à l'aide des nouvelles technologies qui assurent la virtualisation et la programmabilité des réseaux IoT. Cependant, en partant de l'état de l'art, il y a eu peu de travaux de recherche qui se sont focalisés sur la QdS des communications IoT en termes de respect des délais critiques, de garantie d'un certain débit, et d'un taux de réception des paquets élevé. Ces différents réseaux logiques sont appelés des slices du réseau dont chaque slice correspond à un réseau virtuel de bout en bout entre un noeud IoT et un service réseau en s'appuyant sur la même infrastructure réseau physique. Étant donné que le nombre de périphériques IoT connectés augmente rapidement avec le temps, une solution efficace pour garantir la qualité de service consiste à apporter de la flexibilité et une virtualisation des réseaux IoT à l'aide de SDN et du découpage en slices. Cette QdS sera garantie en favorisant les communications urgentes, et une gestion flexible du réseau divisé en plusieurs réseaux virtuels configurés et gérés séparément. Pour chaque slice, une partie des ressources physiques est réservée de bout en bout sur toutes les couches (accès réseau, coeur et cloud) pour répondre aux besoins QdS des applications urgentes et fiables. Dans cette thèse, afin de pouvoir garantir cette QdS pour les communications IoT, il va falloir tout d'abord répondre aux questions suivantes:

- Comment affecter les noeuds IoT aux slices et comment classifier ces slices dans LoRaWAN ?
- Comment réserver les ressources physiques LoRa pour chaque slice et à l'intérieur de chaque slice, comment assurer une allocation optimisée des canaux ?
- Quels sont les paramètres qui impactent la qualité de service de chaque dispositif LoRa et comment optimiser la configuration sans augmenter la complexité du réseau et sans impacter sa performance ?
- L'architecture actuelle de LoRaWAN sera-elle capable de supporter l'utilisation à grande échelle des communications, et comment pourra t-elle suivre les avancées à venir ?

Dans le chapitre II, nous répondons aux deux premières questions en proposant tout d'abord des nouvelles méthodes d'affectation des noeuds aux trois slices définis à la fin du chapitre I. La première étape consiste à associer et détacher un noeud IoT d'une tranche de réseau durant chaque intervalle du temps. Ce mécanisme est réalisé dynamiquement avec une méthode basée sur l'algorithme BIRCH qui regroupe les noeuds selon leur condition QdS, notamment en se basant sur le taux d'urgence défini par le rapport entre le délai instantané et le seuil maximal de délai à ne pas dépasser. Le résultat de cette première partie est un groupe des noeuds associés aux tranches de réseaux dont chacun doit répondre au maximum au besoin des noeuds en délai, en débit et en bonne réception des paquets envoyés. Ensuite, dans la deuxième étape, nous supposons que le serveur LoRa a une vue globale sur le réseau et le besoin de chaque noeud en termes de débit. En se basant sur cette information, le serveur appliquera une estimation sur ces noeuds et réserve les ressources à travers deux stratégies dynamiques:

- La première est nommée *"Dynamic Slicing"* qui estime le besoin de tous les noeuds du réseau appartenant à chaque slice en commençant par le slice le plus urgent et définit le pourcentage des canaux qui vont réserver pour chaque slice. Ce résultat sera appliqué sur toutes les passerelles LoRa de la même manière mais change dynamiquement à chaque intervalle du temps.

- La deuxième stratégie est également dynamique mais plus adaptée pour chaque passerelle, d'où son nom *"Adaptive Dynamic Slicing"*. De la même manière, le serveur commence par le slice plus urgent, mais l'estimation des besoins des noeuds et la réservation sont appliquées individuellement sur chaque passerelle. Le serveur regardera les noeuds qui sont dans la marge de couverture de chaque passerelle, estimera le pourcentage des canaux à réserver pour ce slice et passera ensuite au slice suivant.

Ces deux stratégies dynamiques sont comparées par la suite à une troisième stratégie statique nommée *"Fixed Slicing"* qui réserve les canaux pour chaque slice d'une manière équitable. Ensuite, l'algorithme classifie les noeuds dans chaque slice selon leur niveau d'urgence et commence à associer chaque noeud à la passerelle qui répond le mieux à ses besoins en QdS. Une valeur d'utilité est calculée différemment pour chaque slice en considérant la puissance de réception et le taux de congestion. Suivant la valeur calculée, le choix du canal est réalisé d'une manière qui garantit au maximum la bonne réception du paquet envoyé.

Nous avons prouvé dans les résultats que l'isolation entre les slices du réseau est totalement assurée. En effet, en augmentant le nombre des noeuds, le slice le plus urgent n'a pas été impacté par le trafic envoyé par les membres des autres réseaux virtuels. L'étude paramétrique de l'architecture montre que la meilleure stratégie de configuration d'un noeud LoRa est la méthode dynamique. Cette dernière configure le facteur d'étalement pour un noeud IoT suivant la puissance de réception de la passerelle LoRa la plus proche. En utilisant cette configuration, l'évaluation de performance de chaque stratégie de réservation de ressources a eu lieu. Ces résultats montrent que la stratégie *"Adaptive Dynamic Slicing"* était la plus efficace pour réserver les ressources de chaque slice. Par contre, quand le nombre de noeuds augmente, le slice "Best Effort" (BE) perd plus de 50% du nombre de paquets envoyés par ses membres. Pour améliorer ces résultats, nous avons décidé d'explorer plus en détail les paramètres de configurations de chaque noeud, notamment le facteur d'étalement et de la puissance de transmission, pour améliorer les résultats.

Dans le chapitre III, nous avons retenu la meilleure stratégie *"Adaptive Dynamic Slicing"* et nous sommes focalisés plus ardemment sur comment configurer les dispositifs LoRa pour améliorer leur performance en tenant en considération les caractéristiques du slice du réseau auxquelles ils appartiennent. Nous avons proposé une méthode de configuration qui est adaptée pour chaque slice dont ses membres seront configurés avec un coût différent qui inclue plusieurs objectifs. Cette méthode évite les interférences et augmente la probabilité que le paquet sera bien reçu et décodé au niveau de la passerelle vis à vis des autres paquets reçus simultanément. Dans chaque slice, l'algorithme cherche pour chaque dispositif LoRa, la configuration optimale et la bonne combinaison du facteur d'étalement et la puissance de transmission au lieu d'utiliser le mécanisme ADR qui oblige la configuration d'un noeud par l'une des distribution de la **Table I.4**. Les résultats montrent une amélioration majeure en termes de qualité de service et de consommation énergétique lorsque chaque appareil est configuré avec une configuration optimisée du facteur d'étalement et de la puissance de transmission. Plus précisément, le taux de perte de paquets diminue de 50% (à moins de 30%) pour le même nombre de périphériques IoT. Cependant, il est prévu que le nombre de périphériques augmente avec le temps et dépasse la capacité maximale qui peut être supportée actuellement par une passerelle ce qui va à son tour augmenter le taux de perte des paquets dans les différents slices du réseau. Ce challenge nous a motivé à chercher une réponse et trouver une solution pour réaliser le passage à l'échelle et garantir la QdS et l'efficacité énergétique des dispositifs LoRa.

Après avoir examiné dans le chapitre II les avantages que le découpage en plusieurs slices virtuels apporte à LoRaWAN pour garantir la qualité de service des dispositifs IoT en termes d'urgence et de fiabilité, nous avons amélioré les résultats dans le chapitre III à travers une distribution optimisée des paramètres de la couche physique. Toutefois, des estimations existent qui prévoient 20 à 30 milliards d'appareils IoT connectés d'ici 2022. L'infrastructure actuelle du réseau LoRa ne sera pas capable de prendre en charge les défis à venir lors du déploiement de ce grand nombre de dispositifs IoT. Par conséquent, dans le chapitre IV, nous proposons une architecture basée sur SDN qui répond aux challenges d'évolutivité en traitant les données et la prise de décisions au niveau des passerelles LoRa. Dans ce contexte, c'est chaque passerelle LoRa qui va définir sa stratégie de découpage en slices et l'affectation de ressources qui doivent être réservées après une phase de coordination avec les passerelles voisines. En se rapprochant de la périphérie du réseau, le contrôle des noeuds sera plus flexible à travers SDN et améliorera la fiabilité des communications dans un environnement IoT encombré.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

3GPP	:	3rd Generation Partnership Project 2
5G	:	Fifth Generation vii
ADR	:	Adaptive Data Rate x, xxi, 14, 79
ADS	:	Adaptive Dynamic Slicing 40
API	:	Application Programming Interface 16
BE	:	Best Effort ix, 20, 50, 79
BIRCH	:	Balanced Iterative Reducing and Clustering method using Hierarchies viii, xxi, 25
BS	:	Base Station 5
BW	:	Bandwidth 12
CaF	:	Carrier Frequency 12
Cat-M	:	Categories for Machines 2
CF	:	Clustering Feature 29
CR	:	Coding Rate 14
CRC	:	Cyclic Redundancy Check 10
CSS	:	Chirp Spread Spectrum 10
DA	:	Dynamic Adaoptive 36, 60
DBPSK	:	Differential Binary Phase Shift Keying 9
DR	:	Dynamic Random 36, 60
DRX	:	Discontinuous Reception 5
DS	:	Dynamic Slicing 40

E2E	:	End-to-End v
EC	:	Error Correction 14
eMBB	:	enhanced Mobile Broadband xx
eNB	:	eNodeB 5
EPA	:	Amplifier Added Efficiency 36
EPC	:	Evolved Packet Core 5
FDPS	:	Frequency Domain Packet Scheduling 7
FEC	:	Forward Error Correction 10
FP7	:	Seventh Framework Program 2
FS	:	Fixed Slicing 40
GMM	:	Geometric Mean Method xxi, 49
GSM	:	Global System for Mobile communications 4
GW	:	Gateway 5
H2H	:	Human-to-Human 2
IoT	:	Internet of Things v, vi
LoRa	:	Long Range vii, 10
LoRaWAN	:	Long Range Radio Wide Area Network v, vi, 9, 10
LPWAN	:	Low Power Wide Area Network vii, 1
LS	:	Linear Sum 29
LTE-M	:	LTE for Machines vii, 2
M2M	:	Machine-to-machine xx
MAC	:	Medium Access Control 10
MCE	:	Management and Control Entity 28
MLE	:	Maximum Likelihood Estimation 25, 49, 56
mMTC	:	massive Machine-Type Communications xx
NB-IoT	:	Narrowband Internet of Things vii, 2
NFV	:	Network Function Virtualization xx
NS	:	Network Slicing 15
NS3	:	Network Simulator 3 8

P2P	:	Point-to-Point 9
PDB	:	Packet Delay Budget 29
PDR	:	Packet Delivery Ratio 15
PRB	:	Physical Resource Block 4
QdS	:	Qualité de Service vi
QoS	:	Quality of Service v, xx
RA	:	Reliability Aware 19, 50, 79
RBs	:	Resource Blocks 5
RF	:	Radio Frequency 14
Rx	:	Reception 36
SDN	:	Software Defined Networking v, vi, 15
SF	:	Spreading Factor xxi, 11
SINR	:	Signal to Interference Noise Ratio 13
SNR	:	Signal to Noise Ratio 2
SS	:	Square Sum 29
TDPS	:	Time Domain Packet Scheduling 7
TOA	:	Time on Air 14
TOPG	:	TOPSIS and GMM 49, 60
TOPSIS	:	Technique for Order of Preference by Similarity to Ideal Solution xxi, 49
TP	:	Transmission Power xxi, 12
TTI	:	Transmission Time Interval 5
Tx	:	Transmission 36
UEs	:	User Equipements 2
UNB	:	Ultra Narrow Band 9
URA	:	Urgency and Reliability Aware 19, 50, 79
URLLC	:	Ultra Reliable Low Latency Communications xx

General Introduction

In the era of the internet of things, it is forcasted that the number of machine-type devices will grow and reach 30 billions IoT devices connected by 2022 [36]. Thus, it is expected that machine-to-machine (M2M) communications will pose important challenges on current wireless communications over both licensed and unlicensed frequency bands. To reach an all connected world of humans and machines, network flexibility is needed to be able to manage each network easily with each having conflicting quality of service (QoS) requirements (i.e., ultra-reliable low-latency communications (URLLC), enhanced mobile broadband (eMBB) and massive machine-type communications (mMTC). This objective is possible in next generation networks using major arising technologies namely network functions virtualization (NFV), SDN and network slicing. The present thesis directs towards improving spectrum management and energy efficiency, defined as mandatory requirements, to enable QoS guarantees for IoT communications in low power wide area networks.

Before achieving the objective defined above, advanced and most popular LPWAN technologies are surveyed in Chapter I. Each LPWAN technology operates either in licensed or unlicensed frequency bands. We first review latest research works related to spectrum management in cellular IoT networks using technologies such as LTE-M and NB-IoT. The latter have smaller capacity than technologies operating in unlicensed spectrum which made us also look towards the latest contributions on spectrum management and optimizations proposed for SigFox and LoRa wireless access networks. However, the fact that SigFox is a proprietary technology drove us to choose LoRaWAN, being an alliance with an open approach. To guarantee QoS in LoRaWAN network slicing, IoT service types are divided into three virtual slices with each having different requirements in terms of slicing priority, delay, throughput, and reliability.

In Chapter II, network slicing is implemented in LoRaWAN and its efficiency is investigated over various slicing strategies. The first step involves associating and detaching an IoT node from a network slice during each time interval. We propose to dynamically realize this mechanism with a method based on BIRCH algorithm which groups the nodes based on the urgency rate computed as the ratio between the instantaneous packet delay and the maximum delay threshold that should not be exceeded. The result of this first part is a group of nodes associated with the virtual network slices. The second step is to find an optimal resource reservation strategy which reserves physical channels on LoRa gateways assuming that LoRa server has a global view of the network and the need of each node in terms of throughput. Here, two dynamic slicing strategy are proposed based on maximum likelihood estimation which dynamically adapt to throughput requirements of each slice members at each slicing time interval. After that, the third and final step is to allocate channel resources inside each virtual slice by classifying IoT devices according to the utility value computed based on delay urgency and congestion rate. Finally, the server allocates each LoRa device to the channel providing the highest utility value. Results have shown the utility of virtual network isolation to avoid network performance degradation that comes from the increasing number of IoT devices in a slice over another.

Although network slicing improved QoS results, there was still a room for improving reliability results in LoRa network slices by focusing more on improving devices configuration parameters at the LoRa physical layer. In chapter III, a multi-objective optimization method for LoRa devices configuration is proposed based on TOPSIS and GMM methods. The latter takes into account QoS requirements of each device and configures its parameters accordingly. In each slice, the algorithm searches for the optimal combination of SF and TP configuration for each LoRa device instead of using the ADR mechanism which forces the configuration of a node by one of the distributions listed in **Table. I.4**. The chosen configuration for a device avoids interference and increases the probability of successfully receiving and decoding the packet at the gateway while taking into consideration the other packets received simultaneously. The results show a major improvement in terms of QoS and energy consumption however, it is expected that the number of devices will increase over time and will exceed the maximum capacity that can be currently supported by a gateway which will in turn increase the rate of loss of packets in different network slice. This challenge motivated us to look for an answer and find a solution to meet scalability challenges and guarantee QoS and energy efficiency for LoRa devices.

Despite the improvement achieved in the results obtained using previous contributions and knowing that more than 30 billion IoT devices are estimated to be connected in future generation IoT networks by 2022, scalability remained an important challenge for next generation IoT networks. In a LoRa network slicing scenario, the current centralized architecture of the state of the art will not be able to handle the challenges of network resource management coming ahead in large scale future IoT deployments. Therefore, in Chapter IV, we propose an SDN-based architecture that addresses scalability challenges by processing data and decision-making at the edge of the network. The role of slicing decision making and network configuration is leveraged to the gateway which defines the slicing strategy and the resources that must be reserved after a coordination phase with the other nearby gateways. By getting closer to the network edge, combining network slicing with SDN advantages improve the reliability of communications in a large scale network due its rapid adaptation to changes in a crowded IoT environment.

Chapter I

Low Power Wide Area Network Background

Summary

I.1 Introduction

For the last couple of years, operators needed to address various challenges and complexities in deploying IoT communications within legacy networks. Inevitably, the expected increase in the number of IoT devices causes saturation problems and will have a large impact on current wireless communication systems. IoT devices mainly require long battery life, extended coverage, larger capacity to support billions of devices with low device and deployment cost. Driven from these requirements, various technologies appeared as potential solutions for IoT network deployment. The purpose of this chapter is to introduce the most emerging technologies nowadays for low power wide area networks (LPWAN). We especially focus on the latest research efforts that optimized IoT communications and resource management using emerging technologies that provides virtualization and network softwarization.

I.2 Licensed Spectrum Technologies

In order to meet IoT requirements over cellular networks, various efforts were conducted by standardization organizations and working groups to expand current human-to-human (H2H) communications over LTE for machines (LTE-M), introduced by many cellular operators and companies such as Nokia [82], Ericsson [37] and Qualcomm [111]. However, EXALTED [22] was the first project of the European Union's Seventh Framework Program (FP7) to present LTE-M as a new system that extends LTE specifications and supports future wireless systems with M2M and H2H communications coexistence.

The 3rd Generation Partnership Project (3GPP) has also worked on a group of enhancements in M2M communications over LTE and LTE-A networks [98]. In its Release 12 and 13, 3GPP introduced many categories adapted for M2M communications (Cat-M) that operate in half and full duplex modes. In that sense, Cat-0 was proposed as the first technology in Cat-M which supports lower features in terms of cost, throughput, number of antennas and includes power saving mode enhancements for user equipments (UEs). IoT devices such as sensors and smart meters tend to have much smaller data messages to send and do not need high speed or large bandwidth. Therefore, in its Release 13, 3GPP introduced LTE-M and Narrow-band IoT (NB-IoT) as the two newest technologies to support low-throughput M2M communications. LTE-M provides 1 Mbit/s as downlink and uplink throughput whereas NB-IoT provides few kbit/s throughput due its smaller bandwidth (200 kHz) compared to the one reserved for LTE-M (1.4 MHz). Both technologies operate in much lower signal-to-noise ratio (SNR) than conventional LTE. In Release 14, CaT-M2 is proposed as a new enhancement that supporting 5 MHz bandwidth and higher peak data rates for LTE-M devices. Afterwards, Release 15 came up to enhance coverage for higher devices velocity (200 km/h), to specify new power class (14 dBm) and propose new techniques, such as wake-up signal and relaxed monitoring for cell reselection, to reduce latency and improve spectral efficiency [92]. In **Table I.1**, the licensed technologies for IoT, as well as their respective research works and optimizations, are summarized and described in details in the below subsections.

	Rel. 8	Rel. 12	Rel. 13	Rel. 13	Rel. 14
	Cat-4	Cat-0	LTE-M	NB-IoT	CAT-M2
Bandwidth	20MHz	20MHz	1.4MHz	200kHz	5MHz
DL Rate	150Mbit/s	1Mbit/s	1Mbit/s	25.5kbit/s	4Mbit/s
UL Rate	50Mbit/s	1Mbit/s	1Mbit/s	62,5kbit/s	6Mbit/s
Duplex	Full Duplex	Full Duplex	Half Duplex	Half Duplex	Half Duplex
Number of Antennas	2	1	1	1	1
Tx Power	23dBm	23dBm	20dBm	23dBm	23dBm

Table I.1: LTE Category-M technologies [37] [92]

I.2.1 NB-IoT

NB-IoT [21] [121] is proposed as a new technology that offers ultra-low cost, low energy consumption, low delay sensitivity, low response time and enhanced network architecture [80]. It enables the opportunity to deliver new network services for LPWAN communications. Using its 200 kHz bandwidth on licensed spectrum, NB-IoT offers 20 dB coverage, 200 kbit/s maximum throughput and more than 15 years operation on a single battery charge while keeping the same level of security as LTE [61] without modifying the existing cellular network infrastructure. **Figure I.1** illustrates the three operational scenarios that can be adopted for NB-IoT as follows:

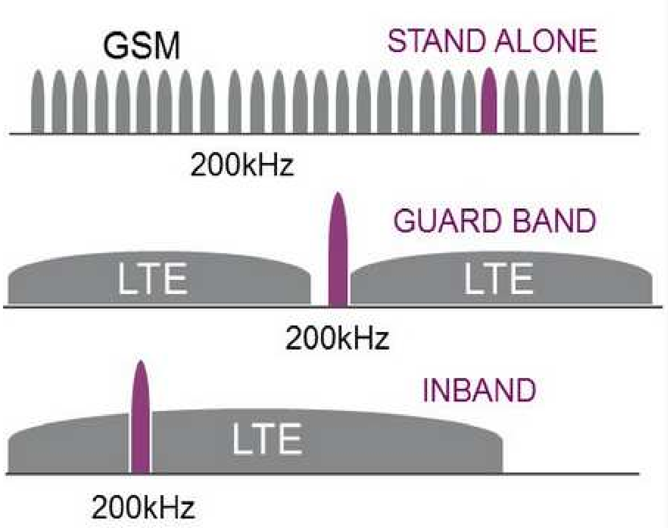


Figure. I.1: Operation modes in NB-IoT [59]

- **Stand alone:** NB-IoT operates as a dedicated carrier where it can occupy one exclusive GSM channel (200 kHz).
- **Guard-band:** NB-IoT operates within the guard band of an existing LTE carrier and uses one physical resource block (PRB) of LTE (200 kHz).
- **Inband:** NB-IoT operates withing the bandwidth of a wideband LTE carrier and uses one PRB of LTE (200 kHz).

Multiple works evaluated the performance of NB-IoT in real case IoT scenarios. In [9], authors developed a testbed that allows NB-IoT devices to repeat signal transmissions operating at very low power to boost the received signal quality. The coverage gain that results from this method was analyzed using real life measurements and was shown to be limited by the channel estimation quality and coherence time. However, transmission signal repetitions increase the energy consumption and the latency in the whole NB-IoT system. Therefore, authors in [7] searched for an energy-latency tradeoff based on channel scheduling through derivation of the expected latency and battery lifetime for each coverage class in NB-IoT systems. In [60], authors analyze the performance of

NB-IoT and the improvement that it provides in terms of coverage and capacity for IoT communications in rural area. For deep indoor communications, authors find NB-IoT more suitable than LTE-M because it provides coverage for more than 95% of the devices due to its Maximum Coupling Loss being 164 dB as compared to LTE-M's 156 dB. However, NB-IoT causes some coverage issues upon NB-IoT deployment due to both large path loss and interference [74]. The performance of NB-IoT is compared to LTE-M for smart city applications, authors find that a battery life time of 8 years can be achieved using both technologies. However, authors show that more devices can be served in an LTE-M network than NB-IoT, while providing a 10 times lower latency [35]. For this reason, we pursued our research studies in this thesis using LTE-M because we didn't want to be limited to indoor IoT applications, we wanted to focus more on a large scale IoT network scenario that includes a massive number of IoT devices.

I.2.2 LTE-M

LTE-M operates on a very small bandwidth and only monitors 6 resource blocks (RBs) per subframe with a coverage enhancement of 15 dB with respect to LTE Release 12. LTE-M is also characterized with an extended Discontinuous Reception (DRX) cycle for both idle and connected mode in order to enable further power savings from the radio perspective. Hence, by increasing DRX cycle from 2.56 seconds to a maximum value of 43.69 minutes, M2M communications with low duty cycle will be efficiently supported [98]. The network architecture of LTE-M is illustrated in **Figure I.2**, uplink traffic originating from IoT devices running various applications can be directly transmitted to the eNodeb (eNB) or through an IoT gateway (GW) that helps the device to store and relay their data to the eNB to increase transmission efficiency. Unlike legacy LTE networks, LTE-M limits the amount of resources that can be allocated to one RB per device in each transmission time interval (TTI). Hence, scheduling decisions are taken differently based on the implemented scheduling algorithm. LTE-M network allows data traffic from the base station (BS) to be routed to the remote IoT server through the evolved packet core (EPC) which includes management and signaling functions of LTE.

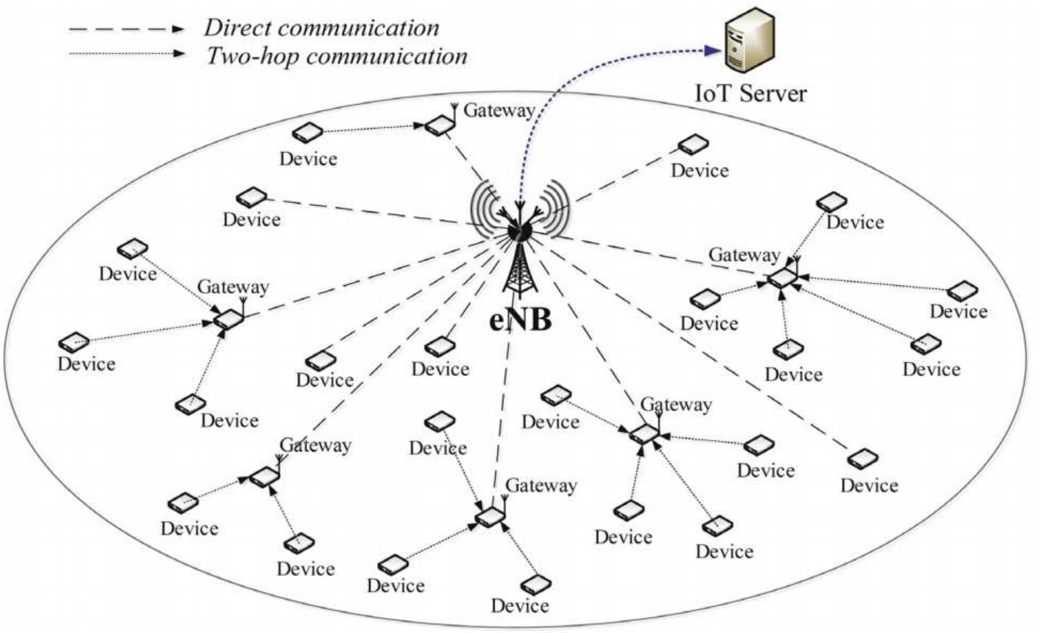


Figure. I.2: LTE-M network architecture [66]

The advantages that LTE-M brings for IoT communications are exploited in various use cases including delay-sensitive applications (ex: instance presence sensors and actuators for emergency alerting) and delay-tolerant applications (ex: temperature monitoring). We first evaluated in depth the performance of LTE-M [26] in terms of coverage and QoS and highlighted its advantages for M2M communications. In [128], LTE-M system performance is evaluated based on field test results in Beijing ring rail lines. Results show the potential that LTE-M provides in terms of reliability and QoS for specific IoT services. Moreover, due to the large variety of M2M applications and its heterogeneous interconnection in the network, a suitable resource access scheme for M2M communications is needed. Authors in [129], proposed a new random access procedure based on the time slot-ALOHA mode of operation to reduce the power consumption of UEs. Nonetheless, research work mainly focused on improving M2M/H2H coexistence to reduce the impact of M2M nodes on H2H communications using the power control scheme proposed in [52]. Moreover, M2M/H2H coexistence can be also improved using resource control in [119] where a Markov model of dynamic resource scheduling is proposed in an LTE cell where M2M transmissions arrive according to a general Markovian arrival process. Furthermore, authors in [3] came up with two schedulers that consider H2H traffic with a periodic M2M traffic produced by smart metering devices only. In [34], authors propose an adaptive scheme that manages LTE-M network resources and avoids fast resource depletion for M2M communications in emergency scenarios.

Nonetheless, various scheduling techniques tackled the QoS of mobile applications such as voice and video call, but did not also consider the QoS and energy consumption of M2M devices. The latter was tackled from different perspectives because M2M devices are battery-driven and should be able to work for long periods of time. In [6], a battery lifetime-aware resource allocation framework is proposed that provides battery lifetime-fairness while reducing maintenance costs of M2M over LTE networks. Additionally, other scheduling methods were also adopted to save energy whether by reducing the number of assigned RBs per device as proposed in [56], by reducing the transmission rate [102] or by reducing transmit power for reliable data transmission [130].

In LTE-M, it is possible that an IoT device reaches its delay limit with a required throughput higher than the one that can be provided through a single RB. Hence, it could be useful to allocate additional RBs for this device for proper scheduling optimization. This use case was not previously considered in LTE-M literature. Therefore, we proposed in this thesis a novel optimization algorithm that jointly provides QoS and energy optimization to IoT devices [30]. The proposed strategy has two-fold objectives:

- To minimize energy consumption by implementing a DRX switching mechanism and extending the DRX cycle length which should provide power savings by putting devices into sleep without affecting their QoS requirements.
- To consider the QoS of LTE-M devices where, depending on the running applications, a part of these devices could impose delay restrictions and need to be urgently served and prioritized in the proposed scheduling strategy.

The global overview of the proposition is illustrated in **Figure I.3** where the scheduler acts differently in both time domain (TDPS) and frequency domain packet schedul-

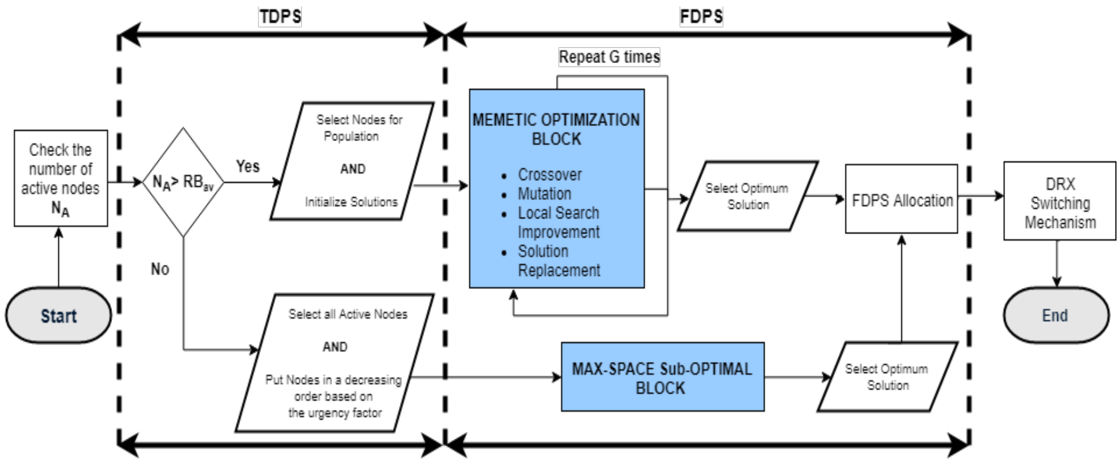


Figure. I.3: Global scheme of LTE-M scheduling algorithm [30]

ing (FDPS). In TDPS, active M2M devices having data packets in their buffer ready for transmission are firstly detected. The algorithm selects afterwards the ones with the best energy and channel conditions in order to reduce the search area and define the best group of devices. We distinguish two scheduling behaviors for FDPS: on one hand, when the number of active devices (N_A) is higher than the number of RBs available in each TTI (RB_{av}), a selection process is needed to define the number of candidate nodes selected for the memetic optimization block which cannot exceed the population size defined in input. Otherwise, all active devices will be selected for FDPS and a sub-optimal algorithm will be launched that leaves a maximum number of contiguous empty RBs for urgent devices with respect to contiguity constraints and gives extra RBs for LTE-M devices, if needed, following to their throughput requirements.

The proposed memetic algorithm is implemented into Network Simulator (NS3) platform and evaluated its performance in a realistic IoT scenario which took into account M2M traffic specifications in terms of infrequent and small packet data transmissions. After analyzing deeply the achieved results of each algorithm, we showed that the proposed scheduling scheme reduced the overall energy consumption in the system and achieved the highest percentage of satisfied devices following to their QoS requirements [30]. However, due to capacity limitations of LTE-M that cannot allow to serve more than 250 devices simultaneously transmitting uplink traffic to a single eNB [75], we decided to move our focus towards emerging unlicensed spectrum technologies, i.e SigFox and LoRaWAN, to be able to reach thousands of IoT devices served in a single cell.

I.3 Unlicensed Spectrum Technologies

Using its low-cost access to the airwaves, tech innovators took advantage of the unlicensed spectrum to propose promising technologies to support IoT communications. In this section, the focus will be on SigFox and LoRaWAN technical overview and their performance in real IoT scenarios.

I.3.1 Sigfox

Sigfox [107] is an ultra narrow-band (UNB) Differential Binary Phase Shift Keying (DBPSK) modulation technology operating on a very small channel bandwidth i.e, 100 Hz in Europe (on a band between 868 and 868.2 MHz) and 600 Hz in USA (on a band between 902 and 928 MHz). Sigfox uses 192KHz of the publicly available band by sending 3 messages using a random frequency to exchange messages over the air. For every transmission, a Sigfox device randomly uses one of the multiple channels available in a bandwidth with a packet duration that goes up to 2 ms [79]. This small bandwidth usage in Sigfox provides the opportunity to concentrate the energy in a very small channel making it very robust against interference. IoT devices transmit short messages in uplink as well as downlink with a throughput that varies between 100 to 600 bits per second depending on the region.

