Jamming on LoRaWAN Networks : from modelling to detection

Ivan Marino Martinez Bolivar

DOCTORAT BRETAGNE LOIRE MATHSTIC

INSA RENNES

THESE DE DOCTORAT DE

L'INSTITUT NATIONAL DES SCIENCES APPLIQUEES RENNES

ECOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : télécommunications

Par

Ivan Marino MARTINEZ BOLIVAR

Jamming on LoRaWAN Networks: from modelling to detection

Thèse présentée et soutenue à Rennes, le 12 janvier 2021
Unité de recherche : Institut d'Electronique et des Technologies du numéRique
Thèse N° : 21ISAR 01 / D21 - 01

Rapporteurs avant soutenance :

Claire GOURSAUD	Maître de Conférence, INSA de Lyon, INRIA
Laurent CLAVIER	Professeur IMT Lille Douai, IRCICA

Composition du Jury :

Président :	Olivier BERDER	Professeur Université de Rennes I, IRISA
Examinateurs :	Maroun CHAMOUN	Professeur Université St Joseph, ESIB, Liban
	Olivier BERDER	Professeur Université de Rennes I, IRISA
Dir. de thèse :	Fabienne NOUVEL	Maître de Conférence/HDR, INSA de Rennes, IETR
Co-dir. de thèse :	Philippe TANGUY	Maître de Conférence, Université Bretagne Sud, LabSTICC

Invité :
Samer LAHOUD — Professeur Université St Joseph, ESIB, Liban

Dedicado a Ana Victoria, mi amada abuelita

REMERCIEMENTS

Au terme de ce mémoire de thèse je tiens à remercier toutes les personnes qui ont contribué et qui m'ont aidé lors de la rédaction.

Tout d'abord, j'adresse mes remerciements à mes deux directeurs de thèse : Mme. Fabienne Uzel de l'INSA de Rennes et M. Philippe Tanguy de l'Université Bretagne Sud. Je suis ravi d'avoir travaillé en leur compagnie car outre son appui scientifique, ils ont toujours été là pour me soutenir et me conseiller au cours de l'élaboration de cette thèse.

Je remercie également professeurs M. Samer Lahoud et M. Melhem El Helou pour l'accueil chaleureux au sein du laboratoire CIMTI à l'USJ de Beyrouth. Ces trois mois de recherche ont été particulièrement enrichissants aussi bien humainement qu'intellectuellement.

Mme. Claire Goursaud et M. Laurent Clavier m'ont fait l'honneur d'être rapporteurs de ma thèse, ils ont pris le temps de m'écouter et de discuter avec moi. Leurs remarques m'ont permis d'envisager mon travail sous un autre angle. Pour tout cela je les remercie.

Je tiens à remercier M. Maroun Chamoun et M. Olivier Berder pour avoir accepté de participer à mon jury de thèse et pour sa participation scientifique ainsi que le temps qu'ils ont consacré à ma recherche.

Enfin, je tiens a remercier ma famille qui m'a aidé tout au long de ma scolarité et qui m'ont donné leur soutien moral tout au long de ces longues années loin de la Colombie. Et je finirais cette liste de remerciements par celui qui m'a apporté une vie heureuse hors du laboratoire. Je lui remercie donc pour sa compréhension sur la rédaction interminable, les corrections de chapitres et les mois passés à Beyrouth qui m'ont permis de finir cette thèse dans un parfait confort.

Ivan MARTINEZ

ABSTRACT

The Internet of Things (IoT) knows a great expansion with a predicted of 75 billion of IoT devices by 2025 [Cisco]. Low Power Wide Area Network (LPWAN) have emerged as one of the most promising IoT wireless technologies. LPWAN technologies are appropriate for devices that require to send small amounts of data over a long range, at a low cost, while maintaining long battery life. Achieving both might sound counter-intuitive, but it is possible through the use of lower frequencies, lower bit rates, and more robust modulation techniques such as Chirp Spread Spectrum (CSS) [1].

Sigfox, LoRaWAN, and Narrowband IoT (NB-IoT) are the three leading LPWAN technologies [2]. While, NB-IoT is a 3rd Generation Partnership Project (3GPP) standard that, as other mobile technologies like 4G and 5G, uses licensed spectrum, Sigfox and LoRaWAN operate in the sub-GHz free band.

As for any communication systems, security is one of the biggest concerns. In LPWAN, security is currently provided by symmetric-key algorithms such as AES 128 at upper levels. In the case of LoRaWAN it offers application level payload encryption and network level integrity. This means that, if implemented well, it can reasonably be secured against upper level attacks such as Distributed Denial of Service (DDoS). But nevertheless, this does not shield it against attacks at a lower level such as jammer-type attacks.

A jamming attack takes place at the PHY layer. It is an external node that sends unauthenticated packets to every wireless station in the network with the aim of disrupting communications by decreasing the Signal-to-noise ratio (SNR). It can be done in an intelligent way, where the jammer knows exactly whether or not a node is transmitting (channel-aware), or in a brute force attack (channel-oblivious) where the jammer starts to transmit on the channel permanently [3].

In this thesis, we focus on LoRaWAN networks under jamming attacks. We address this problem through a holistic approach that encompasses three fundamental axes:

i) Network modeling:

- A new mathematical model for LoRaWAN under jamming.
- An enhancement of a ns3 module for LoRaWAN under jamming.

ii) Performance evaluation of LoRaWAN under jamming:

- Extensive simulation campaigns.
- A test-bed for LoRaWAN under jamming.

iii) Jamming countermeasures:

- a jamming attack detection algorithm using a statistical approach.
- a jamming attack detection algorithm using machine learning.

LIST OF PUBLICATIONS

---

[1] Ivan Martinez, Philippe Tanguy and Fabienne Nouvel, *On the performance evaluation of LoRaWAN under Jamming*. 2019 12th IFIP/IEEE Wireless and Mobile Networking Conference (WMNC), Paris, France. Available: `https://doi.org/10.23919/wmnc.2019.8881830`.

[2] Ivan Martinez, Fabienne Nouvel, Samer Lahoud, Philippe Tanguy and Melhem El Helou, *On the Performance Evaluation of LoRaWAN with Re-transmissions under Jamming*, 2020 IEEE Symposium on Computers and Communications (ISCC), Rennes (online), France. Available: `https://doi.org/10.1109/iscc50000.2020.9219644`.

[3] Ivan Martinez, Fabienne Nouvel, Samer Lahoud, Philippe Tanguy and Melhem El Helou, *Analytical Model of LoRaWAN-like Channel Access with Confirmed Traffic under Jamming*, under revision.

[4] Ivan Martinez, Philippe Tanguy and Fabienne Nouvel, *An RNN Machine Learning Approach for Jamming Detection in LoRaWAN Networks*, to be submitted.

[5] Ivan Martinez, Fabienne Nouvel, J.C Prévotet, *LPWAN networks, défis et opportunuités,* Journée SEOC commune GDR Réseau et SOC2 , Paris, Avril 2019.

Résumé en français

Au début des années 2000, ce que nous appelons aujourd'hui la maison intelligente s'appelait plus communément la domotique. Cette technologie était basée sur les premières normes de technologies sans fil telles que la WiFi et le Bluetooth. Ainsi, au cours de cette décennie, ces technologies ont commencé à être utilisées pour différentes tâches qui nécessitent beaucoup de couverture et nous avons commencé à entendre des termes comme les réseaux *multi-hop*, MANET (*Mobile Ad-hoc Netwokrs*) ou WSN (*Wireless Sensor Networks*). L'idée était d'étendre leur couverture par la mise en œuvre de topologies de réseau de type maillage [4].

À cette époque, il y avait aussi l'accès aux réseaux de téléphones portables tels que 2G et 3G, ils peuvent fournir une plus grande couverture, mais ils exigent beaucoup d'énergie des dispositifs finaux. Par conséquent, les nouvelles exigences en matière d'applications IdO (Internet des Objets) ont mené à l'émergence de nouvelles technologies de communications sans fil : LPWAN. Ce n'est qu'en 2013 que le terme a été inventé [5]. Il s'agit d'un réseau avec une faible consommation d'énergie, une longue portée et des caractéristiques de communication à faible coût, à la fois énergétique et économique. Il fournit une communication à longue portée jusqu'à 40 km dans les zones rurales et 1-5 km dans les zones urbaines [6, 7]. En outre, les noeuds sont conçus pour être très économes en énergie (promettant jusqu'à 10 ans de vie de la batterie).

Ceci dit, de nos jours l'IdO (Internet des Objets) connaît une grande expansion avec une prévision de 75 milliards d'objets connectés [8] en 2025. Aujourd'hui, Sigfox, LoRaWAN (*Long Range Wide Area Networks*) et NB-IoT (*Narrowband IoT*) sont les trois principales technologies LPWAN [2]. Alors que NB-IoT est issue de la norme 3GPP (*Third Generation Partnership Project*) qui, comme d'autres technologies mobiles telles que la 4G et la 5G, utilise un spectre sous licence, Sigfox et LoRaWAN fonctionnent dans une bande de fréquence libre inférieure au 1 GHz.

Comme pour tout système de communication, la sécurité est une exigence importante. Pour LPWAN, la sécurité est actuellement assurée par des algorithmes de cryptographie à clé symétrique tels que l'AES 128. Dans le cas de LoRaWAN, cela permet un chiffrement des données au niveau de l'application et garantit l'intégrité du réseau. Cela signifie que, s'il est bien implémenté, le réseau peut être sécurisé convenablement contre des attaques de haut niveau telles que l'attaque par rejeu et le DDoS (attaque par déni de service distribué). Mais néanmoins, cela ne le protège pas contre les attaques de bas niveau telles que les attaques par brouillage.

Une attaque par brouillage affecte la couche PHY. C'est un nœud externe qui envoie des paquets non authentifiés dans le but de perturber les communications, générant des interférences qui réduisent le rapport signal sur bruit (RSB). Cela peut être réalisé de manière intelligente, lorsque le brouilleur sait exactement si un nœud transmet ou non (écoute du canal), ou par une

attaque aveugle (sans connaissance du canal), le brouilleur commençant à transmettre sur le canal en permanence [3].

Dans cette thèse, nous nous concentrons sur les réseaux LoRaWAN soumis à des attaques par brouillage. Nous abordons ce problème à travers une approche holistique qui englobe trois axes principaux: (i) modélisation du réseaux, (ii) évaluation de la performance, et (iii) détection du *jamming*.

Réseaux LoRaWAN

La pile protocolaire de LoRaWAN est représentée en Figure 1. LoRa définit la couche PHY et LoRaWAN fait référence à la couche MAC (*Medium Access Control*), qui définit la façon dont le support est partagé entre la passerelle et les nœuds terminaux.

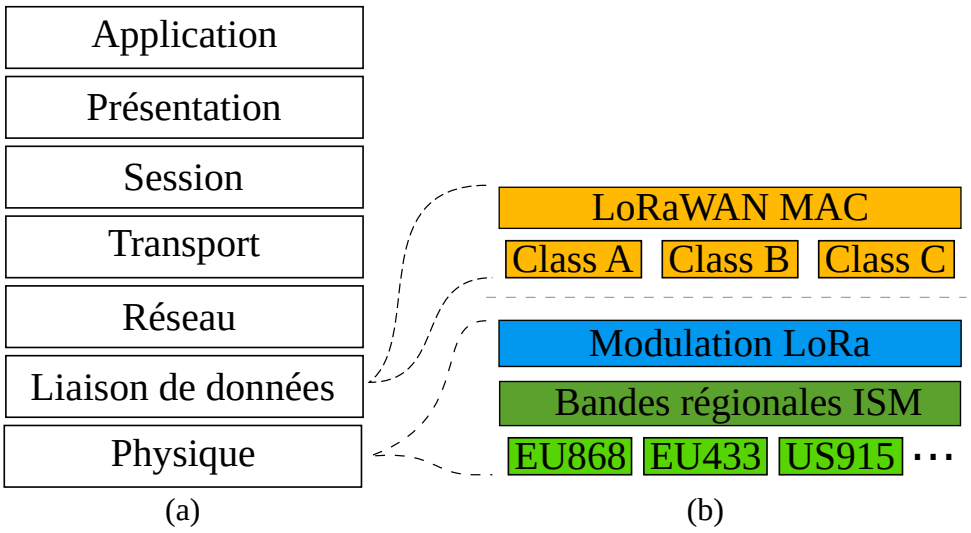


Figure 1: Pile protocolaire du LoRaWAN : (a) modèle OSI, (b) LoRaWAN

LoRaWAN est un protocole basé sur Aloha, dans lequel l'instant d'accès au canal est sélectionné aléatoirement. Il existe trois modes dans lesquels les nœuds peuvent être configurés : Classe A, pour les transmissions en liaison montante peu fréquentes, Classe B pour les cas d'utilisation nécessitant des transmissions en liaison descendante plus fréquentes et, Classe C pour les nœuds en écoute continue. L'architecture du réseau loRaWAN est représentée dans la figure 2 ; un réseau LoRaWAN est formé par:

- *End-device* / Utilisateur/ Nœud : il est chargé de collecter les données et de les envoyer à la passerelle.
- Passerelle: Elle reçoit/envoit des données entre le réseau principal (IP) et les utilisateurs. Elle n'effectue aucun filtrage de trames provenant de ED ou des couches supérieures.
- Serveur de réseau : c'est le nœud le plus intelligent du réseau Il est responsable de la gestion de toutes les passerelles du réseau, et doit effectuer le routage les messages entrants/sortants du serveur d'application.

- Serveur d'application : c'est un logiciel offrant un contexte d'exécution pour des composants applicatifs.
- Serveur d'authentification : il est en charge de gérer l'authentification de chaque utilisateur-nœud dans le réseau.

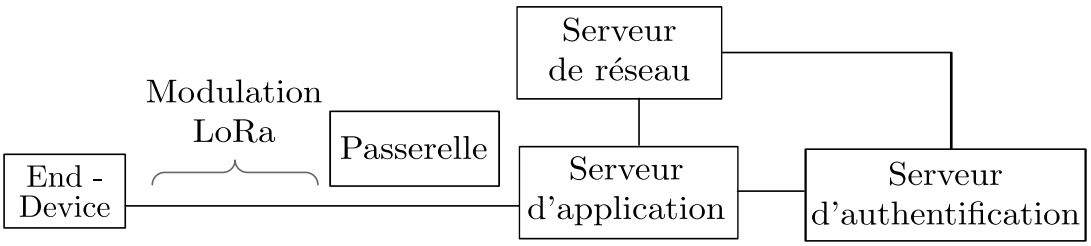


Figure 2: Architecture de LoRaWAN

Dans la Classe A, les transmissions se font en suivant un protocole Pure Aloha. Comme le montre la figure 3, l'utilisateur a son propre cycle de transmission et chaque fois qu'une transmission est effectuée, la passerelle peut envoyer des informations en utilisant deux courtes fenêtres de réception en liaison descendante (RX1 et RX2). Si des paquets supplémentaires doivent être envoyés par la passerelle, ils doivent être reportés jusqu'au prochain événement de transmission (respect du *duty-cycle*).

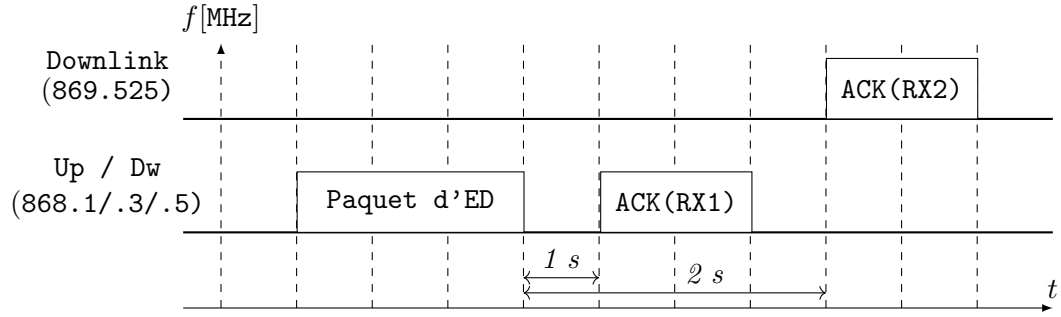


Figure 3: LoRaWAN Classe A

Modulation LoRa

La modulation LoRa utilise une technique du type CSS (*Chirp Spread Spectrum*) qui représente les symboles comme des changements instantanés de la fréquence d'un chirp. Un chirp est alors un signal dont la fréquence change linéairement selon le facteur d'étalement. Ainsi, un changement instantané de sa fréquence constitue un symbole [9]. Le fait de distribuer un symbole sur la bande passante du spectre rend le signal plus résistant aux bruits sélectifs en fréquence, aux interférences multi-canaux et à l'effet Doppler. Cependant, cela conduit à une diminution de l'efficacité spectrale et induit des temps de transmission beaucoup plus longs.

Le SF (*Spreading Factor*) est alors un paramètre ajustable qui indique le nombre de bits d'un symbole. Ce paramètre permet donc de calculer le temps de transmission ToA (*Time on*

Table 1: *Time on air* (ToA) et SF

s	T_{sym} $[ms]$	T_{pre} $[ms]$	$T_{CAD}[ms]$	T_{packet} $(50\ B)[ms]$	T_{packet} $(250\ B)[ms]$
7	1.024	12.544	0.128	97.536	389.376
8	2.048	25.088	0.2304	174.592	686.592
9	4.096	50.176	0.4352	328.704	1229.824
10	8.192	100.352	0.8448	616.448	2254.848
11	16.384	200.704	1.664	1150.917	4100.096
12	32.768	401.408	3.3024	2138.112	7544.832
Equation.	(2.2)	(2.4)	(2.7)	(2.3)	(2.3)

Air) par paquet en fonction de sa longueur. La table 1 présente différents paramètres temporels basés sur ce ToA. Nous considérons 8 octets comme longueur de préambule, une bande passante de 125 kHz, un taux de codage $CR = 1$ et une charge utile de 50 et 250 `bytes` respectivement.

Sécurité en LoRaWAN

Les procédures de sécurité dans LoRaWAN sont mises en œuvre conformément à l'algorithme AES-128 bits. Il fournit la confidentialité des données de bout en bout (chiffrement de charge utile) et l'intégrité des données au niveau du réseau (authentification des données). La spécification définit deux façons pour connecter les nœuds au réseau ; OTAA (*Over The Air Activation*) et ABP (*Activation By Personalisation*). La première définit une procédure pour dériver les clés de session à partir d'une clé pré-activée partagée entre l'utilisateur et le serveur d'application, et la seconde suppose que les clés de session sont stockées des deux côtés avant de commencer la communication. Avec ABP, les utilisateurs utilisent les mêmes clés de session tout au long de leur durée de session.

Il existe également différents paramètres liés aux réseaux LoRaWAN comme le duty-cycle, qui est une restriction liée au temps maximum qu'un nœud peut transmettre et l'ADR (*Adaptive Data Rate*) qui est un algorithme dédié au contrôle des paramètres de transmission des nœuds.

Modélisation des réseaux

La modélisation des réseaux constitue la contribution principale de cette thèse. Nous proposons deux outils permettant d'évaluer l'impact des attaques de brouillage sur les réseaux LoRaWAN.

Tout d'abord nous proposons un modèle mathématique qui considère un réseau LoRaWAN. Le modèle permet de dimensionner un réseau avec une seule passerelle et plusieurs nœuds bidirectionnels sous l'attaque de brouilleurs du type *channel-oblivious jammer*. Le modèle considère également un mécanisme de re-transmission qui vise à améliorer la probabilité de succès du message. Pour ce faire, toutes les transmissions bien reçues côté serveur de réseau sont accusées avec l'envoi d'un paquet ACK. Pour le développent de ce modèle nous avons pris plusieurs hypothèses simplificatrices:

- Tous les paquets sont envoyés de façon aléatoire selon un processus de Poisson (paquets d'utilisateurs, jammers et ACKs).

- Un canal idéal: toutes les transmissions qui se déroulent en parallèle (même canal, SF et s'ils coïncident dans leur temps de transmission, aussi minime soit-il.) sont considérées comme perdues.
- Une seule fenêtre de transmission (RX1) : nous considérons que la passerelle transmet les ACK en utilisant un canal dédié, contrairement à LoRaWAN qui le fait le fait en considérant deux fenêtres de transmission (RX1 et RX2).
- Les ACK sont envoyés systématiquement après chaque réception en utilisant le même SF utilisé par l'utilisateur.

L'objectif de ce modèle mathématique est de fournir une estimation rapide des performances d'un réseau en présence de brouilleurs. Il permet donc d'obtenir quatre métriques d'évaluation différentes : (i) *throughput* des utilisateurs, définit comme le débit des paquets [packets/s], (ii) le goodput des utilisateurs ($\mathcal{O}_u$) définit comme le débit des messages d'utilisateur [messages/s], (iii) la probabilité de succès par message ($\mathcal{P}_m$) et (iv) l'espérance de re-transmissions ($\mathcal{E}(r)$).

Dans un second temps, nous étendons un modèle de simulation de LoRaWAN existant [10–13] basé sous ns3 qui permet non seulement de modéliser le *channel-oblivious jammer*, mais aussi le brouilleur qui a la capacité de détecter l'activité dans le réseau et de le brouiller en conséquence. Contrairement au modèle mathématique, ce modèle permet de modéliser un réseau LoRaWAN en considérant:

- Plusieurs passerelles.
- Modèle de propagation.
- Modèle d'interférence qui prenne en compte le *capture effect* et des SF non orthogonaux.
- Modèle de consommation de courant électrique des nœuds.
- Limitations matérielles des nœuds et de la passerelle : nombre de canaux de réception limité.
- Un système de retransmission de trames qui peut être paramétré en fonction du pourcentage des paquets qui doivent être re-transmis.

La figure 4 représente les principales classes et connexions du module ns3. C'est une collection de classes qui travaillent ensemble pour modéliser les utilisateurs, les passerelles et le serveur réseau à différents niveaux. Il comprend les niveaux PHY, MAC et le niveau Application pour chaque type d'appareil. Certaines classes modélisent la pile de protocole sur un dispositif donné tandis que d'autres sont utilisées pour modéliser d'autres aspects comme le modèle de propagation ou la consommation d'énergie.

Comme nous l'avons vu, dans le modèle mathématique, nous avons pris diverses hypothèses simplificatrices pour rendre le modèle résolvable, comme l'utilisation d'une fenêtre de réception unique dans une bande de fréquence séparée et l'orthogonalité des facteurs d'étalement. Par conséquent, une comparaison de ce système avec un modèle LoRaWAN plus précis devient

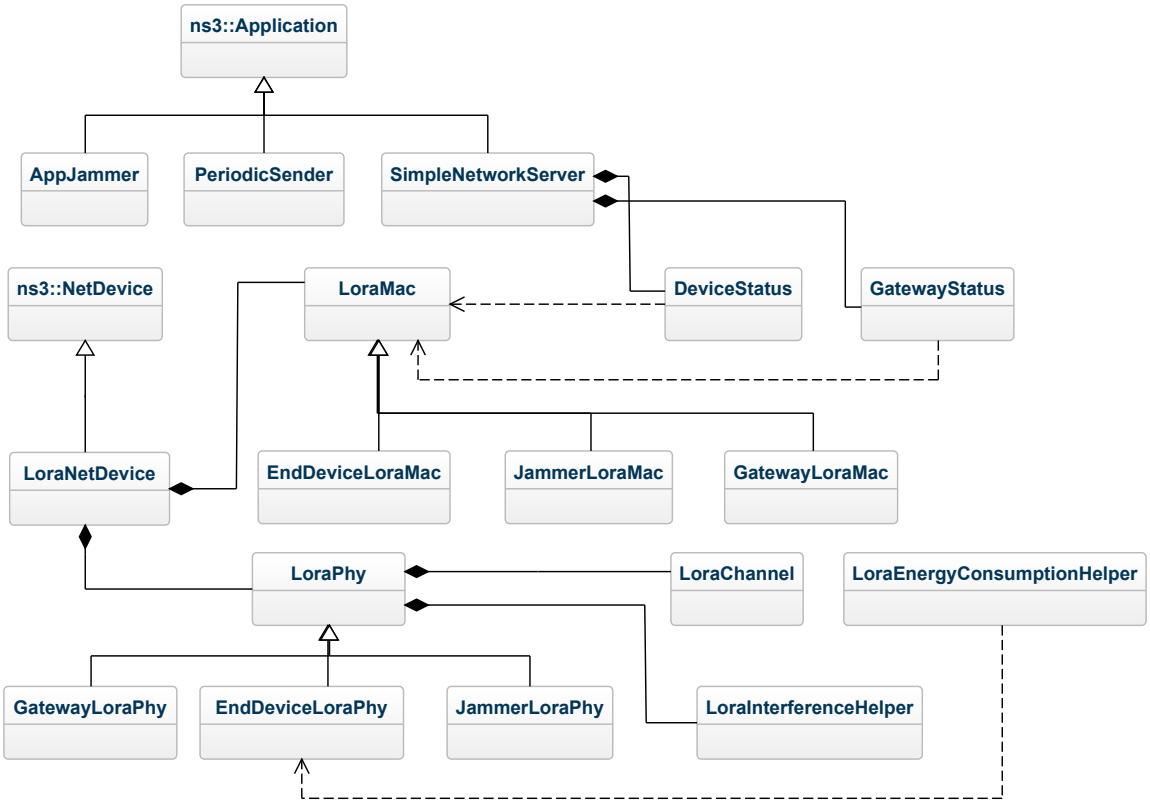


Figure 4: Schéma UML du module ns-3.

nécessaire pour vérifier si les résultats obtenus sont similaires à ceux qui seraient obtenus dans LoRaWAN.

Nous avons effectué une campagne de simulation dans ns3 pour comparer les deux systèmes. Les paramètres de simulation sont résumés en Table 2. Nous avons considéré différents scénarios en faisant varier le nombre d'utilisateurs $N_u = 1, 250, 500, ..., 4000$, le nombre maximum de retransmissions par message ($0 < r \leq 8$). La longueur des paquets utilisateur (l_u) est fixé à 50 bytes, la longueur des paquet ACK (l_u) est fixé à 10 bytes, le duty-cycle pour les utilisateurs (d_u) est de 0.01.

En ce qui concerne les canaux RF utilisés, nous considérons que les transmissions en uplink sont faites en utilisant trois canaux : 861.1/.3/.5 MHz avec un BW de 125 kHz. Nous considérons également un modèle de pertes (PL) du type *outdoor* avec les paramètres présentés dans la table 5.1:

$$PL = 10n \log_{10}(d) + PL_0 + L_h \log_{10}(h_{ED}) + X_\sigma \tag{1}$$

où, d est la distance entre la passerelle et l'utilisateur, PL_0 est le PL à une distance de référence d_0, L_h est la perte supplémentaire due à la hauteur de l'antenne ED h_{ED}, et le bruit blanc est représenté par une variable aléatoire Gaussienne X_sigma d' écart type $sigma$ (en dB).

Nous considérons que la passerelle est capable de décoder jusqu'à 18 paquets en parallèle (3 canaux et 6 SF) dans les deux cas (modèle analytique et de simulation) et une distribution

Table 2: Paramètres de simulation

	Paramètre	Valeur
Réseau	N_u	$1, 250, \ldots, 4000$
	l_u	50 bytes
	l_a	10 bytes
	d_u	0.01
	s	7 - 12
	ρ_u^s	$1/6\ \forall s$
	T_u	82.17 ms - 1.81 s
	T_{ack}^s	30,976 ms - 0.82 s
	r	0 - 8
	radius	5 km

	Paramètre	Valeur
Path-loss	d_0	40 m
	PL_0	140.7
	n	3.12
	L_h	-4.7
	h_{ED}	3 m
	σ	9.7 dB
Bande	Up / Dw	868.1/.3/.5 MHz
	Downlink	869.525 MHz
Simulation time		10 h

uniforme de SF. Pour la simulation en ns3 nous ajoutons la possibilité d'avoir deux fenêtres de réception (RX1 et RX2) comme indiqué dans la spécification. Les résultats obtenus sont représentes dans la figure 5. Nous présentons trois métriques d'évaluation : le *goodput* des utilisateurs, la probabilité de collision de paquets et la probabilité de succès des ACKs.