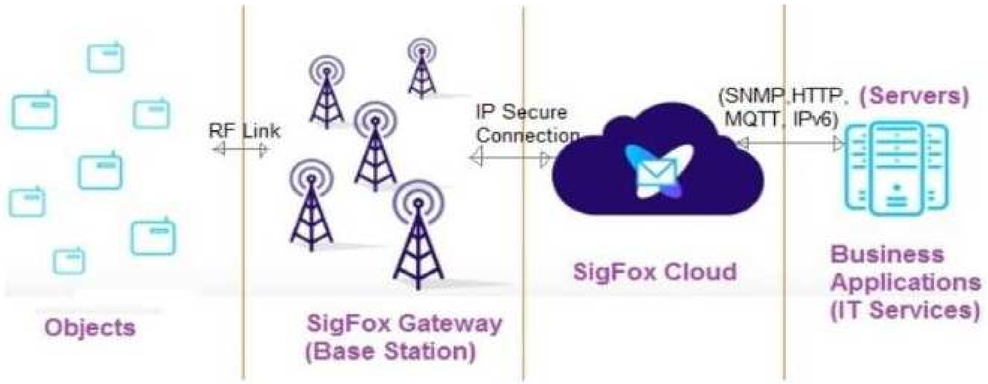


Figure. I.4: SigFox architecture [25]

The architecture of Sigfox network is illustrated in **Figure I.4**. It adopts a star topology where IoT devices transmit their messages to the Sigfox network when the radio signal sent reaches the BS within the range. A message can be received by multiple BSs deployed by Sigfox network operators which detect, demodulate and report the messages to the Sigfox Cloud using point-to-point (P2P) links. The Sigfox cloud then pushes the messages to many customer servers and IT platforms. The time on air of a packet is 6 seconds [118] where 6 messages can be transmitted per hour with a payload of 4, 8, or 12 bytes. However, in this thesis we are looking towards simulating a larger variety of applications with higher throughput requirements than the one provided by Sigfox. In addition, Sigfox protocol is proprietary which prevented scientists from working on this technology in their research studies. Therefore, we preferred to look towards the possibility of working in LoRa wide area networks (LoRaWAN) for this thesis contributions.

I.3.2 LoRaWAN

LoRa is a shortcut name for **Lo**ng **Ra**nge and a proprietary spread spectrum physical layer that derives from Chirp Spread Spectrum (CSS) modulation as described in the IEEE standard 802.15.4 [45]. CSS modulation transmits symbols by encoding them into multiple signals of increasing or decreasing radio frequencies making signals more robust to multi-path interference, Doppler shifts and fading [10]. Moreover, Forward Error Correction (FEC) and Cyclic Redundancy Check (CRC) techniques are also implemented in LoRa to improve receiver's sensitivity and the robustness of communications. Knowing that LoRa is proprietary and capable of communicating with any other Mediaum Access Control (MAC) layer, LoRa Alliance defines LoRaWAN MAC as an open source protocol built on top of LoRa physical layer. The former defines the communication protocol and system architecture for the network, whereas the latter enables the long-range communication link. LoRaWAN supports low-power and long-range communications where a set of $K = \{1, 2, ..., k\}$ IoT devices transmit directly to $M = \{1, 2, ..., m\}$ LoRa GWs in a star of stars topology before forwarding data to a backbone infrastructure. LoRa architecture is shown in **Figure I.5** where low throughput traffic is uploaded by each IoT device (thing) to the cloud application servers via IP networks. Within the backbone network, operators servers perform authentication, validation, and forward the packets to the application servers. The latter connects to the backbone network to receive the data and send back the packets in downlink via LoRa GWs.

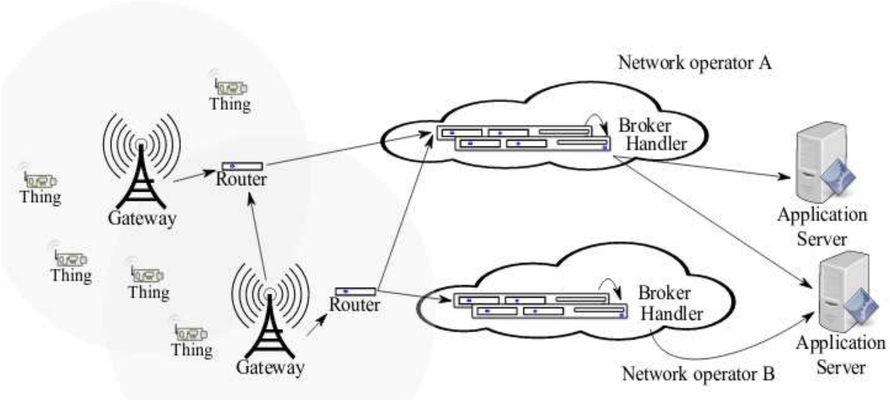


Figure. I.5: LoRaWAN architecture [10]

IoT communications are bidirectional where each LoRa device $k \in K$ is characterized with specific parameters that needs to be optimized to meet the requirements of each application in terms of coverage, achieved throughput and energy consumption. In the following we expound LoRa settings and their impact on network performance:

- **Spreading Factor (SF):** SF parameter is by definition the logarithm, in base 2, of the number of chips per symbol and impacts the duration of a Lora chip. Each device k adopts specific SF configuration for information transmission. LoRa spreads each symbol in a rate of 2^{SF} chips per symbol with $SF = \{7, ..., 12\}$ resulting a data rate computed as written in **Eq. I.1** below:

$$r_{k,c} = SF.\frac{R_{chip}}{2^{SF}} \quad bits/s \tag{I.1}$$

where R_{chip} denotes the chip rate and $r_{k,c}$ the data rate achieved by a device k on channel c of LoRa GW m. Depending on the transceiver model, SF configuration varies from 7 to 12 in a way that higher SF values correspond to more robust communications but lower data rates whereas lower SF values increase the rate and reduce the a active time on air. Some research works ([13] and [12]) claim that SFs are orthogonal to each other, whereas others [23] show that unperfect orthogonality happens between SFs leading to interference between packets. In this thesis, interference in LoRaWAN is considered and will be described in more details later in the interference section below.

- **Transmission Delay:** Transmission delay parameter $d_{k,c}$ denotes the transmission delay of a packet with a length of L bits uploaded by device k to one of the channels c that belongs to GW m.

$$d_{k,c} = \frac{L}{r_{k,c}} \quad seconds \tag{I.2}$$

- **Transmission Power (TP):** Transmission power parameter defines the energy consumed by an IoT device and can be set between -4 and 20 dBm with a step of 1 dB, however in LoRa configuration, TP values vary between 2 and 14 dBm.

- **Carrier Frequency (CaF):** Three different radio bands are available for LoRaWAN (137-175 MHz, 410-525 MHz and 820-1020 MHz). In this thesis, we work on European frequency bands where operators work in in the 863-870 MHz frequency band. Here, specific duty cycles are imposed on IoT devices by the European frequency regulations where each device transmits on a certain frequency in a way respected by both GWs and devices. LoRaWAN channels have a duty-cycle as low as 1% which means that during the last 3600 seconds, a device must never have transmitted more than 36 seconds in total.

Spreading Factor	Sensitivity (dBm)
SF7	-130.0
SF8	-132.5
SF9	-135.0
SF10	-137.5
SF11	-140.0
SF12	-142.5

Table I.2: List of parameters

- **Radio Bandwidth (BW):** Based on the transceiver model, operators may choose one of the 10 available bandwidth values that varies from 7.8 kHz to 500 kHz. European frequency regulations imposes that the bandwidth adopted for each channel is 125 kHz. Increasing this bandwidth improves the data rate of LoRa device on the expanse of sensitivity. Moreover, increasing the SF value configured on IoT device also reduces the transmitted data rate, increases the strength of the signal and offers a better sensitivity at the GW receiver as shown in **Table I.2**.

- **co-SF and inter-SF Interference:** LoRa GWs are unable to decode two packets if both are received on the same channel with the same SF configuration. This mechanism leads to packet loss of both packets due to **co-SF** interference. On the other hand, **inter-SF** collisions happen between two packets if they were simultaneously received on the same channel with different SFs and are shown to cause packet loss [23]. Signal to Interference Noise Ratio (SINR) varies based on SF configuration on each device. The assumptions in [73] are followed where a packet should survive interference that comes from other LoRa transmissions. Each device configured with $SF = i$ experiences a SINR value computed based on **Eq. I.3**:

$$SINR_{i,j} = \frac{P_i^{rx}}{\sigma^2 + \sum_{n \in \partial_j} P_n^{rx}} \tag{I.3}$$

where P_i^{rx} is the power of the packet i under consideration sent by device with $SF = i$, σ^2 the lognormal shadowing component and P_n^{rx} the power of one interfering packet $n \in \partial_j$ configured with $SF = j$. Each element in the **Table I.3** [44] denotes the minimum signal power margin threshold $V_{i,j}$, with $i, j \in \{7, ..., 12\}$, that a packet sent with $SF = i$ must have in order to be decoded successfully over every interfering packet with $SF = j$. Hence, packet survives interference with all interfering packets if, considering all combinations of SF, a higher power margin value (dB) is satisfied than the corresponding co-channel rejection value. One thing to note is that values in below matrix are not symmetric because the power needed to decode a packet is higher when a packet intercepts another one configured with smaller SF. This is because the smaller SF configuration, the stronger the signal power. Taking for example SF8 and SF10, a higher power is needed to decode packets if a packet configured with SF8 intercepts another configured with SF10

(30 dB). However, if the opposite case happened, a smaller power margin value (22 dB) will be needed to decode the SF8 packet intercepted by SF10 packet.

Desired Packet \ Interferer Packet	SF_7	SF_8	SF_9	SF_{10}	SF_{11}	SF_{12}
SF_7	-6	16	18	19	19	20
SF_8	24	-6	20	22	22	22
SF_9	27	27	-6	23	25	25
SF_{10}	30	30	30	-6	26	28
SF_{11}	33	33	33	33	-6	29
SF_{12}	36	36	36	36	36	-6

Table I.3: Cochannel rejection (dB) for all combinations of spreading factor for the desired and interferer packets

- **Propagation Loss model:** The log-distance propagation loss model is adopted to evaluate the performance of LoRa devices in a dense environment and is expressed following to the **Eq. I.4** below:

$$L = L_0 + 10 \cdot \Gamma \cdot \log_{10}\left(\frac{d}{d_0}\right) \tag{I.4}$$

where L denotes the path Loss (dB), d the length of the path in meters (m), Γ represents the path loss distance exponent, d_0 the reference distance in meters (m) and L_0 the path loss at reference distance (dB).

- **Coding Rate (CR):** CR is computed based on **Eq. I.5** in which the redundancy of the error correction (EC) code is determined and varies between 1 and 4.

$$CR = \frac{4}{4+EC} \quad with \quad EC = 1,2,3,4 \tag{I.5}$$

- **Adaptive Data Rate (ADR):** ADR is a mechanism for optimizing throughput, energy consumption and time on air (TOA) in LoRaWAN and is generally more efficient for static devices having stable radio frequency (RF) conditions. Depending on the conditions of the environment between the IoT device and the GW, network sever will determine SF and TP values to work on between one of the combinations shown in **Table I.4** below.

Spreading Factor	Transmission Power (dBm)
SF 7	TP 2
SF 8	TP 5
SF 9	TP 8
SF 10	TP 11
SF 11	TP 14
SF 12	TP 14

Table I.4: ADR parameters configurations

> ADR is highly efficient and very effective in LoRa parameters configuration to maximize battery lifetime, range and overall network capacity. LoRa network server can manage the achieved throughput and the output transmission power used for the communication for each LoRa device individually. The better the coverage the lower the SF and TP configuration. ADR computes the median SNR value of the last 10 received uplink packets, compares it against the SNR limit for each SF and decides afterwards the best configuration.

Multiple research works in the literature evaluated LoRa networks performance [123] [64] [117]. Other research studies focused on evaluating LoRa scalability [78] while considering co-SF interference that comes from collisions when using the same SF configuration on the same channel [43] whereas others assumed that SFs on a channel are perfectly orthogonal [13] [12]. SF represents the ratio between the chirp rate and the data symbol rate and affects directly the data rate and the range that a LoRa device can reach away from a LoRaWAN GW. Moreover, co-SF interference directly impacts communication reliability, reduces the packet delivery ratio (PDR) successfully decoded at the GW [24] and limits the scalability of a LoRa network when increasing the number of devices [122]. Therefore, scalability should be considered in any upcoming study related to SF configuration strategies and network deployments. Some study examples focused on finding the optimal transmitter parameter settings that satisfy performance requirements using a developed link probing regime [11]. In [67], the authors analyze several SF configuration strategies where a group of LoRa devices can be configured with similar or heterogeneous SFs based on their position from the GW. The goal is to find the scheme that gives the best PDR. However, the impact of SF and TP configuration on network slicing has not been previously tested by the research community.

I.4 Towards enabling programmability in IoT networks

In traditional IoT networks, each equipment requires to be configured separately. This makes maintaining, configuring and adapting network devices to the changes that happen in the device, an expensive and time consuming task [8]. To tackle this problem, Software Defined Networking (SDN) emerged as a promising solution towards enabling programmability, flexibility and virtualization. Nowadays, including various IoT use cases in a single network is not straightforward due to their heterogeneous QoS requirements. Hence, it is hard for operators to guarantee QoS requirements of each service. Network slicing (NS) provides for each use case isolated network resources based on its specific needs. This section defines both SDN and NS paradigms and explores research works that integrates virtualization in IoT networks.

I.4.1 Software Defined Networking

SDN is an approach for network management that enables programmability and decouples the data plane from the control plane without one restricting the growth of other. In a network that requires fast adaptation due to the increasing number of connected devices, managing these elements becomes complex especially in IoT where each device may install various IoT applications and settings. To counter this problem, SDN emerged as new paradigm [91] that brings the ability to dynamically control the network programmatically through software applications [50].

The network architecture illustrated in **Figure I.6** shows how SDN decouples the control plane from the data plane [8] by moving the control logic into the SDN control layer. The control plane programmatically control network resources through a logically abstracted view of the network and expose this view to the application plane where operators configure IoT applications through northbound application programming interfaces (API). This facilitates the task for operators to monitor each softwarized network and to define the forwarding rules based on the traffic and the requirements of devices. OpenFlow [76], is a multi-vendor protocol which defines the interface between the control plane and SDN switches to instruct each switch on how to handle incoming data packets

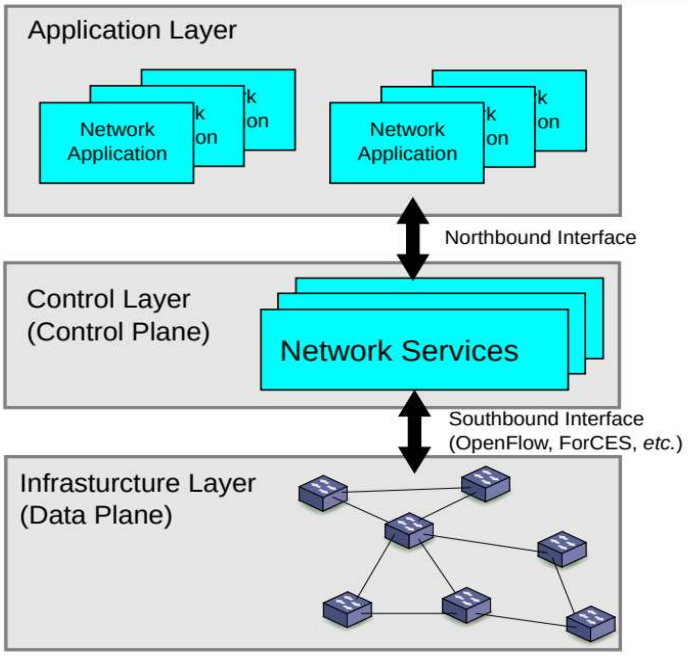

Application Layer
Network Application
Network Application
Northbound Interface
Control Layer (Control Plane)
Network Services
Southbound Interface (OpenFlow, ForCES, *etc.*)
Infrasturcture Layer (Data Plane)

Figure. I.6: SDN architecture [16]

through the southbound API. Hence, the data plane of the network will be only responsible for monitoring local information, gathering statistic and forwarding the traffic according to rules received from the centralized controllers.

I.4.2 Network Slicing

Softwarized and virtualized networks enabled the ability to support heterogeneous services running on top of the same physical infrastructure with each having isolated slice created and managed in an "on demand" manner. Network slicing is an E2E concept covering all network layers and segments. This means that slicing, performed on access, core and transport networks, will provide specific hardware requirements (bandwidth, radio resources, processing power, storage, etc.) across multiple operators [46].

By isolating virtual resources with network slicing, various use cases illustrated in **Figure I.7**, can be served with specific QoS requirements in terms of urgency, throughput and reliability, in a way that removes the impact that may come from a slice over another. However, managing each slice and finding the appropriate amount of resources

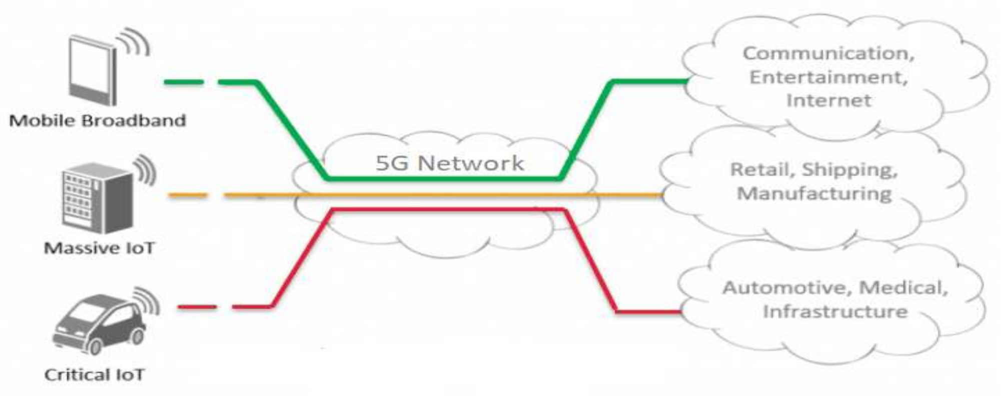

Mobile Broadband
Massive IoT
Critical IoT
5G Network
Communication, Entertainment, Internet
Retail, Shipping, Manufacturing
Automotive, Medical, Infrastructure

Figure. I.7: Network Slicing architecture

that should be allocated to each slice, remains an important challenge due to physical resources limitation and the various amount of services required in IoT scenarios. If the requirements for those virtual networks were properly instantiated on physical network infrastructure through orchestrated SDN and carefully designed, network may consume more resources than anticipated, becomes slower, unreliable and impacts other network slices performance. To tackle this challenge, multiple solutions were proposed by the research community to optimize network management in IoT networks and will be listed in detail in the following section.

I.4.3 Network Slicing and SDN integration in IoT

In large scale IoT networks, the cloud-based server should be able to acknowledge more messages as the number of IoT devices in the network increases. Hence, network flexibility is required and potentially reached using network slicing and SDN to provide heterogeneous QoS requirements through isolated E2E virtual networks controlled with SDN to facilitate the task for operators to manage IoT networks. The latter is composed of multi-networks supporting applications with various QoS requirements in terms of reliable delivery and minimum delay [125]. Therefore, authors proposed in [85] a multi-layered IoT architecture involving SDN that is able to cope with various identified IoT challenges, i.e. designing a system able to cope with numerous use cases, ensuring QoS for IoT, controlling congestion and avoiding side effects on legacy services. Moreover, various research works highlighted the efficiency of SDN in IoT networks in terms of security [40], improving transmission quality [110] and scalability through cloud-based solutions [113]. In [120], authors proposed a novel IoT network slicing creation system based on SDN and NFV emerging technologies which provides management flexibility in a centralized fashion. However, all previous solutions are not effective enough to be deployed in upcoming IoT challenges. Therefore, new slicing strategies should be adopted to cope with the fast changes in a more congested IoT environment and to create network slices and allocate physical resources accordingly.

I.4.4 Defining IoT Virtual Slices

In all contributions of this thesis, the first challenge was to propose a classification of IoT devices based on which each service will have isolated and virtualized network resources. Based on the IoT QoS requirements [2] [39], one can note that IoT devices can be classified into three categories proposed in **Table I.5** below:

QCI	Slice Name	Resource Type	Priority	Packet Delay Budget (ms)	PER %	Example Services
71	URA	GBR	1	100	10^{-3}	Real time, alarm monitoring
72	RA	GBR	2	200	10^{-3}	Real time, live monitoring
73	BE	nGBR	3	300	10^{-6}	Delay tolerant, metering

Table I.5: IoT QCIs table

Urgency and Reliability-Aware (URA) slice: requires the highest slicing priority due to urgency and reliability requirements of its members. Some examples of these applications are: surveillance and alarm monitoring. Based on **Eq. I.6**, U_{URA} is computed to define the utility for critical communications with w_{ld} and w_r the weights of load and reliability, $\sigma_r = SINR_{k,l,m}/SINR_{max}$ the rate of reliability of SINR that a device k achieves on a flow $f_{k,l,m}$ over the highest flow reliability that can be achieved through slice l and δ_r, a binary variable that guarantees a minimum threshold when searching for the highest reliability links.

$$U_{URA} = \delta_r(\sigma_r w_r + \sigma_{ld} w_{ld}) \quad with \quad \delta_r \in \{0,1\} \tag{I.6}$$

Reliability-Aware (RA) slice: requires lower priority consideration and are less critical in terms of delay. This slice presents a trade-off between reliability and load, i.e: health sensors and home security systems.

$$U_{RA} = \sigma_r w_r + \sigma_{ld} w_{ld} \tag{I.7}$$

Best Effort (BE) slice: requires the lowest priority due to their non-guaranteed data rate and delay-tolerant QoS requirements, i.e: smart metering applications.

$$U_{BE} = \sigma_{ld} w_{ld} \tag{I.8}$$

I.5 Problem statement and contributions

After presenting LPWAN technologies summarized in **Table I.6**, we have chosen to work in this thesis on LoRaWAN because its a more scalable technology operating in unlicensed spectrum. Unlike cellular IoT where only hundreds of IoT devices can be simulated in a single cell, Lora is able to serve thousands of IoT devices while also being an alliance with an open approach (instead of the proprietary one SigFox). However, in the state of the art, there's an obvious lack in providing QoS in IoT communications, which till now is limited to just reliability, meaning it's limited to just guaranteeing the delivery of a packet to the base station without considering throughput and delay constraints of the running application. Since the number of connected IoT devices is rapidly growing, an efficient solution to guarantee QoS is by bringing virtualization to IoT networks using SDN and network slicing. The motivation behind it is to improve server level from end to end across multiple network layers. This guarantees QoS requirements for IoT devices running urgent and reliable applications. We mainly answer the following questions:

- How to assign IoT devices to virtual slices and how to classify these slices in LoRaWAN ?
- How to reserve LoRa physical resources for each slice and inside each slice, how to efficiently allocate each device to the appropriate channel ?

- What are the parameters that impact QoS of each LoRa device and how to optimize this configuration in a way that doesn't increase network complexity and without impacting network performance ?
- Is LoRaWAN architecture capable of supporting IoT communications in large scale IoT deployments and how to efficiently meet the upcoming challenges ?

Features	**LTE Cat-1**	**LTE-M**	**NB-IOT**	**SIGFOX**	**LORAWAN**
Spectrum	Licensed	Licensed	Licensed	Unlicensed	Unlicensed
Modulation	OFDMA	OFDMA	OFDMA	UNB/GFSK/BPSK	CSS
Rx Bandwidth	20 MHz	1.4 MHz	200 KHz	100 Hz	125-500 KHz
Data Rate	10Mbps	200Kbps-1Mbps	20Kbps	100bps	290bps-50Kbps
Max nb of Msgs/day	Unlimited	Unlimited	Unlimited	140 msgs/day	Unlimited
Max Output Power	23-46 dBm	20 dBm	20 dBm	20 dBm	20 dBm
Link Budget	130 dB	146 dB	150 dB	151 dB	154 dB
Power Efficiency	Low	Medium	Medium High	Very High	Very High
Interference Immunity	Medium	Medium	Low	Low	Very High
Coexistence	Yes	Yes	No	No	Yes
Security	Yes	Yes	Yes	No	Yes
Mobility/localization	Mobility	Mobility	Limited Mobility, No localization	Limited Mobility, No localization	Yes

Table I.6: LPWAN technologies comparison for IoT communications

I.6 Conclusion

LPWAN technologies are being more deployed nowadays in IoT networks due to their efficiency in meeting QoS and energy constraints. However, this proliferation of IoT technologies poses co-existence challenges as they differ in their settings where ones operate in licensed frequency spectrum and others have the ability to communicate via free frequencies spectrum. In this chapter, we presented LPWAN technologies specifications and we listed the latest research work that evaluates their performance and optimization efforts in improving IoT communications. In the next chapter, we answer the first two questions by first proposing a new methods for assigning IoT devices to the three virtual slices that we have previously defined for IoT communications. Next, we implement network slicing where virtual networks share the same physical LoRaWAN infrastructure and we evaluate their performance over different SF configurations. We show the impact of traditional IoT networks on energy consumption and how in this chapter, the proposed new dynamic slicing and resource allocation strategy contributes in efficiently using LoRa resources and prioritizing urgent communications over delay tolerant IoT applications.

Chapter II

Adaptive Dynamic Network Slicing in LoRa Networks

Summary

II.1 Introduction

In Chapter I, the problem of providing QoS and flexible resource management for IoT communications is clearly stated. More specifically, three main issues should be tackled towards achieving this goal in LoRa networks:

- Finding the best way to assign IoT devices to the appropriate virtual slice that meets their specific QoS requirements.
- Due to capacity constraints and the limited number of channels on LoRa GWs, it is not straightforward to decide on how the amount of resources should be reserved while avoiding resource starvation for any of LoRa virtual slices.
- Inside each slice, one should define a strategy on how to classify IoT devices and allocate intra-slice channels accordingly.

These three problems are directly related in a way that inter-slice resource reservation and intra-slice resource allocation impact not only reliability and QoS, but also the energy consumption of IoT devices. Few research works recently tackled network slicing in IoT and focused on machine critical communications over various wireless networks. The work in [81] introduced a slicing infrastructure for 5G mobile networking and summarized research efforts to enable E2E NS between 5G use cases. Furthermore, authors in [41] and [97] adopted NS in LTE mobile wireless networks. The former proposed a dynamic resource reservation for M2M communications whereas the latter presents a slice optimizer component with a common objective in both papers to improve QoS in terms of delay and link reliability. In a 5G wearable network, the authors took advantage of slicing technology to enhance the network resource sharing and energy-efficient utilization [54]. Moreover in [31], the authors perform slicing in virtual wireless sensor networks to improve lease management of physical resources with multiple concurrent application providers. In [57], authors proposed several slicing methods for URLLC scenarios which require strong latency and reliability guarantees. Nowadays, guaranteeing service requirements in LoRaWAN with traffic slicing remains as open research issues [1]. Our main contributions with respect to the surveyed literature are stated as follows:

1. Network slicing is implemented in LoRaWAN where virtual slices are created and devices are assigned to one slice using a balanced iterative reducing and clustering method using hierarchies (BIRCH) method. The performance of LoRa virtual slices is investigated over different SF configurations in order to evaluate system performance and find the one that serves best LoRa devices in each slice.

2. A dynamic inter-slicing algorithm is proposed where the bandwidth will be similarly reserved on all LoRa gateways based on a maximum likelihood estimation (MLE) and then the latter is improved and extended with an adaptive dynamic method that considers each LoRa gateway separately and reserves its bandwidth after applying MLE on the devices in its range. Both dynamic slicing propositions will be compared to a straightforward fixed slicing strategy in which the GW's bandwidth is equally reserved between slices.

3. An energy model for LoRaWAN is integrated in NS3 based on LoRa energy specifications to analyze the energy consumed in each slice and an intra-slicing algorithm is proposed that meets the QoS requirements of each slice in an isolated manner.

The remainder of this chapter is organized as follows. Section II.2 and II.3 respectively present the LoRa system model and the network slicing problem established in this paper. In Section II.4, the slicing algorithm is proposed and implemented over the LoRa module of NS3 simulator [73]. The performance evaluation of the algorithm and simulation results are analyzed and carried out through various scenarios in Section II.5. Finally, Section II.6 concludes this chapter.

II.2 Modeling Network Slicing in LoRaWAN

LoRa network consists of a set of $K = \{1, 2, ..., k\}$ IoT devices, $M = \{1, 2, ..., m\}$ LoRa GWs plotted over a cell and connected to external LoRa Servers via fronthaul links. A slicing framework is defined that consists of a set of L virtual slices such that $L =$

$\{1, 2, ..., l\}$ can be created on physical network hardware. Each GW is characterized with a fixed number of channels $C = \{c_1, ..c_{c'}.., c_C\}$ in which part of these channels $C_{l,m}$ will be reserved for each slice $l \in L$ on GW $m \in M$ as illustrated in **Figure II.1** below.

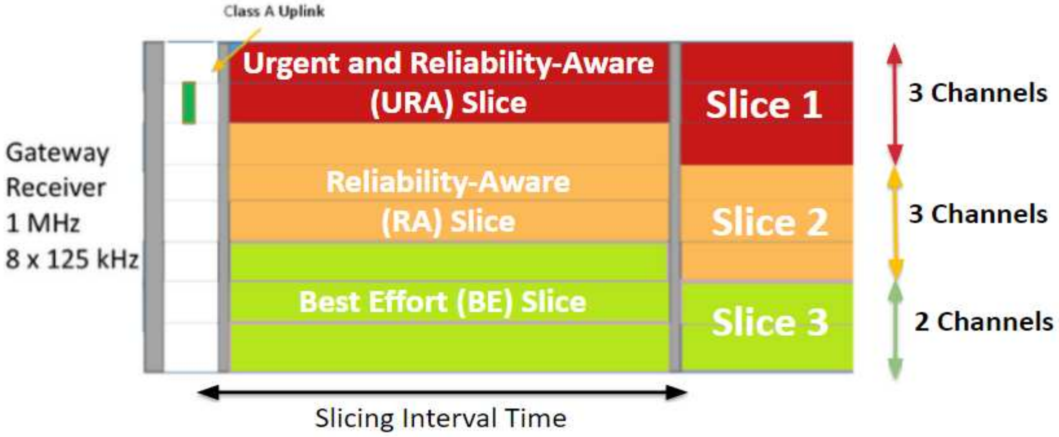


Figure. II.1: Channels slicing example over a LoRa GW

Compared to Sigfox [131], NB-IoT [93] and other IoT technologies, LoRa is more resilient to interference and jamming [18] thanks to its ability to efficiently trade communication range with high data-rate. Network slicing mainly brings flexibility to the network by virtually reserving physical resources in order to meet the QoS requirements of each slice. In IoT, each device requires specific QoS requirements in terms of delay and reliability depending on the running IoT application. Hence, the channels of each GW are divided into l slices with $l \in L$, as shown in **Figure II.2** below. The main goal behind slicing is to virtually split the network by reserving resources for each slice on the same physical device with each slice l characterized by a priority sp_l and a bandwidth $b_{l,m}$ at the GW level. A set of virtual flows F is defined where a device k associated to slice l generates a flow $f_{k,l,m}$ that goes from device k to LoRa servers through the GW m and is characterized by a utility metric $U_{k,l,m}$. LoRa GWs in range will receive the packets. However, unlike traditional LoRa networks, only one GW forwards the packet to LoRa servers to avoid duplicated packets.

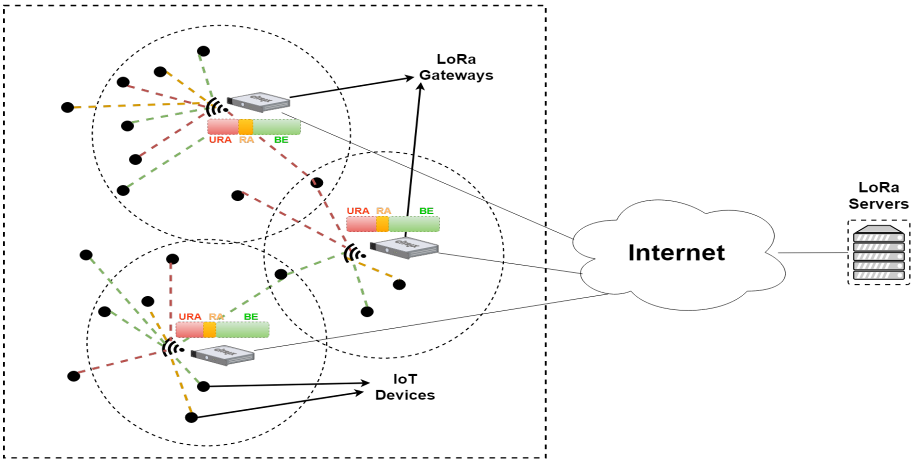


Figure. II.2: IoT slicing architecture in LoRa networks

II.3 Problem formulation

Optimizing network slicing in IoT is a threefold problem and involves: *1) LoRa devices admission and association to slices*; *2) Finding the best inter-slicing resources reservation strategy*; *3) Intra-slice resources allocation algorithm.*
First, L slices are defined based on the delay urgency factor and reliability requirements of each device. Each device is assigned next to the slice that meets best its service latency requirement. It is noteworthy that in IoT, the delay urgency and reliability represents the major key factors to define the priority of a device over another without neglecting the service type and the congestion that results from the large amount of IoT devices.

Based on throughput requirements of each slice, slicing rate is estimated to define capacity c_l that needs to be reserved for each slice l. Each GW m reserves for each slice, some of its physical receiving channels. Finally, intra-slice resource allocation is optimized in the third step by assigning each device in slice l to the most efficient virtual flow with the highest utility metric. Let $\alpha_{k,l} \in \{0, 1\}$ be a binary variable that indicates whether a device k is associated with a flow $f_{k,l,m} \in F$. The goal is to maximize the number of LoRa devices assigned to virtual flows in a way that maximizes the utility function adopted by each slice members. Therefore, the slicing and resource allocation problem for IoT can be formulated as:

$$Max \sum_{k \in K} \sum_{l \in L} \alpha_{k,l} U_{k,l,m}, \forall m \in M \tag{II.1}$$

subject to

$$C1 : \sum_{l \in L} \alpha_{k,l} = 1, \forall k \in K \tag{II.2a}$$

$$C2 : \sum_{k \epsilon K} \beta_{k,m} p_{k,l,m} \leq P_m^{max}, \forall m \in M, \forall l \in L \tag{II.2b}$$

$$C3 : \sum_{k \epsilon K} \alpha_{k,l} \beta_{k,m} r_{k,l,m} \leq R_{l,m}^{max}, \forall l \in L, \forall m \in M \tag{II.2c}$$

$$C4 : \beta_{k,m} = \begin{cases} 1 & \text{if device } k \text{ is assigned to gateway } m. \\ 0 & \text{Otherwise.} \end{cases} \tag{II.2d}$$

Knowing that multiple virtual network slices are isolated and built on top of a common physical gateway, (II.2a) ensures that each device should always choose exactly one and only network slice even if the latter was implemented on different physical gateways. Hence in a multi-gateway scenario, the device assigned to a slice will only have the option to choose between the flows that lead to the channels reserved for that slice. The total transmission power of each GW m is limited in constraint (II.2b). Moreover, constraint (II.2c) guarantees the sum of uplink traffic sent by slice members do not exceed the maximum data rate capacity of the slice that can be sent through each gateway. Constraint (II.2d) ensures binary-association values $\beta_{k,m}$ between a physical IoT device k and a physical LoRa gateway m.

II.4 Proposed Method

In LoRa networks, the general control plane and resource management module are centralized and moved to a management and control entity (MCE) in the cloud to ensure an efficient coordination of resources. Hence, LoRa servers will be the final decision maker in assigning the devices to the appropriate slice and defining the gateway that will transmit the packet following to a three-steps optimization algorithm. In the first step, each device will be assigned to the slice that meets its QoS requirements based on BIRCH method. Next, after assigning each device to its corresponding slice, GW resources will be dynamically reserved for each slice based on MLE before finally forwarding the packet to LoRa servers through the GW that provides the maximum utility value.

II.4.1 BIRCH-based Slicing Admission

Due to the large number of connected devices in IoT, BIRCH algorithm is adopted [126] which belongs to the agglomerative hierarchical clustering family and was proven as the best available clustering method for handling large datasets [127]. The main goal behind this method is to assign IoT devices to LoRa slices by checking their QoS requirements and moving from a large set of devices to a group of subsets with similar QoS requirements. The most urgent devices are the ones that have the closest instant delay d_k to their packet delay budget (PDB) and are assigned the highest priority. u_k denotes the urgency factor of device k with $u_k = d_k/PDB_k$. Given K_l devices in a cluster l, the latter will be considered as a utility point u_k of each device in a cluster with $\in K_l$. Each node in the CF-tree is a cluster of subclusters defined by a clustering feature (CF) as follows:

$$CF = (K_l, LS, SS) = (K_l, \sum_{k=1}^{K_l} u_k, \sum_{k=1}^{K_l} u_k^2) \tag{II.3}$$

where K_l denotes the number of devices in the cluster, LS the linear sum of the K_l utility points and SS the square sum of K_l utility points. BIRCH dynamically builds a CF-tree at each time a new device is inserted based on two parameters: a branching factor B and a threshold T. Each parent node contains a maximum number of B childs and a single child contains at most T entries. In this problem, B represents the number of L slices created with K_l the group of devices admitted to slice l. Hence, l nodes derive from the root representing the slices created with each slice is made up of a group of subclusters. Therefore, entries in CF-tree are not considered as devices but as a set of subclusters C that belongs to slice l and groups devices with nearly similar utility points. In **Pseudo-code 1**, the algorithm scans the clusters from the root **(line 3)** and recursively traverses down the CF-tree and chooses the closest node at each level with the smallest average inter-cluster distance D as follows:

Pseudo-code 1 BIRCH-based Slicing Admission algorithm

```
   Input  : Set of devices $K$, diameter $D$, branching factor $L$, threshold $T$
1  begin
2      Initialize as many clusters as devices
       for each $k \in K$ do
3          Start from root
           Search for closest child node according to $D$
           Search for closest subcluster according to $D$
           if number of entries $< T$ then
4              Add $k$ to subcluster $C_{l,l}$
               Update CF of $C_{l,l}$
5          else if number of childs $< B$ then
6              Create a new subcluster $C_{l,l'}$
               Add $k$ to $C_{l,l'}$
               Update CF of the parent node $S_l$
7          else if number of parents $< B$ then
8              Split child nodes and redistribute CF entries according
               to closest $D$
9          else
10             Split parent nodes
11         end
12     end
13     Update CF entries in CF-tree
14 end
   Output: Set of groups $G_l$(l=1,2,...,L)
```

$$minD = \left(\frac{\sum_{k=1}^{K_l} \sum_{k'=K_l+1}^{K_l+K_{l'}} (u_k - u_{k'})}{K_l K_{l'}} \right)^{1/2}, \forall k \in K_l, \forall k' \in K_{l'} \tag{II.4}$$

After defining the candidate child, a test is performed to find the closest CF-entry and defines if the device can be added to the child without violating the threshold condition. If so, the algorithm groups the node with the chosen entry and updates the CF-entry of the candidate subcluster **(line 4)**. If not, a new entry is created for the node inside the candidate child node without breaking the branching factor condition **(line 5-6)**. Otherwise, the child node is splitted and the utility points are redistributed based on the closest distance criteria to obtain a set of new subclusters that do not break the branching factor constraint **(line 7-8)**. In case the number of childs already reached the maximum, the parent nodes are splitted and the childs are redistributed to the closest parents **(line 9-10)**. After inserting the CF-entry, all CF informations of the path are updated from the inserted information to the root **(line 13)**.

II.4.2 Dynamic MLE-based Inter-Slicing Algorithm

Knowing the physical capacity C limitation in terms of radio channel resources of a GW m, The goal of this scheme is to estimate and reserve the appropriate resources by finding the maximum likelihood buffer demands for each slice l starting by the one with the highest slicing priority. In this work, the traffic that needs to be uploaded follows a Poisson distribution and LoRa servers are aware of the amount of data stored in the buffer B_i of each slice member.

Lemma 1. *Let T_i be the throughput needed by each device $i, \forall i \in K_l$ captured at each slicing interval time and identified by a corresponding probability distribution. For a fixed physical capacity, the optimum slicing strategy is to virtually reserve resources for each slice based on the mean throughput of its members.*

Proof: We consider T_i follows a Poisson distribution P(λ) where λ denotes the throughput needed by device i assigned to slice $l, \forall i \in K_l$. Let $f(T_i|\lambda)$ be a probability density function similar to $L(\lambda|T_i)$ that represents the likelihood of λ given the observed throughput.

$$\begin{aligned}
L(\lambda|T_1, T_2, ..., T_{K_l}) &= f(T_1|\lambda)f(T_2|\lambda)....f(T_{K_l}|\lambda) \\
L(\lambda|T_1, T_2, ..., T_{K_l}) &= \prod_{i=1}^{K_l} \frac{e^{-\lambda}\lambda^{T_i}}{T_i!} \\
logL(\lambda|T_1, T_2, ..., T_{K_l}) &= log\left[\prod_{i=1}^{K_l} \frac{e^{-\lambda}\lambda^{T_i}}{T_i!}\right] \\
logL(\lambda|T_1, T_2, ..., T_{K_l}) &= \sum_{i=1}^{K_l} log\left[\frac{e^{-\lambda}\lambda^{T_i}}{T_i!}\right] \\
logL(\lambda|T_1, T_2, ..., T_{K_l}) &= \sum_{i=1}^{K_l} \left[log(e^{-\lambda}) + log(\lambda^{T_i}) - log(T_i!)\right] \\
logL(\lambda|T_1, T_2, ..., T_{K_l}) &= \sum_{i=1}^{K_l} \left[-\lambda + T_i log\lambda - log(T_i!)\right]
\end{aligned}$$

To find the maximum likelihood parameter, we apply the first derivative and solve it to zero.

$$\begin{aligned}\frac{\partial logL(\lambda|T_1,T_2,...,T_{K_l})}{\partial\lambda} &= \sum_{i=1}^{K_l}[-1+\frac{T_i}{\lambda}] \\ &= -K_l + \frac{\sum\limits_{i=1}^{K_l} T_i}{\lambda} = 0 \\ \widehat{\lambda} &= \frac{\sum\limits_{i=1}^{K_l} T_i}{K_l}, \forall i \in \{1,...,K_l\}\end{aligned}$$

To prove that the $\widehat{\lambda}$ is the maximum value, we apply a second derivative as follows:

$$\frac{\partial^2 logL(\lambda|T_1,T_2,...,T_{K_l})}{\partial^2_\lambda} = -\frac{\sum\limits_{i=1}^{K_l} T_i}{\lambda^2}, \forall l \in L$$

The optained result is always a negative number which indicates that $\widehat{\lambda}$ is maximum and the optimal parameter to consider. Hence, the best slicing decision is to consider the mean throughput $\widehat{\lambda_l}$ of slice l members $\forall l \in L$. However, slices are not equal in terms of priority. Therefore, GW resources will be dynamically allocated to the most urgent slice starting by the channel with the highest reliability. Let $\Theta_l = \widehat{\lambda}_l / \sum\limits_{l=1}^{L} \widehat{\lambda}_l$ be the slicing rate based on which the algorithm reserves for each slice a capacity $c_{l,m} = c_m.\Theta_l, \forall l \in L$.

Pseudo-code 2 summarizes the inter-slicing algorithm and starts with the most critical slice **(line 2)**. Depending on the slicing strategy, the algorithm equally reserves the bandwidth between slices based on a straightforward *"Fixed Slicing"* **(line 14-16)** or estimates the needed throughput $\widehat{\lambda_i}$ of all slice l members in the case of *"Dynamic Slicing"* strategy, defines Θ_l for channels reservation and reserve a part of the bandwidth on all LoRa GWs in a similar manner **(line 3-7)**. If the *"Adaptive Dynamic Slicing"* was adopted, slicing rate of each slice Θ_l varies from a GW to another because in this case, MLE estimates throughput of each slice members deployed in the range of the corresponding GW m **(line 8-14)**. The algorithm moves next to the following slice, repeats the process and stops when no resources are left for reservation.

Pseudo-code 2 Dynamic and Adaptive Dynamic Inter-Slicing Strategies

```
Input  : Capacities $c_m$, $c'_n$;
         Number of slices $L$;
         Set of Throughput Requirements $T_l$
begin
    Put slices in decreasing order based on priority $sp_l$
    if method=DS then
        for each GW m do
            for each slice $l \in L$ do
                Apply MLE Estimation based on the throughput
                  required by all slice $l$ members
                  Define Slicing Rate $\Theta_l$ and Reserve capacity $c_{l,m}$
            end
        end
    else if method=ADS then
        for each GW m do
            for each slice $l \in L$ do
                Apply MLE Estimation based on the throughput
                  required by slice $l$ members in the range of GW $m$
                  Define Slicing Rate $\Theta_l$ and Reserve capacity $c_{l,m}$
            end
        end
    else
        Reserve capacity $c_{l,m}$ equally between slices
    end
end
Output: Set of resources reserved for each slice $l$
```

II.4.3 Intra-Slicing Resource Allocation Algorithm

After reserving the radio resources for each slice, the goal next is to maximize the utility function of slice members. In the previous subsection I.4.4, utility function for each slice is computed based on multiple criteria weights for reliability and load and are respectively manipulated using the analytical and hierarchy process approach. The latter is proved as a very decent method for multi-criteria decisions and was adopted in many IoT applications [114]. The algorithm searches in each slice for the gateway that offers the most robust and reliable link with lowest delay [99], finds the highest U_{URA} metric and allocates resources accordingly. Increasing the number of devices will decrease the reliability of links due to congestion. It happens sometimes for devices that are more tolerant to delay, the most reliable link may be overloaded due to the increasing number of devices and should not be taken into consideration. Instead, another channel should be available that gives the best trade-off solution when computing U_{RA} metric and offers the highest reliability with the lowest possible load. In BE slice, IoT devices runs delay-tolerant applications with higher packet delay budget. Therefore, only the load is considered in this slice utility U_{BE} without taking reliability into consideration.

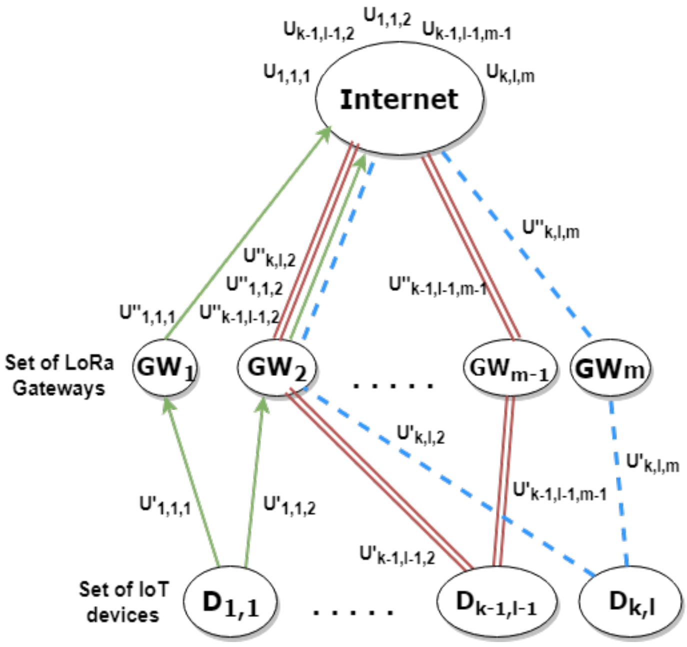


Figure. II.3: Flow modeling for IoT network slicing

In **Figure II.3**, a directed network $N = (V, E)$ is considered, where each device k is a source node s uploading traffic to external server considered as sink node t such that $s, t \in V$. Moreover, each GW m is considered as edge node and bounded by the amount of flow allowed in each slice l. In the latter, the flow that maximizes the utility function of each device k is selected. Without loss of generality, it is assumed that no edges enter the sources or exist sinks. For each edge, the respective utilities $U'_{k,l,m}$ and $U''_{k,l,m}$ are computed in the network based on **Eq. II.5** below:

$$U_{k,l,m} = U'_{k,l,m} + U''_{k,l,m} \tag{II.5}$$

Each LoRa device k assigned to slice l searches for the most efficient virtual flow through

GW m with the objective to find the highest utility metric $U_{k,l,m}$ as shown in the **Pseudo-code 3** below.

Pseudo-code 3 Max-Utility Intra-Slice Resource Allocation

```
Input  : Set of LoRa devices $K$, GWs $M$, slices $L$
         and channels $c$
begin
    Initialize flow utilities to null for all $e \in E$
    for each slice $l \in L$ do
        Put devices in decreasing order based on $u_k$
        for each device $k \in K_l$ do
            Draw network $N(V, E)$
            Find path with the highest utility $U_{k,l,m}$
            Allocate device $k$ to $f_{k,l,m}$
            Update capacity $c_{l,m}$
        end
    end
end
Output: Max-Utility flows allocation for LoRa devices
```

II.5 Simulation Results

In uplink, centralized servers enable the opportunity to make efficient slicing configurations based on data traffic in the buffer of each LoRa device. In this work, LoRa model is adopted [73] to simulate the network in the open source NS3 simulator [83]. For additional implementation details, one can refer to the work done in [72] by Magrin et al. in which a complete description of the model is included and integrated in NS3 platform. Each simulation is replicated 50 times and results are plotted with 95% confidence intervals with respect to the parameters shown in the first section of **Table II.1**.

The experiment is realized in a realistic LoRa scenario where devices are choosing a random time for transmission but periodically uploading to LoRa servers small packet payloads that varies from 10 to 20 Bytes. Simulations start with 100 devices to emulate a load of one due to the legal duty-cycle limitations of 1% in the European region [5]. The maximum number connected to a single gateway is limited to 1000 devices following to the scalability study in [55]. LoRa servers allow 8 MAC retransmissions for IoT devices before defining a packet delivery failure. Moreover, LoRa devices and gateways are both placed over a cell of 10 KM radius following to a uniform random distribution. Each device is configured with spreading factors that varies from 7 to 12 when uploading traffic to LoRa GWs. Each GW is characterized by 8 receiving channels in the 867-868 MHz european sub-band. Based on the **Eq. II.6** below, energy consumption is evaluated when the number of LoRa devices increases in each slice.

$$E_k = \frac{p_i^{tx} + p_i^{rx}}{V + epa} . d_{tx/rx} \tag{II.6}$$

where E_k is the energy consumed by an IoT device, V the LoRa supply voltage, EPA the amplifier's added efficiency, d_{tx} the duration of transmission, p_i^{rx} the power of reception and p_i^{tx} the power of transmission that varies between 2 and 14 dBm based on the SF i with $i \in \{7, .., 12\}$ adopted. Based on LoRa ADR, for each SF a static power value (dBm) is configured for transmission (Tx) and reception (Rx). An energy module for LoRa module is integrated in NS3, inspired by the one that already exists for Wifi, and is characterized with specific energy parameters and power model for LoRa [10] as listed in the second section of **Table II.1** below.

II.5.1 Proof of Isolation

The very first step before investigating slicing strategies is to prove the isolation concept. Assuming that all devices are uploading packets to a single LoRa GW. The number of LoRa devices is fixed to 20 in URA slice and the rest of devices in the network are assigned to RA and BE slices. **Figure II.4** proves the isolation concept because when the number of devices increases in RA and BE slices, URA members are not affected and the PLR percentage remained constant and nearly null whereas PLR increased in RA and URA virtual slices in a more congested scenario.

II.5.2 SF Configuration Variation

In this section, the performance of LoRa slices is evaluated with different SF configurations for a fixed number of 300 devices. Three major configuration strategies are

Simulation Parameters	
Simulation Time	300 seconds
Slicing Interval Time	50 seconds
Cell Radius	10 KM
Number of replications	50
MAC retransmissions	8
LoRa devices and GWs distribution	Random Uniform
Propagation loss model	Log-distance
Bandwidth	125 kHz
Spreading Factor	{7,8,9,10,11,12}
Confidence intervals	95%
European ISM sub-band	863-870 MHz
Power Consumption Parameters [10]	
Battery Maximum Capacity	950 mAh
LoRa Supply Voltage	3.3V
Amplifier Power's added Efficiency	10%
Connected (Tx/Rx-SF7)	2 dBm
Connected (Tx/Rx-SF8)	5 dBm
Connected (Tx/Rx-SF9)	8 dBm
Connected (Tx/Rx-SF10)	10 dBm
Connected (Tx/Rx-SF11-12)	14 dBm
Standby	0.09 mW
Sleep	0 mW

Table II.1: Chapter II simulation parameters

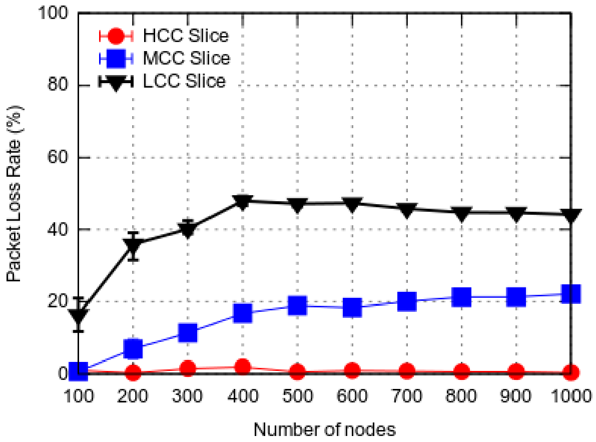


Figure. II.4: Proof of isolation

considered, namely *static* configuration where all devices in the cell are configured with the same SF, *dynamic − random* (DR) where each device randomly picks a SF value and finally the *dynamic − adaptive* (DA) where each LoRa device estimates the best SF configuration depending on the receiving power measured from the gateway. In static configurations, the test is repeated for each SF value. However, regarding dynamic configurations, a device with a powerful receiving signal picks a small SF value whereas edge nodes are generally configured with larger SF values. **Table II.2** and **Table II.3** include the mean PLR% for each SF configuration with a fixed and variant packet transmission intervals respectively. Packets may be lost when the gateway is saturated due to the load in the network (Congestion PLR%), due to co-channel rejection (Interference PLR%) or due to lack of sensitivity when the packet is out of range, or also if it doesn't reach the gateway due to an appropriate SF configuration (Sensitivity PLR%).

II.5.2.1 Fixed Packets Transmission Period

In this subsection, a decent comparison is performed between SF configuration methods for a fixed packet transmission interval. Each device randomly selects a time for transmission and then it periodically uploads a packet each 50s. $Static - SF12$ scored the highest PLR percentage. By adopting this configuration, packets transmitted occupy the spectrum for the longest time on air. Therefore, the highest impact on PLR% was reached due to congestion. Packets arrive at constant intervals and cannot be decoded due to gateway saturation. It is noteworthy to mention that no packets were lost due to lack of sensitivity because increasing the spreading factor increases at its turn the range and the probability for successfully decoding a packet. Unlike $static - SF12$, devices with $static - SF7$ configuration lost more than half of the packets. However this time, the main loss was due to lack of sensitivity for packets that are mainly transmitted by edge nodes and cannot reach the gateway because SF7 offers the shortest range capability between SF configurations. Following these assumptions, one can now understand why $static - SF9$ could be placed as a trade-off between range and spectrum occupation with the best overall PLR% between the measured static configurations. As previously mentioned, increasing SF configuration also increases the spectrum time occupation of packets sent, which also increases the interference PLR% because the probability of receiving packets with the same SF configuration at the same time will also increase.

Table II.2 illustrates PLR percentage for each category in each slice. Results show that DA configuration was the most reliable technique because SFs are dynamically

	Slice Name	Static						Dynamic	
		SF7	SF8	SF9	SF10	SF11	SF12	Random	Adaptive
Mean PLR %	Overall	54.14	39.24	39.03	43.93	78.19	94.15	43.02	30.07
Sensitivity PLR %	Overall	76.14	61.75	28.84	2.06	0	0	19.63	0
	URA	17.99	17.90	17.99	19.74	0	0	18.73	0
	RA	26.98	26.91	25.97	24.21	0	0	27.75	0
	BE	55.03	55.20	56.04	56.05	0	0	53.52	0
Congestion PLR %	Overall	22.16	32.35	53.78	63.61	61.9	69.53	69.51	86.43
	URA	0.12	0.62	2.91	6.91	11.08	15.9	8.99	8.48
	RA	0.41	1.75	9.42	30.65	46.75	49.76	36.28	34.76
	BE	99.47	97.63	87.66	62.44	42.17	34.34	54.73	56.76
Interference PLR %	Overall	0.89	4.87	16.15	33.32	37.30	30.47	9.84	12.39
	URA	7.45	11.85	13.35	15.33	16.44	20.05	16.16	15.43
	RA	42.40	42.21	40.01	35.88	30.08	28.01	35.12	36.29
	BE	50.15	45.83	46.64	48.78	53.48	51.95	48.72	48.28

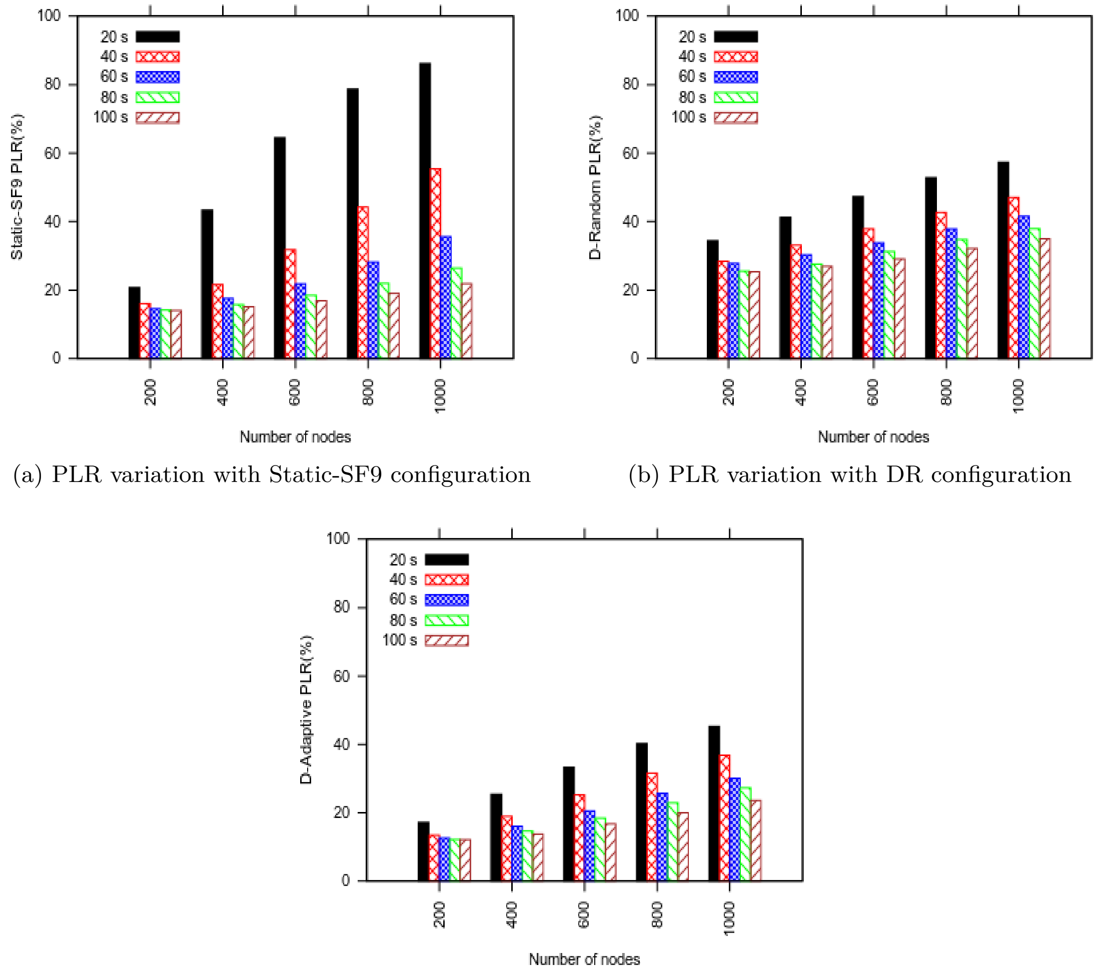
Table II.2: Packet Loss Rate variation with various SF configurations

configured on LoRa devices by measuring the receiving power that a GW gets from the device depending on its position. The advantages that the latter configuration presents are two-fold: first, depending on how far the device is from the gateway, a smaller distance requires a smaller SF configuration and secondly, the fact of adopting different SFs configuration reduces interference PLR and the probability of collisions. Regardless of the adopted SF configuration method, the urgency character of URA slice members explains the low percentage in terms of PLR compared to RA and BE slices. Urgent packets are not sent as often as other slices which reduces the probability of packets collision.

II.5.2.2 Variant Packets Transmission Interval

In **Figure II.5**, In this section, simulation is repeated with different transmission time interval. $static - SF9$ is considered as the best static configuration and is compared to DR and DA dynamic configurations when the packets transmission period increases. It is noteworthy that regardless of the adopted configuration, increasing packets transmission interval decreases PLR due to traffic intensity decrease. This can be shown with the decreasing behavior of all configurations for a common set of devices simulated. For all SF configurations and transmission intervals, DA always had the best SF distribution specially for high transmission intervals. This proves the utility of the former in realistic scenarios where congestion is normally higher due to the massive number of IoT devices.

In **Table II.3**, results show that reducing congestion has the same impact on network performance of each slice. For each transmission interval, it is shown how the percentage of PLR is distributed on each slice. Regardless of packets transmission intensity, the PLR percentage decreased in all configurations in URA slice and had the smallest impact on its communications reliability. Therefore, based on all performance results, DA configuration is adopted for the following simulations in which we compare the performance of fixed and dynamic slicing strategies in LoRaWAN.



(a) PLR variation with Static-SF9 configuration

(b) PLR variation with DR configuration

(c) PLR variation with DA configuration

Figure. II.5: Performance study with/without considering load in metric calculations

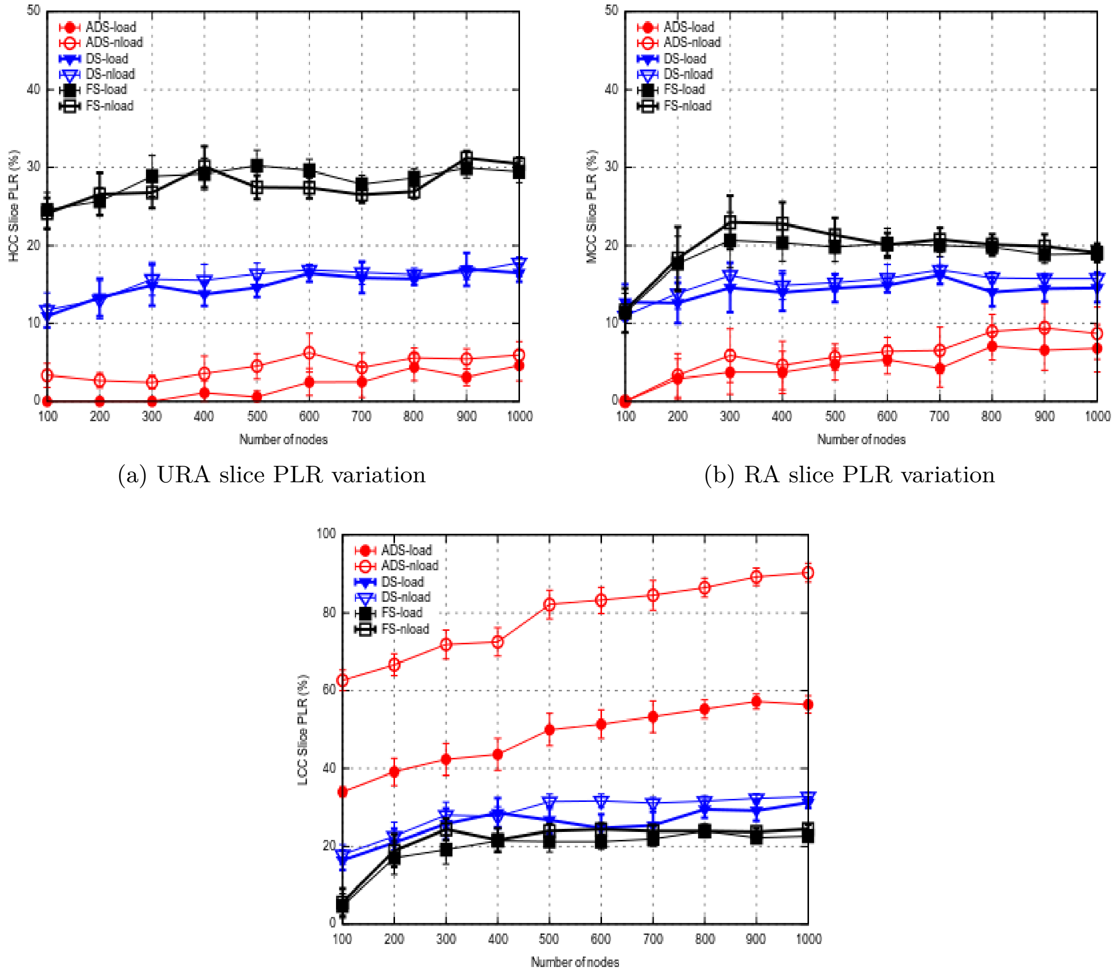
PLR %	PTP (s)	Static-SF9			Dynamic-random			Dynamic-adaptive		
		URA	RA	BE	URA	RA	BE	URA	RA	BE
Sensitivity PLR %	20	24.68	22.98	52.34	18.86	26.12	55.02	0	0	0
	40	20.35	25.81	53.84	18.69	27.43	53.88	0	0	0
	60	19.97	24.15	55.88	18.25	27.36	54.39	0	0	0
	80	18.23	24.22	58.46	18.52	26.82	54.66	0	0	0
	100	17.32	23.92	57.85	18.80	26.96	54.24	0	0	0
Congestion PLR %	20	11.57	45.75	42.68	10.46	39.39	50.15	10.93	39.30	49.77
	40	8.00	37.26	54.74	8.78	36.33	54.89	8.88	36.73	54.39
	60	5.77	24.71	69.52	8.01	31.49	60.50	7.92	32.38	59.70
	80	3.95	13.74	82.31	6.45	29.25	64.30	6.30	29.40	64.30
	100	3.13	8.33	88.55	5.29	23.84	70.87	5.11	24.16	70.73
Interference PLR %	20	15.92	32.20	51.88	16.32	35.63	48.05	16.12	36.77	47.11
	40	15.82	35.53	48.65	15.66	34.95	49.40	15.43	36.98	47.59
	60	15.56	37.12	47.32	14.92	36.28	48.80	15.14	35.81	49.05
	80	14.50	38.37	47.13	15.62	36.58	47.81	15.63	36.20	48.17
	100	14.42	38.14	47.44	15.91	36.16	47.93	13.93	36.53	49.55

Table II.3: PLR variation with various SF configurations

II.5.3 Fixed vs Dynamic vs Adaptive-Dynamic Slicing Strategies

Following to previous simulations, $dynamic-adaptive$ SF configuration is adopted which has proved its worthiness for this study. The goal in this section is to evaluate the performance of the $fixed$ (FS), $dynamic$ (DS) and the $adaptive-dynamic$ slicing (ADS) strategies. With FS, the number of receiving paths is reserved in an equal manner and is compared to DS and ADS strategies where slicing decisions are performed using MLE throughput estimation for each slice starting with the one with the highest priority. Moreover, the impact of adding load metric to utility calculations is studied for each slicing strategy when the number of LoRa devices assigned to each slice increases. Each slice in a LoRa gateway suffers from congestion, decreasing with it the probability of successfully decoding the packet.