En ce qui concerne le *goodput*, nous avons un comportement similaire dans les deux modèles. Nous avons donc une courbe ascendante jusqu'à ce que nous atteignions un point où le *goodput* atteint un pic et la valeur commence à baisser. Cependant, ce point de *goodput* maximum est obtenu pour un nombre beaucoup plus élevé d'utilisateurs dans la simulation de LoRaWAN. En effet, le *goodput* maximum atteint à $N_u = 2000$, contrairement au résultats obtenus avec le modèle mathématique où ce point correspond à $N_u = 1000$.

Pour expliquer ce comportement, nous nous référerons aux deux hypothèses prises en compte pour le modèle mathématique : l'utilisation d'une seule fenêtre de réception (RX1) avec un canal de fréquence séparé et la non prise en compte du *capture effect*.

Tout d'abord, l'utilisation d'une première fenêtre de réception dans la même bande que les transmissions en liaison montante devrait faire diminuer le *goodput* car les paquets ACKs génèrent des collisions avec les paquets utilisateurs. Ceci peut être vu clairement dans la Figure 5 (c). Comme nous pouvons le voir, quand il y a un faible nombre d'utilisateurs (et donc une charge de trafic utilisateur plus faible) les deux modèles estiment la même probabilité de succès ACK, cependant quand le nombre d'utilisateurs augmente, cette probabilité diminue rapidement dans le LoRaWAN simulé.

Deuxièmement, le *capture effect* augmente la probabilité d'avoir un paquet bien décodé à la passerelle même en cas de collision. Cet effet est clairement présenté dans la Figure 5 (b). En effet, pour $N_u > 1250$, la probabilité de collision est plus faible dans le cas de LoRaWAN rendant le *goodput* obtenu plus élevé pour des charges de trafic utilisateur plus élevées.

Ensuite, en conséquence de ces deux facteurs, le premier abaisse la performance du réseau et le second fait le contraire; la cellule LoRaWAN simulée trouve un point d'équilibre moyen, où pour moins d'utilisateurs ($N_u < 1000$) elle suit presque le même comportement que le modèle mathématique. Au contraire, pour $N_u > 1000$, le fait de considérer *capture effect* devient beaucoup plus prépondérant que la probabilité de collisions dues aux ACKs envoyés

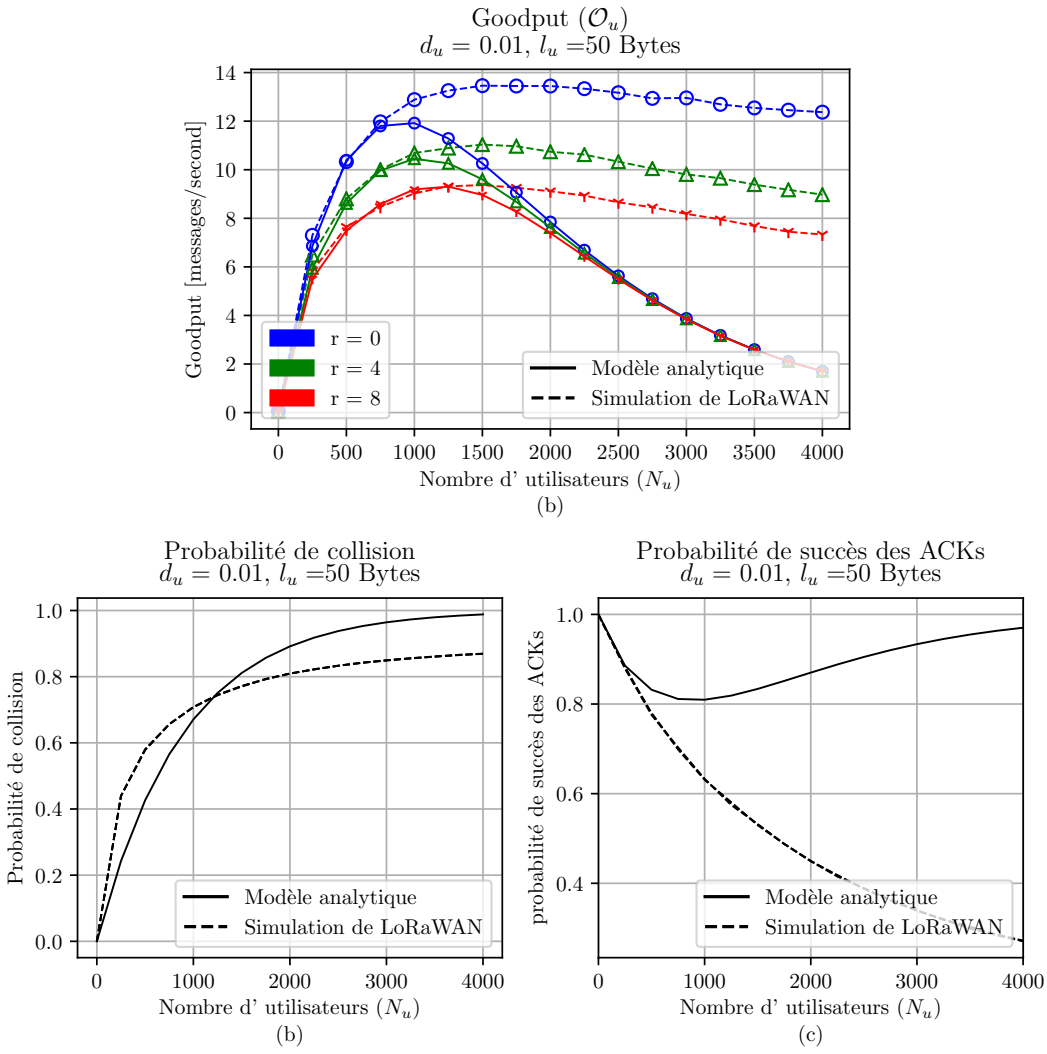


Figure 5: Comparaison du modèle mathématique et simulateur: (a) *Goodput*, (b) Collision Probabilité de collision and (c) Probabilité de succès des ACKs (RX1 en LoRaWAN).

dans la première fenêtre de réception. Cela signifie que le *goodput* obtenu ne diminue pas aussi rapidement que dans un réseau Aloha pur.

L'analyse mentionnée ci-dessus nous permet de conclure que le modèle analytique fonctionne bien pour des charges de trafic plus faibles lorsque les deux hypothèses fortes prises en considération sont valides. À partir de ce point, notre modèle est plus pessimiste.

Évaluation de la performance

Nous présentons également une analyse approfondie de l'évaluation des performances de LoRaWAN en présence de brouilleurs. Nous considérons plusieurs scénarios:

i) Un réseau LoRaWAN avec des limites matérielles et des canaux réalistes.

ii) Un réseau LoRaWAN avec plusieurs passerelles.

iii) Différentes stratégies de brouillage.

iv) L'impact de la position géographique des brouilleurs.

v) L'analyse de la consommation de courant électrique des nœuds et la durée de vie de la batterie.

Nous présentons ci-après les résultats obtenus pour les points (i) et (v) car ils sont les plus marquants, les autres peuvent être observés plus en détail dans le chapitre 5.

LoRaWAN avec des limites matérielles et des canaux réalistes.

Ce scénario prend en compte différentes particularités d'un déploiement réel. Pour cela, nous utilisons exclusivement le module de simulation en ns3 décrit précédemment. Nous présenterons d'abord un réseau LoRaWAN sans brouillage, puis nous évaluerons les performances du réseau avec brouillage. Nous considérons une cellule LoRaWAN composée de plusieurs utilisateurs et d'une passerelle. Les nœuds sont répartis uniformément autour de la passerelle dans un rayon de 5 km. Les nœuds sont statiques et configurés pour utiliser le meilleur SF possible comme fonction de leur position et de la sensibilité de la passerelle. Le modèle de propagation est le même que celui utilisé dans la simulation précédente.

Ainsi, selon le modèle de propagation et la distribution uniforme des nœuds, les SF sont répartis comme suit : $0.33, 0.22, 0.1, 0.09, 0.19, 0.07$. En ce qui concerne le profil d'application des nœuds, nous avons considéré une longueur de paquet l_u de 50 bytes, que tous les utilisateurs sont configurés pour utiliser un duty-cycle d_u de 0.01. Ainsi, selon le SF, le temps de transmission T_u varie de 82.17 ms à 1.81 s.

Pour la liaison montante et les premières transmissions d'ACK (RX1), la cellule fonctionne dans la bande 868 MHz, trois sous-bandes sont considérées : 868.1, 868.3 et 868.5 MHz, chacune avec une bande passante de 125 kHz, tous les nœuds appartiennent à la classe A. Pour le deuxième ACK (RX2), un canal distinct fonctionnant dans la bande de 869.525 MHz est utilisé et un SF 12 est considéré. La longueur du paquet ACK (l_a) est fixée à 25 bytes, et le temps de transmission ACK T_{ack} varie de 46.33 ms à 1.15 s en fonction du SF.

Contrairement au scénario précédent, dans cette simulation nous considérons une passerelle avec des limitations matérielles. Par conséquent, nous considérons le Semtech SX1301 qui ne peut pas décoder 18 paquets en parallèle mais 8.

Pour le LoRaWAN sans brouillage, nous avons simulé une cellule avec N_u variant de 1 à 2000 et r variant de 0 à 64. Ensuite, nous avons simulé une cellule LoRaWAN sous l'attaque de brouilleurs émettant soit sur la bande 868 MHz (trois canaux) ou sur la bande de 869.525 MHz. Le nombre d'utilisateurs est fixé à 100. Le réseau est attaqué par 10 brouilleurs par canal dont la charge de trafic cumulée varie de 0 à 2. La longueur du paquet du brouilleur l_{ju}/l_{ja} est définie sur 50 bytes avec un temps de transmission T_j variant de 82.17 ms à 1.81 s (en fonction du SF).

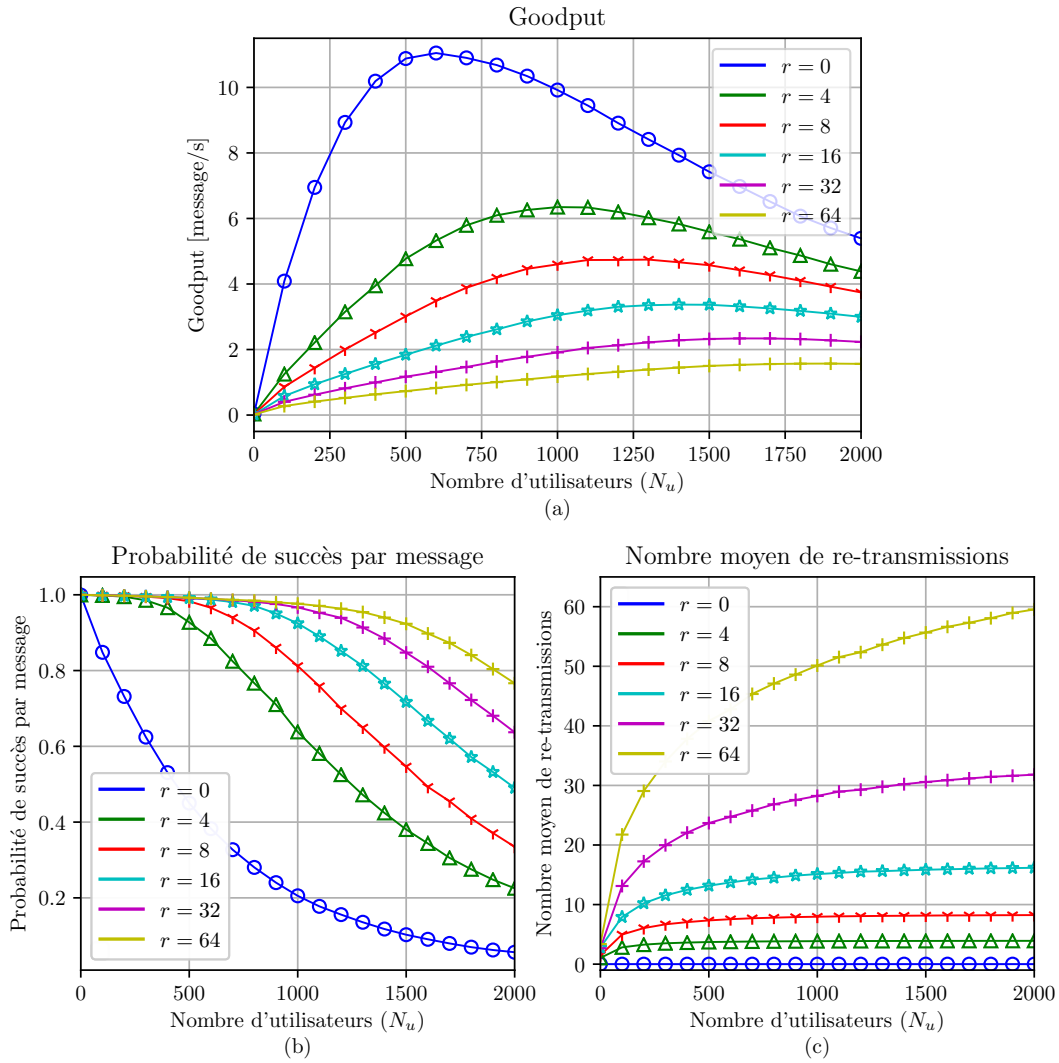


Figure 6: Performance de LoRaWAN avec re-transmission considérant des limites matérielles et des canaux réalistes: (a) *Goodput*, (b) Probabilité de succès par message et (c) Nombre moyen des re-transmissions.

Réseau LoRaWAN sans brouillage

La figure. 6 présente l'évaluation des performances d'une cellule LoRaWAN sans brouilleurs en tenant compte des limites matérielles et des canaux réalistes. Différentes configurations de réseau variant entre N_u et r ont été considérées.

La figure 6 (a), présente le goodput. Pour $r = 0$, tous les messages sont envoyés une seule fois, ce qui signifie que le nombre de paquets et de messages est le même. Par conséquent, un comportement classique d'un réseau de type Aloha peut être observé, atteignant le maximum *goodput* à 11 `[messages/s]` pour $N_u = 600$.