Simulation results in **Figure II.6** prove the efficiency of load consideration when computing the mean values of slices with and without considering load in metric calculations. Being load-aware improves reliability in the network. When congestion in the network increases, the traffic is balanced to the corresponding slice but on a less-loaded gateway. Reliability on all slices improved especially in BE slice because most of its members lose their packets due to congestion. In a comparison between each slicing strategy, ADS with load consideration showed the most reliable performance for URA



(a) URA slice PLR variation

(b) RA slice PLR variation

(c) BE slice PLR variation

Figure. II.6: PLR in each slice with various slicing strategies

and *RA* slice as plotted in **Figure II.6a** and **Figure II.6b** respectively. This returns for example to the case of *URA* slice where the sporadic nature of packet transmissions requires low latency and high reliability with unsteady throughput needs. Therefore, an appropriate estimation of throughput improves slicing and should be considered on each GW separately because it differs from a GW to another. Moreover, **Figure II.6c** shows that considering load in metric calculations scored approximately 50% improvement in the PLR% of *BE* slice members. However, this did not prevent *ADS* from being the lowest reliable strategy in *BE* slice. The reason returns to the fact that *ADS* prioritizes a slice over another and reserves for it the needed bandwidth unlike *FS* where the bandwidth is equally reserved between slices. *BE* members do not always get the needed bandwidth required for transmission when a small capacity is fixed for this slice. The performance of each slice is evaluated next using *ADS* with a load strategy in terms of energy consumption and the percentage of devices that respected their delay deadlines.

II.5.3.1 Percentage of unserved nodes in delay

The efficiency of *ADS* is mainly shown in **Figure II.7** below. With *ADS*, LoRa devices had the highest percentage of devices that respected their delay deadlines compared to *DS* and *FS* strategies with an unserved rate that never exceeded 10% of the total number of packets transmitted. This highlights the importance of including urgency priority in slicing strategies and considering reliability in intra-slice resource allocation algorithm due to its direct impact on the spreading factor configuration and the spectrum occupation time.

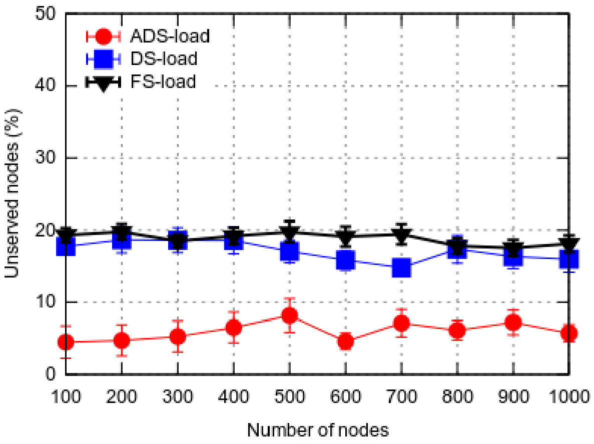
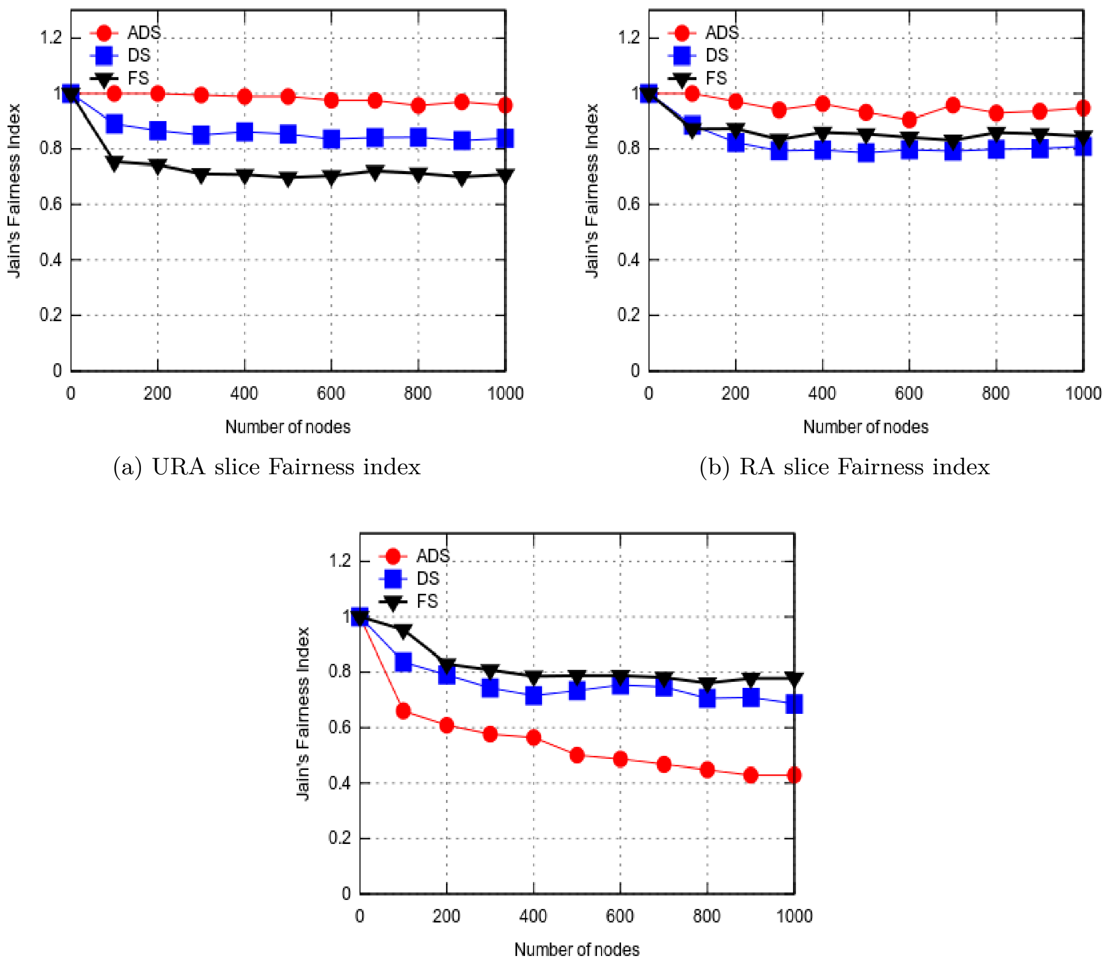


Figure. II.7: Percentage of unserved nodes

II.5.3.2 Jain's Fairness index

The goal of this study is to measure the metric that identifies underutilized channels in each slice with FS, DS and ADS strategies. Based on **Eq. II.7**, we evaluate in **Figure II.8** the Jain's fairness index of each slicing strategy as follows:

$$Fairness_{index} = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2} \tag{II.7}$$



(a) URA slice Fairness index

(b) RA slice Fairness index

(c) BE slice Fairness index

Figure. II.8: Fairness evaluation in each slice with various slicing strategies

where x_i denotes the normalized throughput of each IoT device and n is the total number of active devices in each LoRa slice. Jain's fairness index varies between 0 and 1 with 1 being perfectly fair. ADS strategy provides the best distribution compared to DS and FS strategies as plotted in **Figure II.8a** and **Figure II.8b** below. With FS strategy, resources are divided equally between URA, RA and BE slices. This explains fairness results of FS that are quite similar in all simulated slices. It is noteworthy to mention performance degradation of ADS and DS strategies when moving from urgent to less urgent slices. This is normal due to slicing priority consideration where resource reservation algorithm begins with the most critical slice. However, ADS always had a clear upper hand over DS strategy in urgent slices except for BE slice where less channels are reserved for its members as shown in **Figure II.8c** below.

II.5.3.3 Energy Consumption

When increasing the number of nodes, the total energy consumed increases for all the simulated slices, as plotted in **Figure II.9** below. However, *URA* slice always consumed less energy even when the number of its LoRa members increased. This returns to relation between SF and TP configuration shown in the second section of **Table II.1**. Increasing SF will increase the transmission power and the energy consumption of a slice member. Therefore, the consideration of reliability in utility calculations forces delay-sensitive devices to take the most reliable path with the lowest spreading factor values and transmission power compared to *RA* and *BE* slices.

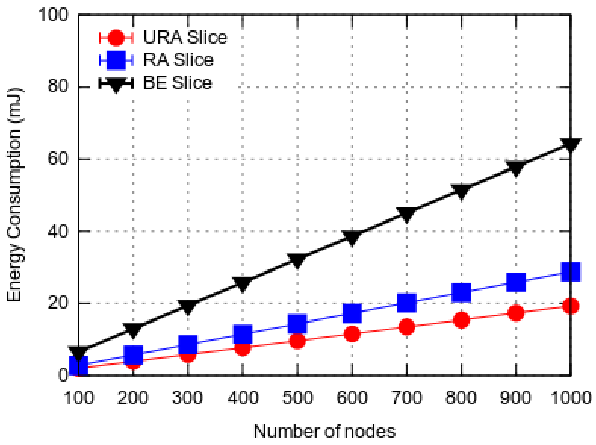


Figure. II.9: Mean energy consumption variation

II.6 Conclusion

In this chapter, network slicing is implemented and investigated in centralized standard LoRa architecture in which inter-slice resource reservation and intra-slice resource allocation methods are both proposed and optimized with respect to the QoS requirements of each slice members. Various slicing strategies are compared after proving the isolation concept between each of LoRa virtual slices. Based on the results obtained, the adaptive dynamic network slicing appears to be the best slicing method between the ones compared in this chapter. However, it is shown in **Figure II.6** that more than 50% of the packets were lost in BE slice. We believe that these results can still be improved if LoRa parameters of IoT devices were efficiently optimized to improve network performance in each slice. In the following chapter, instead of considering the adaptive data rate mechanism which jointly increases or decreases both spreading factor and transmission power of an IoT device, we propose a slice-based optimization method that finds the best combination between LoRa SF and TP parameters in a way that maximizes network performance of each slice in a LoRaWAN smart city scenario.

Chapter III

Joint QoS and Energy aware Optimization in LoRa Network Slicing

Summary

III.1 Introduction

The solution that we proposed in Chapter II using network slicing, has shown its worthiness in providing urgency and reliability in LoRa networks. In this context, an urgent packet will always have a part of LoRa resources reserved to guarantee its arrival to the gateway. However, after analyzing in depth reliability results, we have noticed that there is still room for improvement to reduce the percentage of packets lost in the network. Hence, we decided to investigate more in depth on how LoRa parameters impact QoS of an IoT device and how to configure the latter properly in a network slicing scenario.

Reliability in LoRa does not depend only on just successfully delivering a packet to a channel above sensitivity, it also depends on the configuration of other packets received at the same time on a LoRa channel which may cause significant packet losses due to co-SF and intra-SF interference. The former happens when two packets configured with same SF are simultaneously received at the same channel whereas the latter happens when the interfere packet is decoded with different SF configuration. Many research studies focused on proposing various SF configurations and distribution strategies over multiple network deployments [84] with the goal to overcome capacity limits [116] and to provide a trade-off solution that minimizes energy consumption while maximizing reliability [62]. However, SF is not the only parameter that should be taken into consideration when optimizing LoRa configuration.

Increasing TP of a device is also important to increase SNR and the chance of decoding one of the packets upon interference. However, one should also not forget on battery constraints that should be respected to avoid depleting the battery lifetime of IoT devices. In some works, authors showed the importance of configuring IoT devices with a proper combination between SF and TP parameters to improve scalability of LoRaWAN [89] and to avoid performance degradation and unfairness that happens in LoRa network if IoT devices configure SF and TP locally [94]. LoRa originally includes a link-based adaptation of SF and TP configurations using the ADR mechanism. Many works tried to propose modified and improved ADR algorithms with the goal to increase reliability and energy-efficiency without taking into consideration the possibility of intra-SF and inter-SF collisions [58] [108] [95]. The latter can be decreased with the knowledge of the entire network or by finding the optimum configuration after testing all combinations of LoRa parameters that respects specific thresholds [11]. However, this method is considered as time consuming because sometimes, achieving multi-objectives in terms of reliability and energy-efficiency do not always require tuning parameters, especially on IoT devices placed at the edge of their communication range [18]. In [65], the performance of the official ADR mechanism proposed by LoRa is evaluated and shows the impact of different configurable parameters in terms of slow convergence rate which introduces higher energy consumption and packet losses.

All works previously mentioned from the literature improved LoRaWAN performance using various optimization strategies. However, the random-based access nature in IoT network gives the motivation to optimize network slicing with a slice-based parameters configuration that treats each virtual slice differently without considering all IoT devices as devices belonging to the same LoRa network. The goal behind this proposition is to improve QoS of IoT devices and limit interference and collisions in each LoRa virtual network. This chapter contributions extend the previous one by considering smart city applications belonging to different QoS classes and are stated as follows:

1. We include QoS in LoRa, which was previously considered as a best effort technology, with the goal to test the flexibility that network slicing provides in terms of traffic management and QoS integration.

2. We apply ADS, found to be the best slicing strategy in Chapter II, where the bandwidth is efficiently reserved on each LoRa GW separately based on MLE estimation. The goal of this scheme is to avoid channels starvation while considering the exact need of each slice starting by the one with the highest slicing priority.

3. We propose TOPG as a novel slicing optimization method that is based on Tech-

nique for Order of Preference by Similarity to Ideal Solution (TOPSIS) and Geometric Mean Method (GMM). The proposed method efficiently configures LoRa SF and TP parameters and improves the performance of each slice in terms of QoS, reliability and energy consumption.

The remainder of this chapter is organized as follows. We devote Section III.2 and III.3 to respectively describe the network slicing system model in a smart city scenario and the multi-objective optimization problem established in this chapter. Section III.4 presents the proposed slicing and optimization algorithm implemented over the LoRa module of NS3 simulator [73]. The performance evaluation of the algorithm and simulation results are analyzed and carried out through various scenarios in Section III.5. Finally, Section III.6 concludes the chapter.

III.2 Modeling Network Slicing in LoRa-based Smart City Network

In a smart city network deployed with LoRa, various use cases are enabled for citizens in terms of mobility, smart home, health and many other fields. However, due to the heterogeneity of these applications, a single smart city network is unable to support all of these traffic types within a network without compromising QoS for any of them. In case of an accident, a connected vehicle should immediately communicate the information to the people involved and responsible for emergency situations. However, this information could be lost or arrived without respecting the required delay in urban cities. Hence, the focus here is on applying traffic slicing in smart city scenarios, virtually isolated, and with specific QoS thresholds. In **Table I.5**, the key QoS requirements of URA, RA and BE slices were previously defined in Chapter I with each having running various IoT applications illustrated in **Figure III.1**. One of the listed use cases is smart mobility, where, an increase in communication delay between two vehicles or between a vehicle and its base station may result a dangerous accident and should be provided with the highest levels of urgency and reliability. On the other hand, some smart city applications only require best effort behavior like metering and actuating to measure the consumption data of different resources like electricity, water, gas and heating power.

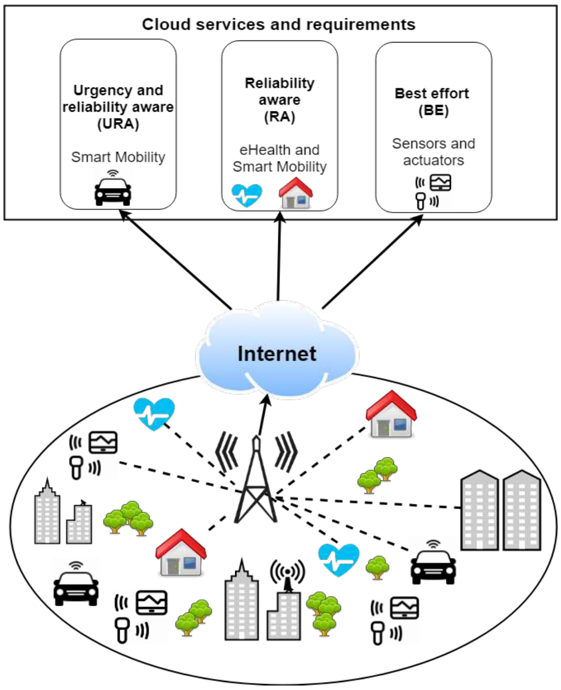

Cloud services and requirements
Urgency and reliability aware (URA)
Smart Mobility
Reliability aware (RA)
eHealth and Smart Mobility
Best effort (BE)
Sensors and actuators
Internet

Figure. III.1: Smart city applications in LoRa-based network

Figure III.2 illustrates how IoT devices are connected to a LoRa GW in the actual standard architecture (**Figure III.2a**) and configured with one of the SF-TP combinations, listed in **Table** I.4. The server aims to increase both SF and TP values simultaneously to increase signal robustness and decode packets at larger distance from the GW. However, when network slicing is applied on a Lora gateway (**Figure III.2b**), ADR mechanism becomes inefficient specially if the device in question belongs to a slice having specific QoS thresholds that needs to be respected before reaching external LoRa servers through the internet. With traffic slicing, the receipt of urgent communications is now guaranteed at the GW level. However, overestimating SF and TP configurations leads to an increase in energy consumption due to the longer activity time for an IoT device when uploading a packet with high SF configuration. Moreover, if a high SF is configured, achieved throughput may be lower than the one that needs to be guaranteed in the corresponding slice. Hence, for each slice, one should not be limited to discrete SF and TP values proposed by LoRa ADR mechanism. This work enables the possibility to define specific slice-based SF and TP combination to be configured on an IoT device in a way that respects its QoS thresholds.

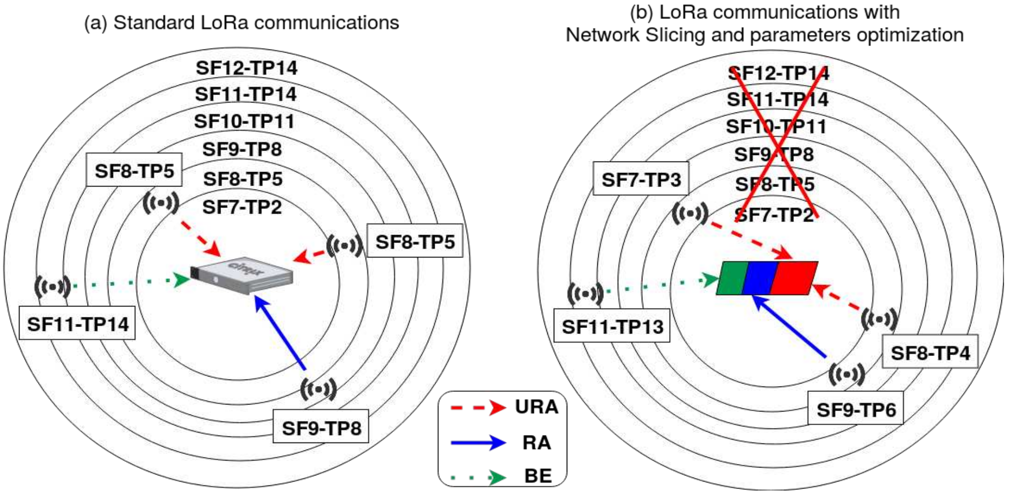


Figure. III.2: (a) Standard LoRa and (b) LoRa network slicing with parameters optimization

In this Smart City network, we assume that centralized LoRa servers are aware of the QoS required by each device in terms of delay, throughput and reliability. Moreover, LoRa servers are responsible for defining resource reservation strategies on LoRa gateways (GWs) and on configuring the devices with SF and TP parameters. Let N(V,K) be a directed LoRa network including V={S,M,C} components and consists of S LoRa servers, $M = \{m_1, ..m_{m'}.., m_M\}$ denotes the set of LoRa gateways and $C = \{c_1, ..c_{c'}.., c_C\}$ denotes the set of channels on each gateway. Let $K = \{k_1, ..k_{k'}.., k_K\}$ be the set of IoT devices connected to the gateways and belongs to the set of slices L. Each slice is defined based on delay, throughput and reliability requirements of IoT applications [2]. It is noteworthy that to improve communications in an IoT environment, multiple objectives should be reached. More precisely, we jointly consider in this chapter QoS, energy, and reliability requirements as major key factors and objectives to optimize parameters configuration of an IoT device belonging to a slice with a specific slicing priority sp_l. On each LoRa gateway, a slicing rate is estimated based on the throughput required by the devices active in each slice l in order to define capacity c_l that needs to be reserved. Each gateway has a fixed number of C channels with $C_{l,m}$ the set of channels reserved for slice l on GW m. We search to jointly optimize QoS and network slicing energy efficiency by assigning slice members with the proper SF and TP configurations.

However, solving this multi-objective problem is challenging. Therefore, the goal in this chapter is to optimize parameters selection after evaluating the cost and benefits in each slice. We added σ_1, σ_2 and σ_3 as constant variables to equally distribute the weight between objective functions and we introduced $\alpha_{k,l} \in \{0, 1\}$ and $\beta_{C_l,m} \in \{0, 1\}$ as two binary decision variables that respectively indicates the admission of device k to slice l and the reservation of a channel C_l on GW m.

III.3 Multi-Objective Problem Formulation

Network slicing optimization in IoT is a twofold problem and involves:

1. *Finding the best inter-slicing resources reservation strategy*
2. *Configuring each slice member with the optimum SF and TP parameters*

The goal in this chapter is to optimize the global performance of each slice in terms of QoS, energy, and reliability. This turns the second problem of finding the best SF and TP configuration for an IoT device into a multi-objective problem formulated as follows:

III.3.1 QoS in a LoRa slice

Each device k adopts a specific SF configuration for information transmission. The configuration of SF is very crucial because the latter is directly related to throughput $r_{k,c}$ and transmission delay $d_{k,c}$ previously defined in Section **I.3.2**. Based on these values, we model in **Eq. III.1** the QoS cost as:

$$\begin{aligned} &QoS_{k,c} = \overline{r_{k,c}} + (1 - \overline{d_{k,c}}) \\ &Maximize \sum_{k \in K} \alpha_{k,l} QoS_{k,c}, \forall l \in L, \end{aligned} \tag{III.1}$$

where $QoS_{k,c}$ denotes the benefits that should be maximized in each slice and respectively includes $\overline{d_{k,c}}$ and $\overline{r_{k,c}}$ normalized by dividing $r_{k,c}$ and $d_{k,c}$ values by the highest throughput and delay that can be achieved over a wireless LoRa link.

III.3.2 Interference in a LoRa slice

In LoRaWAN, the reason for loosing a packet uploaded by an IoT device is three-fold:

1. when a packet is received under-sensitivity if the device was out of range or configured with bad SF and TP values. This is indicated by $PLR'_{k,c}$ denoted as binary variable as follows:

   $$PLR'_{k,c} = \begin{cases} 0 & \text{if device } k \text{ successfully reaches } c \in C_{l,m} \\ 1 & \text{Otherwise} \end{cases}$$

   The output of this variable mainly depends on the sensitivity of the gateway that increases alongside an increase in SF configuration [73].

2. when packets are lost due to co-SF interference that happens between two devices simultaneously transmitting with the same SF. Based on random access formula [109], the probability of the latter G_{SF} depends on the number of packets generated during the transmission of one packet with the same SF and is written in **Eq. III.2** below:

   $$PLR''_{k,c} = 1 - e^{-2G_{SF}} \tag{III.2}$$

3. when a collision happens between two packets transmitted with different spreading factors leading to a potential loss due to inter-SF interference. In this case, the packet survives interference if its signal power was higher than the power margin value (dB) needed to decode a packet from its interferer. Based on the power margin matrix, previously explained in **Table I.3**, $PLR'''_{k,c}$ is modified to indicate if a packet survives inter-SF interference on a channel or not.

   $$PLR'''_{k,c} = \begin{cases} 0 & \text{if device } k \text{ survives interference} \\ 1 & \text{Otherwise} \end{cases}$$

The goal here is to find the configuration that maximizes the chances of decoding a packet upon its reception at the GW level. After combining all these objectives, the reliability cost is finally modeled in **Eq. III.3** with the objective to find the configuration that can minimize the probability of loosing a packet due to co-SF interference, inter-SF interference or low channel sensitivity:

$$\begin{aligned} & PLR_{k,c} = PLR'_{k,c} + PLR''_{k,c} + PLR'''_{k,c} \\ & Minimize \sum_{k \in K} \alpha_{k,l} PLR_{k,c}, \forall c \in C_{l,m}, \forall l \in L \end{aligned} \tag{III.3}$$

III.3.3 Energy Consumption in a LoRa slice

Increasing the SF reduces the transmitted data rate and decreases the transmission delay and signal strength whereas higher TP increases SNR and the energy consumption of an IoT device. The latter is defined in **Eq. II.6** and is affected by both SF and TP values. Accordingly, we compute the energy of a LoRa device during a slicing interval time following to **Eq. III.4** with the objective of minimizing energy consumption in a LoRa slice without degrading QoS performance:

$$Minimize \sum_{k \in K} \alpha_{k,l} E_{k,c}, \forall c \in C_{l,m}, \forall l \in L \tag{III.4}$$

Due to the multi-objectivity of the problem, we search to find the optimum slicing strategy with the proper SF and TP configurations that simultaneously maximize QoS benefits of each slice and minimize energy and reliability costs without under optimizing a function over another. This multi-objective problem is formulated subject to the constraints below:

$$C1 : \sum_{l \in L} \alpha_{k,l} = 1, \forall k \in K \tag{III.5a}$$

$$C2 : b_{l,m} \cap b_{l',m} = \emptyset, \forall l, l' \in L, \forall m \in M \tag{III.5b}$$

$$C3 : 0 \leq P_{k,C_l} \leq P_k^{max}, \forall m \in M, \forall l \in L \tag{III.5c}$$

$$C4 : \sum_{k \epsilon K} \alpha_{k,l} \beta_{C_l,m} R_{k,c} \leq R_{l,m}^{max}, \forall l \in L, \forall m \in M \tag{III.5d}$$

$$C5 : \alpha_{k,l} \in \{0,1\}, \forall k \in K, \forall l \in L \tag{III.5e}$$

$$C6 : \beta_{C_l,m} = \begin{cases} 1 & \text{if channel belongs to slice } l \text{ on GW } m. \\ 0 & \text{Otherwise.} \end{cases} \tag{III.5f}$$

The first constraint (III.5a) ensures that each device should always choose exactly one and only network slice even if the latter was implemented on different physical gateways. Moreover, a perfect isolation is guaranteed in (III.5b) between two bandwidth parts assigned for two different slices regardless if the latter was reserved on the same or on two different gateways. The transmission power of each device is limited in constraint (III.5c). Furthermore, constraint (III.5d) guarantees the sum of uplink traffic sent by slice members which do not exceed the maximum data rate capacity of the slice that can be sent through each gateway. Constraint (III.5e) ensures binary association values of device k to slice l and constraint (III.5f) ensures binary reservation values of a channel that belongs to slice l on a LoRa GW m.

III.4 Proposed Method

In this section, we expound the proposed slicing and configuration mechanism, illustrated in **Fig. III.3**, that will optimize LoRa network slicing by catching up to the multi-objective optimization problem in finding the appropriate resource reservation and the best configuration to adopt for IoT devices.

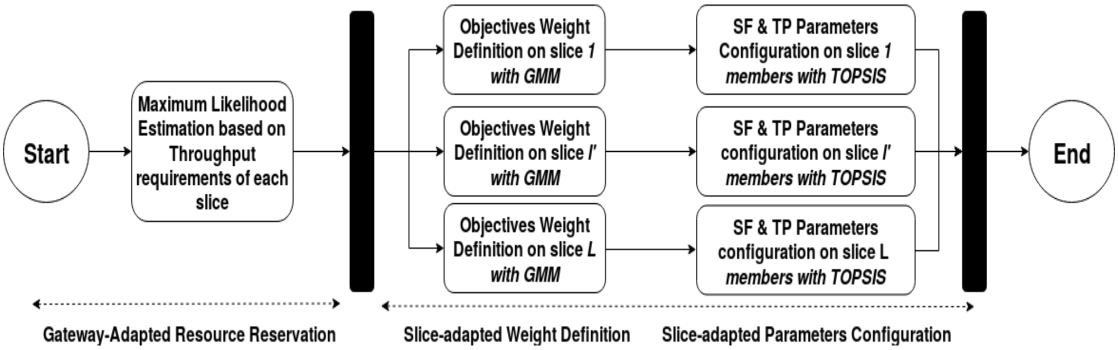

Start
Maximum Likelihood Estimation based on Throughput requirements of each slice
Objectives Weight Definition on slice 1 with GMM
Objectives Weight Definition on slice l' with GMM
Objectives Weight Definition on slice L with GMM
SF & TP Parameters Configuration on slice 1 members with TOPSIS
SF & TP Parameters configuration on slice l' members with TOPSIS
SF & TP Parameters configuration on slice L members with TOPSIS
End
Gateway-Adapted Resource Reservation
Slice-adapted Weight Definition
Slice-adapted Parameters Configuration

Figure. III.3: The proposed slicing and TOPG optimization algorithm

The first problem appears in finding a decent slicing strategy to split the physical network in a way that avoids resource starvation. To this manner, we consider the adaptive dynamic slicing strategy that we previously proposed in Section II.4.2. The centralized controller estimates the need in throughput using MLE for each virtual slice to define the rate of channels that should be reserved for the devices starting with the slice with the highest slicing priority sp_l. In each slice, we propose a novel slice-based TOPG method that combines GMM [124] and TOPSIS [106] optimization algorithms. GMM is adopted to define the weight values based on the objectives importance in each slice. These values are next imported to a TOPSIS-based optimization method that searches for the best SF and TP configuration which meets utility requirements of each slice members.

III.4.1 The Proposed TOPG Optimization Algorithm

After defining slicing objectives, we next need to adapt the weight of every objective before optimizing SF and TP parameters configurations in a way that meets best the requirements of the corresponding slice. To do this, we propose an optimization algorithm based on GMM and TOPSIS methods.

Let A_l=$(a_{ij,l})_{n\times n}$ be a judgment matrix where $a_{ij,l} > 0$ and $a_{ij,l} \times a_{ji,l} = 1$. Each value $a_{ij,l}$ measures the importance of an objective i over another objective j for each slice l. Based on the importance values in each slice, a priority vector is derived and denoted as $\psi_l = (\psi_{1,l}, \psi_{2,l}, ..., \psi_{(n-1),l}, \psi_{n,l})$, where $\psi_l \geq 0$ and $\sum\limits_{i=1}^{n} \psi_i = 1$, from the decision matrix A_l. With GMM, weight configuration for each objective is defined as an objective function of the following optimization problem:

$$\begin{cases} Minimize \sum\limits_{i=1}^{n} \sum\limits_{j>i} [ln(a_{ij,l}) - (ln(w_{i,l}) - ln(w_{j,l}))]^2 \\ s.t. \quad w_{i,l} \geq 0, \sum\limits_{i=1}^{n} w_{i,l} = 1, \forall l \in L \end{cases}$$

which have a unique solution and can be simply solved by the geometric means of the rows of each slice's decision matrix A_l:

$$w_{i,l} = \frac{\sqrt[n]{\prod_{j=1}^{n} a_{ij}}}{\sum\limits_{i=1}^{n} (\sqrt[n]{\prod_{j=1}^{n} a_{ij}})} \tag{III.6}$$

After finding the objective weights for each slice, we import the weight vector of each slice into a decision matrix D_l, which consists of a set of possible alternatives A_x as shown in the below matrix:

$$D_l = \begin{array}{c} \\ \\ \end{array} \begin{array}{c|cccc|} \mathit{Alternatives} & w_{1,l} & .. & w_{n-1,l} & w_{n,l} \\ A_1 & a_{1,1} & .. & a_{1,n-1} & a_{1,n} \\ .. & .. & .. & .. & .. \\ .. & .. & .. & .. & .. \\ A_{m-1} & a_{m-1,1} & .. & a_{m-1,n-1} & a_{m-1,n} \\ A_m & a_{m,1} & .. & a_{m,n-1} & a_{m,n} \end{array}$$

where each value $a_{x,y}$ represents a parameter configuration of a device with $y \in \{1, 2, ..., n\}$ defines the objective and $x \in \{1, 2, ..., m\}$ denotes a combination of SF $i \in I = \{7, ..., 12\}$ and TP discrete values $j \in J = \{2, ..., 14\}$ in dBm among which LoRa servers need to assign the device with the best configuration based on W_l, the set of objectives weight values of the corresponding slice. TOPSIS method requires normalized values $\overline{a_{x,y}}$ in D_l with the goal is to find the alternative with the shortest distance from positive ideal solution and the one with the largest distance from the negative ideal solution.

$$\overline{a_{x,y}} = \frac{a_{x,y}}{\sqrt{\sum\limits_{x=1}^{m} a_{x,y}^2}}, \quad \mathit{with} \quad x \in \{1, ..., m\}, y \in \{1, ..., n\} \tag{III.7a}$$

In other terms, the goal is to find the best configuration that maximizes QoS benefits and minimizes the costs in terms of PLR and energy consumption. For each positive ideal solution A^+ and negative ideal solution A^-, normalized weight rating $v_{x,y}$ can be determined using the following equations:

$$v_{x,y} = w_{x,l}\overline{a_{x,y}}, \quad with \quad x \in \{1, ..., m\}, y \in \{1, ..., n\} \tag{III.7b}$$

$$A^{+} = (v_1^{+}, v_2^{+}, ..., v_n^{+}) \tag{III.7c}$$

$$A^{-} = (v_1^{-}, v_2^{-}, ..., v_n^{-}) \tag{III.7d}$$

where V_y value results using equations

$$V_y^{+} = \left\{ \max_x v_{x,y}, y \in Y_1; \min_x v_{x,y}, y \in Y_2 \right\} \tag{III.7e}$$

$$V_y^{-} = \left\{ \min_x v_{x,y}, y \in Y_1; \max_x v_{x,y}, y \in Y_2 \right\} \tag{III.7f}$$

where Y_1 and Y_2 respectively respect benefit and cost criterias. We calculate next the euclidean distance from the positive ideal solution and negative ideal solution of each alternative; respectively as follows:

$$d_i^{+} = \sqrt{\sum_{j=1}^{n} (d_{i,j}^{+})^2} \tag{III.7g}$$

$$d_i^{-} = \sqrt{\sum_{j=1}^{n} (d_{i,j}^{-})^2} \tag{III.7h}$$

where $d_{x,y}^{-} = V_y^{+} - v_{x,y}$, with $x = 1, ..., m$ and $d_{x,y}^{-} = V_y^{-} - v_{x,y}$, with $x = 1, ..., m$.

$$\zeta_x = \frac{d_x^{-}}{d_x^{+} + d_x^{-}} \tag{III.7i}$$

We finally rank the configurations according to the relative closeness previously calculated and we assign each device with the configuration that provides the highest value ζ_x due to its closest position to the positive ideal solution.

III.4.2 Complexity Analysis

We evaluate the complexity of the proposed algorithm briefly listed in **Pseudo-code 4** compared to other configuration methods implemented in this study.

Pseudo-code 4 Adaptive Slicing and (SF-TP) Configuration

```
Input  : Capacities $c_m$; Number of slices $L$;
         Set of Throughput Requirements $T_l$
begin
    Put slices in decreasing order based on priority $sp_l$
    for each GW m do
        for each slice $l \in L$ do
            Apply MLE based on the throughput required by slice
             $l$ members in the range of GW $m$.
             Define slicing rate $\Theta_l$.
             Reserve bandwidth capacity $c_{l,m}$.
        end
    end
    for each GW m do
        for each slice $l \in L$ do
            Apply GMM to define $W_{l,n}$ of each objective.
        end
    end
    Sort devices in $K_{l,m}$ based on urgency factor $u_k$.
    for each device $k \in K_{l,m}$ do
        Sort channels in $C_{l,m}$ based on link budget.
        for each Channel c in $C_{l,m}$ do
            if config=false then
                Apply TOPSIS to define (SF-TP) parameters:
                 $SF_k$,$TP_k$=TOPSIS($w_{l,1}, .., w_{l,n}$)
            else
                config=true;
                 Configure the device with $SF_k$ and $TP_k$.
            end
        end
    end
end
Output: Set of resources reserved for each slice $l$.
        (SF-TP) parameters configuration for each device $k$.
```

One primar method (*static*) is to statically configure all the devices with the same SF and TP configuration. The latter has a constant complexity of $O(1)$ due to its simplicity. Similarly, same complexity analysis is applied for dynamic random (DR) and dynamic adaptive (DA) methods because in DA, centralized LoRa servers assign a specific TP value based on the SF assigned for the device. The latter is determined based on the distance between the device and its closest GW. Whereas in *DR*, the controller randomly select SF and TP values for all IoT devices in the network. The complexity of the proposed dynamic algorithm supported by (TOPG) is compared to the one supported by an optimal method (*optimal*). The latter includes a complete TOPSIS algorithm where all alternatives are tested with each including a different combination of SF and TP parameters. The complexity of the *optimal* algorithm is calculated as follows: an attribute normalization and weighting which result is $O(n^2)$, the algorithm complexity ranking which result is $O(1)$, the complexity of a positive-negative ideal solution and the distance to alternative solutions is $O(n)$. Hence, the overall complexity of the *optimal* and the proposed *TOPG* configuration is $O(n^2)$ [51]. However, instead of testing all possibilities of SF and TP configurations with the *optimal* algorithm, complexity is reduced in *TOPG* because the server reduces the search space to SF values that respect the guaranteed bit rate threshold. This reduces computation time without highly affecting QoS performance as will be shown in the following section.

III.5 Simulation Results

In uplink, centralized LoRa servers enable the opportunity to make efficient slicing decisions and optimum parameters configuration based on the knowledge of the data in the buffer of each LoRa device. We implemented our methods in the open source NS3 simulator [83] using LoRa model that was firstly developed by authors in [73]. The first section of **Table III.1** gives a brief of LoRa parameters implemented in this work. Simulations are replicated 50 times with 95% confidence interval and are realized in realistic LoRa scenarios. We assume that devices are defining a random time for transmission but periodically uploading small packet payloads of 18 Bytes following to the work done in [10]. LoRa devices and gateways are both placed over a cell of 7.5 KM radius based on a uniform random distribution. Each device is configured with spreading factors that varies from 7 to 12 when uploading traffic to LoRa GWs. Each GW is characterized by 8 receiving channels wih each channel having a bandwidth of 125 kHz in the 867-868

MHz european sub-band.

The second section of **Table II.1** summarizes LoRa energy model parameters. Based on the **Eq. III.8** below [83], we seek to evaluate the energy consumed when we increase the number of LoRa devices in each slice.

$$E_{k,l,m} = \frac{p_i^{tx} + p_i^{rx}}{V + epa}.d_{tx/rx} \tag{III.8}$$

where $E_{k,l,m}$ is the energy consumed by an IoT device, V the LoRa supply voltage, epa the amplifier's added efficiency, d_{tx} the duration of transmission, p_i^{rx} the power of reception and p_i^{tx} the power of transmission that varies between 2 and 14 dBm depending on the configuration strategy adopted. We integrate an energy module for the LoRa module in NS3 similar to the one that already exists for Wifi and we applied energy parameters and the power model specified for LoRa in [10] and [15]. In the following, we start by a proof of isolation and we highlight the importance of finding proper SF-TP combination with a parameters study in which we focus on showing the impact of SF and TP on energy consumption, mean PLR and the percentage of devices that respected GBR and PDB.

III.5.1 Parameters Study

In this section, we investigate the performance of each slice when we put in place different SF-TP configuration strategies for a fixed number of 300 devices. We first study *static* configurations in which all devices in the cell are configured with one of the following SF-TP combinations (i.e., SF7-TP2, SF8-TP5, SF9-TP8, SF10-TP11, SF11-TP14 and SF12-TP14). Then, we study the impact of TP variation for *static* configuration compared to three types of dynamic configuration strategies namely, *DR* where each device randomly picks a SF and TP values, *DA* where each LoRa device dynamically adapts device parameters to one of the SF-TP configurations depending on the highest receiving power measured from the gateway and we compare them with *TOPG* where dynamic slicing is supported with the proposed GMM and TOPSIS optimization.

Simulation Parameters	
Simulation Time	600 seconds
Slicing Interval Time	50 seconds
Cell Radius	7.5 KM
Number of replications	50
LoRa devices and GWs distribution	Random Uniform
Propagation loss model	Log-distance
Bandwidth	125 kHz
Spreading Factor	{7,8,9,10,11,12}
Confidence intervals	95%
European ISM sub-band	863-870 MHz
Power Consumption Parameters [10] [15]	
Battery Maximum Capacity	950 mAh
LoRa Supply Voltage	3.3V
Amplifier Power's added Efficiency	10%
Connected (Tx/Rx-SF7 to SF12)	1.58 to 25.11 mW
Standby	0.09 mW
Sleep	0 mW

Table III.1: Chapter III simulation parameters

III.5.1.1 Proof of Isolation

The very first step before investigating the strategies that can be used to configure SF and TP parameters is to prove the isolation concept between virtual slices in LoRa. Assuming that all devices are transmitting with the same DA configuration, we consider a single LoRa GW scenario in which we fix 20 LoRa devices for URA slice and we increase the number of devices. Therefore, all the devices that are left are assigned now to RA and BE slices. **Fig. III.4** proves the isolation concept because when the number of devices increases in RA and BE slices, URA slice members were not affected and the percentage of PLR remained constant and nearly null whereas PLR increased in RA and BE slices in a more congested scenario.

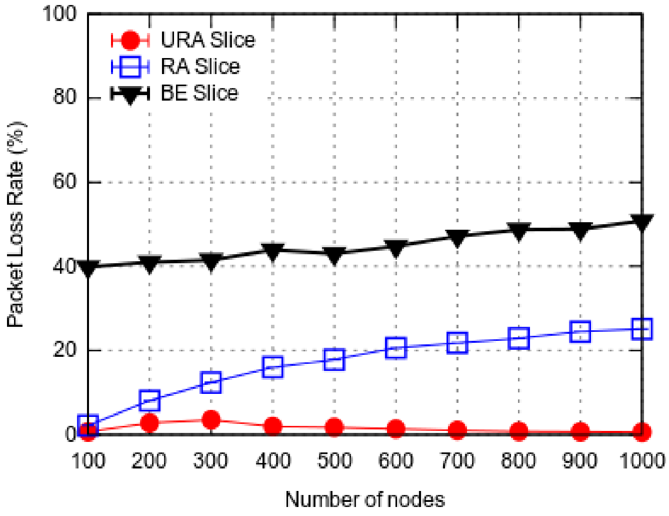


Figure. III.4: Proof of isolation

III.5.1.2 Parameters study with Static SF-TP Configuration

Performance comparison between various *static* configuration strategies is summarized in **Table III.2** below and evaluated in terms of QoS for a fixed packet transmission interval. When *static* configurations are adopted, all devices in the cell are configured with one of the following SF-TP combinations (i.e., SF7-TP2, SF8-TP5, SF9-TP8, SF10-TP11, SF11-TP14 and SF12-TP14). Results show that increasing the SF improves QoS metrics in terms of throughput and delay except for SF11 and SF12 where performance degrades tremendously. With high SF configurations, sensitivity is improved but the energy increases as well because with this configuration IoT devices occupy the spectrum for the longest time on air. This explains the increase in PLR and the probability that packets with same SF interfere upon transmission. However, with small SF configurations, energy is reduced with an improved QoS performance compared to high SF configurations. However, more than 50 % are lost due to lack of sensitivity, which means that a large number of packets are lost because they were not successfully received and decoded by the gateway.

Regarding QoS, increasing the SF reduces the throughput and increases the transmission delay. This explains why the percentage of devices that respect PDB decreases due to the increase in transmission delay. However, knowing that throughput decreases when SF increases, it is noteworthy to mention that the percentage of devices that respect GBR is not affected and improves with SF. This is because a higher SF with higher TP helps more devices to deliver the required throughput while improving at the same time packets sensitivity. This clearly explains the low values in PLR and highlights the trade-off that some configurations deliver in terms of QoS, reliability and energy. Therefore, we pursue this study with $(SF9 - TP8)$ configuration due to its trade-off performance that this configuration provides between QoS, energy consumption and having the best overall PLR% between the ones simulated with *static* strategies.

	Slice Name	Static					
		SF7-TP2	SF8-TP5	SF9-TP8	SF10-TP11	SF11-TP14	SF12-TP14
Devices that respect GBR (%)	Overall	2.9	6.21	14.65	23.08	0	0
Devices that respect PDB (%)	Overall	41.15	30.7	13.85	12.3	0	0
Mean Packet Loss Rate (%)	Overall	78.37	58.68	20.46	23.73	47.73	70.89
	URA Slice	6.94	6.80	10.23	3.33	5.33	5.94
	RA Slice	10.34	10.89	16.91	10.50	10.61	18.22
	BE Slice	82.71	82.31	72.87	86.16	84.07	75.84
Mean Energy Consumption (mJ)	Total	0.06	0.2	0.73	1.47	3.99	4.41
	URA Slice	0.01	0.04	0.16	0.28	0.67	0.74
	RA Slice	0.02	0.06	0.23	0.55	1.07	1.47
	BE Slice	0.03	0.1	0.35	0.64	2.26	2.21

Table III.2: Parameters study with static SF-TP configurations strategies

III.5.1.3 Parameters study with Dynamic SF-TP Configuration

After defining $(SF9 - TP8)$ as the best *static* configuration, we compare the latter to dynamic configurations. First, we highlight in this study the impact of increasing TP for *static* configurations before comparing its performance to DA, DR and the proposed $TOPG$ method. Based on the results shown in **Table III.3** below, one can conclude the importance of efficiently identifying TP parameter due to its direct impact on QoS performance metrics. The results of each slice show the efficiency of URA compared to RA and BE slices in terms of reliability and energy consumption due to slicing priority consideration in MLE resource reservation mechanism.

	Slice Name	Static-SF9					Dynamic		
		TP2	TP5	TP8	TP11	TP14	DR	DA	TOPG
Devices that respect GBR (%)	Overall	6	9.35	14.65	23.04	37.67	7.65	16.75	60.99
Devices that respect PDB (%)	Overall	0.45	1.7	13.85	19.32	29.86	76.3	85.73	94.8
Mean Packet Loss Rate (%)	Overall	61.77	45.96	20.46	12.3	9.59	20.86	12.26	4.37
	URA Slice	6.85	8.75	10.23	9.45	3.27	12.32	0.67	6.18
	RA Slice	15.54	16.00	16.91	15.24	5.84	23.69	0.97	11.13
	BE Slice	77.61	75.24	72.87	75.31	90.89	64	98.37	82.69
Mean Energy Consumption (mJ)	Total	0.18	0.37	0.73	1.46	2.91	3.53	1.04	1.8
	URA Slice	0.04	0.08	0.16	0.31	0.62	0.64	0.22	0.25
	RA Slice	0.06	0.12	0.23	0.46	0.92	1.1	0.33	0.49
	BE Slice	0.09	0.17	0.35	0.69	1.38	1.80	0.49	1.06

Table III.3: Complete Parameters Study with static and dynamic SF-TP configuration strategies

Increasing TP for SF9 configuration will increase packets arriving above sensitivity and improves the rate of devices that guaranteed delay and throughput on the expanse of energy consumption. This highlights the motivation for optimizing SF and TP parameters and the utility to sometimes increase TP for an IoT device, if the latter improves its QoS with respect to GBR and PDB thresholds defined for the slice it belongs to. Based on what was previously mentioned, IoT devices are configured with the highest TP for *static* ($SF9 - TP14$) configuration because it gives the best QoS performance for LoRa slices. The latter will be evaluated in depth in the following section compared to DA, DR and the proposed $TOPG$ methods.

Regarding dynamic configurations, DA was the best strategy in terms of energy compared to DR and $TOPG$ because in DA, the centralized server dynamically configures LoRa devices with one of the SF-TP combinations defined by LoRa. It measures the receiving power that a GW gets from the device depending on its position and configure the parameters accordingly. The advantages that dynamic configurations brings to LoRa are two-fold: first, depending on how far the device is from the gateway, a smaller distance requires a smaller SF configuration which also mean smaller TP and energy consumption. Secondly, the fact of adopting different SFs configuration reduces the probability of collisions and the percentage of packets lost due to interference. However, similar to *static* configurations, DA is weak in terms of QoS. This is also due to unefficient SF-TP distribution where it could be useful to improve QoS by keeping the same SF with higher TP value instead of increasing both SF and TP as it's done in *DA* method. Moreover, when devices are close to the gateway, it could be also interesting to reduce the TP to save energy without degrading QoS performance of IoT devices. *TOPG* results in **Table III.3** clearly show the potential that this method brings and requires further evaluation in complete simulations due to the trade-off results that were achieved in terms of QoS, reliability and energy consumption.

III.5.2 Performance Evaluation of SF-TP Configurations

Following to previous simulation results, we focus in this section on evaluating the proposed *TOPG* configuration method that proved its worthiness for this study. We run now simulations starting by 100 devices over a network of four gateways managed by a centralized LoRa server and we increase the number of devices until the maximum number connected to a single gateway is reached and limited to 1000 devices, as shown in the scalability study in [55]. A load of one is emulated due to the legal duty-cycle limitations of 1% in the European region [5].

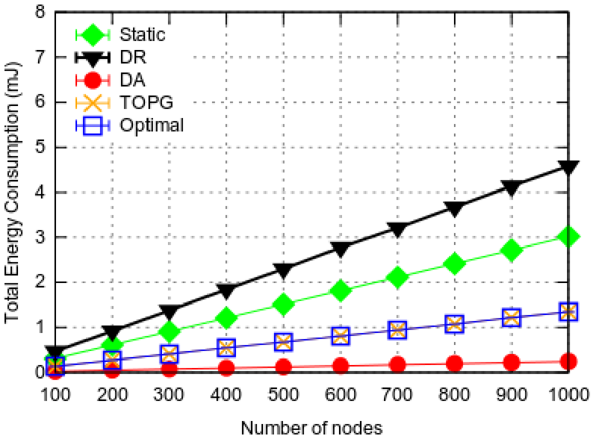
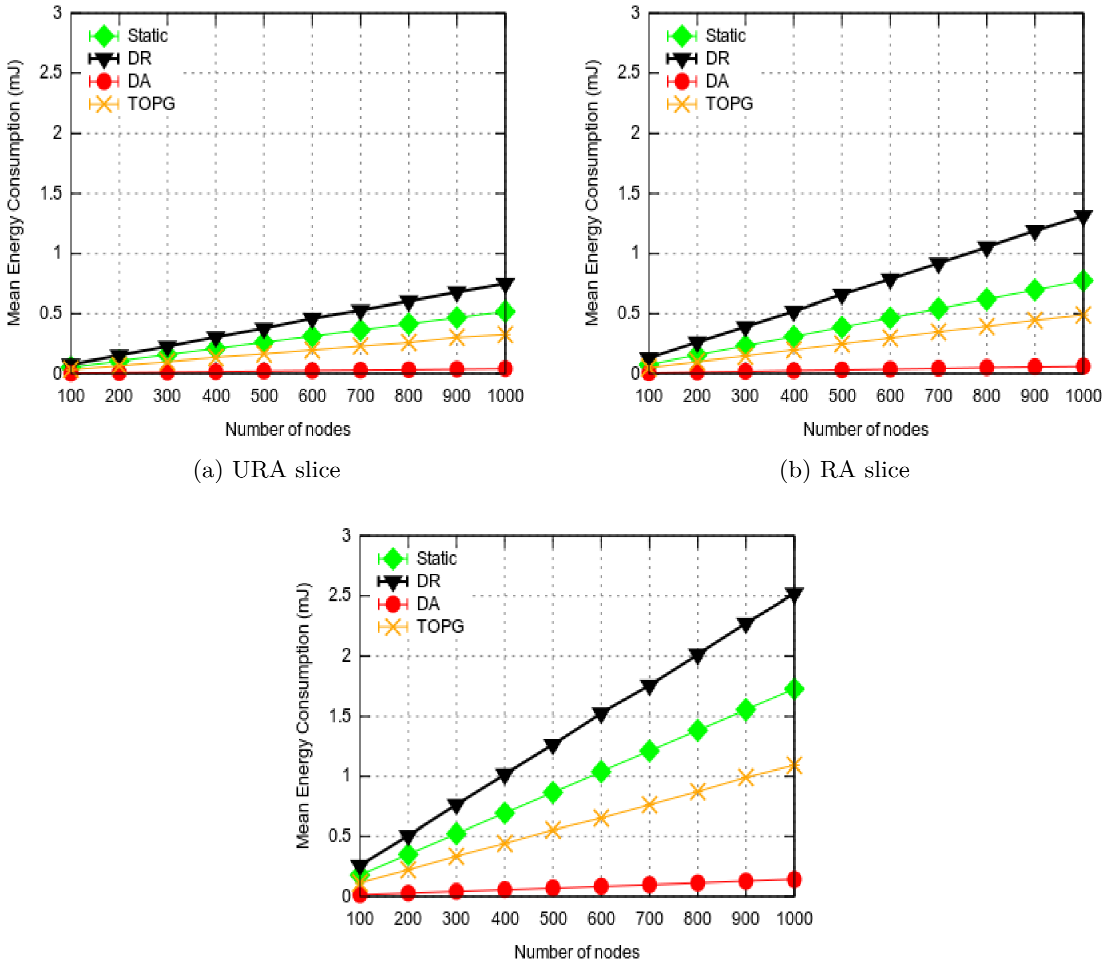


Figure. III.5: Total energy consumption variation

III.5.2.1 Total Energy Consumption

In **Fig. III.5**, when the number of devices increases, the total energy consumed increases as well regardless of the adopted SF-TP configuration. *DR* scored the highest energy consumption whereas *DA* was the most energy efficient method because it configures for each device the minimum TP required. Moreover, it is normal that *TOPG* algorithm consumes more energy because it configures SF and TP parameters while also considering QoS requirements of IoT devices in each slice.



(a) URA slice

(b) RA slice

(c) BE slice

Figure. III.6: Mean energy consumption in each slice with different SF-TP configurations

For further investigation, energy consumption is evaluated in each slice which also increased when the number of LoRa devices increases. In **Fig. III.6a**, *URA* slice members scored the lowest energy consumption between the simulated slices. The reason for this result is due to the higher impact that SF parameter provides by letting IoT devices occupy the spectrum for a smaller duration of time even if configured with higher *TP* values. This also explains why even in *RA* and *BE* slices, *TOPG* always had a higher energy consumption than DA and lower than DR and *static* configuration methods. RA and BE slice members consumed more energy compared to URA as shown in **Fig.** III.6b and **Fig.** III.6c respectively. This returns to GMM method that considers a slice-based configuration that gives higher importance for reliability and QoS in utility calculations. Hence, a higher weight is provided for QoS and reliability that forces delay-sensitive devices to take the most reliable gateway with the lowest SF-TP values compared to RA and BE slice members.

III.5.2.2 Packet Loss Rate

In this section, packet loss rate for each configuration algorithm is evaluated. Results shown in **Fig.** III.7 prove the efficiency of the proposed optimization method in reducing PLR compared to static and dynamic configuration strategies. With *static* ($SF9-TP14$) configuration, PLR reached more than 30 % of packets due to congestion. However, the worst result was scored with DR method where IoT devices lost approximately 40 % of their packets due to wrong configurations that lead to intra-SF and inter-SF collisions. The *optimal* configuration had the lowest PLR percentage between the simulated strategies but with higher complexity compared to the proposed $TOPG$ configuration. This puts the latter as a trade-off solution between performance and computation time.

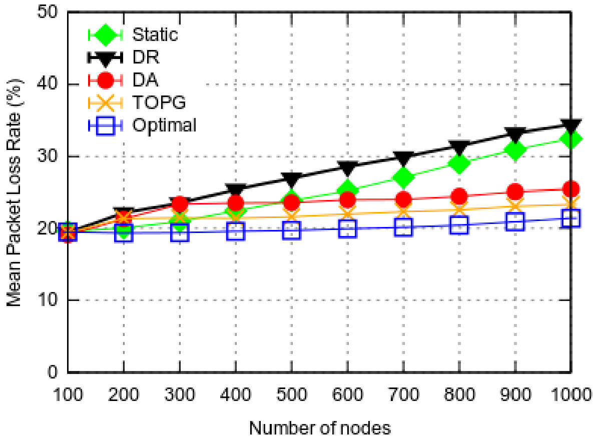
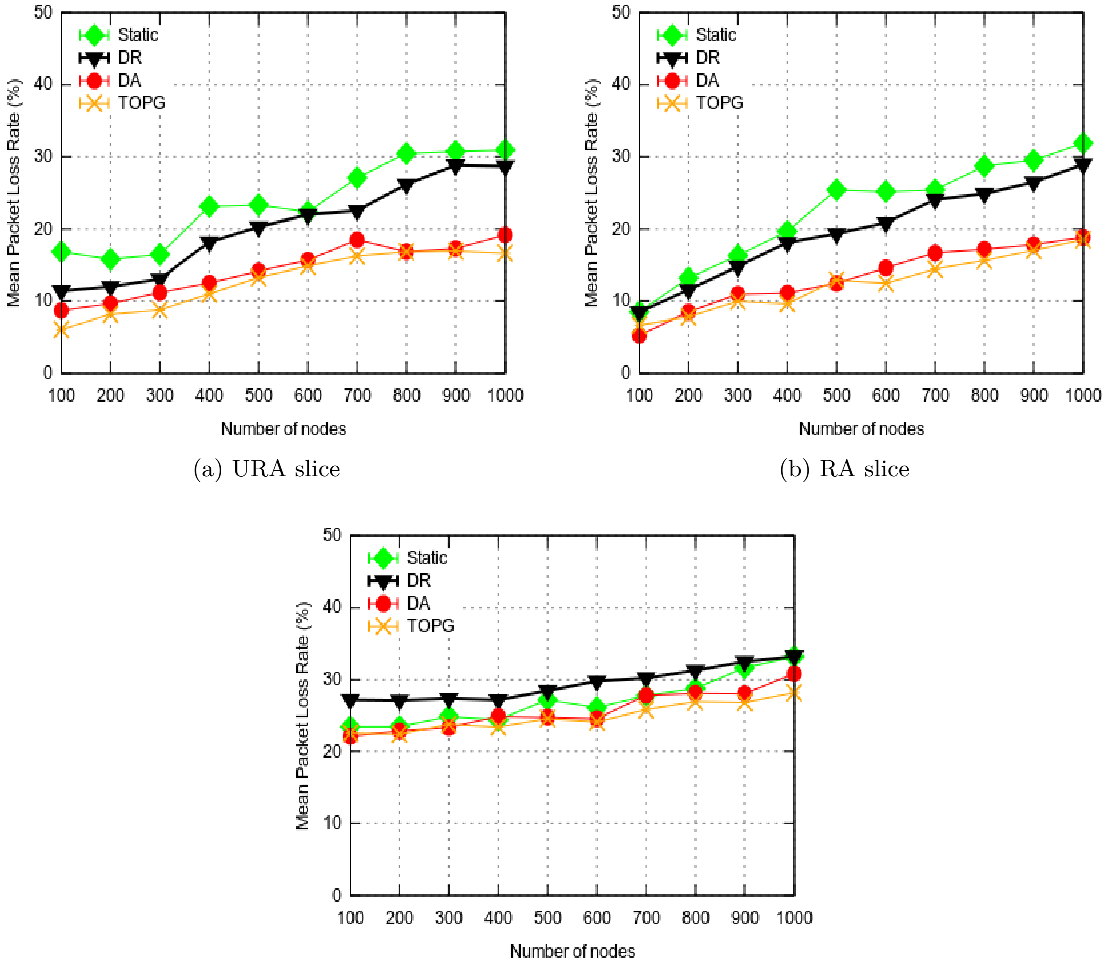


Figure. III.7: PLR variation

We also look towards mean PLR results in URA, RA and BE slices illustrated in **Fig.** III.8a, **Fig.** III.8b and **Fig.** III.8c respectively. Here, URA and RA slice members requiring urgent and reliable communications are more prioritized in terms of resource reservation than the best effort slice resulting lower PLR regardless of the method adopted for SF-TP configuration. This returns to the efficiency of the estimation method that avoids resource starvation and dynamically reserves physical channels on LoRa gateways following to the throughput requirements of each slice members. Additionally, the efficiency of the proposed configuration method can also be concluded which gave the lowest PLR with *TOPG* with a rate that did not bypass 20% in URA and RA slices and 30% in the BE slice.



(a) URA slice

(b) RA slice

(c) BE slice

Figure. III.8: Mean PLR in each slice with different SF-TP configurations

III.5.2.3 Percentage of Unserved devices in Delay

In **Fig. III.9**, *static* configuration had the worst results with 15% of devices that did not respect their delay thresholds. Results of *DR* and *static* configurations were nearly similar and return to their static and random configuration nature that do not take into consideration the link quality neither QoS requirements of IoT devices. With *DA* configuration, IoT devices had much better results compared to the previous configurations with a rate that did not exceed 10% of the devices violating their delay thresholds. However, unlike *DA* where the controller jointly increases or decreases SF-TP combination for an IoT device, the proposed *TOPG* algorithm searches for the best SF-TP combination based on the objectives and the weight defined by GMM method. *TOPG* sometimes modify TP for a device instead of increasing both SF and TP parameters like in the case of *DA* configuration. This explains the slight improvement and the decrease in the percentage of devices that violated their PDB in *TOPG* with less computation complexity than the *optimal* configuration.

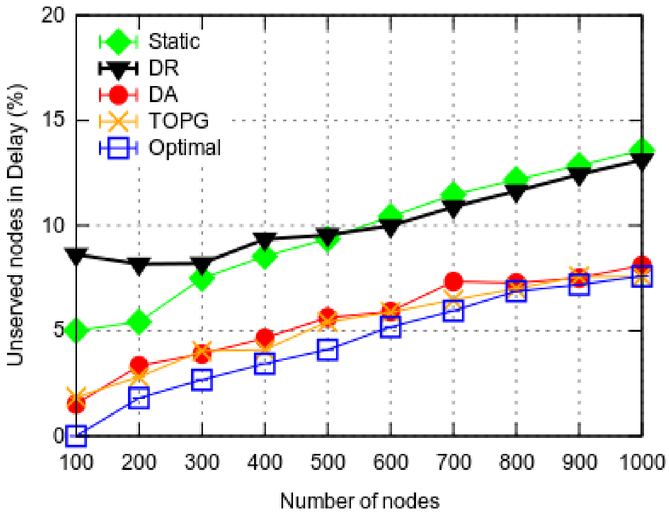


Figure. III.9: Percentage of unserved nodes in delay

III.5.2.4 Percentage of Unserved devices in Throughput

Further improvement in throughput is achieved in **Fig. III.10** where *TOPG* came as the second best configuration method behind the *optimal* algorithm. The former scores nearly similar results with less computation time. With both *TOPG* and *optimal* algorithms, the rate of devices that did not guarantee their throughput did not exceed 30% even in a very congested scenario. This mainly highlights the efficiency of using TOPSIS instead of testing all SF and TP combinations. Moreover, *static* and *DR* configurations had the worst results with a rate that exceeded 50% of the devices that violated the GBR defined in each slice. With *DA*, smaller SF values provide an achievable throughput that can be sometimes very high compared to the one that needs to be guaranteed. This is also true with smaller SF parameters where in both cases, an IoT device with *DA* configuration is assigned a specific TP for each SF parameter. However with the proposed algorithm, *TOPG* provides the guaranteed throughput with an efficient SF or TP variation. With *TOPG* optimization, a proper SF and TP combination is found that guarantees throughput while saving lots of energy for each slice members.