Au contraire, pour $r > 0$ on note que, comme r augmente le goodput obtenu diminue. Cette diminution est due à deux raisons : (i) Les ACKs envoyés dans la même bande et les SF identiques aux paquets de données des utilisateurs provoquent des collisions avec les paquets

des utilisateurs, et (ii) les ACKs envoyés dans la bande de 869.525 MHz sont envoyés avec $SF = 12$. Par conséquent, le GW est verrouillée pendant un temps considérable pour envoyer un seul paquet ACK (1.15 s). Cela signifie que à mesure que le nombre d'utilisateurs augmente, sa réactivité diminue à mesure qu'elle devient saturée. Ensuite, les utilisateurs commencent à faire des re-transmissions inutiles même lorsque le paquet d'origine est bien arrivé.

À la figure 6 (b), nous pouvons voir que la Probabilité de Succès du Message se dégrade rapidement en augmentant le nombre d'utilisateurs pour $r = 0$. Par exemple, il tombe à 0.37 quand $N_u = 600$. Cette réduction devient encore plus importante pour les réseaux avec $N_u \geq 1500$ où ce nombre est réduit à seulement 0.1 ou moins.

D'autre part, pour $r > 0$, on peut voir que plus r est élevé, plus la probabilité de succès sur les messages sera élevée, augmentant ainsi la fiabilité du réseau. En effet, la sélection de r devrait tenir compte du nombre d'utilisateurs qu'un opérateur de réseau est disposé à servir. Par exemple, il n'y a aucun intérêt à sélectionner $r = 8$ pour $N_u \geq 1400$ puisque la probabilité de succès du message sera réduite à 0.6 ou moins.

La figure. 6 (c) présente le nombre moyen de re-transmissions par message en fonction de r et N_u. On peut voir que ce nombre augmente proportionnellement au nombre d'utilisateurs desservis dans le réseau. Il y a donc un compromis entre le goodput et le nombre moyen de re-transmissions. En conséquence, si un opérateur est disposé à fournir un goodput adéquat, il doit décider soigneusement le nombre de re-transmission que l'application doit soutenir.

Enfin, nous pouvons conclure que les limitations matérielles ont un impact significatif sur la performance finale atteinte. En effet, le maximum *goodput* obtenu dans cette cellule LoRaWAN est 11 `[messages/s]` pour seulement $N_u = 600$, alors que dans le réseau LoRaWAN simulé précédemment (cas sans considérer les limitation du matérielle) le *goodput* maximum atteint était 13.46 `[messages/s]` à $N_u = 2000$. Cette différence est principalement due à la capacité limitée de la passerelle, puisqu'elle passe de 18 paquets en parallèle à seulement 8.

De plus, nous avons également constaté que la façon dont les fenêtres de réception sont gérées a une incidence importante sur la performance du réseau en cas de retransmission. En effet, une cellule LoRaWAN où tous les ACK sont envoyés sur un canal séparé avec le plus grand SF possible réduit considérablement son *goodput*. Par exemple, pour $r = 4$ dans la cellule LoRaWAN simulée précédemment, le *goodput* maximum était 10.85 `[messages/s]` pour $N_u = 1500$, alors que le *goodput* obtenu dans la cellule LoRaWAN avec des limitations matérielles est seulement 6.25 `[messages/s]` pour $N_u = 1000$.

Réseau LoRaWAN sous brouillage

Dans ce scénario, nous considérons un réseau avec 100 nœuds. Dans ce cas, le réseau n'est pas surchargé. Comme dans le cas de l'absence de brouillage, nous avons également évalué trois métriques de performance : (i) *goodput*, la probabilité de succès par message et (iii) le nombre moyen de re-transmissions. Ensuite, nous modifions le nombre de re-transmissions maximum autorisé et la charge de trafic du brouilleur agrégé ($\overline{G_{ju}}/\overline{G_{ja}}$) entre 0 et 2.

La figure 7 présente les résultats obtenus. Il est évident que, quand les brouilleurs transmettent sur la bande 868 MHz (.1/.3/.5), le plus grand $\overline{G_{ju}}$ est, plus le *goodput* est bas, et plus

le nombre moyen de re-transmissions est élevé. En effet, pour tous les cas, le *goodput* tombe à près de zéro lorsque $overline G_{ju} = 1.0$. Cependant, le réseau peut atténuer le problème des brouilleurs lorsqu'ils brouillent modérément en autorisant les rediffusions. Par exemple, pour un $\overline{G_{ju}}$ de 0.2, la probabilité de succès passe de 0.44 pour $r = 0$ à 0.91 pour $r = 4$. Ce chiffre peut être amélioré pour $r = 32$, où une probabilité de succès proche de 1 est atteinte.

Au contraire, pour les brouilleurs qui émettent sur la bande de 868.525 MHz, l'impact sur le *goodput* réseau est beaucoup moins important. En effet, pour $r = 4$ les *goodput* passent de 1.19 à 1.01 `[messages/s]` dans le pire des cas. Quant à la probabilité de succès par message, nous pouvons voir qu'elle reste constante quelle que soit la valeur de $\overline{G_{ja}}$. En outre, le nombre moyen de re-transmissions augmente bien que les performances du réseau ne s'améliorent pas.

En résumé, une cellule LoRaWAN avec $N_u = 100$ et des re-transmissions est raisonnablement résistante au brouillage sur les canaux de liaison montante si la charge de trafic cumulée des brouilleurs est inférieure à 0,2 avec $r = 4$. En ce qui concerne les brouilleurs en liaison descendante, nous avons vu que le *goodput* obtenu n'est pas affecté quelle que soit la charge de trafic des brouilleurs, cependant si les messages sont systématiquement acquittés l'impact sur le nombre de re-transmissions (et donc la consommation d'énergie) n'est pas négligeable.

Consommation de courant électrique et durée de vie

Jusqu'à présent, nous avons concentré notre attention sur l'évaluation de la performance au niveau du réseau. Dans cette section, nous allons évaluer un autre aspect, qui est d'une importance particulière dans les réseaux LoRaWAN, la consommation de courant électrique des utilisateurs. Pour cela, nous utiliserons le module ns3, nous donnerons également une estimation de la durée de vie de la batterie d'un dispositif.

Notre modèle de consommation de courant électrique est basé sur la machine d'état pour la couche PHY présentée dans la figure 8. Par conséquent, le courant total consommé par un utilisateur donné dépend du temps qu'il consacre à chaque état. Pour notre simulation, nous considérons les valeurs de consommation présentées dans le tableau 3. Nous considérons le module SX1272 [14] avec une puissance de transmission de 14 dBm et un schéma de modulation avec taux de codage de 4/5.

Table 3: Consommation de courant électrique en fonction des différents états

État	Consommation de courant [mA / s]
Tx	83
Rx	32
Standby (STB)	32
Sleep	45×10^{-3}

En fonction de ces niveaux de consommation, nous calculons la consommation moyenne par

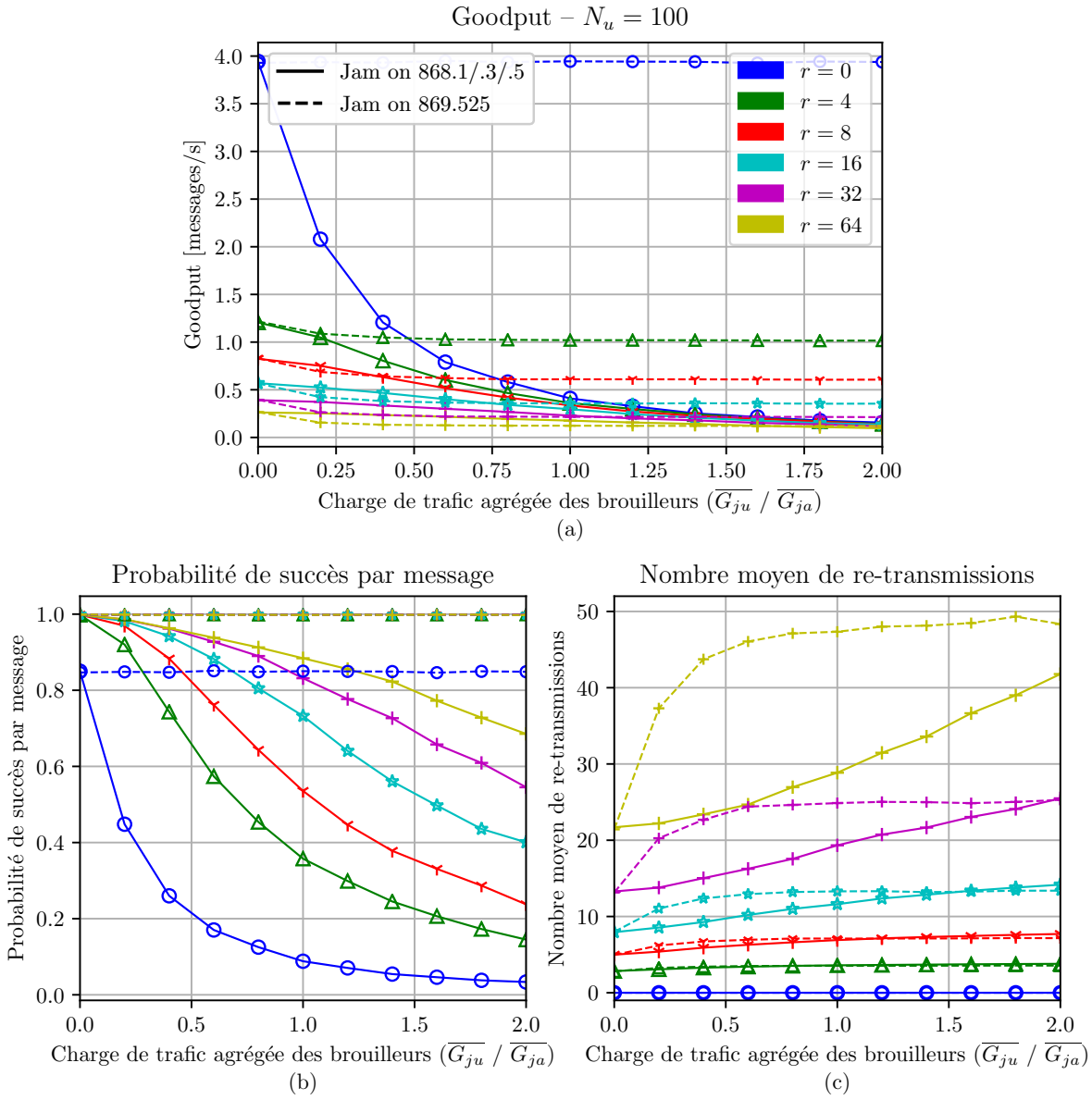


Figure 7: Performance d'un réseau LoRaWAN avec brouilleurs et $N_u = 100$: (a) *Goodput*, (b) Probabilité de succès par message and (c) Nombre moyen de re-transmissions.

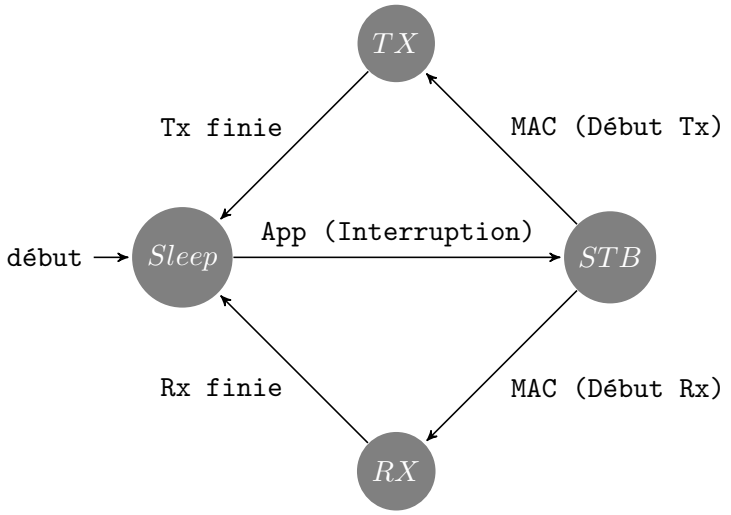


Figure 8: Machine d'état pour la couche PHY des utilisateurs

paquet envoyé (I_{avg}) au cours d'une session donnée :

$$I_{avg} = \frac{1}{T_{app}} \sum_{i=1}^{N_{states}} T_i \cdot I_i \tag{2}$$

où T_{app} est le temps d'application en secondes, défini comme le temps entre deux transmissions de paquets consécutives, N_{states} est le nombre d'états par lesquels un utilisateur passe pour transmettre un paquet, T_i est la durée passée dans l'état i et I_i est la consommation de courant électrique [mA/s] dans l'état i.

Par conséquent, si nous considérons seulement l'état de transmission, nous pouvons facilement voir que la consommation actuelle d'un paquet donné dépend de son ToA, qui dépend à son tour de la longueur et du SF du paquet. Une comparaison de cette consommation de courant électrique est présentée dans la figure 9.

Cependant, pour envoyer un seul message, il arrive que les utilisateurs doivent le renvoyer si le réseau est congestionné. Ainsi, dans ce qui suit, nous allons considérer le LoRaWAN avec un scénario de canal réaliste afin d'estimer la consommation par message. Comme dans la section précédente, nous présenterons d'abord un déploiement sans brouilleurs, puis nous le comparerons avec un réseau attaqué. Ensuite, ces résultats seront extrapolés pour estimer la durée de vie de la batterie.

En plus des mesures de performance présentées précédemment, nous extrayons également la consommation moyenne de courant par message. Il comprend tous les états (STB, TX, RX, *Sleep*) par lesquels un utilisateur a dû passer pour envoyer chaque paquet, ainsi que le nombre de fois qu'il a dû envoyer ce message. La figure 10 montre les résultats obtenus pour les deux scénarios. Il représente la consommation de courant électrique moyenne d'un utilisateur donné en fonction du nombre maximal de re-transmissions (r) autorisées dans le réseau.