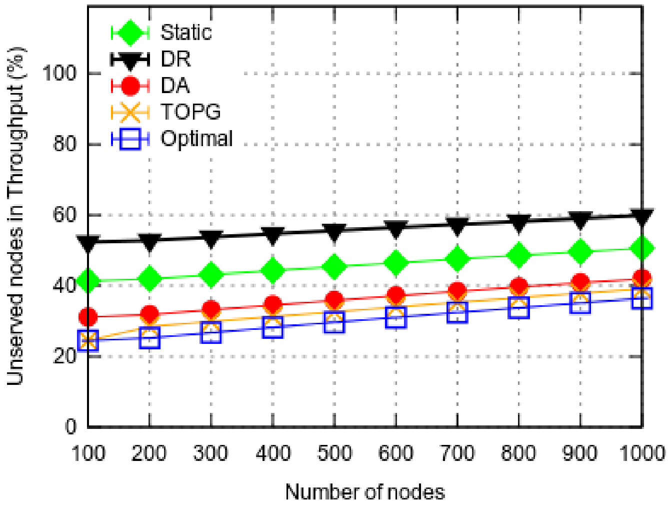


Figure. III.10: Percentage of unserved nodes in throughput

III.6 Conclusion

This chapter highlights the utility of supporting the adaptive dynamic slicing strategy with a slice-based parameters optimization that searches for the best SF and TP configuration for each device depending on the slice that it belongs to. Results show major improvement in terms of QoS, reliability and energy consumption when each device is configured with the proper SF and TP combination. More specifically, the rate of packets lost decreases from 50% to less than 30 % for the same number of IoT devices. However, it is expected that the number of devices will increase and bypass 1000 devices, found to be the maximum capacity in the scalability study realized by authors in [55]. LoRaWAN will suffer from congestion when the number of devices increase in the network. Hence, we believe that centralized servers will practically face major difficulties in managing and properly isolating Lora slices as well as configuring each IoT devices with the proper parameters configuration. Hence, the goal in the next chapter is to propose a distributed strategy supported by SDN which should meet scalability and capacity requirements of LoRaWAN in large scale IoT deployments.

Chapter IV

Distributed Network Slicing in large scale LoRaWAN

Summary

IV.1 Introduction

After evaluating in Chapter II the assets and the usability of network slicing in guaranteeing QoS for LoRa devices in terms of urgency and reliability, we have shown next, in Chapter III, that further improvement can be reached if an optimized SF and TP distribution is taken into consideration. However, due to the vast popularity that IoT is gaining, estimations forecast that 20 to 30 billion IoT devices will be connected by 2022 [36]. There is some doubts about how to deal with the rapid development of LoRaWAN knowing that the current LoRa architecture won't be capable of supporting upcoming scalability challenges in large scale LoRa deployments despite the advantages brought to LoRaWAN with our previous contributions.

In [77], Mikhaylov et al. present an analytical scalability analysis that measures the maximum throughput that can be transmitted by a single LoRa device. The capacity of the latter is analyzed by Augustin et al. in [5] as the superposition of independent ALOHA-based networks. Moreover, Bora et al. in [14] performed practical experiments to study the limit on the number of transmitters supported in LoRa based on an empirical model and built LoRaSim simulator with the goal of studying LoRaWAN scalability. Unlike [14], Van den Abeele et al. in [115] adopted the LoRaWAN MAC protocol in NS3 module to analyze its scalability with thousands of end devices and showed the impact of downstream traffic on packet delivery ratio (PDR) of confirmed upstream traffic. In [89], Petajajarvi et al. showed that LoRa scalability can be improved using an optimized configuration of LoRa parameters (spreading factor (SF), transmission power (TP) and coding rate (CR)). Any misconfiguration of one of these parameters will lead to degradation in PDR and unfairness in LoRaWAN [94] network performance. The current cloud-based server cannot meet scalability challenges in properly allocating network resources and configuring IoT devices as their number in the network increases. Hence, flexibility in managing network resources is required using emerging technologies in IoT, namely network slicing and software defined networking (SDN), to provide heterogeneous QoS requirements through isolated End-to-End (E2E) network slices and an optimized resource allocation and network configuration strategies.

Knowing that SDN in itself is not the solution for the slicing problem, it provides the potential of enabling simplified resource management, distributed control and communications between LoRa GWs. The crucial role that SDN plays in improving IoT network is highlighted in the literature in terms of security [40], improving transmission quality [110] and scalability through cloud-based solutions [113]. However, even with SDN, it is impractical to assume that the centralized network server is capable of acknowledging messages received from billions of devices given their limited physical bandwidth and computational capacity [48]. Luo et al. in [71], proposed an SDN-based testbed with semi-centralized and distributed SDN control for underwater wireless sensor networks. Moreover, Reynders et al. [96] proposed a distributed scheduling solution to improve LoRaWAN reliability and scalability. Hence, decentralized optimizations could be the solution for this slicing problem. There has been previous attempts to evaluate decentralized architecture in LoRaWAN. Lin et al. proposed in [68] a conceptual architecture design of a blockchain built-in solution for LoRaWAN servers to improve network coverage and build the trust of private network operators. Moreover, Durand et al. improved in [32] LoRaWAN security and its resilience against threats by practically implementing decentralized LoRa architecture using a blockchain-connected packet forwarding application. In [88], Pankratev et al. compared IoT technologies for data exchange in decentralized systems and highlight the advantages of using distributed architecture using Bluetooth, ZigBee and WiFi technologies but scheduled decentralized LoRaWAN for their future research work.

In large scale IoT, devices cannot be controlled by a single network entity. Hence, due to the lack of information for a GW regarding IoT devices managed by the other GWs, game theory can be used, as a popular framework [53], to effectively analyze the interactive decision making of GWs with conflicting interests [103]. Each GW tries to find the best resource reservation strategy for its virtual slices and the best parameters configuration for devices in its range. Multiple works proposed game-theoretic models

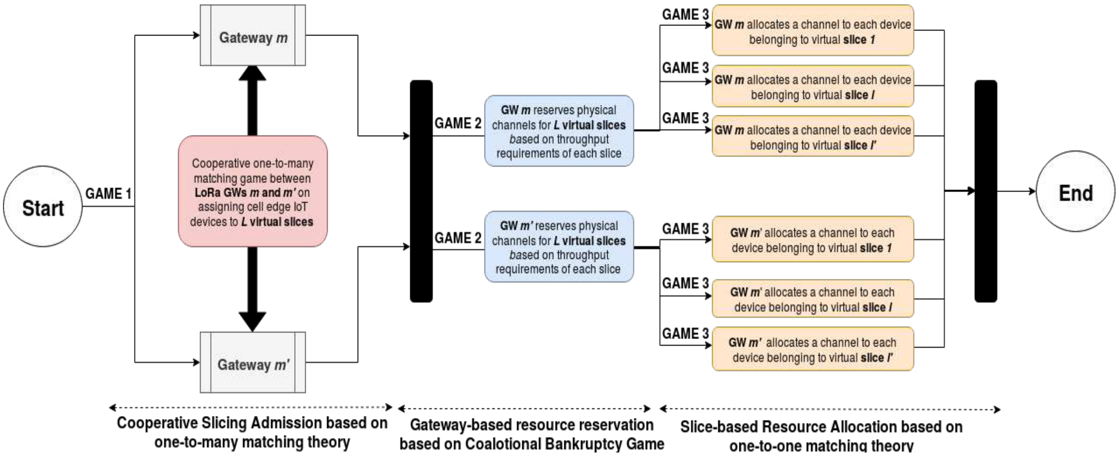

Start
GAME 1
Gateway m
Cooperative one-to-many matching game between LoRa GWs m and m' on assigning cell edge IoT devices to L virtual slices
Gateway m'
GAME 2
GW m reserves physical channels for L virtual slices based on throughput requirements of each slice
GAME 2
GW m' reserves physical channels for L virtual slices based on throughput requirements of each slice
GAME 3
GW m allocates a channel to each device belonging to virtual slice 1
GAME 3
GW m allocates a channel to each device belonging to virtual slice l
GAME 3
GW m allocates a channel to each device belonging to virtual slice l'
GAME 3
GW m' allocates a channel to each device belonging to virtual slice 1
GAME 3
GW m' allocates a channel to each device belonging to virtual slice l
GAME 3
GW m' allocates a channel to each device belonging to virtual slice l'
End
Cooperative Slicing Admission based on one-to-many matching theory
Gateway-based resource reservation based on Coalotional Bankruptcy Game
Slice-based Resource Allocation based on one-to-one matching theory

Figure. IV.1: The proposed distributed multi-game for slicing admission, resources reservation and resources allocation

to optimize resource management problems in distributed systems. Gu et al. [47] tackled radio and computational resource allocation problem to optimize users satisfaction and improve network performance in IoT fog computing. In [17], Bui et. al propose an optimization method for smart traffic lights control and improve traffic flow in real time by applying game theory. Touati et al. [112] proposed a matching game approach for mobile user association and resource allocation in IEEE 802.11 wireless networks. Moreover, a distributed resource allocation and orchestration approach is proposed by Doro et al. [33] to allow dynamic and flexible resource management in softwarized networks. Chen et al. [20] formulated two resource management games and proposed distributed algorithms to optimize link selection and power allocation and improve network localization. However, few works addressed resource management problem in LoRaWAN, Sharma et. al in [105] optimized resource allocation among available network servers by forming self-enforcing agreement via game theory modeling. Moreover, Haghighi et. al proposed in [49] a GW-centric distributed approach for radio selection based on game theory to minimize energy consumption at LoRa sensor nodes.

Unlike our previous contributions in Chapter II and Chapter III where we tackled resource reservation problem in centralized LoRa architecture. In this chapter, we will try to prevent potential challenges that may appear due to the increasing congestion in large scale IoT deployments by leveraging computational intelligence to LoRa gateways and moving closer to the edge. This should improve network performance and reduce computational complexity in next generation IoT networks. The proposed multi-game slicing admission control, resource reservation and resource allocation is illustrated in **Fig. III.3**. Our main contributions with respect to the surveyed literature are stated as follows:

1. We propose a coalitional game (**GAME 1**) where LoRa GWs coordinate to improve network reliability. GWs assign devices to the requested slice and compete when assigning cell edge devices to the most efficient virtual slice. This game provides a better flexibility in managing traffic coming from heterogeneous IoT applications and guarantees their required QoS with complete isolation between each virtual slice.

2. We formulate the slicing problem as a Bankruptcy game (**GAME 2**) and propose an inter-slice resource reservation that builds on previous matching game results. Here, on each GW, coalitions including slice members compete for gaining access to LoRa physical channels. The goal of this scheme is to avoid channels starvation by providing fair resource reservation for each slice.

3. We propose an intra-slice resource allocation based on one-to-one matching theory (**GAME 3**) with optimized configuration. In each slice, the proposed method efficiently configures SF and TP parameters for a device before being assigned to a LoRa channel.

The remainder of this chapter is organized as follows. We devote Section IV.2 for describing the distributed SDN-based network slicing architecture for LoRaWAN and defining the network slicing model. In Section IV.3, the multi-objective network slicing problem is established with respect to various constraints. Section IV.4 presents the coalitional multi-game theory proposed for resource reservation and supported by an optimized slice-based resource allocation algorithm. The latter is simulated over the LoRa module of NS3 simulator [73] before evaluating its performance and analyzing simulation results carried out through various scenarios in Section IV.5. Finally, Section IV.6 concludes the paper.

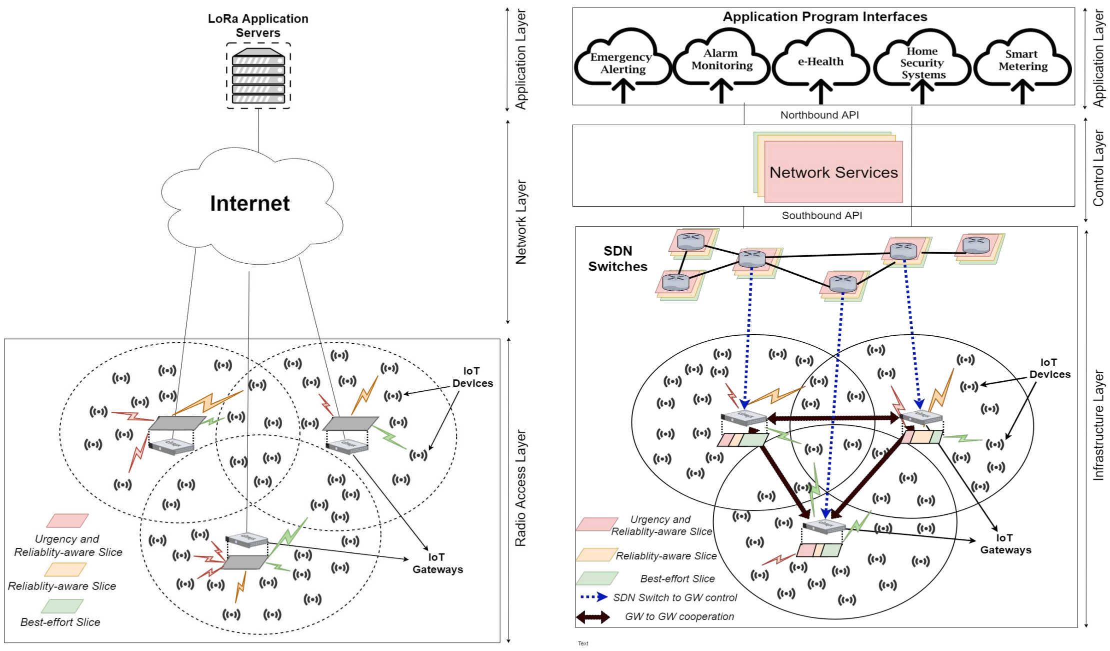
IV.2 Distributed SDN-based architecture for IoT

In this section, we first present the distributed SDN-based architecture compared to the centralized non-SDN LoRa existing actually in the IoT market. Similar to previous chapters, virtual slices are next defined and integrated in densified LoRa networks before finally presenting the slicing system model.

IV.2.1 LoRa SDN-Based Architecture

In **Fig. IV.2a**, the standard LoRa architecture is illustrated and is originally designed as centralized and non-SDN architecture in which End-to-End (E2E) network slicing can be implemented and managed by the centralized network server with a global overview of the network. The latter is responsible for estimating and reserving physical resources on LoRa GWs for each slice based on QoS requirements of IoT devices. However, in large scale dense deployments, network complexity significantly increases which degrades network performance specially when more edge devices are positioned in the range of multiple GWs simultaneously. Hence, communication reliability decreases leading to an increase in packets loss due to interference and misconfiguration of LoRa SF and TP parameters. This motivates the idea of integrating SDN in a distributed network slicing and enabling cooperation between LoRa GWs via SDN switches.

With the distributed SDN-based architecture, illustrated in **Fig. IV.2b**, LoRa application servers are replaced by application program interfaces (APIs) to provide efficient communication regarding which IoT services and applications LoRaWAN should be providing. SDN simplifies network management and enables distributed solutions for resource and network configurations. In this context, slicing admission control and resource reservation decisions are delegated to the GWs which cooperate between each others using game theory framework. Each IoT device randomly runs an application belonging to one of URA, RA or BE slices previously defined in chapter I and listed in **Table I.5**. The GW forms coalitions with devices having similar QoS requirements and assign each device to one of the channels reserved for his virtual slice.



(a) Centralized non-SDN architecture

(b) Distributed SDN-based architecture

Figure. IV.2: Centralized non-SDN vs Distributed SDN-based network slicing architecture in LoRaWAN

IV.2.2 Slicing System Model

In IoT, we assume that IoT devices are initially associated to the closest gateway in the network with ADR configuration. The latter does not influence the architecture but is a mechanism that optimizes LoRa SF and TP parameters based on its power received from the closest GW. However in ADR, QoS is not considered in each slice in terms of delay, throughput and reliability. Let $M = \{m_1, ..m_{m'}.., m_M\}$ be the set of LoRa GWs that cooperate to optimize network configuration with $K = \{k_1, ..k_{k'}.., k_K\}$ denotes the set of IoT devices that should be admitted to one of the L virtual slices with $L = \{l_1, ..l_{l'}.., l_L\}$ grouping $K_{l,m}$ IoT devices belonging to the same virtual slice on GW m. Upon resource reservation, GW physical resources are divided and virtually reserved for each slice $l \in L$ having a set of channels $C_{l,m}$ completely isolated on each GW $m \in M$. Improving IoT communications requires QoS consideration while decreasing at the same time energy consumption for each device. Therefore, this requires the formulation of a multi-objective optimization that considers resource reservation and allocation in a LoRa network slicing scenario.

IV.3 Problem Formulation

Optimizing LoRa parameters configuration and resource reservation impacts QoS, energy consumption and reliability performance for IoT devices belonging to a coalition with specific slicing priority sp_l. However, solving this multi-objective problem is proven to be NP-Hard in similar problems by Amichi et al. [4] for LoRa parameters configuration and by Liu et al. [69] for channel resources and power allocation. This problem is also more challenging due to the maximum number of physical channels q_m that can be used for transmission on each GW m. The goal is to first control IoT devices admission where some of these devices are positioned in the range of multiple GWs. The latter cooperate to assign these devices to the most appropriate virtual slice before optimizing the reservation and allocation of channel resources. We consider $\alpha_{k,c} \in \{0,1\}$ and $\beta_{l,m} \in \{0,1\}$ as two binary decision variables that respectively indicates the admission of device k to a channel $c \in C_{l,m}$ and the association of a slice l to GW m. Based on SF and TP configuration, each device $k \in K_{l,m}$ achieves specific throughput and delay formulated in **Eq. IV.1** and **Eq. IV.2** below:

$$r_k = SF.\frac{R_c}{2^{SF}}.CR \quad with \quad k \in K_{l,m} \tag{IV.1}$$

$$d_k = \frac{L}{r_{k,c}} \quad with \quad k \in K_{l,m} \quad and \quad c \in C_{l,m} \tag{IV.2}$$

Hence, the first objective is presented in **Eq. IV.3** which consists on improving QoS of a slice when QoS of all its members are also improved based on slice specific thresholds.

$$\begin{aligned} &Maximize \quad u_{QoS}^{K_{l,m}} \\ &with \quad u_{QoS}^{K_{l,m}} = \sum_{k \in K_{l,m}} \alpha_{k,c}(\overline{r_{k,c}} + (1 - \overline{d_{k,c}})), \\ &\forall c \in C_{l,m}, \forall l \in L, \forall m \in M \end{aligned} \tag{IV.3}$$

where $u_{QoS}^{K_{l,m}}$ denotes the QoS metric that measures satisfaction rate of slice l members over a LoRa GW m in terms of throughput and delay normalized into $\overline{r_{k,c}}$ and $\overline{d_{k,c}}$ respectively. Configuring the device with lowest SF and TP configurations may lead to packet reception errors due to inter and intra-SF interference. The former denoted as $PLR'_{k,c}$, depends on collisions that happen on a GW channel between two devices configured with the same SF based on random access formula [109]. The latter, denoted as a binary variable $PLR''_{k,c}$, depends on low Signal-to-interference-plus-noise ratio (SINR) [73] and indicates if the transmitting packet survives interference table [44] that comes from other LoRa transmissions with each having different SF and TP configurations. The packet survives interference with all interfering packets if, considering all combinations of SF, a higher power margin value (dB) is satisfied than the corresponding co-channel rejection value. Moreover, $PLR'''_{k,c}$ denotes a binary variable that indicates if a packet arrives to the GW above or below sensitivity thresholds [73]. The latter is an additional factor that leads to a loss of a packet due to the lack of sensitivity. Thus, finding the proper configurations of a device is crucial because increasing SF will also increase receiver's sensitivity allowing a packet to be transmitted at a wider range.

Based on what was previously mentioned, we define $PSR_{k,c}$ formulated in **Eq. IV.4** below, with the objective of maximizing the packet success rate $PSR_{k,c}$ of a packet transmitted by an IoT device $k \in K$ over a channel $c \in C_{l,m}$.

$$\begin{aligned} &PSR_{k,c} = (1 - PLR'_{k,c}) + PLR''_{k,c} + PLR'''_{k,c} \\ &with \quad k \in K \quad and \quad c \in C_{l,m} \end{aligned} \tag{IV.4}$$

The second objective is defined in **Eq. IV.5** below as maximizing the reliability of a transmission by optimizing parameters configuration if the latter improves $PSR_{k,c}$ of a transmission.

$$\begin{aligned} &Maximize \quad u_{Rel}^{K_{l,m}} \\ &with \quad u_{Rel}^{K_{l,m}} = \sum_{k \in K_{l,m}} \alpha_{k,c} PSR_{k,c}, \\ &\forall c \in C_{l,m}, \forall l \in L, \forall m \in M \end{aligned} \tag{IV.5}$$

where $u_{Rel}^{K_{l,m}}$ denotes the reliability metric that slice l members achieve over a LoRa GW m including sensitivity, inter-SF and intra-SF interference estimations. It is noteworthy to mention that overestimating SF, TP and CR configurations leads to an increase in energy consumption due to the longer activity time for an IoT device when uploading a packet with high SF configuration. Therefore, we also consider energy in utility calculations based on the power model adopted in [15]. Two energy states for LoRa IoT device are assumed in which the energy of a transmission is formulated in **Eq. IV.6** below and computed based on $P_{k,c}^{tx}$ and $P_{k,c}^{sleep}$ power values respectively denoting the power consumed in active or in sleep mode.

$$E_{k,c} = P_{k,c}^{tx} T_{active} + P_{k,c}^{sleep} T_{sleep} \tag{IV.6}$$

The problem of minimizing energy consumption is transformed into a maximization problem following to **Eq. IV.7** below:

$$\begin{aligned} &Maximize \quad u_{Energy}^{K_{l,m}} \\ &with \quad u_{Energy}^{K_{l,m}} = \sum_{k \in K_{l,m}} \alpha_{k,c}(1 - \overline{E_{k,c}}) \\ &\forall c \in C_{l,m}, \forall l \in L, \forall m \in M \end{aligned} \tag{IV.7}$$

where $u_{Energy}^{K_{l,m}}$ denotes the energy consumed by a slice l over a LoRa GW m including $\overline{E_{k,c}}$, the normalized energy value of a packet that varies depending on SF and TP parameters configuration.

According to the multi-objective problem, QoS, reliability and energy objectives are turned into a single objective function with the goal to maximize the global utility value of each GW m. The latter is shown in **Eq. IV.8** below and formulated in subject to the constraints below:

$$\begin{aligned} &Maximize \quad U_m \\ &with \quad U_m = \sum_{l \in L} \beta_{l,m}(u_{QoS}^{K_{l,m}} + u_{PSR}^{K_{l,m}} + u_{Energy}^{K_{l,m}}), \\ &\forall m \in M \end{aligned} \tag{IV.8}$$

$$C1 : \sum_{c \in C_{l,m}} \alpha_{k,c} = 1, \forall k \in K \tag{IV.9a}$$

$$C2: K_{l,m} \cap K_{l',m} = \emptyset, \forall l, l' \in L, \forall m \in M \tag{IV.9b}$$

$$C3: K_{l,m} \cap K_{l,m'} = \emptyset, \forall l \in L, \forall m, m' \in M \tag{IV.9c}$$

$$C4: C_{l,m} \cap C_{l',m} = \emptyset, \forall l, l' \in L, \forall m \in M \tag{IV.9d}$$

$$C5: 0 \leq P_{k,c} \leq P_k^{max}, \forall k \in K, \forall c \in C_{l,m} \tag{IV.9e}$$

$$C6: \sum_{k \epsilon K} \alpha_{k,c}\beta_{l,m}r_{k,c} \leq R_c^{max}, \forall l \in L, \forall m \in M, \forall c \in C_{l,m} \tag{IV.9f}$$

Knowing that multiple virtual network slices are built on top of a common physical gateway, (IV.9a) ensures that each device should always choose exactly one and only channel reserved for a virtual slice $l \in L$ on GW m. (IV.9b) and (IV.9c) controls the formation of coalitions upon IoT devices admission into virtual slices. The former guarantees that two coalitions of devices belonging to slice l and l' do not have common IoT devices in the range of GW m whereas the latter guarantees that IoT devices belonging to the same slice l cannot be shared between two LoRa GWs $m, m' \in M$. Moreover, perfect isolation is guaranteed in (IV.9d) between two set of physical channels belonging to different slices $l, l' \in L$ over a GW m. The transmission power of each device is limited in constraint (IV.9e). And finally, constraint (IV.9f) guarantees that the sum of uplink traffic sent by slice members do not exceed the maximum data rate capacity of the slice that can be sent through each gateway. All the constraints previously mentioned should be respected by each GW in the following multi-game proposition. Due to lack of information between GWs in large scale LoRaWAN, a cooperative multi-game is next proposed to maximize the utility function of each GW when optimizing slicing admission, inter-slice resource reservation and intra-slice resource allocation.

IV.4 Proposed Approach

In this section, the proposed multi-game slicing admission control, resource reservation and resource allocation are summarized and illustrated in **Fig. IV.1**. The first step is formulated as a *cooperative slicing admission based on one-to-many matching game* to define how IoT edge devices are assigned to virtual slices and GWs. The result of the game is used as an input for *Bankruptcy game* to reserve physical channels on GWs based on throughput requirements of each slice members. Each GW virtually splits its bandwidth between various networks that are isolated with each having heterogeneous degree of importance in terms of QoS, energy and reliability. After reserving channels for each slice, an *inter slice resource allocation based on one-to-one matching game* is formulated between the set of channels reserved for a slice and its IoT devices members.

IV.4.1 Cooperative Slicing Admission via Matching Theory

A key problem in dense IoT deployments is mainly found in the edge of each cell where IoT devices are located in the range of multiple GWs simultaneously. However, the received power could not be considered as the only metric for devices slicing admission specially in large scale deployments where congestion increases and more packets are lost due to interference. Hence, finding the proper device configuration has a major impact on communications reliability and the probability of loosing a packet due to inter or intra-SF interference [95]. Therefore, based on the configuration of each device, cooperation between GWs becomes mandatory to control slicing admission where it may be useful to transfer a device initially from GW m to GW m' while keeping it assigned to the same virtual slice $l \in L$ if this move improves the probability of successfully decoding its packet. The slicing admission control problem is formulated to a college admission problem, introduced in [42], and resolved based on one-to-many matching theory. In this framework, three components are needed: 1) the set K of IoT devices acting as students; 2) the set M of LoRa GWs acting as colleges, and 3) the preference relations for IoT devices and GWs which are built based on the preferences over one another. IoT devices can be initially in the range of multiple GWs with the latter having a fixed quota $q_{l,m}$

on the number of channels that can be reserved for each slice. However, upon stability, each device has the possibility of being assigned to one virtual slice l on GW m. LoRa GWs forms initially L coalitions including IoT devices with the highest metric in each slice $l \in L$.

Pseudo-code 5 Cooperative Slicing Admission algorithm based on one-to-many matching theory

```
Input  : Set of gateways $M$ and slices $L$; Set of devices $K$;
         Set of channels $C_{l,m}$;
begin
    Construct $KLIST_{k,l,m}$,$\forall k \in K_l, l \in L$ and $m \in M$
    Construct $GWLIST_{l,m}$,$\forall l \in L$ and $m \in M$
    while $K$ devices are not on waiting lists do
        for each GW $m$ in $M$ do
            Receive preference of $K$ for each slice $l$
            GW $m$ creates new waiting list including best devices between
            the preferred and new applicants based on $q_{l,m}$
            GW $m$ rejects the rest of devices

        end
    end
    Each GW $m \in M$ forms $K_{l,m}$ coalitions in $S_{partial}$.
    $S_{partial}$=$\{K_{1,m}$,$K_{l',m}$,...,$K_{l,m}\}$.
    LoRa GWs cooperate to transfer IoT devices.
    while Stable $S_{final}$ not found do
        Preferred transfers of devices are indicated to GWs
        for each GW $m$ in $M$ do
            Receive transfer request of $k$ to slice $l$ on GW $m'$
            if $U_{l,m'} > U_{l,m}$ then
                Transfer device $k$ from slice $l$ on GW $m$ to slice $l$ on GW $m'$.
            else
                IoT device remains assigned to the same coalition $K_{l,m}$.
            end
        end
    end
end
Output: Stable matching $S_{final}$ of coalitions including $K_{l,m}$ devices assigned to slice $l \in L$ on GW $m \in M$.
```

The following matching game is modeled and summarized in the **Pseudo-code 5** with the goal to prove that a stable matching can be found between GWs and IoT edge devices. LoRa GWs cooperate to maximize the utility value of IoT devices by assigning edge devices to the slice l on GW m that offers the maximum reliability and respects QoS thresholds of the corresponding slice. Each GW $m \in M$ has an objective to reduce congestion and to form coalitions for each slice members in a way that maximizes its utility U_m. To resolve this *slicing admission game*, a preference relation $\succeq$ is first defined as a binary relation over the coalitions of devices belonging to a slice $l \in L$ on GW $m \in M$. Based on these preferences, IoT devices and GWs can rank one another. Each GW $m \in M$ defines a preference relation set $KLIST_{k,l,m}$ over the set of devices which are members of coalitions $K_{l,m}$, such that, for two devices $k, k' \in K_{l,m}$ and $k \neq k'$, the following **Eq. IV.10** is approved:

$$k \succeq_m k' \iff U_m(k) \geq U_m(k') \tag{IV.10}$$

where $U_m(.)$ is given by the utility computed for a device k which is affected by the other devices that exists in $K_{l,m}$ **(line 3)**. Moreover, the preference relation set $GWLIST_{l,m}$ which ranks LoRa GWs based on the utility that depends on the configuration adopted by each device **(line 3)** and evaluates its utility metric in terms of QoS, reliability and energy consumption. The primary stable matching solution $S_{primary}$ is found with the deferred acceptance method of Gale and Shapely [42] repeated until there is no pair of devices $k, k' \in K$ assigned to slice l on GWs $m \in M$ and $m' \in M$ respectively, although k prefers m' to m, i.e. $m' \succeq_k m$ and k' prefers m to m', i.e. $m \succeq_{k'} m'$ **(line 4-7)**. However, due to the dependency between the preferences of the students, the possibility to find an acceptable and a stable solution for the one-to-many matching game becomes more complex [100]. Hence, transferring a device from a coalition to another while keeping the device assigned to the same virtual slice is proposed to overcome this challenge. The transfer of devices between virtual slices is defined based on the framework of the coalitional game theory [53]. The latter is a pair (N, V) where:

- N is the finite set of players, i.e. IoT devices.
- V is the mapping that assigns devices for every coalition $K_{l,m}$ that groups devices belonging to slice l on GW m with $U_{l,m}$ denotes the utility value of each coalition.

The objective of this game is to enable the opportunity for LoRa GWs to cooperate and to make a decision on accepting or refusing to transfer the device from $K_{l,m}$ to $K_{l,m'}$ if the latter improves the utility of both coalitions **(line 8-17)**. The process is proven to convert to a final solution S_{final} [101] and is repeated until convergence. S_{final} includes the final coalitions $K_{l,m}$ of IoT devices assigned to each virtual slice $l \in L$ on LoRa GW $m \in M$ **(line 18)**.

IV.4.2 Bankruptcy Resource Reservation game

After defining coalitions of devices that belongs to each slice, each LoRa GW will reserve a number of channels for each coalition of devices belonging to the same virtual slice based on throughput requirements of the latter. In other terms, each slice experiencing higher traffic load requires a higher number of channels compared to the less loaded slice. However, the number of channels required by the sum of virtual slices is higher than the number of physical channels that actually exists for each LoRa GW. Therefore, the resource reservation problem is modeled as bankruptcy situation which tries to predefine in a fairly manner how to ration the amount of channels among the group of IoT devices belonging to the same slice with each having different demands in throughput. The key denotations and description of the bankruptcy game are defined in **Table IV.1**.

Variable	Bankruptcy Game	Resource Reservation Game
k	Total number of players	Total numbers of IoT devices in a range of GW
$K_{l,m}$	Set of players	Set of coalitions including IoT Devices belonging to slice l on GW m
S	Coalition in the game	Set of coalitions including IoT slice members
X	Total money the company owes	Total number of channels on GW m
Y	Total amount of money claimed by companies	Total number of claimed channels by each slice
x_i	Minimum money needed by each player	The number of channels needed by each slice
y_i	Claimed money of each player	Claimed extra channels of each slice
$C - \sum\limits_{i \in N} b_i$	Money(estate)	Total number of additional claimed channels
$\psi_i(v)$	Solution of money distributed to each player	Additional channels reserved for each slice

Table IV.1: Bankruptcy Game description

Based on the bankruptcy problem, modeled as a triple (N, C, g), where $N = \{1, ..., n\}$ is the set of players, i.e. coalition of slice members, C represents the benefit, i.e. the total set of physical channels that are available on LoRa GW $m \in M$, and $g = \{g_1, ..., g_n\}$ is the vector of claims of each coalition of IoT devices. Based on O'Neill approach [87], a bankruptcy game (N,v) can be defined for every bankruptcy problem (N, C, g) where v is a characteristic function with 2^n possible coalitions with players in a game. Moreover,

X is defined as a positive integer, b_i denotes the minimum number of channels that each virtual slice needs to operate and y_i the number of channels actually claimed by each slice members, where $\sum\limits_{i \in N} b_i \leq X$ and $\sum\limits_{i \in N} c_i \geq X - \sum\limits_{i \in N} b_i$. The total amount of money which the company owns denotes the total number of additional channels claimed by each virtual slice and is denoted by Y with $Y \geq X - \sum\limits_{i \in N} b_i$. Due to the direct relationship between throughput requirements and the amount of channels that are requested by each slice, the number of channels requested increases when slice members requires higher traffic rate. The rate required by each slice l is denoted as $R_{i,j}$, the sum of throughput requirements of the coalition i including the set of IoT devices j that are members of $K_{l,m}$ group. Hence, based on what was previously mentioned, the additional claimed channels y_i for a coalition $i \in N$ belonging to slice $l \in L$ is obtained through **Eq. IV.11** while also respecting **Eq. IV.12** below:

$$y_i = \left(\frac{\sum\limits_{j=1}^{j=K_{l,m}} R_{i,j}}{\sum\limits_{i=1}^{i=N} \sum\limits_{j=1}^{j=K_{l,m}} R_{i,j}}\right) * (Y - n) + 1, \forall i \in N, \forall j \in K_{l,m} \tag{IV.11}$$

$$\sum_{i \in N} y_i = Y \tag{IV.12}$$

The characteristic function of this bankruptcy game can be particularly introduced for all possible coalitions in **Eq. IV.13** below [90]:

$$v(S) = max\{o, X - \sum_{i \in N} c_i - \sum_{i \notin S} y_j\} \tag{IV.13}$$

where based on **Eq. IV.14**, no channels are reserved for a virtual slice l on GW m if it doesn't have any IoT devices active in the range of the corresponding GW.

$$v(\phi) = 0 \tag{IV.14}$$

and in **Eq. IV.15** below:

$$v(N) = X - \sum_{i \in N} x_i \tag{IV.15}$$

which indicates that if all devices are member of one coalition, the latter will have the total number of the extra channels available for reservation.

After computing the characteristic function of all possible coalitions, the stable number of channels that should be reserved for each virtual slice l on GW m is computed based on the Shapley value [104]. The latter is a game theory concept which was proposed by Shapley with the goal of finding the fairest allocation of collectively gained profits between several collaborative players based on the relative importance of every player regarding the cooperative activities. In this framework, the number of channels that each slice will get represents the average payoff of the coalition. Hence, the number of channels reserved of each coalition of devices i with $i \in N$ that belongs to slice $l \in L$ is given using function $\varphi_i(v)$ in **Eq. IV.16** below:

$$\begin{aligned} &\varphi_i(v) = \sum_{S \subset N, i \in N} \left(\tfrac{(|S|-1)!(n-|S|)!)}{n!}(v(S) - v(S-i))\right) \\ &with \quad i \in N \end{aligned} \tag{IV.16}$$

where $|S|$ indicates the number coalitions in the set and

$$\varphi_i(v) = X - \sum_{i \in N} y_i \quad with \quad i \in N$$

IV.4.3 Inter-slice Resource allocation via Matching Theory

After reserving the set of channels for each virtual slice, the output of the bankruptcy game is brought up to the inter-slice resource allocation phase. Here, each GW is now aware of the set of channels $C_{l,m}$ reserved for the coalition of devices that belongs to slice $l \in L$ on GW $m \in M$. To solve this channel allocation problem, a modified deferred acceptance algorithm is proposed due to the merits that the one-to-one matching approach brings in providing distributed solution with tractable computation complexity. The latter is originally defined as a mathematical framework to analyze and optimize the allocation problem among players and resources. A one-to-one matching game is a two-sided assignment problem between two disjoint sets of players, in which each individual of a set has preferences over the individuals of the opposite set where the output of this game is an IoT device assigned to one of the channels that were virtually reserved for his slice. With this strategy, complete isolation is provided between channel resources of each slice on LoRa GWs. The proposed algorithm for this framework is described in the **Pseudo-code** 6.

Pseudo-code 6 Resource Allocation algorithm based on one-to-one matching theory

```
Input : Set of gateways $M$ and slices $L$;
        Slice members $K_{l,m}$;
        Set of channels $C_{l,m}$;
        Set of quotas $Q$
begin
    Construct $KLIST_{k,l,m}$,$\forall k \in K_{l,m}, l \in L$ and $m \in M$
    Construct $CHLIST_{l,m}$,$\forall l \in L$ and $m \in M$
    Construct $NMLIST_k$ with $k \in W$.
    while $NMLIST_k$ not empty do
        GW $m$ receives preference of $K_{l,m}$ IoT devices.
        for each device $k \in K_{l,m}$ do
            Put $CHLIST_{l,m}$ in decreasing order.
            Device $k$ makes an offer to first channel in $CHLIST_{l,m}$.
            if quota $q_{l,m}$ is respected then
                if $c'_{l,m}$ is more preferred than $c_{l,m}$ then
                    Keep the preferred channel $c'_{l,m}$;
                    Reject channel $c_{l,m}$;
                    Apply $TOPG(k, c_{l,m})$;
                    Allocate device $k$ to $c'_{l,m}$;
                    Remove device $k$ from $NMLIST_k$.
                    Decrease quota $q_{l,m}$
                else
                    Keep the preferred channel $c_{l,m}$.
                end
            else
                Remove $k$ from $NMLIST_k$.
            end
        end
    end
end
Output: Set of channels $C_{l,m}$ allocated in each slice $l$.
```

After the end of the bankruptcy game, the algorithm receives the set of channels $C_{l,m}$ for each virtual slice l on GW m as well as the quotas $q_{l,m}$ that denotes the number of channels reserved for each slice. The inter-slice resource allocation algorithm is launched between the set of IoT devices and the physical channels that were previously reserved with respect to the maximum quota of the gateway. The latter executes the initialization of algorithm and starts by building with preference relations of devices and physical channels **(line 2)**. The former list is denoted as $KLIST_{k,l,m}$ whereas the latter is denoted $CHLIST_{l,m}$. For each slice, all devices that did not match a channel are stored in a new list denoted as $NMLIST$ **(line 2)**. While devices exist in $NMLIST$, the algorithm propose the most preferred channel to the device with the highest utility in the slice **(line 4-7)**. This way, the most urgent device is given the highest priority upon channel allocation. However, in LoRa allocating an IoT device is not enough to improve QoS of the device because the latter is related to its configuration in terms of spreading factor and transmission power. This is why configuration of each device matched to a channel will be optimized with $TOPG$ that was previously proposed in [29] to optimize SF and TP configuration with respect to the QoS thresholds of the slice that is assigned to **(line 6)**. Next, the allocated device is removed from $NMLIST$ and the available quota $q_{l,m}$ for the corresponding slice is reduced by one channel **(line 7)**. The matching process is repeated for each slice members and ends when $NMLIST$ is empty **(line 11)** or when physical channels are completely assigned by IoT devices to reach a weak Pareto optimal stable matching **(line 15)** [63].

IV.5 Simulation Results

As most of traffic in IoT comes from uplink communications, we focus more on the uplink traffic that comes from IoT sensors assigned to one of the virtual slices that were previously defined in **Table I.5**. Simulations, replicated 20 times with 95% confidence interval in realistic LoRa scenarios, are implemented over the open source NS3 simulator [83]. To implement SDN in our simulations, OpenFlow protocol version 1.3 [19] is adopted alongside the LoRa model proposed by Magrin et al. in [73] using the following SDN [38] and LoRaWAN [72] source codes on Github. We assume that each GW has a complete knowledge on the buffer of the IoT devices existing in its range. This assumption has been previously considered by Liu et al. in [70] and showed its worthiness in improving energy efficiency and QoS effectiveness in wireless IoT networks. Next, bankruptcy is applied to reserve channel resources for each virtual slice based on data transmitted by its members. In this context, GWs cooperate to assign edge devices, then reserve the bandwidth and apply proper network configuration in a distributed manner. The latter should improve network performance because GWs will have the ability to be closer to IoT devices in terms of network slicing management and to adapt faster to their QoS requirements. Hence, in large scale networks, making slicing decisions at the GW level reduces complexity due to the cooperation that happens between GWs instead of transmitting all information to the centralized SDN controller.

In this work, simulations are replicated 20 times with 95% confidence interval in realistic LoRa scenarios. All application and simulation parameters are summarized in **Table III.1**. However, this time we will evaluate the new distributed proposition in large scale where we increase the number of IoT devices into 5000 devices in the network. At each replication, devices are distributed based on the uniform random distribution over a cell of 10 km whereas GWs positions are fixed and spaced 2.5 km apart. Each GW is characterized by 8 receiving channels with each having a bandwidth of 125 kHz in the 867-868 MHz European sub-band. Regarding application settings, packets are transmitted at a random time but in a periodic manner with fixed payloads of 18 Bytes following to the work done in [10]. LoRa parameter settings are respected with SF configurations that vary between 7 and 12 and 2 to 14 dBm for TP configurations. Based on each configuration, the lifetime of a device varies and is evaluated in each slice using **Eq. IV.17** below [86].

$$\begin{aligned} & L_{k,l,m} = \tfrac{BC_k}{E_{k,l,m}(1-ds)} \\ & with \quad E_{k,l,m} = \tfrac{p_i^{tx}+p_i^{rx}}{V+epa}.d_{tx/rx} \end{aligned} \tag{IV.17}$$

where $L_{k,l,m}$ denotes the battery lifetime of an IoT device k belonging to slice l having a maximum capacity and a discharge safety ds fixed in this work to 10 % of the maximum capacity of the battery BC_k. $L_{k,l,m}$ computation depends on the energy consumed by an IoT device $E_{k,l,m}$, V the LoRa supply voltage, epa the amplifier's added efficiency,

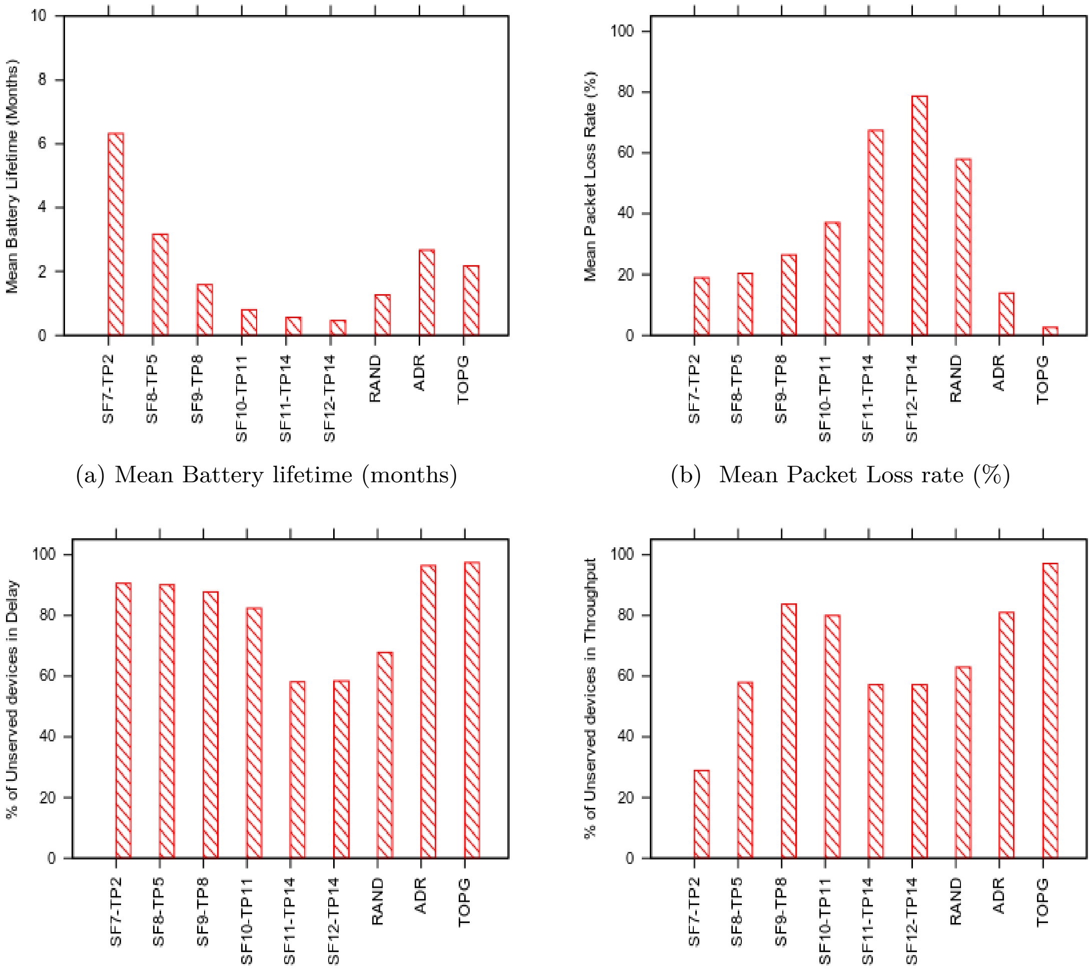
d_{tx} the duration of transmission, p_i^{rx} and p_i^{tx} that finally denote the power of reception and transmission respectively and vary between 2 and 14 dBm depending on the configuration strategy adopted. Using **Eq. IV.17** and the energy module that already exists in NS3 for Wifi, we integrate an energy model for LoRa using the energy parameters and the power model specified in [10] and [15] respectively.

In the following, the proposed distributed slicing strategy is evaluated over various parameters configuration methods. The best configuration found will be adopted in a second performance study that compares *static*, *probabilistic*, *centralized* and *distributed* slicing strategies and evaluates their impact on the battery lifetime, communications reliability and the percentage of devices that respect throughput and delay thresholds.

IV.5.1 Performance Study with various configurations strategies

In **Fig. IV.3**, performance results are evaluated in LoRa with the proposed distributed slicing for a fixed number of 2500 devices. Each simulation is replicated 20 times using different configurations that assign SF and TP parameters based on static, dynamic or optimized strategies. With *static* configurations, all devices in the cell are configured with one of the following SF-TP combinations (i.e., SF7-TP2, SF8-TP5, SF9-TP8, SF10-TP11, SF11-TP14 and SF12-TP14). The latter are compared to the random configuration denoted as $RAND$ where each device randomly picks SF and TP values, the adaptive data rate ADR which is the standard configuration adopted now by LoRa that dynamically assigns to the device one of the SF-TP configurations depending on the highest receiving power measured from nearby GWs and finally $TOPG$, an optimized configuration method proposed in [29] based on GMM and TOPSIS optimization algorithms. The latter starts from ADR configuration and modifies SF or TP configuration to improve QoS in each slice.

A higher spreading factor for LoRa results a longer time on air (ToA). Thus, TP increases with increasing SF because LoRa radio module will need more time to send the same amount of data. Therefore, with *static* configurations, when all devices are



(a) Mean Battery lifetime (months)

(b) Mean Packet Loss rate (%)

(c) % of devices served in delay

(d) % of devices served in throughput

Figure. IV.3: Performance evaluation of 2500 devices simulated with various SF-TP configuration strategies

configured with one of SF-TP combinations, battery lifetime increases with low SF and TP values as shown in **Fig. IV.3a**. However, for higher *static* configurations, the mean lifetime of a device is lower when a device is configured randomly or with *ADR*, which was the most energy efficient configuration, where each device is configured based on its distance to the most reliable GW. However, knowing that *TOPG* is less efficient in terms of energy than *ADR*, QoS performance was highly improved in terms of reliability and the percentage of devices that respected GBR and delay thresholds as respectively shown in **Fig. IV.3b**, **Fig. IV.3c** and **Fig. IV.3d** respectively. *TOPG* modifies *TP* or *SF* to ensure the receipt of a packet above the sensitivity level and to avoid the loss of packets received simultaneously at a channel and configured with the same or different SFs. Hence, instead of loosing both packets, at least one of the packets will be decoded successfully. Moreover, configuring all IoT devices with same configuration is proven to be inefficient in terms of QoS. For example, configuring all devices with static $SF7-TP2$ combination increases the rate of packets lost due to interference for nearby devices and to lack of sensitivity for cell edge devices. Similarly, $RAND$ configuration do not take into consideration the position of the device and QoS thresholds of the ones transmitting in each LoRa slice. This leads to extreme misconfigurations and explains the lowest performance results in terms of QoS. ADR dynamically assigns the proper combination to the device based on its position which highly improved reliability but do not consider QoS thresholds defined for each LoRa slice. This highlights the advantages of considering an optimized configuration where, unlike ADR, $TOPG$ assigns IoT device configuration in a way that respects QoS thresholds defined for each slice. Therefore, $TOPG$ is adopted for the following slicing performance study because it highly improves the performance of each slice members compared to *static*, $RAND$ and ADR configurations.

IV.5.2 Centralized vs Distributed Slicing

Based on the scalability study performed in [55], LoRa scalability varies for different IoT applications. In some applications, high spreading factors cannot be used due to violation of radio duty cycle by the message transaction period. Hence, knowing the variety of IoT use cases and when end devices density increases as well in the network, it looks impractical to manage IoT communications in a centralized manner specially if the SDN controller configures edge devices with high spreading factors which at its turn increases packet error rate and collisions [116]. Moreover, in large scale networks, the SDN controller should be able to acknowledge more messages as the number of IoT devices in the network increase. This increases signaling cost and slicing decision time and hence affects delay performance. Instead, performing resource reservation in a distributed manner while being closer to end devices could save lots of time and energy. For this reason, delay performance is evaluated for URA slice members as it requires the respect of the lowest PDB. The goal here is to show the efficiency of considering distributed slicing for large scale networks. **Fig. IV.4** validates this assumption when the distributed slicing respected more URA PDB threshold when IoT devices density increases in the network.

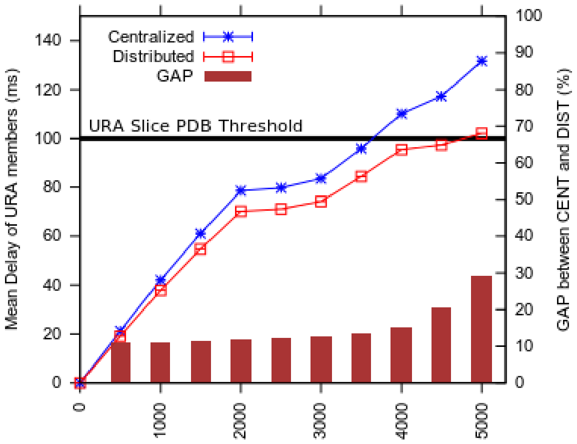
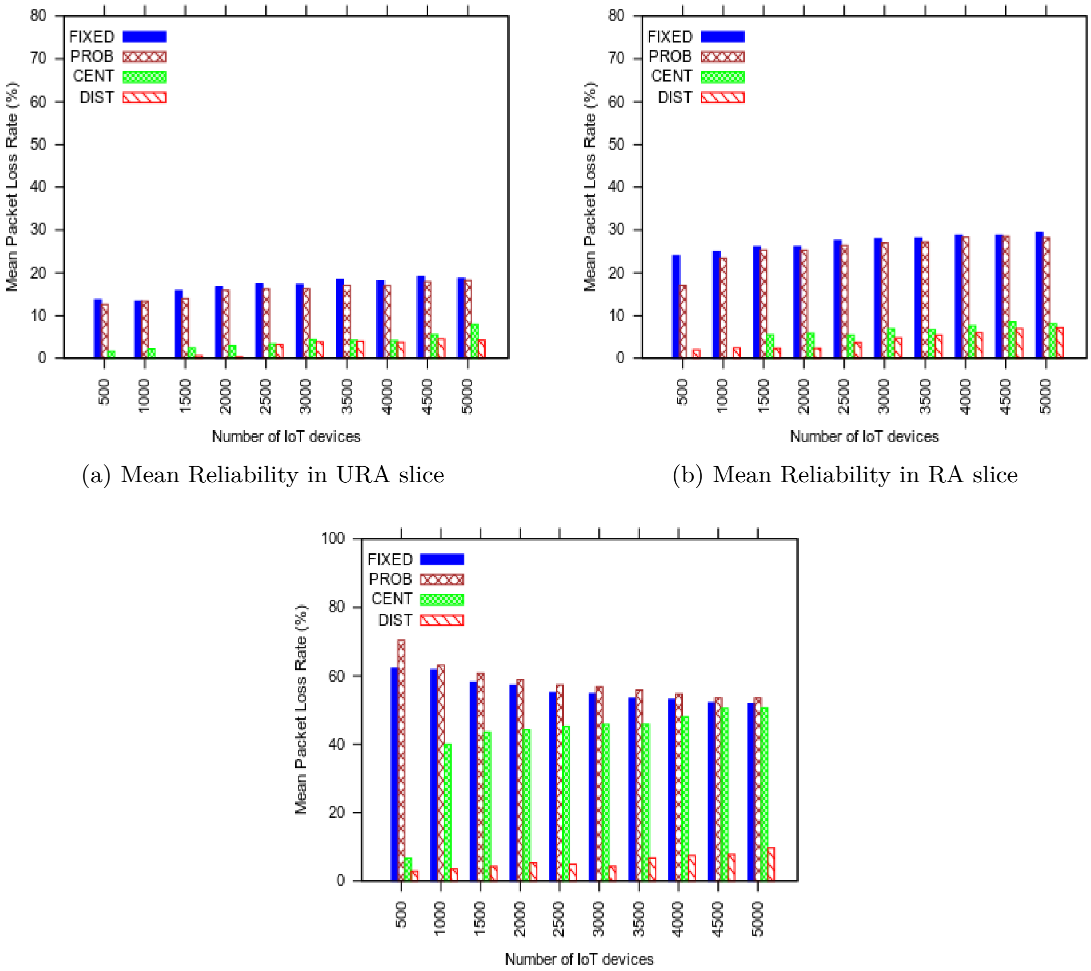


Figure. IV.4: Mean URA slice members delay

IV.5.3 Performance Study with various network slicing strategies

In this section, the performance in LoRa slices is evaluated for various slicing strategies. The first denoted as $FIXED$ is a strategy where the centralized SDN controller reserves the channels equally between LoRa slices. Hence, for each GW, the total number of channels is divided by the number of slices and reserved accordingly. The second strategy is derived from the literature [41] and denoted as $PROB$ because the number of channels reserved for each slice is defined based on the probability that the traffic generated at a time instant is less than the maximum throughput that can be uploaded through the reserved channels. The third strategy is using centralized network slicing denoted as $CENT$ where slicing decisions are performed on all GWs using maximum likelihood throughput estimation while having a global overview of IoT devices positions and their QoS requirements [28]. With $CENT$, slicing decisions are taken by the centralized controller and transmitted to the GW which reserves the channels for each slice starting by the one with the highest slicing priority. We compare these strategies to the proposed distributed slicing algorithm, denoted as $DIST$, where unlike $CENT$, slicing and resources allocation decisions are taken by each GW in a distributed manner.

Simulation is performed with respect to the parameters summarized in **Table II.1** when the number of devices increases till it reaches 5000 devices deployed in LoRa network. The goal is to evaluate how slicing performance varies for each strategy in large scale LoRaWAN in terms of reliability, battery lifetime and respecting QoS thresholds.



(a) Mean Reliability in URA slice

(b) Mean Reliability in RA slice

(c) Mean Reliability in BE slice

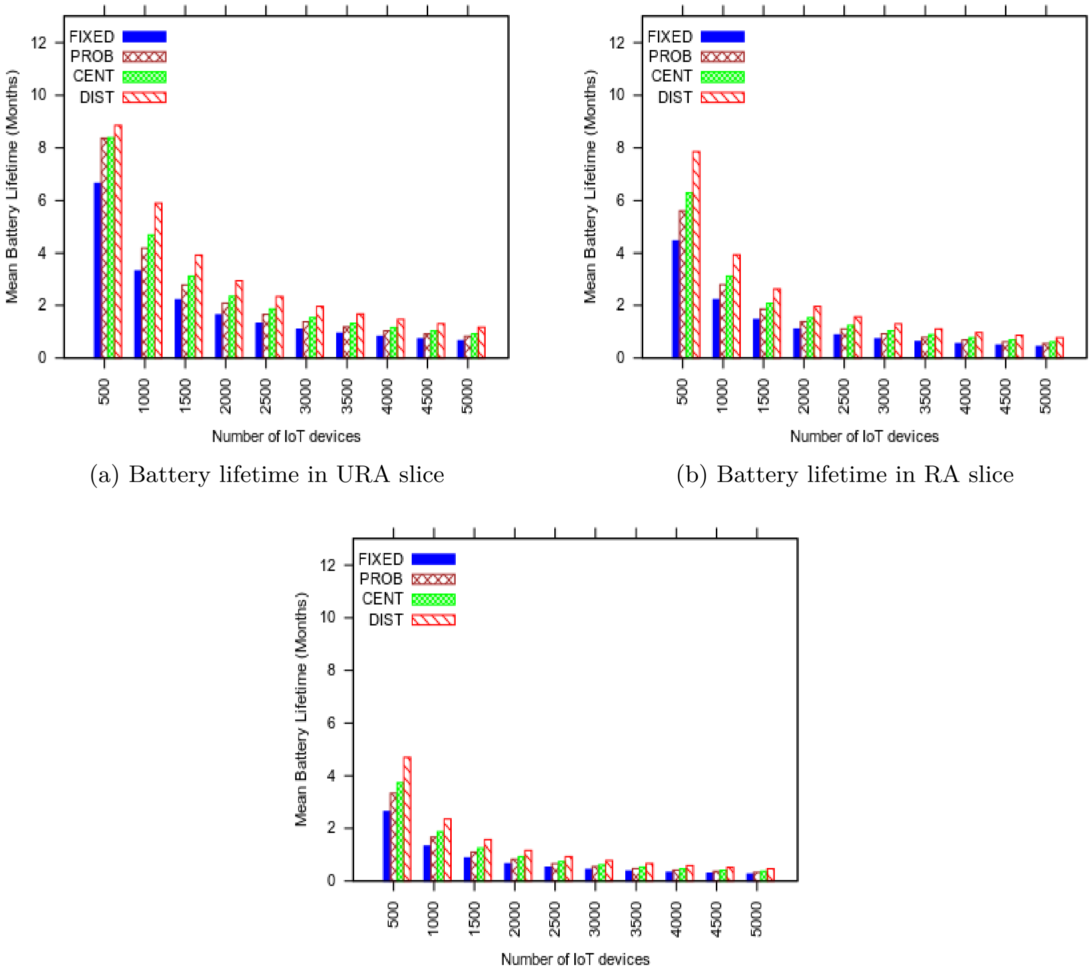
Figure. IV.5: Reliability performance study in every slice with various slicing strategies

IV.5.3.1 Slice-based Reliability

In **Fig. IV.5**, reliability performance is evaluated for each slice. The best reliability performance, illustrated in **Fig. IV.5a**, is achieved with *URA* slice members compared to *RA* and *BE* slices regardless of the adopted strategy. This proves the utility of forming virtual isolated networks over physical LoRa equipments. *FIXED* strategy scored the least reliable results overall because the slicing is static and does not adapt to slicing admission. The latter depends on the running application on each device where some transmissions could be urgent or just requiring a best effort behavior. Moreover, **Fig. IV.5a** and **Fig. IV.5b** show that although $PROB$ slightly improved reliability performance for URA and RA slices, $PROB$ had the worst performance in BE slice due to the over estimation of the amount of channels that should be reserved for urgent and reliable slices. $CENT$ and $DIST$ proved their efficiency reducing PLR compared to $FIXED$ and $PROB$ strategies. However, when congestion increases in LoRa, the efficiency of $DIST$ method is clearly highlighted in **Fig. IV.5c** and proved to be the best slicing strategy between the ones simulated. This specially appeared in the BE slice where with $CENT$ more than 50% of packets uploaded by BE members are lost with high congestion scenarios whereas with $DIST$ strategy PLR is limited to 10% of the total number of packets uploaded. This result returns to the fast adaptation and the fair resources reservation that the distributed method provides.

IV.5.3.2 Slice-based Battery Lifetime

In **Fig. IV.6**, when the number of devices increases, the mean battery lifetime of devices, measured in months, decreases as well regardless of the adopted slicing strategy. $FIXED$ and $PROB$ were the two strategies with the shortest mean battery lifetime for all LoRa slices simulated in **Fig. IV.6a**, **Fig. IV.6b** and **Fig. IV.6c** respectively. Both slicing strategies do not efficiently adapt to throughput requirements of each slice members. However, knowing that urgent communications are less frequent in IoT, battery lifetime decreased in RA and BE because part of the channels for URA are sometimes left unused and kept RA and BE slice members active for a longer time. This wasn't the case for $CENT$ and $DIST$ strategies which estimates and adapts to throughput requirements of each slice members. With $CENT$, an IoT device, if positioned on the edge in the range of two GWs, the centralized controller estimates its throughput twice for both GWs upon resource reservation. Hence, in large scale where the number of edge devices increases in the network, $DIST$ slicing strategy is more efficient and reduces the probability of misconfiguring SF and TP parameters, because with $DIST$, each GW reserves its own channels and the throughput requirement of each edge device is only considered by the GW which is assigned to.



(a) Battery lifetime in URA slice

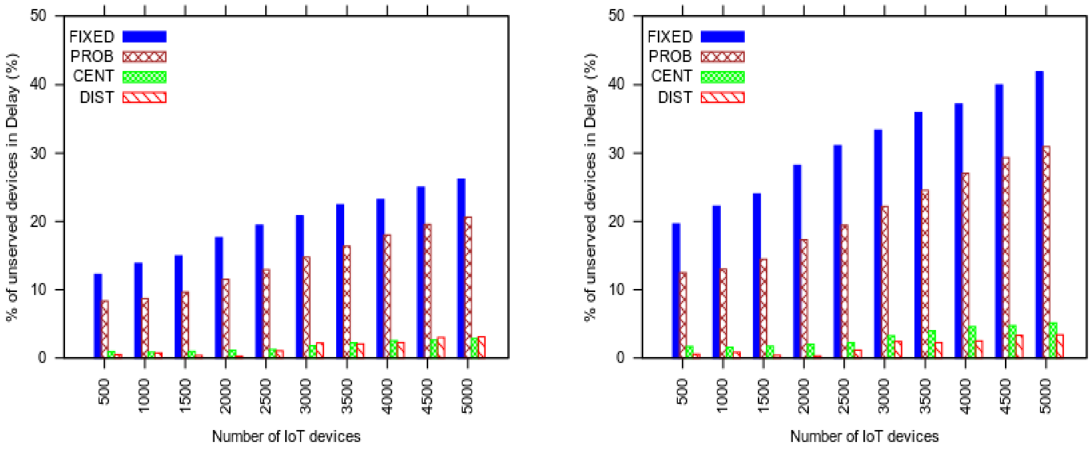
(b) Battery lifetime in RA slice

(c) Battery lifetime in BE slice

Figure. IV.6: Mean Battery lifetime in every slice with various slicing strategies

IV.5.3.3 Percentage of unserved devices in Delay

In **Fig. IV.7**, $FIXED$ configuration had the worst results with 20% of devices that did not respect their delay thresholds in URA slice and 40% in RA slice. Here, BE slice is not considered for all strategies because no delay restrictions are imposed on BE slice members. $PROB$ was more efficient and improved delay performance in both URA and RA slices as illustrated in **Fig. IV.7a** and **Fig. IV.7b** respectively. The latter show the efficiency of $CENT$ and $DIST$ slicing strategies. In both URA and URA slices, the rate of unserved devices in delay did not exceed 10% of the packets transmitted with *CENT* and *DIST* slicing strategies. This highlights the importance of considering slice-based resource reservation and network optimization for both methods. However, with *DIST* slicing, decisions are taken faster and closer to the infrastructure which improves the rate of devices that respected PDB thresholds when reducing signaling information transmitted to the centralized controller.



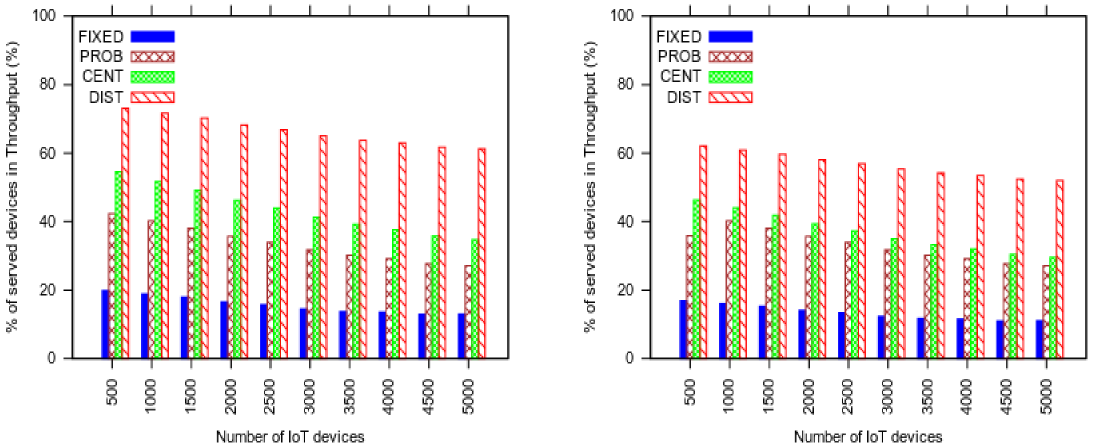
(a) Percentage of unserved nodes in delay for URA slice

(b) Percentage of unserved nodes in delay for RA slice

Figure. IV.7: Percentage of unserved nodes in delay

IV.5.3.4 Percentage of served devices in Throughput

Further improvement in throughput is achieved in **Fig. IV.8** where *DIST* was also the best slicing strategy compared to the ones simulated in this work. The rate of devices that respected GBR in *URA* and *RA* slices is respectively illustrated for each slicing strategy in **Fig. IV.8a** and **Fig. IV.8b**. The proposed *DIST* slicing scored the best performance in both slices and the was the only strategy that had more than 50% of devices respecting GBR thresholds even in a very congested scenario. This mainly highlights the efficiency of *DIST* method when combining distributed slicing strategy with slice-based configuration optimization. Its fast adaptation to QoS requirements of IoT applications installed on end devices provides the most suitable SF and TP configurations in large scale IoT network.