De la figure 10 (a) nous pouvons voir que, comme prévu, la consommation de courant est inversement proportionnelle à la qualité du canal du réseau, et que l'ajout de re-transmissions

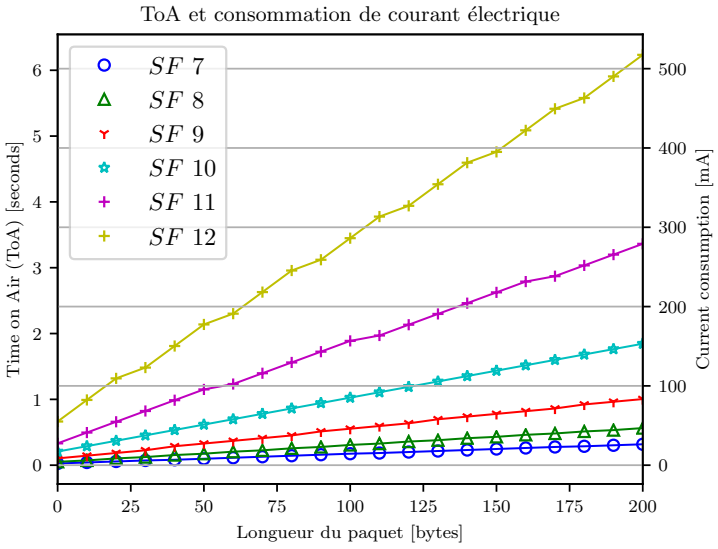


Figure 9: Temps de transmission (ToA) et consommation de courant électrique en fonction de la longueur du paquet

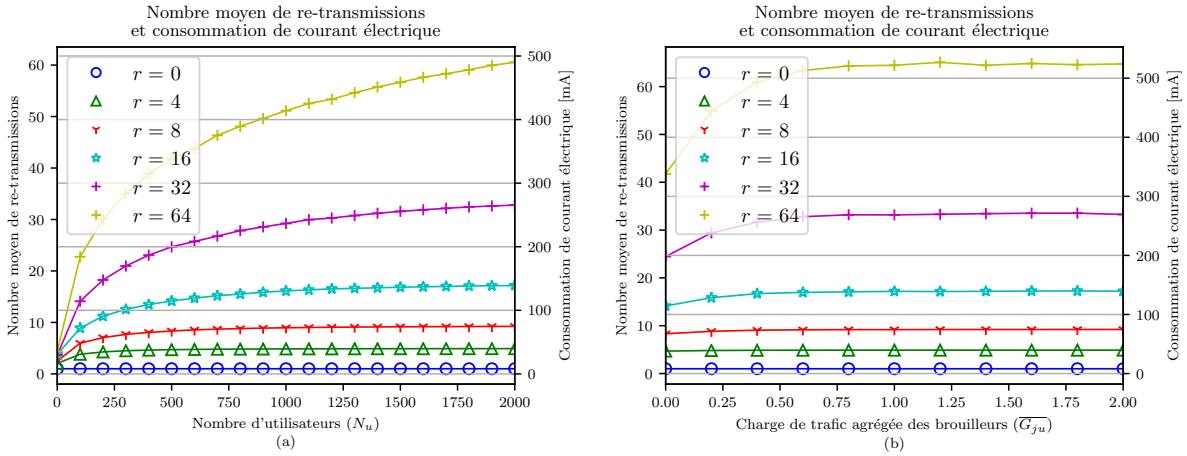


Figure 10: Consommation de courent électrique moyenne et nombre moyen de re-transmissions : (a) pas de jamming, (b) jamming dans la voie montante.

de paquets l'incrémente drastiquement. En fait, pour le pire des scénarios ($r = 64, \overline{G_{ju}} = 2$), la consommation de cour par message est 518 [mA], lorsque chaque message est retransmis 60 fois en moyenne.

De même, lorsque le réseau est attaqué par des brouilleurs comme représenté dans la figure 5.17 (b), nous remarquons que la consommation de courant électrique augmente à mesure que le réseau est saturé et qu'elle est également directement proportionnelle au nombre maximum de re-transmissions autorisées dans le réseau. En fait, pour le pire des scénarios ($r = 64, G_j = 1$), la consommation de courant électrique par message est de 486 [mA], lorsque chaque message est retransmis 60 fois en moyenne.

En ce qui concerne l'estimation de la durée de vie de la batterie d'un utilisateur, elle est obtenue à partir de l'équation (3) pour $C_{battery} = 2400$ mAh et I_{avg} extrapolés à partir des

scénarios de simulation précédents. Pour estimer la durée de vie d'un utilisateur fonctionnant sur batterie, nous avons considéré une fonction linéaire de la capacité de la batterie ($C_{battery}$) [mAh] :

$$T_{lifetime} = \frac{C_{battery}}{I_{Avg}} \tag{3}$$

Nous considérons différentes configurations réseau : plusieurs passerelles ($N_{gw} = 1, 2$), deux SF ($S = 7, 12$), différents paramètres de retransmission ($r = 0, 4, 8$, $pr = 100\%$), profils d'application ($T_{app} = 30$ s, 10 min, 1 h, $l_u = 50$ `bytes`). Quant à la charge de trafic des brouilleurs, nous la définissons comme $G_j = 0, 0.2$.

Comme présenté dans le tableau de la thèse 5.5, nous considérons d'abord un réseau LoRaWAN avec et sans re-transmission de paquets avec une seule passerelle qui n'est pas attaquée par les brouilleurs. Pour ce cas, une durée de vie de la batterie est estimée à 14.97 mois et peut être obtenue en envoyant des messages très fréquemment ($T_{app} = 10$ min) avec un SF 7, et jusqu'à 76.94 mois pour les communications peu fréquentes ($T_{app} = 1$ jour). Avoir un SF plus élevé réduit considérablement la durée de vie de la batterie. En effet, ces chiffres peuvent être réduits à 0.84 et 72.57 mois respectivement.

Table 4: Durée de vie de la batterie

Paramètres de réseau	Durée de vie de la batterie [mois]					
$\{r, \overline{G_j}, N_{gw}\}$	$T_{app} = 10$ min		$T_{app} = 1$ h		$T_{app} = 1$ day	
	SF 7	SF 12	SF 7	SF 12	SF 7	SF 12
$\{0, 0, 1\}$	14,97	0,84	72,17	30,65	76,94	72,57
$\{0, 0.2, 1\}$	12,90	0,70	71,24	27,36	76,89	71,72
$\{0, 0, 1\}$	22,10	1,39	74,08	40,39	77,03	74,34
$\{0, 0.2, 2\}$	17,85	1,05	73,11	34,85	76,98	73,45
$\{8, 0, 1\}$	2,25	0,11	49,66	5,87	75,42	51,24
$\{8, 0.2, 1\}$	2,15	0,10	48,79	5,61	75,34	50,40
$\{8, 0, 1\}$	2,76	0,13	53,22	7,10	75,74	54,69
$\{8, 0.2, 2\}$	2,50	0,12	51,50	6,47	75,59	53,03
$\{16, 0, 1\}$	1,14	0,05	36,61	3,05	73,76	38,36
$\{16, 0.2, 1\}$	0,92	0,04	32,36	2,46	72,95	34,08
$\{16, 0, 1\}$	1,02	0,05	34,36	2,73	73,35	36,09
$\{16, 0.2, 2\}$	1,30	0,06	39,19	3,47	74,17	40,93

En ce qui concerne les scénarios où le réseau est en cours de brouillage, il peut être clairement vu que l'autonomie de la batterie est réduite car les utilisateurs sont obligés de faire plus de retransmission en moyenne. Par exemple, la durée de vie d'un réseau avec $r = 16$ et $T_{app} = 1$ h, est réduite de 36.61 à 32.36 mois lorsque la charge de trafic des brouilleurs (G_j) est réglée à 0.2 pour un utilisateur utilisant un SF de 7. Cette réduction devient encore plus importante lorsque le DE utilise un SF 12, dans ce cas la durée de vie de la batterie est seulement de 3.05mois dans le cas sans brouillage et de 2.46 lorsqu le réseau est brouillé.

En résumé, nous avons montré qu'une batterie de 2400 mAh peut atteindre une durée de vie de la batterie jusqu'à 73.45 mois si les utilisateurs font des transmissions très peu fréquentes et si un SF 7 est utilisé. Au contraire, si un réseau avec un système de re-transmission ($r = 16$)

est modérément brouillé, et qu'un SF 12 est utilisé à la place, une durée de vie de la batterie de quelques jours est atteinte.

Contre-mesures de brouillage

Une analyse de plusieurs contre-mesures qui pourraient être appliquées au réseau LoRaWAN est présentée. Ces contre-mesures sont classés en trois catégories : (i) détection, qui vise à fournir des alertes précoces afin de prendre d'autres mesures, (ii) contre-mesures pro-actives qui sont mises en œuvre avant que l'attaque ne se produise réellement et (iii) contre-mesures réactives qui améliorent la résilience du réseau en prenant des mesures spécifiques pendant l'attaque. Nous fournissons également un état de l'art sur les principaux mécanismes de détection proposés pour LoRAWAN et d'autres réseaux IdO en cours de brouillage. Nous avons également constaté qu'il existe de nombreuses métriques comme le taux de paquets mal décodes, la consommation énergétique ou le taux d'occupation du canal qui peuvent être utilisées pour atteindre cet objectif.

Dans un deuxième temps, nous examinons plusieurs contre-mesures pro-actives possibles, comme les préambules authentifiés et le saut de fréquence *(frequency hopping)*. Ensuite, nous nous concentrons sur les contre-mesures réactives et nous présentons brièvement trois mécanismes possibles qui peuvent être mis en œuvre : l'ADR sensible au brouillage, la cartographie de la zone de brouillage et les nœuds de détection externes.

Ensuite, nous proposons et évaluons deux approches de détection de brouillage appliquées à des ensembles de données issues des scénarios de simulation présentés au chapitre 5.

Détection de brouillage en utilisant un approche statistique

Premièrement, on propose d'utiliser l'EWMA (*Exponentially Weighted Moving Average chart*), qui est une approche statistique classiquement utilisée dans les processus de contrôle industriel. Il utilise la moyenne glissante et une limite de contrôle pour détecter les anomalies. La statistique EWMA $z(t)$ est alors calculée comme suit :

$$z(t) = \lambda \cdot x(t) + (1 - \lambda) \cdot z(t-1) \quad t = 1, 2, 3 \ldots n \tag{4}$$

où $x(t)$ est l'observation au moment t, n représente le nombre d'observations à surveiller incluant $z(0)$ et λ est la constante $(0 \lambda leq 1)$ utilisée pour déterminer l'importance des observations actuelles et historiques. Par exemple, lorsque *lambda* a une valeur de 1, cela signifie que seules les mesures récentes influencent l'EWMA.

La détection se fait en comparant la statistique $z(t)$ avec deux valeurs : UCL (*Upper Control Level*) et LCL (*Lower Control Level*), qui peuvent être déterminées à l'aide de l'équation ci - dessous :

$$UCL_z = z_0 + f \cdot \sigma_z \qquad LCL_z = z_0 - f \cdot \sigma_z \tag{5}$$

où z_0 est la valeur cible et est définie en fonction de la connaissance préalable du système, f

est une valeur de réglage généralement définie aux limites de contrôle 3-sigma et σ_z est l'écart type de $z(t)$. La variance estimée de la statistique de l'EWMA peut être estimée :

$$\sigma_z^2 = \sigma_x^2 \cdot \left(\frac{\lambda}{2 - \lambda} \right) \tag{6}$$

où σ_x est l'écart type de la série d'observations $x(t)$ obtenu à partir des données historiques.

Nous proposons que l'algorithme de détection soit implémenté dans le serveur réseau. Comme mesure de performance, nous utilisons l'IAT (*Inter Arrival Time*), et le RSSI (*Received Signal Strength Indicator*), qui sont tirés des scénarios de simulation précédents et du banc de test LOFRASEA (*LOrawan FRamework for SEcurity Audit*). La détection est divisée en deux phases : (i) la phase d'initialisation, où les valeurs de seuil UCL et LCL sont calculées et (ii) la phase de détection, où l'EWMA est utilisée pour détecter les anomalies en les comparant avec les valeurs de seuil.

Détection de brouillage en utilisant un approche d'apprentissage automatique

Nous proposons également d'utiliser une famille de réseaux de neurones appelés RNN (*Recurent Neural Networks*) qui emploient une architecture récursive utilisée pour les séries temporelles. Ce mécanisme de détection est capable d'apprendre le comportement normal du réseau, qui peut ensuite être utilisé pour détecter les anomalies.

Pour ce propos, nous adaptons la méthodologie précédemment proposée dans [15]. Comme le montre la figure 11, le mécanisme de détection comporte deux phases : (i) une phase d'apprentissage où le RNN apprend le comportement normal du réseau. Dans notre cas, cela correspond à des traces précédemment enregistrées d'un réseau qui n'a pas été attaqué, et (ii) à la phase de détection, où le RNN compare le comportement normal prédit avec une trace qui contient des anomalies. Ensuite, la sortie est l'erreur entre ces deux séries. Pour la détection, l'erreur est comparée à un seuil τ.

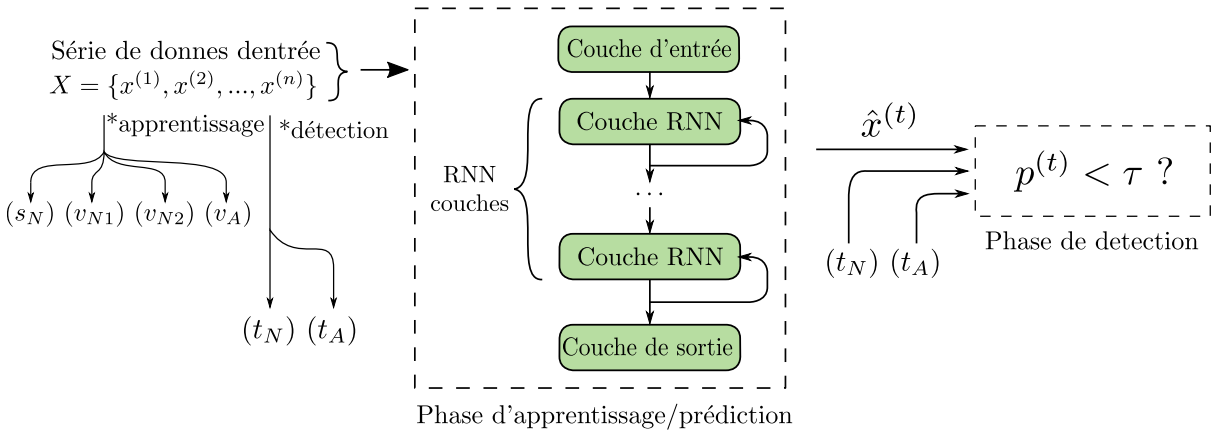


Figure 11: Model de détection avec réseau RNN

En tant que modèle de prédiction, nous utilisons trois réseaux RNN : Vanilla, LSTM et GRU (une description détaillé de ces trois est présentée dans le chapitre 6). Ensuite, nous calculons

la distribution des erreurs de prédiction pour détecter les anomalies.