(a) Percentage of served nodes in throughput for URA slice

(b) Percentage of served nodes in throughput for RA slice

Figure. IV.8: Percentage of served nodes in throughput

IV.6 Conclusion

In this chapter, distributed slicing is proposed to prevent potential challenges that may appear due to the increasing congestion in large scale IoT deployments and to reduce network complexity. The proposed SDN-based architecture improves LoRaWAN scalability and performance in large scale scenarios by leveraging computational intelligence to LoRa gateways and moving closer to the edge. Here, LoRa GWs defines locally the slicing strategy and the resources that should be reserved for each LoRa slice after a coordination phase with neighboor gateways to improve network reliability and to avoid resource starvation that may happen due to congestion in large scale LoRa deployments. Numerical results have shown that the proposed distributed slicing strategy outperformed the centralized one in more congested scenarios due to its faster adaptation to IoT devices QoS requirements.

General Conclusion and Perspectives

To support efficient IoT communications with guaranteed QoS requirements, new contributions are needed to provide flexible resource management in the network and optimize its configuration dynamically. In this thesis, we proposed new ideas that improve spectrum management, QoS consideration and energy efficiency in IoT networks using network slicing and software defined networking. The proposed solutions are implemented over LoRaWAN due its low power, wide area, open alliance and its potential to support large scale IoT deployments.

After dividing IoT services into three class of services based on urgency and reliability, network slicing is first implemented in the centralized LoRa architecture where each class of services belongs to a network slice. The latter are proved to be completely isolated to protect urgent and critical IoT communications from being impacted by less prioritized IoT devices. Next, various static and dynamic slicing strategies are compared with different spreading factor distributions. Results show that the adaptive dynamic slicing and configuration method was the best in terms of QoS, reliability and energy consumption. With this strategy, the centralized server defines, for each GW, how channels are reserved for the virtual slices configured on that GW based on an MLE estimation of devices throughput requirements in its range.

To improve these results, we extended the previous method with a slice-based optimization that improves spreading factor and transmission power parameters configuration at the physical layer. While respecting QoS thresholds of each slice, IoT devices are configured with *TOPG*, based on TOPSIS and GMM optimization methods. The proposed *TOPG* method finds the configuration that responds best to multiple objectives in terms of improving QoS, avoiding interference and reducing the energy consumption of each slice members.

Despite improving network performance with the previous contributions, LoRaWAN will still come up short in meeting scalability challenges in next generation large scale IoT networks. Therefore, we finally proposed an SDN-based distributed LoRaWAN architecture and slicing strategy improved reliability performance for 5000 devices deployed in the network. When slicing and configuration decisions are leveraged to the edge, LoRa GWs will be able to apply the needed optimization faster instead of just sending all information to the server.

For future works, the focus should go towards investigating the findings of this thesis and implementing network slicing in real test-bed implementations. We will work on practically proving the slicing concept in all LoRa architecture layers. Moreover, some further improvements could be also realized by integrating artificial intelligence with machine learning tools to enable rapid analysis, prediction, and decision making. Finally, we also intend to work on other big challenges in IoT such as improving security measures and ensuring interoperability in next generation IoT networks.

Bibliography

[1] Ferran Adelantado, Xavier Vilajosana, Pere Tuset-Peiro, Borja Martinez, Joan Melia-Segui, and Thomas Watteyne. Understanding the limits of lorawan. *IEEE Communications Magazine*, 55(9):34–40, 2017.

[2] Basim KJ Al-Shammari, Nadia Al-Aboody, and Hamed S Al-Raweshidy. Iot traffic management and integration in the qos supported network. *IEEE Internet of Things Journal*, 5(1):352–370, 2018.

[3] Bhagya Amarasekara, Chathurika Ranaweera, Rob Evans, and Ampalavanapillai Nirmalathas. Dynamic scheduling algorithm for lte uplink with smart-metering traffic. *Transactions on Emerging Telecommunications Technologies*, 2017.

[4] Licia Amichi, Megumi Kaneko, Nancy Rachkidy, and Alexandre Guitton. Spreading factor allocation strategy for lora networks under imperfect orthogonality. 2019.

[5] Aloÿs Augustin, Jiazi Yi, Thomas Clausen, and William Townsley. A study of lora: Long range & low power networks for the internet of things. *Sensors*, 16(9):1466, 2016.

[6] Amin Azari and Guowang Miao. Network lifetime maximization for cellular-based m2m networks. *IEEE Access*, 2017.

[7] Amin Azari, Guowang Miao, Cedomir Stefanovic, and Petar Popovski. Latency-energy tradeoff based on channel scheduling and repetitions in nb-iot systems. In

2018 IEEE Global Communications Conference (GLOBECOM), pages 1–7. IEEE, 2018.

[8] Theophilus Benson, Aditya Akella, and David A Maltz. Unraveling the complexity of network management. In *NSDI*, pages 335–348, 2009.

[9] Yihenew Dagne Beyene, Riku Jantti, Kalle Ruttik, and Sassan Iraji. On the performance of narrow-band internet of things (nb-iot). In *2017 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, 2017.

[10] Norbert Blenn and Fernando Kuipers. Lorawan in the wild: Measurements from the things network. *arXiv preprint arXiv:1706.03086*, 2017.

[11] Martin Bor and Utz Roedig. Lora transmission parameter selection. *International Conference on Distributed Computing in Sensor Systems*, 2017.

[12] Martin Bor, John Edward Vidler, and Utz Roedig. Lora for the internet of things. *International Conference on Embedded Wireless Systems and Networks*, 2016.

[13] Martin C Bor, Utz Roedig, Thiemo Voigt, and Juan M Alonso. Do lora low-power wide-area networks scale? In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 59–67. ACM, 2016.

[14] Martin C Bor, Utz Roedig, Thiemo Voigt, and Juan M Alonso. Do lora low-power wide-area networks scale? In *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pages 59–67. ACM, 2016.

[15] Chaillout Bouguera, Diouris and Andrieux. Energy consumption modeling for communicating sensors using lora technology. In *IEEE Conference on Antenna Measurements and Applications (CAMA)*. IEEE, 2018.

[16] Wolfgang Braun and Michael Menth. Software-defined networking using openflow: Protocols, applications and architectural design choices. *Future Internet*, 6(2):302–336, 2014.

[17] Khac-Hoai Nam Bui, Jai E Jung, and David Camacho. Game theoretic approach on real-time decision making for iot-based traffic light control. *Concurrency and Computation: Practice and Experience*, 29(11):e4077, 2017.

[18] Marco Cattani, Carlo Alberto Boano, and Kay Römer. An experimental evaluation of the reliability of lora long-range low-power wireless communication. *Journal of Sensor and Actuator Networks*, 6(2):7, 2017.

[19] Luciano Jerez Chaves, Islene Calciolari Garcia, and Edmundo Roberto Mauro Madeira. Ofswitch13: Enhancing ns-3 with openflow 1.3 support. In *Proceedings of the Workshop on ns-3*, pages 33–40. ACM, 2016.

[20] Junting Chen, Wenhan Dai, Yuan Shen, Vincent KN Lau, and Moe Z Win. Resource management games for distributed network localization. *IEEE Journal on Selected Areas in Communications*, 35(2):317–329, 2017.

[21] Min Chen, Yiming Miao, Yixue Hao, and Kai Hwang. Narrow band internet of things. *IEEE access*, 5:20557–20577, 2017.

[22] N Chu et al. Exalted: Expanding lte for devices. *European Commission for Information Society and Media*, pages 1–141, 2012.

[23] Daniele Croce, Michele Gucciardo, Stefano Mangione, Giuseppe Santaromita, and Ilenia Tinnirello. Impact of lora imperfect orthogonality: Analysis of link-level performance. *IEEE Communications Letters*, 22(4):796–799, 2018.

[24] Daniele Croce, Michele Gucciardo, Ilenia Tinnirello, Domenico Garlisi, and Stefano Mangione. Impact of spreading factor imperfect orthogonality in lora communications. In *International Tyrrhenian Workshop on Digital Communication*, pages 165–179. Springer, 2017.

[25] Anshuman Dash, Satyajit Pal, and Chinmay Hegde. Ransomware auto-detection in iot devices using machine learning. *International Conference on Business Analytics and Intelligence (ICBAI)*, 2018.

[26] Samir Dawaliby, Abbas Bradai, and Yannis Pousset. In depth performance evaluation of lte-m for m2m communications. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2016 IEEE 12th International Conference on*, pages 1–8. IEEE, 2016.

[27] Samir Dawaliby, Abbas Bradai, and Yannis Pousset. Scheduling optimization for M2M communications in LTE-M. In *2017 IEEE International Conference on Consumer Electronics (ICCE)*, pages 126–128. IEEE, 2017.

[28] Samir Dawaliby, Abbas Bradai, and Yannis Pousset. Adaptive dynamic network slicing in lora networks. *Future generation computer systems*, 2019.

[29] Samir Dawaliby, Abbas Bradai, and Yannis Pousset. Network slicing optimization in large scale lora wide area networks. In *Wireless and Mobile Computing, Networking and Communications (NETSOFT), 2019 IEEE 12th International Conference on*. IEEE, 2019.

[30] Samir Dawaliby, Abbas Bradai, Yannis Pousset, and Christian Chatellier. Joint energy and qos-aware memetic-based scheduling for m2m communications in lte-m. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2018.

[31] Carmen Delgado, María Canales, Jorge Ortín, José Ramón Gállego, Alessandro Redondi, Sonda Bousnina, and Matteo Cesana. Joint application admission control and network slicing in virtual sensor networks. *IEEE Internet of Things Journal*, 5(1):28–43, 2018.

[32] Arnaud Durand, Pascal Gremaud, and Jacques Pasquier. Resilient, crowd-sourced lpwan infrastructure using blockchain. In *Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems*, pages 25–29. ACM, 2018.

[33] Salvatore D'Oro, Laura Galluccio, Sergio Palazzo, and Giovanni Schembra. A game theoretic approach for distributed resource allocation and orchestration of softwarized networks. *IEEE Journal on Selected Areas in Communications*, 35(3):721–735, 2017.

[34] Ahmad Hani El Fawal, Ali Mansour, Mohamad Najem, Frédéric Le Roy, and Denis Le Jeune. Lte-m adaptive enodeb for emergency scenarios. In *2017 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 536–541. IEEE, 2017.

[35] Mohieddine El Soussi, Pouria Zand, Frank Pasveer, and Guido Dolmans. Evaluating the performance of emtc and nb-iot for smart city applications. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–7. IEEE, 2018.

[36] Ericsson. Internet of Things Forecast. https://www.ericsson.com/en/mobility-report/internet-of-things-forecast, 2019. [Online; accessed January-2019].

[37] ERICSSON. Cellular iot alphabet soup, February 2016.

[38] Chaves et al. Ofswitch13 - openflow 1.3 module for ns-3, 2019.

[39] TR ETSI. 103 467 v1.1.1:" speech and multimedia transmission quality (stq). *Quality of Service aspects for IoT*, 2018.

[40] Ivan Farris, Tarik Taleb, Yacine Khettab, and Jae Seung Song. A survey on emerging sdn and nfv security mechanisms for iot systems. *IEEE Communications Surveys & Tutorials*, 2018.

[41] Yasser Gadallah, Mohamed H Ahmed, and Ehab Elalamy. Dynamic lte resource reservation for critical m2m deployments. *Pervasive and Mobile Computing*, 40:541–555, 2017.

[42] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 120(5):386–391, 2013.

[43] Orestis Georgiou and Usman Raza. Low power wide area network analysis: Can lora scale? *IEEE Wireless Communications Letters*, 6(2):162–165, 2017.

[44] Claire Goursaud and Jean-Marie Gorce. Dedicated networks for iot: Phy/mac state of the art and challenges. *EAI endorsed transactions on Internet of Things*, 2015.

[45] IEEE 802 Working Group et al. Ieee standard for local and metropolitan area networks—part 15.4: Low-rate wireless personal area networks (lr-wpans). *IEEE Std*, 802:4–2011, 2011.

[46] GSMA. Network Slicing use cases requirements. https://www.gsma.com/futurenetworks/wp-content/uploads/2018/06/Network-Slicing-Use-Case-Requirements-_-FInal-.pdf, April 2018. [Online; accessed 19-august-2019].

[47] Yunan Gu, Zheng Chang, Miao Pan, Lingyang Song, and Zhu Han. Joint radio and computational resource allocation in iot fog computing. *IEEE Transactions on Vehicular Technology*, 67(8):7475–7484, 2018.

[48] Piyush Gupta and Panganmala R Kumar. The capacity of wireless networks. *IEEE Transactions on information theory*, 46(2):388–404, 2000.

[49] Mo Haghighi, Zhijin Qin, Davide Carboni, Usman Adeel, Fengrui Shi, and Julie A McCann. Game theoretic and auction-based algorithms towards opportunistic communications in lpwa lora networks. In *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pages 735–740. IEEE, 2016.

[50] Evangelos Haleplidis, Kostas Pentikousis, Spyros Denazis, J Hadi Salim, David Meyer, and Odysseas Koufopavlou. Software-defined networking (sdn): Layers and architecture terminology. Technical report, 2015.

[51] Hamdani and Retantyo Wardoyo. The complexity calculation for group decision making using topsis algorithm. In *AIP Conference Proceedings*, volume 1755, page 070007. AIP Publishing, 2016.

[52] Safa Hamdoun, Abderrezak Rachedi, and Yacine Ghamri-Doudane. A flexible m2m radio resource sharing scheme in lte networks within an h2h/m2m coexistence scenario. In *Communications (ICC), 2016 IEEE International Conference on*, pages 1–7. IEEE, 2016.

[53] Zhu Han, Dusit Niyato, Walid Saad, Tamer Başar, and Are Hjørungnes. *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge university press, 2012.

[54] Yixue Hao, Daxin Tian, Giancarlo Fortino, Jing Zhang, and Iztok Humar. Network slicing technology in a 5g wearable network. *IEEE Communications Standards Magazine*, 2(1):66–71, 2018.

[55] Jetmir Haxhibeqiri, Floris Van den Abeele, Ingrid Moerman, and Jeroen Hoebeke. Lora scalability: A simulation model based on interference measurements. *Sensors*, 17(6):1193, 2017.

[56] Yeonjune Jeong, Mihui Kim, Min Young Chung, Tae-Jin Lee, and Hyunseung Choo. Frequency-domain packet scheduling for low papr in 3gpp lte uplink. *International Journal of Smart Home*, 7(1):163–172, 2013.

[57] Anders Ellersgaard Kalør, René Guillaume, Jimmy Jessen Nielsen, Andreas Mueller, and Petar Popovski. Network slicing for ultra-reliable low latency communication in industry 4.0 scenarios. *arXiv preprint arXiv:1708.09132*, 2017.

[58] Dae-Young Kim, Seokhoon Kim, Houcine Hassan, and Jong Hyuk Park. Adaptive data rate control in low power wide area networks for long range iot services. *Journal of computational science*, 22:171–178, 2017.

[59] Anna Larmo. Narrowband IoT in the cloud. https://www.ericsson.com/en/blog/2016/9/narrowband-iot-in-the-cloud, 2016. [Online; accessed 01-September-2016].

[60] Mads Lauridsen, István Z Kovács, Preben Mogensen, Mads Sorensen, and Steffen Holst. Coverage and capacity analysis of lte-m and nb-iot in a rural area. In *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, pages 1–5. IEEE, 2016.

[61] Alexandru Lavric and Valentin Popa. Lora™ wide-area networks from an internet of things perspective. In *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–4. IEEE, 2017.

[62] Xuan-Chien Le, Baptiste Vrigneau, Matthieu Gautier, Malo Mabon, and Olivier Berder. Energy/reliability trade-off of lora communications over fading channels. In *International Conference on Telecommunication*, 2018.

[63] Tuan LeAnh, Nguyen H Tran, Walid Saad, Long Bao Le, Dusit Niyato, Tai Manh Ho, and Choong Seon Hong. Matching theory for distributed user association and resource allocation in cognitive femtocell networks. *IEEE Transactions on Vehicular Technology*, 66(9):8413–8428, 2017.

[64] Lingling Li, Jiuchun Ren, and Qian Zhu. On the application of lora lpwan technology in sailing monitoring system. In *Wireless On-demand Network Systems and Services (WONS), 2017 13th Annual Conference on*, pages 77–80. IEEE, 2017.

[65] Shengyang Li, Usman Raza, and Aftab Khan. How agile is the adaptive data rate mechanism of lorawan? *arXiv preprint arXiv:1808.09286*, 2018.

[66] Jia-Ming Liang, Po-Yen Chang, and Jen-Jee Chen. Energy-efficient scheduling scheme with spatial and temporal aggregation for small and massive transmissions in lte-m networks. *Pervasive and Mobile Computing*, 52:29–45, 2019.

[67] Jin-Taek Lim and Youngnam Han. Spreading factor allocation for massive connectivity in lora systems. *IEEE Communications Letters*, 22(4):800–803, 2018.

[68] Jun Lin, Zhiqi Shen, Chunyan Miao, and Siyuan Liu. Using blockchain to build trusted lorawan sharing server. *International Journal of Crowd Science*, 1(3):270–280, 2017.

[69] Xiaolan Liu, Zhijin Qin, Yue Gao, and Julie A McCann. Resource allocation in wireless powered iot networks. *IEEE Internet of Things Journal*, 2019.

[70] Zhiqiang Liu, Bin Liu, and Chang Wen Chen. Buffer-aware resource allocation scheme with energy efficiency and qos effectiveness in wireless body area networks. *IEEE Access*, 5:20763–20776, 2017.

[71] Hanjiang Luo, Chao Liu, and Yongquan Liang. A sdn-based testbed for underwater sensor networks. 2019.

[72] David Magrin. LoRaWAN Module Documentation. https://github.com/signetlabdei/lorawan/blob/master/doc/lorawan.rst, 2016. [Online; accessed 27-November-2018].

[73] Davide Magrin, Marco Centenaro, and Lorenzo Vangelista. Performance evaluation of lora networks in a smart city scenario. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–7. IEEE, 2017.

[74] Nitin Mangalvedhe, Rapeepat Ratasuk, and Amitava Ghosh. Nb-iot deployment study for low power wide area cellular iot. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–6. IEEE, 2016.

[75] Pavel Masek, Jiri Hosek, and Marek Dubrava. Influence of m2m communication on lte networks. In *Sbornik prispevku studentske konference Zvule*, volume 1, page 2014, 2014.

[76] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008.

[77] Konstantin Mikhaylov, Juha Petaejaejaervi, and Tuomo Haenninen. Analysis of capacity and scalability of the lora low power wide area network technology. In *European Wireless 2016; 22th European Wireless Conference*, pages 1–6. VDE, 2016.

[78] Konstantin Mikhaylov, Juha Petäjäjärvi, and Janne Janhunen. On lorawan scalability: Empirical evaluation of susceptibility to inter-network interference. In *Networks and Communications (EuCNC), 2017 European Conference on*, pages 1–6. IEEE, 2017.

[79] Hussein Mroue, A Nasser, Sofiane Hamrioui, Benoît Parrein, Eduardo Motta-Cruz, and Gilles Rouyer. Mac layer-based evaluation of iot technologies: Lora, sigfox and

nb-iot. In *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*, pages 1–5. IEEE, 2018.

[80] Kishor Krishnan Nair, Adnan M Abu-Mahfouz, and Samuel Lefophane. Analysis of the narrow band internet of things (nb-iot) technology. In *2019 Conference on Information Communications Technology and Society (ICTAS)*, pages 1–6. IEEE, 2019.

[81] Akihiro Nakao, Ping Du, Yoshiaki Kiriha, Fabrizio Granelli, Anteneh Atumo Gebremariam, Tarik Taleb, and Miloud Bagaa. End-to-end network slicing for 5g mobile networks. *Journal of Information Processing*, 25:153–163, 2017.

[82] Nokia Networks. Lte-m, optimizing lte for the internet of things, 2016.

[83] NSNAM. Ns-3.29 documentation, September 2018.

[84] Moises Nunez Ochoa, Arturo Guizar, Mickael Maman, and Andrzej Duda. Evaluating lora energy efficiency for adaptive networks: From star to mesh topologies. In *Wireless and Mobile Computing, Networking and Communications (WiMob),*, pages 1–8. IEEE, 2017.

[85] Nathalie Omnes, Marc Bouillon, Gael Fromentoux, and Olivier Le Grand. A programmable and virtualized network & it infrastructure for the internet of things: How can nfv & sdn help for facing the upcoming challenges. In *Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on*, pages 64–69. IEEE, 2015.

[86] Omni. Battery life calculator, April 2019.

[87] Barry O'Neill. A problem of rights arbitration from the talmud. *Mathematical Social Sciences*, 2(4):345–371, 1982.

[88] Denis A Pankratev, Andrey A Samsonov, and Anastasiia D Stotckaia. Wireless data transfer technologies in a decentralized system. In *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pages 620–623. IEEE, 2019.

[89] Juha Petäjäjärvi, Konstantin Mikhaylov, Marko Pettissalo, Janne Janhunen, and Jari Iinatti. Performance of a low-power wide-area network based on lora technology: Doppler robustness, scalability, and coverage. *International Journal of Distributed Sensor Networks*, 13(3):1550147717699412, 2017.

[90] Manuel Pulido, Joaquín Sánchez-Soriano, and Natividad Llorca. Game theory techniques for university management: an extended bankruptcy model. *Annals of Operations Research*, 109(1-4):129–142, 2002.

[91] Fernando MV Ramos, Diego Kreutz, and Paulo Verissimo. Software-defined networks: On the road to the softwarization of networking. *Cutter IT journal*, 2015.

[92] Rapeepat Ratasuk, Nitin Mangalvedhe, Zhilan Xiong, Michel Robert, and David Bhatoolaul. Enhancements of narrowband iot in 3gpp rel-14 and rel-15. In *2017 IEEE Conference on Standards for Communications and Networking (CSCN)*, pages 60–65. IEEE, 2017.

[93] Rapeepat Ratasuk, Benny Vejlgaard, Nitin Mangalvedhe, and Amitava Ghosh. Nb-iot system for m2m communication. In *Wireless Communications and Networking Conference (WCNC), 2016 IEEE*, pages 1–5. IEEE, 2016.

[94] Brecht Reynders, Wannes Meert, and Sofie Pollin. Range and coexistence analysis of long range unlicensed communication. In *Telecommunications (ICT), 2016 23rd International Conference on*, pages 1–6. IEEE, 2016.

[95] Brecht Reynders, Wannes Meert, and Sofie Pollin. Power and spreading factor control in low power wide area networks. In *Communications (ICC), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

[96] Brecht Reynders, Qing Wang, Pere Tuset-Peiro, Xavier Vilajosana, and Sofie Pollin. Improving reliability and scalability of lorawans through lightweight scheduling. *IEEE Internet of Things Journal*, 5(3):1830–1842, 2018.

[97] Pedro HA Rezende and Edmundo RM Madeira. An adaptive network slicing for lte radio access networks. In *Wireless Days (WD), 2018*, pages 68–73. IEEE, 2018.

[98] Alberto Rico-Alvarino, Madhavan Vajapeyam, Hao Xu, Xiaofeng Wang, Yufei Blankenship, Johan Bergman, Tuomas Tirronen, and Emre Yavuz. An overview of 3gpp enhancements on machine to machine communications. *IEEE Communications Magazine*, 54(6):14–21, 2016.

[99] Boris Rogier. Network performance : Links between latency throughput and packet loss, 2016.

[100] Alvin E Roth and Marilda Sotomayor. a study in game-theoretic modeling and analysis. *Econometric Society Monographs*, 18, 1990.

[101] Walid Saad, Zhu Han, Rong Zheng, Mérouane Debbah, and H Vincent Poor. A college admissions game for uplink user association in wireless small cell networks. In *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*, pages 1096–1104. IEEE, 2014.

[102] Nitin Salodkar, Abhay Karandikar, and Vivek S Borkar. A stable online algorithm for energy-efficient multiuser scheduling. *IEEE Transactions on Mobile Computing*, 9(10):1391–1406, 2010.

[103] Prabodini Semasinghe, Setareh Maghsudi, and Ekram Hossain. Game theoretic mechanisms for resource management in massive wireless iot systems. *IEEE Communications Magazine*, 55(2):121–127, 2017.

[104] Lloyd S Shapley. A value for n-person games. *Classics in game theory*, page 69, 1997.

[105] Vishal Sharma, Gaurav Choudhary, Ilsun You, Jae Deok Lim, and Jeong Nyeo Kim. Self-enforcing game theory-based resource allocation for lorawan assisted public safety communications. *arXiv preprint arXiv:1804.07204*, 2018.

[106] Hsu-Shih Shih, Huan-Jyh Shyur, and E Stanley Lee. An extension of topsis for group decision making. *Mathematical and Computer Modelling*, 45(7-8):801–813, 2007.

[107] SA Sigfox. Sigfox technology overview, 2018.

[108] Mariusz Slabicki, Gopika Premsankar, and Mario Di Francesco. Adaptive configuration of lora networks for dense iot deployments. In *16th IEEE/IFIP Network Operations and Management Symposium (NOMS 2018)*, pages 1–9, 2018.

[109] Andrew S Tanenbaum. a d. wetherall. *Computer networks. 5th ed. Boston: Pearson*, 2011.

[110] Fengxiao Tang, Zubair Md Fadlullah, Bomin Mao, and Nei Kato. An intelligent traffic load prediction based adaptive channel assignment algorithm in sdn-iot: A deep learning approach. *IEEE Internet of Things Journal*, 2018.

[111] Qualcomm Technologies. Leading the lte iot evolution to connect the massive internet of things, June 2017.

[112] Mikael Touati, Rachid El-Azouzi, Marceau Coupechoux, Eitan Altman, and Jean-Marc Kelif. A controlled matching game for wlans. *IEEE Journal on Selected Areas in Communications*, 35(3):707–720, 2017.

[113] Mostafa Uddin, Sarit Mukherjee, Hyunseok Chang, and TV Lakshman. Sdn-based multi-protocol edge switching for iot service automation. *IEEE Journal on Selected Areas in Communications*, 2018.

[114] Omkarprasad S Vaidya and Sushil Kumar. Analytic hierarchy process: An overview of applications. *European Journal of operational research*, 169(1):1–29, 2006.

[115] Floris Van den Abeele, Jetmir Haxhibeqiri, Ingrid Moerman, and Jeroen Hoebeke. Scalability analysis of large-scale lorawan networks in ns-3. *IEEE Internet of Things Journal*, 4(6):2186–2198, 2017.

[116] Nadège Varsier and Jean Schwoerer. Capacity limits of lorawan technology for smart metering applications. In *Communications (ICC), 2017 IEEE International Conference On*, pages 1–6. IEEE, 2017.

[117] Nuttakit Vatcharatiansakul, Panwit Tuwanut, and Chotipat Pornavalai. Experimental performance evaluation of lorawan: A case study in bangkok. In *Computer*

Science and Software Engineering (JCSSE), 2017 14th International Joint Conference on, pages 1–4. IEEE, 2017.

[118] Benny Vejlgaard, Mads Lauridsen, Huan Nguyen, István Z Kovács, Preben Mogensen, and Mads Sorensen. Coverage and capacity analysis of sigfox, lora, gprs, and nb-iot. In *2017 IEEE 85th vehicular technology conference (VTC Spring)*, pages 1–5. IEEE, 2017.

[119] VM Vishnevsky, KE Samouylov, VA Naumov, A Krishnamoorthy, and N Yarkina. Multiservice queueing system with map arrivals for modelling lte cell with h2h and m2m communications and m2m aggregation. In *International Conference on Distributed Computer and Communication Networks*, pages 63–74. Springer, 2017.

[120] Meng Wang, Bo Cheng, Xuan Liu, Yi Yue, Biyi Li, and Junliang Chen. Poster: A sdn/nfv-based iot network slicing creation system. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 666–668. ACM, 2018.

[121] Y-P Eric Wang, Xingqin Lin, Ansuman Adhikary, Asbjörn Grövlen, Yutao Sui, Yufei Blankenship, Johan Bergman, and Hazhir S Razaghi. A primer on 3gpp narrowband internet of things (nb-iot). *arXiv preprint arXiv:1606.04171*, 2016.

[122] Antoine Waret, Megumi Kaneko, Alexandre Guitton, and Nancy El Rachkidy. Lora throughput analysis with imperfect spreading factor orthogonality. *arXiv preprint arXiv:1803.06534*, 2018.

[123] Andrew J Wixted, Peter Kinnaird, Hadi Larijani, Alan Tait, Ali Ahmadinia, and Niall Strachan. Evaluation of lora and lorawan for wireless sensor networks. In *SENSORS, 2016 IEEE*, pages 1–3. IEEE, 2016.

[124] Anupma Yadav and S.C Jayswal. Using geometric mean method of analytical hierarchy process for decision making in functional layout. *International Journal of Engineering Research and Technology (IJERT)*, 2:5, 2013.

[125] Zainab Zaidi, Vasilis Friderikos, Zarrar Yousaf, Simon Fletcher, Mischa Dohler, and Hamid Aghvami. Will sdn be part of 5g? *IEEE Communications Surveys & Tutorials*, 2018.

[126] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. In *ACM Sigmod Record*, volume 25, pages 103–114. ACM, 1996.

[127] Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1(2):141–182, 1997.

[128] Hongli Zhao and Hailin Jiang. Lte-m system performance of integrated services based on field test results. In *2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*. IEEE, 2016.

[129] Yifeng Zhao, Huayu Yang, Kai Liu, Lianfen Huang, Mingjun Shi, and Meng Zhang. A random-access algorithm based on statistics waiting in lte-m system. In *2017 12th International Conference on Computer Science and Education (ICCSE)*, pages 214–218. IEEE, 2017.

[130] Zhenyu Zhou, Junhao Feng, Yunjian Jia, Shahid Mumtaz, Kazi Mohammed Saidul Huq, Jonathan Rodriguez, and Di Zhang. Energy-efficient game-theoretical random access for m2m communications in overlapped cellular networks. *Computer Networks*, 2017.

[131] Juan Carlos Zuniga and Benoit Ponsard. Sigfox system description. *LPWAN@ IETF97, Nov. 14th*, 2016.

List of Publications

Journal Articles

Samir Dawaliby, Abbas Bradai, and Yannis Pousset. Adaptive dynamic network slicing in LoRa networks. *Future Generation Computer Systems*, 2019.

Samir Dawaliby, Abbas Bradai, and Yannis Pousset. Distributed Network Slicing in Large Scale IoT based on Coalitional Multi-Game Theory. *IEEE Transactions on Network and Service Management*, 2019.

Samir Dawaliby, Abbas Bradai, and Yannis Pousset. Joint Spreading Factor and Transmission Power Optimization in LoRa Network Slicing. *Internet of Things*, 2019.

Samir Dawaliby, Abbas Bradai, Yannis Pousset, and Christian Chatellier. Joint Energy and QoS-Aware Memetic-Based Scheduling for M2M Communications in LTE-M. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2018.

International Conferences

Samir Dawaliby, Abbas Bradai, and Yannis Pousset. In depth performance evaluation of LTE-M for M2M communications. In *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–8. IEEE, 2016.

Samir Dawaliby, Abbas Bradai, and Yannis Pousset. Scheduling optimization for M2M communications in LTE-M. In *2017 IEEE International Conference on Consumer Electronics (ICCE)*, pages 126–128. IEEE, 2017.

Samir Dawaliby, Abbas Bradai, and Yannis Pousset. Network Slicing Optimization in Large Scale LoRa Wide Area Networks. In *IEEE Conference on Network Softwarization (NETSOFT)*, 2019.

Samir Dawaliby, Abbas Bradai, Yannis Pousset, and Roberto Riggio. Dynamic Network Slicing for LoRaWAN. In *2018 14th International Conference on Network and Service Management (CNSM)*, pages 134–142. IEEE, 2018.