Résultats

Afin d'évaluer les deux algorithmes, nous utilisons la méthodologie *precision and recall*. Les résultats sont comparés en utilisant les termes : vrais positifs (tp), vrais négatifs (tn), faux positifs (fp), et faux négatifs (fn). Les termes positifs et négatifs font références à la prédiction du classificateur (parfois appelée attente), et les termes vrais et faux à la question de savoir si cette prédiction correspond à l'observation. Par conséquent, nous calculons trois mesures : Précision, Recall et $F_1 - \text{score}$, définies comme suit :

Precision: est la fraction des instances pertinentes parmi les instances récupérées

$$\text{Precision} = \frac{tp}{tp + fp} \tag{7}$$

Recall: également connu sous le nom de sensibilité, est la fraction du nombre total d'instances pertinentes qui ont été effectivement récupérées.

$$\text{Recall} = \frac{tp}{tp + fn} \tag{8}$$

$F_1 - \textbf{score:}$ est la moyenne de la précision et du Recall. La valeur la plus élevée possible d'un score F_1 est 1, ce qui indique une précision et un Recall parfaits, et la valeur la plus basse possible est 0, si la précision ou le rappel sont zéro.

$$F_1 - \text{score} = \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})} \tag{9}$$

Trois ensembles de données ont été utilisés pour évaluer nos algorithmes de détection, ils ont été tirés des différentes campagnes de simulation décrites précédemment et du banc d'essai décrit dans la Section 6.4.4.2. Nous choisissons comme paramètres d'évaluation les paramètres Inter Arrival Time (IAT) et Received Signal Strength Indicator (RSSI) :

i) Réseau chargé à partir du scénario réaliste décrit dans la section 5.3 : il s'agit d'une cellule LoRaWAN avec $N_u = 1000$, et 5 brouilleurs sur le canal de liaison montante avec $d_{ju} = 0.5$. Le temps de simulation est fixé à 1500 s et l'attaque dure pour 200 s.

ii) Réseau moins chargé du scénario réaliste de simulation de canal de la section 5.3 : il s'agit d'une cellule LoRaWAN avec $N_u = 100$, et 5 brouilleurs sur le canal de liaison montante avec $d_{ju} = 0.5$. Le temps de simulation est fixé à 1500 s et l'attaque dure pour 200 s.

iii) Traces de réseau réelles : C'est une cellule LoRaWAN avec 5 utilisateurs et un brouilleur. Ces traces sont issues du banc de test matériel. Les utilisateurs sont configurés pour utiliser un SF unique, qui est défini à 12 et le profil de l'application est considéré comme un $T_{app} = 1$ m. Le brouilleur est également configuré pour utiliser un SF de 12, et envoie

un paquet chaque 30 s. La campagne de mesure est considérée sur une durée de 3h et le brouilleur est allumé pendant 1h.

Pour chacun des ensembles de données décrits ci-dessus, nous dérivons plusieurs réalisations en considérant les cas avec et sans brouillage afin de former les six séries temporelles nécessaires pour évaluer les algorithmes :

- Comportement normal (pas de brouillage) : *normal train* (s_N), *normal validation-1* (v_{N1}), *normal validation-1* (v_{N2}), et *normal test* (t_N).
- Comportement d'anomalie (avec brouillage) : *anomalous validation* (v_A), and *anomalous test* (t_A).

La table 5 rapporte les résultats des mécanismes de détection EWMA et RNN sur les trois jeux de données décrits précédemment : précision, *recall* et score F1. Les résultats sont rapportés après avoir choisi la valeur de seuil (UCL et τ) avec le F_1-score maximum.

Table 5: Comparaison des deux approches de détection : EWMA et RNN

	Algorithme	Simulation ($d_{ju} = 1$) - IAT	Simulation ($d_{ju} = 1$) - RSSI	Simulation ($d_{ju} = 0.5$) - IAT	Simulation ($d_{ju} = 0.5$) - RSSI	Test-bed IAT	Test-bed RSSI
Precision	EWMA	0.90	0.85	0.82	0.69	0.95	0.93
	LSTM	0.97	0.88	0.94	0.88	0.98	0.96
	GRU	0.88	0.75	0.93	0.85	0.96	0.97
	Vanilla	0.87	0.86	0.82	0.69	0.91	0.93
Recall	EWMA	0.89	0.81	0.80	0.58	0.96	0.94
	LSTM	0.98	0.91	0.93	0.89	0.96	0,98
	GRU	0.87	0.64	0.88	0.77	0.98	0.97
	Vanilla	0.85	0.94	0.78	0.74	0.95	0.97
F_1-score	EWMA	0.90	0.83	0.81	0.63	0.97	0.94
	LSTM	0.98	0.90	0.94	0.88	0,97	0.97
	GRU	0.88	0.69	0.90	0.81	0.96	0.97
	Vanilla	0.86	0.90	0.80	0.71	0.93	0.95

Pour tous les cas, l'algorithme LSTM surpasse les autres algorithmes de détection. En effet, lorsqu'il est appliqué à l'ensemble de données du réseau chargé ($d_{ju} = 1$), une précision de 0.98 et un recall de 0.98 sont obtenus lorsque la mesure IAT est considérée. Les deux autres modèles RNN considérés obtiennent également de bons résultats, mais l'approche GRU montre une meilleure performance que Vanilla dans tous les cas.

En ce qui concerne la métrique utilisée, on peut voir que, dans tous les cas, il est préférable d'utiliser l'IAT. Ceci nous permet d'inférer qu'il existe une dépendance temporelle beaucoup plus faible en RSSI, analyse observée à la fois pour l'EWMA et pour les RNNs considérés. En outre, nous considérerons également que les algorithmes ont été appliqués à l'ensemble des ensembles de données. Par conséquent, nous n'avons pas fait de filtre en tenant compte de chacun des utilisateurs. Dans une mise en œuvre réelle, cela pourrait être une très bonne option qui pourrait donner de meilleurs résultats même en utilisant le RSSI.

Conclusions et perspectives

Les réseaux LoRaWAN gagnent en popularité dans l'environnement IoT parce qu'ils offrent une large couverture, un faible coût de mise en œuvre et une spécification ouverte. Un tel succès s'accompagne cependant de nombreux défis à relever. Le brouillage est présenté comme un obstacle majeur qui devrait être considéré pour les futures versions de la spécification. Dans cette thèse, ce problème a été abordé en suivant une approche holistique. Nos contributions sont brièvement décrites comme suit :

i) nous proposons d'abord un modèle analytique qui permet d'évaluer la performance d'un réseau LoRaWAN soumis à des attaques de brouillage. Pour ce modèle, nous prenons plusieurs hypothèses simplificatrices qui, bien qu'elles rendent le résultat obtenu inférieur aux performances d'un réseau réel, permettent une bonne approximation pour des charges de trafic plus faibles.

ii) nous améliorons un simulateur événementiel précédemment développé sous ns3 pour LoRaWAN. Nous ajoutons à ce module plusieurs fonctions qui permettent de simuler et d'estimer la performance réseau d'un réseau LoRaWAN sous l'attaque de brouilleurs.

iii) comme on peut le voir à l'annexe A, lors du développement du modèle de simulation en ns3, nous avons pu constater que les deux modèles d'interférence existants fournissent une estimation partielle des collisions. La raison de cela est que le modèle le plus utilisé dans la littérature [1] est conçu pour un cas idéal dans lequel il existe une pseudo-orthogonalité des facteurs d'étalement.Un modèle plus récent [16] basé sur des mesures physiques indique que cette pseudo-orthogonalité n'existe pas dans les déploiements réels. Par ailleurs, le deuxième modèle n'est pas complet car il ne prend pas en compte toutes les combinaisons possibles des différents facteurs d'étalement. Cela ouvre différentes possibilités de recherche qui permettront d'améliorer ces modèles d'interférence à l'avenir.

iv) nous utilisons ces deux outils de modélisation de réseau pour effectuer une évaluation complète des performances de LoRaWAN sous brouillage. Nous examinons six scénarios différents : (i) un réseau LoRaWAN avec un canal idéal, (ii) un réseau LoRaWAN avec des limites matérielles et des canaux réels, (iii) un réseau LoRaWAN avec une diversité de passerelles, (iv) une stratégie de brouillage, (v) l'impact géographique des brouilleurs et (vi) la consommation de courant électrique et l'autonomie de la batterie d'un nœud.

v) nous avons proposé et évalué deux approches de détection de brouillage appliquées à différents ensembles de données à partir des scénarios de simulation précédents. Celles-ci utilisent la moyenne glissante et une limite de contrôle pour détecter les anomalies. Ensuite, nous avons utilisé une famille de réseaux neuronaux appelés RNN qui emploient une architecture récursive, largement utilisée pour les séries chronologiques. En outre, nous avons montré que l'EWMA donnait des résultats similaires par rapport au RNN. Par conséquent, nous avons démontré que le mécanisme de détection de ML proposé est capable d'apprendre le comportement normal du réseau, qui peut ensuite être utilisé pour

détecter les anomalies. Notre approche RNN donne des résultats prometteurs pour les ensembles de données d' un réseau réel.

vi) Enfin, nous avons présenté une description du banc d'essai LOFRASEA, qui est un environnement logiciel et matériel libre qui permet de mettre en œuvre et d'évaluer un réseau LoRaWAN sous brouillage.

Ces travaux peuvent être prolongés en tenant compte des problèmes de recherche suivants:

i) Ajouter l'effet du *capture effect* et des deux fenêtres de réception dans le modèle mathématique présenté dans la thèse.

ii) Étudier le *capture effect* et son impact possible sur les performances du réseau en cas de brouillage. Ceci peut être fait en considérant une distribution de brouilleurs différente d'une distribution uniforme comme ce que nous considérons dans cette thèse.

iii) Étudier l'effet que l'utilisation de l'algorithme ADR peut avoir sur les attaques de brouillage. En fait, comme d'autres auteurs le laissent entendre, il n'offre pas les meilleurs résultats parce qu'il ne tient pas compte de l'équilibre correct dans la répartition des utilisateurs parmi les SFs. Par conséquent, la performance du réseau pourrait être encore pire dans un réseau avec brouilleurs, ce qui peut impacter la consommation d'énergie et la durée de vie d'être affecté.

iv) Envisager un déploiement réel plus large pour l'évaluation des algorithmes de détection de brouillage proposés ici, ainsi que l'extension des mesures de performance utilisées pour détecter le brouillage

En plus de ces problèmes de recherche ouverts, nous avons également identifié plusieurs fonctionnalités de sécurité qui peuvent être ajoutées à la spécification LoRaWAN:

i) Ajout d'une version étendue de l'ADR qui considère le brouillage.

ii) L'impact de l'ajout de nœuds supplémentaires en charge de la détection de la qualité du canal afin d'améliorer la résilience du réseau.

iii) Utiliser une version allégée du CSMA afin d'accroître la résilience et la fiabilité du réseau.

iv) Accroître la sécurité du réseau et des applications des utilisateurs de Classe B, particulièrement en ce qui concerne la diffusion de messages du type *Beacon*.

Enfin, nous croyons fermement que, si le cas d'utilisation le justifie, les développeurs devraient envisager d'inclure des règles de sécurité réseau supplémentaires telles que :

i) Ajout d'un mécanisme de détection de brouillage au niveau réseau comme ceux proposés dans cette thèse

ii) Ajout de nœuds supplémentaires spécifiquement déployés pour détecter la qualité du canal, ceux-ci pouvant être utilisés pour déclencher des alertes précoces et prendre d'autres mesures.

Contents

Abbreviations

3GPP 3rd Generation Partnership Project

ABP Activation by Personalization

ADR Adaptive Data Rate

AES Advanced Encryption Standard

AIDS Anomaly-based intrusion detection system

BPR Bad Packet Ratio

CAD Channel Activity Detection

CBR Channel Busy Ratio

CMAC block-Cipher-based MAC

CSMA Carrier sense multiple access

CSS Chirp Spread Spectrum

CST Carrier Sense Time

CUSUM Cumulative Sum Control Chart

DDoS Distributed Denial of Service

EC-GSM-IoT Extended Coverage - GSM - Internet of Things

ECA Energy Consumption Amount

ECC Electronic Communications Committee

ED End Device

eDRX Extended idle-mode DRX cycle

eMTC enhanced Machine-type Communication

ERP Effective Radiated Power

ETSI European Telecommunications Standards Institute

EUI Extended Unique Identifier

EWMA Exponentially Weighted Moving Average

fNS forwarding Network Server

GPL General Public License

GPS Global Positioning System

GRU Gated Recurrent Unit

GSM Global System for Mobile Communications

H2M Human-to-Machine

IAT Inter Arrival Time

IIoT Industrial Internet of Things

IoT Internet of Things

IoT-A Internet of Things Architecture

IT Inactivity Time

LBT Listen-Before-Talk

LCL Lower Control Limit

LOFRASEA The LOrawan FRamework for SEcurity Audit

LoRa Long Range

LPWAN Low Power Wide Area Network

LSTM Long short-term memory

LTE Long Term Evolution

M2M Machine-to-Machine

MAC Medium Access Control

MANET Mobile ad hoc networks

MCS Modulation and Coding Scheme

MIC Message Integrity Code

MITM Man in the Middle

ML Machine Learning

NB-IoT Narrowband IoT

NFC Near Field Communication

ns3 Network simulator 3

OSI Open System Interconnection

OTAA Over the Air Activation

p-CSMA persistent CSMA

PAPR Low Peak to Average Power Ratio

PDR Packet Data Rate

PL Packet Losses

PSM Power Saving Mode

QoS Quality of Service

RFID Radio-frequency identification

RNN Recurrent Neural Networks

RSS Received Strength Signal

RSSI Received Signal Strength Indicator

SBC Single Board Computer

SDR Software Define Radio

SF Spreading Factor

SFD Start Data Frame

SINR Signal-to-interference-plus-noise ratio

SNR Signal-to-noise ratio

sNS serving Network Server

SPC Statistical Process Control

SVM Support Vector Machine

TCP/IP Transmission Control Protocol/Internet Protocol

ToA Time-on-Air

TP Transmission Power

UCL Upper Control Limit

UE User Equipment

VANET Vehicular Ad-Hoc Network

WHAN Wireless Home Area Network

WLAN Wireless Local Area Network

WPAN Wireless Personal Area Network

WSN Wireless Sensor Networks

WWAN Wireless Wide Area Networks

List of Figures

List of Tables

Notation Summary

Notation	Definition	Notation	Definition
N_u, N_{ju}, N_{ja}	Number of: users, jammers on uplink and, downlink	$\overline{\mathcal{G}_u}$	User throughput
F	Number of frequency channels for uplink	$\mathcal{O}_u$	User goodput
$s \in \mathcal{S}$	Set of orthogonal Spreading Factors	$\mathcal{P}_m$	Message success probability
r	Maximum number of allowed re-transmissions	$\mathcal{E}(r)$	Expected/Average number of re-transmissions

N_u, N_{ju}, N_{ja}	Number of: users, jammers on uplink and jammers on downlink
λ_u, λ_{ju}, λ_{ja}	Packet generation rate per user, jammer on uplink and jammer on downlink
ρ_u^s, ρ_{ja}^s, ρ_{ju}^s	Ratio of users, jammers on uplink and jammers on downlink transmitting on SF s
r_u^s, r_{ju}^s, r_{ja}^s	Average packet arrival rate on a given SF s and frequency channel per user, jammer on uplink and jammer on downlink
T_u^s, T_{ack}^s T_{ju}^s, T_{ja}^s	Packet time on air of: users, ACK, jammers on uplink and jammers on downlink transmitted with SF s
T_{vju}^s, T_{vu}^s, T_{vdw}^s	Vulnerability periods: user – jammer on uplink, user – user on uplink, and ACK – jammer on downlink on a given SF s
G_u^s, G_{ju}^s, G_{ja}^s, G_{ack}^s	Normalised traffic load of: users, jammers on uplink, jammers on downlink and, ACK packets on a given SF s
$\overline{G_j}$	Aggregated traffic load of jammers on downlink and uplink channels
G_{dw}^s, G_{up}^s	Normalised traffic load of the uplink and downlink channels on a given SF s
$\mathcal{P}_u^s$	Probability that a user packet is correctly received given that it is transmitted with SF s
$\mathcal{P}_A^s$	Probability that a user packet is correctly acknowledged given that it is correctly received with SF s
Q_{TX}^s	Probability that the GW transmit an ACK to acknowledge a user packet that has been correctly received with SF s
Q_{RX}^s	Probability of successful reception of an ACK packet given that it was transmitted by the GW with SF s

INTRODUCTION

Context

The IoT knows a great expansion with a predicted of 75 billion of IoT devices [8] by 2025. LPWAN have emerged as one of the most promising IoT wireless technologies. LP-WAN technologies are appropriate for devices that require to send small amounts of data over a long range, at a low cost, while maintaining long battery life. Achieving both might sound counter-intuitive, but it is possible through the use of lower frequencies, lower bit rates, and more robust modulation techniques such as CSS [1].

Sigfox, LoRaWAN, and NB-IoT are the three leading LPWAN technologies [2]. While, NB-IoT is a 3GPP standard that, as other mobile technologies like 4G and 5G, uses licensed spectrum, Sigfox and LoRaWAN operate in the sub-GHz free band.

As for any communication systems, security is one of the biggest concerns. In LPWAN, security is currently provided by symmetric-key algorithms such as AES 128 at upper levels. In the case of LoRaWAN it offers application level payload encryption and network level integrity. This means that, if implemented well, it can reasonably be secured against upper level attacks such as DDoS. But nevertheless, this does not shield it against attacks at a lower level such as jammer-type attacks.

A jamming attack takes place at the PHY layer. It is an external node that sends unauthenticated packets to every wireless station in the network with the aim of disrupting communications by decreasing the SNR. It can be done in an intelligent way, where the jammer knows exactly whether or not a node is transmitting (channel-aware), or in a brute force attack (channel-oblivious) where the jammer starts to transmit on the channel permanently [3].

In this thesis, we focus on LoRaWAN networks under jamming attacks. We address this problem through a holistic approach that encompasses three fundamental axes:

i) Network modelling:

- A new mathematical model for LoRaWAN under jamming.
- An enhancement of a ns3 module for LoRaWAN under jamming.

ii) Performance evaluation of LoRaWAN under jamming:

- Extensive simulation campaigns.
- A test-bed for LoRaWAN under jamming.

iii) Jamming countermeasures:

- a jamming attack detection algorithm using a statistical approach.
- a jamming attack detection algorithm using machine learning.

Organisation of the thesis

The remainder of this thesis is organised as follows: In Chapter 2, we introduce the general context of the thesis. We first present an extensive review of the different IoT existing technologies. Then, a comparison of the use-cases present in the current market is provided, hence we conclude which are the most appropriate technologies for each case. Finally, we present a brief description of the security protocols used in the most widely used IoT technologies.

In Chapter 2, we focus in the description of LoRaWAN. There we address different particularities of this type of networks: (i) the LoRa modulation, (ii) the LoRaWAN specification, (iii) the ADR and (iv) duty-cycle limitations, and band regulations.

In Chapter 3, we present an overview of the main security features of the LoRaWAN specification, an analysis of its vulnerabilities, and a revision of the main attacks that have already been documented in the literature. We will put special emphasis in jamming attacks and how they can affect LoRaWAN. We provide a taxonomy of the different jamming attacks and their relationship with loRaWAN. And finally, we will present the general threat model used throughout the theses.

In chapter 4, we focus on network modelling of LoRaWAN under jamming. We first present a detailed description of both our novel mathematical model and an open source event driven simulator written in the Network simulator 3 (ns3) [17].

In chapter 5, we present the performance evaluation of LoRaWAN under jamming. We first use our mathematical model and event driven simulator environment to evaluate network performance. And finally we introduce a test-bed that allows to implement basic jamming-attacks with LoRa hardware.

In chapter 6, we focus on jamming countermeasures. We first present two non-invasive countermeasures (without changing the LoRaWAN specification): (i) Gateway diversity, (ii) message re-transmissions. And finally, we focus on jamming detection. then we introduce two detection approaches: (i) an algorithm based on statistical detection and (ii) a machine learning approach using RNN.

Finally, Chapter 7 concludes this thesis with an overall summary with the main discoveries and the scope for future work is also mentioned.

Chapter 1

Theoretical Framework

In this chapter, an overview of IoT is presented. We will focus on its evolution from Wireless Sensor Networks (WSN) to the actual paradigm of LPWAN. We will also provide a subjective classification of the different use-cases and application domains of these networks. Then, we focus on LPWAN and its security challenges and we summarize and discuss some relevant and open issues in the security research field.

1.1 Evolution and challenges of IoT

The idea behind Internet of Things is to interconnect all sort of things without the need of human intervention. A "thing" can be a person with a connected pacemaker, an animal with a biochip, a vehicle with an air pollution sensor, a humidity sensor in a tomato crop, and the list can go on forever. The IoT is then an umbrella term that encompasses different concepts such as *smart home*, *smart city*, *smart health*, its main objective is to connect everything to the internet without the need for human intervention.

Making this possible requires the convergence of multiple technologies. Just to name a few, we have: real-time analytics, machine learning, commodity sensors, embedded systems and communications technologies. At first, IoT was mostly related with *smart home* applications. Hence, services were provided through Machine-to-Machine (M2M) connections. These systems typically connect devices such as lighting control systems, music streaming control, security cameras, *etc.* The user is able to control these sensors/actuators through an smartphone app or a Web site. However, the scale of devices was very limited. As time has passed, these simple M2M applications were added to applications with a bigger scale like smart grids and smart cities. Thus, the number of connected devices have grown exponentially.

Consequently, the needs towards IoT communications technologies is very different depending on the type of application. The data rate needed, can vary from a few bytes per day in a smart metering application to hundreds of megabytes in remote monitoring. With regards to latency demands, it varies from several hours in applications such as waste management to less than one millisecond in e-health interventions. In respect of connectivity, it may vary from robot movements in Industrial Internet of Things (IIoT) applications to various kilometers in agriculture applications. An finally, in terms of network scale, it varies from less than 10 devices

in *smart home* applications to hundreds of thousands in *smart city* environments.

In the next section broad view of the world of IoT will be given, and we will try to classify the majority of current application domains and use cases.

1.2 Application domains and use-cases

Classifying all the IoT application domains is not an easy task. In fact, many taxonomies have been proposed in the literature, we can find works that provide classifications based on, for example, general aspects of the IoT [18–21], IoT sensor types [22], deployment scenarios [23], or some others related to specific application domains like, the e-health [24, 25] or the IIoT [26]. Having said that, in the following paragraphs we will try to do a general characterization of the main application domains and its respective use-cases. This will be done by defining the main network requirements of each application domain.

Table 1.1: IoT Application domains and use cases

Application Domain	Use Case	Coverage				Data rate			Mobility	Low Latency
		(+5km)	(200m - 1km)	(10 - 100m)	(<10m)	<1kb/s	1kb/s - 500kb/s	>500kb/s		
Smart Home	Video Surveillance			•				•		
	Lightning			•		•				
	Heating			•		•				
	Smoke Detection			•		•				
	Smart Appliances			•			•			
Smart City	Smart Grids	•				•	•			
	Street Lighting	•				•	•			
	Water Level monitoring	•				•	•			
	Road Signalization	•				•	•			
	Air Quality monitoring	•				•	•			
	Public transport	•					•		•	•
	Waste management	•				•	•			
Healthcare	Remote health monitoring			•	•			•	•	•
	Sport/fitness connected			•	•		•		•	•
	Remote Surgery			•	•			•	•	•
	Chronic disease monitoring			•	•				•	•
	Drug provisioning			•	•	•	•		•	•
Agriculture	Smart Irrigation	•				•				
	Wind monitoring	•				•				
	Soil moisture monitoring	•				•				
	Crop management	•				•				
	Livestock Monitoring	•				•			•	•
Industrial	Logistics		•				•			
	Warehousing	•	•				•			
	Tracking Assets	•	•			•	•		•	•
	Remote Diagnosis			•		•	•			•
	Energy Efficiency		•	•		•	•			

1.2.1 Smart Home

The term *smart home* refers to the use of IoT Technologies in a residence in order to enable remote monitoring and management of the house's functions. The main objective of this application domain is to provide the customer with a variety of functionalities allowing to manage the entire house environment by means of sensors and automated systems. Typical uses cases of smart home are lighting, heating and smart appliances. It can also be extended to video surveillance or autonomous demand-side energy management systems. Consequently, this kind of use-cases demands from the network, small coverages ranging from 10 m to 100 m, very small data rates ($<1kb/s$) when it comes to sensor systems and high data rates in specific applications like video surveillance. Mobility and low latency are not specially required. As shown in [19], typical Network technologies used in Smart Home environments are WiFi, Bluetooh, and IEEE 802.15.4.

1.2.2 Smart City

A smart city is a city that uses a large number of interconnected sensors to manage and monitor its assets and resources in an efficient way. In this context, the role of IoT Technologies is to connect these different sensors to the network and make possible the provision of different services. Typical use cases of Smart City are public transport systems, smart grids, smart lighting and waste management.

As it can be seen, the action area of these application domains is far wider than those reviewed in smart home environments. As a result, from the network point of view these application demand network technologies with much wider coverage. Depending on each use case, data rate requirements can be high, but in most cases, as shown in [27], few kb/s are sufficient for most cases. Regarding mobility and low latency, for some use cases like public transport is a must, but for others like waste management is not specially required.

Typical network technologies used in this application domains are those offering wide area coverage like traditional Global System for Mobile Communications (GSM) (EC-GSM-IoT) networks, and more recently LPWAN networks like SigFox or LoRaWAN.

1.2.3 Healthcare

The incorporation of communication technologies in the health sector have been widely studied in literature for decades. The IoT is an enabler of many medical applications, such as vital signal monitoring, fitness programs and remote drug provisioning [24].

The main characteristic of these IoT devices is that they are by nature wearable, meaning that the action range of a typical device is of the order of a few metres or even less. Depending on each specific use case, it is also expected that these devices have high reliability. From the network point of view, some uses cases like remote surgery require very low latency and high data rates. Some others like sport/fitness bracelets demand very low data rates and mobility. With regards to Network Technologies used in this application domain, according to [24], IEEE 802.15.4 and Bluetooth are typically used in the development of these kind devices.

1.2.4 Agriculture

Agriculture industries are a vital part of our society, however, they are also traditionally disconnected. The role of IoT technologies in this context is to provide farmers with accurate data allowing them to take well-informed decisions. There are lot of use cases in this specific domain. IoT technologies can help farmers in different activities like, irrigation, wind monitoring and livestock monitoring.

The main characteristic of these IoT devices is that the vast majority of them must have an action range of the order of hundred of metres or even kilometres. For example, in a use case like water irrigation, the network is expected to sent trigger signals to a mesh of devices that can be deployed in an area of hundreds of square kilometres. Some other uses cases like real time livestock tracking may also require mobility and low latency.

Consequently, technologies used in this application domain are typically those that provide wide area coverage like EC-GSM-IoT, LoRa and SigFox, those that can be easily scaled by using a mesh topology like 802.15.4 and Radio-frequency identification (RFID) for tracking use cases [28].

1.2.5 Industrial

The so-called IIoT or *Industry 4.0*, is an umbrella term that encompasses different technologies that allow automation and data exchange in manufacturing environments. It incorporates technologies like, machine learning, big data, cloud computing, cyber-physical systems and wireless technologies.

The IIoT can be used in each of the links of a certain value chain no matter the market sector. As an illustration of this, let us consider the study made by *Zhuming Bi et al* in [29] where the impact of the Internet of Thing in different Enterprise Systems was investigated. They showed that there is an heterogeneous environment where different wireless technologies converge and the proper use of each technology depends on the specific environment. For example RFID is more suitable for supply chain management, and WSN networks like Zigbee fit better in data collection functionalities.

1.3 IoT Communication Technologies

As pointed out before, there is no consensus on a unique IoT architecture. One of the reasons for that is that there are many communication technologies available in the market, each one with its own characteristics, advantages and disadvantages depending on the application domain and the use case.

1.3.1 IoT Generic architecture

Generally speaking, a network architecture is an abstraction model that allows to divide the services offered by a communication system into layers. Each layer is in charge of providing services to the layer above it and to request services from the layer below. As we have seen,

contrary to traditional Internet, IoT provides M2M and Human-to-Machine (H2M) connectivity. Providing these connectivity leads to a massive traffic. Therefore, the classical TCP/IP architecture does not suits the needs of IoT. Therefore, there is no consensus about the architecture that should be used to develop or deploy IoT technologies. Meanwhile, there are some projects like One M2M, FIWARE or Internet of Things Architecture (IoT-A) trying to design a common architecture based on specific requirements at different abstraction levels [30]. Various models have been proposed in the literature [19, 31].

As shown in Figure 1.1 the most used are the 3-layer model, the 5-layer model and the classic Open System Interconnection (OSI) model. The basic architecture model proposed in the literature is a three-layer. It consists of: perception, network and application layers:

1. Perception layer: It is the physical layer that senses the environment to perceive the physical properties (e.g. temperature, speed, humidity, *etc*). For that it uses different technologies (e.g., GPS, RFID, *etc*).

2. Network Layer: It is the layer in charge of getting data from the perception layer and transmitting it to the application layer through various network technologies (e.g., Wi-Fi, Long Term Evolution (LTE), LoRaWAN, *etc.*). It is also in charge of data management task like data storage.

3. Application Layer: It is the layer that is in charge of delivering application-specific services to the user. Its importance lies in that it makes possible all the application domains and use cases previously presented.

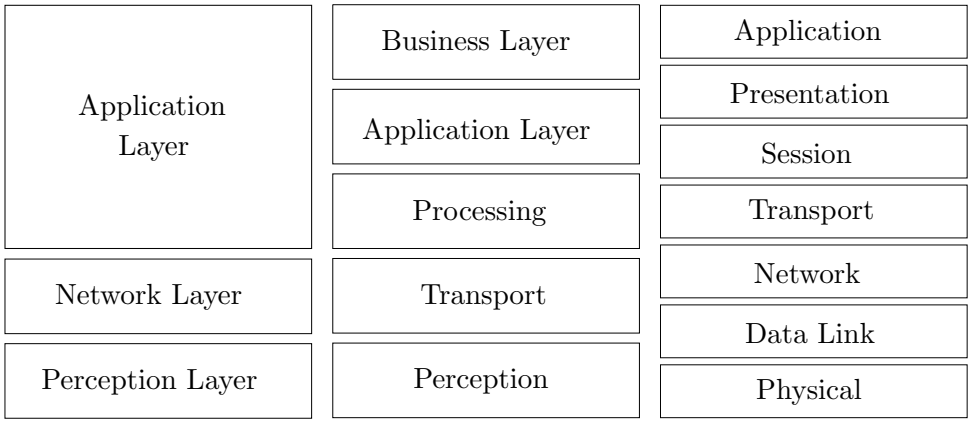


Figure 1.1: Different IoT Architecture models.

Another proposed layered architecture is the five-layer architecture presented in figure 1.1 (b). The five layers are from top to bottom: business, application, service management, object abstraction, and object. The functions of perception, transport (i.e., network layer) and application layers are the same as in the 3-layer architecture. The remaining layers of the architecture are:

1. Processing layer: It is responsible of offering different services related to data management, it covers storing, analysing, and processing data.
2. Business layer: Its work covers the overall IoT system actions and functionality. The application layer sends the data to the business layer whose role is to build business models, graphs, and flowcharts to analyse data, in order to play a role in decision making about business strategies and road-maps.

Finally, the classical OSI model is presented in figure 1.1 (c). It is the reference model for any network protocol. The seven layers are from top to bottom: application, presentation, session, transport, network, data link and physical. The functions of Application, Presentation and session are the same as the application layer as in the 3-layer model, transport, network and data link are included in what would be the network layer of the 3-layer model. The lower level of the architecture are:

1. Network: It is in charge of structuring the packets and managing a multi-node network, including addressing, routing and traffic control.
2. Data link: It assures reliable transmission of data frames between nodes connected by a physical layer.
3. Physical: It is in charge of the transmission and reception of raw bit streams over a physical medium.

It is worth nothing that the work presented in this thesis concerns mostly the physical and data link layer. More specifically, in the data link layer, we study the Medium Access Control (MAC) protocol in LoRaWAN, which as we will see in detail is one of the enabling IoT network technologies. Therefore, in the next section we present a comparison of the main IoT technologies used in this layer.

1.3.2 Technologies comparison

Table 1.2 shows a comparison of the most popular communication technologies to date based on different criteria: standard, frequency, band operation, data-rate, coverage, topology and energy consumption.

In addition to the technologies shown in Table 1.2 there are many other technologies in the market supported on top of these standards. A clear example of that is the IEEE 802.15.4 standard, which is probably the most used standard for WSN applications with many technologies based on it. The most popular are: ZigBee, Thread, 6LoWPAN, Wireless Hart and ISA100 [50].

As it can be seen in Table 1.2, there are lot of technologies in the market that can be used to develop and deploy IoT networks. Choosing the wireless technology for an IoT use case requires careful consideration of multiple factors related to the characteristics of each technology. The following is a review of the most important characteristics that should be considered when selecting an IoT technology.

Table 1.2: Main characteristics of existing IoT wireless communication technologies.

Commercial name	Standard	Frequency band [MHz]	Data-rate [Kb/s]	Coverage [km]	Topology	Energy Consumption	Ref
Wireless Local Area Network (WLAN)							
WiFi	IEEE 802.11 (n/ac)	2.4 & 5	6.5 – 69,330	~ 0.2	Star, Mesh	High	[21]
Wi-Fi HaLow	IEEE 802.11.ah	≤ 1	100 - 4,000 5,800 - 78,000	~ 1.5	Star	Low	[32,33]
Wireless Personal Area Network (WPAN)							
LR-WPAN	IEEE 802.15.4 (a/b)	≤ 1 & 2.4	20 - 250	~ 0.05	Star mesh	Medium	[34,35]
Bluetooth	IEEE 802.15.1	2.4	1,000 - 3,000	~ 0.1	Point to point	Low	[35]
BLE	IEEE 802.15.1	2.4	125 – 2,000	~ 0.05	Point to point	Very Low	[35]
Wireless Wide Area Networks (WWAN)							
eMTC	3GPP TR 45.820	3GPP Bands	10 - 1,000	~ 10	Star (LTE)	Low	[36–38]
NB - IoT	3GPP TR 45.820	3GPP Bands	20 - 50	~ 10	Star (LTE)	Low	[36–38]
EC-GSM-IoT	3GPP TR 45.820	3GPP Bands	70 - 240	~ 10	Star (GSM)	Low	[36–38]
Low Power Wide Area Networks (LPWAN)							
DASH7	DASH7	≤ 1	9.6 - 167	~ 2	Star	Low	[39,40]
LoRa	LoRaWAN	≤ 1	0.3 - 50	~ 10	Star	Very Low	[41]
WAVIoT	NB-Fi	≤ 1	0.05 - 0.1	10 - 50	Star	Very Low	[42]
Sigfox	SIGFOX	≤ 1	0.6	~ 13	Star (Operator)	Low	[35]
Weightless W	Weightless SIG	TV White Spaces	1 - 1,000	~ 5	Star	Low	[43]
Weightless N	Weightless SIG	≤ 1	0.1	~ 3	Star	Low	[43]
Weightless P	Weightless SIG	≤ 1	0.2 - 100	~ 2	Star	Low	[43]
Wavenis	Wavenis OSA	≤ 1	9.6 - 19.2	~ 4 (LoS)	Tree, Star (Max 3 hops)	Very Low	[44]
Proximity Networks							
NFC	ISO/IEC 14443 ISO/ 18000-3	13.56	106 - 848	$\sim 0{,}0001$	Point to point Communication	Medium	[45]
Wireless Home Area Network (WHAN)							
Z-Wave	Based on MAC/PHY ITU G.9959	≤ 1	9.6 - 100	~ 0.03	Star, Mesh	Low	[46]
EnOcean	ISO/IEC 14543-3-1	≤ 1	125	0.03 - 0.3	Mesh	Low	[35,47]
Radio-frequency identification Networks (RFID)							
RFID	ISO 11784 ISO 11784/85 ISO 14223 ISO/IEC 18000	0.135 13.56 860-960 2,450	4 - 40	~ 0.00005	Point to point	Very Low	[48]
Magnetic Field Networks							
RuBee	IEEE 1902.1	0.03 0.9	9.6	≤ 0.03		Very Low	
Infra-red Communications							
IrDA	IrDA PHY v1.5	IR Light	9.6 - 96,000	~ 0.00001	Point to point	Very Low	[49]

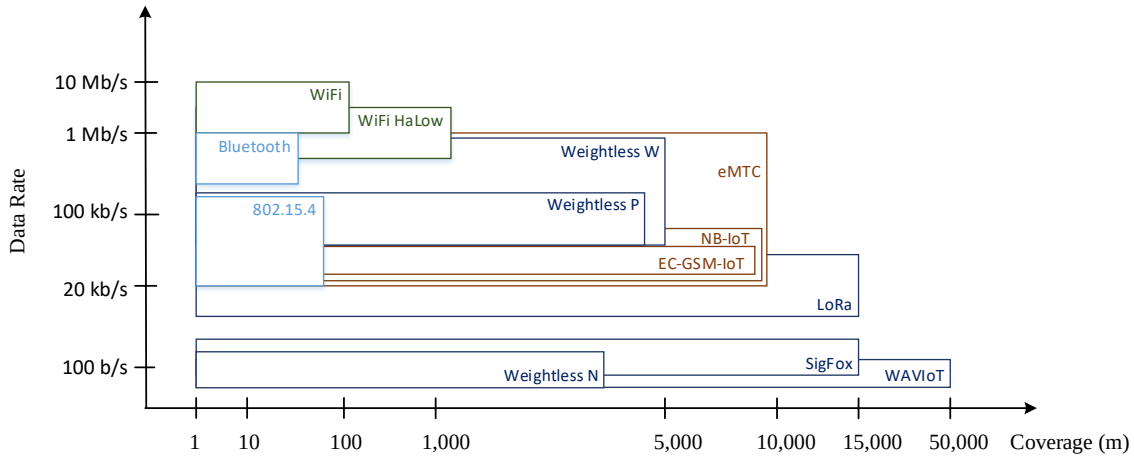


Figure 1.2: Data rate and coverage of IoT wireless technologies.

1.3.3 Coverage and data-rate

The first characteristics we will consider are the coverage area and the data-rate. As shown in Figure 1.2 in most cases greater range can be provided by those technologies with reduced data-rates and vice-versa. Technologies such as Bluetooth, BLE, Wi-Fi (n/ac) and LR-WPAN are more likely to be used in applications where the coverage range is not an issue and relatively high data rates are appreciated. The typical use cases of these technologies can be found in smart-home environments. On the other hand, if the use case requires long coverage and relatively high data rates, technologies like enhanced Machine-type Communication (eMTC) are ideally suited.

Just in the middle of these two groups of technologies we have the LPWAN, they providing long range coverage and modest data-rates. The most popular technologies in this group are LoRa, SigFox, NB-IoT, WAVIoT, Weighless (W/N/P) and Wavenis. Typical use cases of these technologies can be found in smart-city applications. A detailed description and analysis of LPWAN will be presented in Section 1.4.

Finally, technologies such as Near Field Communication (NFC), RFID and IrDA offer low data rates and low coverage. Having these characteristics seems counter-intuitive, but it actually can be an advantage for certain use cases. NFC is a clear example of that, it is been widely used in high-tech credit cards where 10cm coverage means high security and $\sim$ 500 kb/s is enough to do a financial transaction.

1.3.4 Power Consumption

As shown in Table 1.2, the vast majority of technologies used in IoT are battery based. As a result, the power consumption of its protocols should be reduced when compared with other wireless technologies. It is difficult to classify the energy consumption of different IoT Wireless technologies because, as shown in [51,52], it depends on many factors like the amount of energy

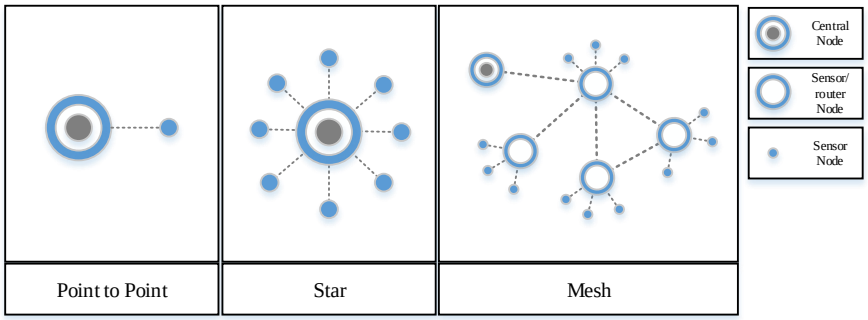


Figure 1.3: IoT Wireless Technologies Topologies.

used on communications, on the acquisition/processing of data and even on the specific radio module used in the solution. As it will be shown latter, in this thesis we propose a simulation environment that includes a current consumption model for LoRaWAN which is one of the most used IoT networking technologies today.

1.3.5 Topology

Generally speaking, a network topology is the arrangement of a network, including its nodes and connecting lines. Thus, we can categorise IoT networks into three basic network topologies; point-to-point, star, and mesh. Figure 1.3 illustrates these three topologies followed by a brief description of each one.

1.3.5.1 Point-to-Point

In a point-to-point or one-hop network, a direct connection between two network nodes is established. Meaning that communication can take place only between these two nodes. An example of this type of network is Bluetooth and NFC. The primary limitation of this topology is that the network cannot scale beyond these two nodes. Consequently, the whole transmission range of the network is defined by the transmission range of a single node.

1.3.5.2 Star

The Star network topology is composed by one central node (hub), to which all nodes in the network are connected. The role of this node is to act as a controller of all communication that takes place within the network. As a result, all peripheral nodes may communicate with others by transmitting to, and receiving from, the central node. The classic example of this topology is the WiFi access technology.

The general performance of this network topology is consistent and predictable, unlike the mesh topology, the number of hops a packet travels is no more than two (one when sending data from one sensor to the hub, or two when sending data between two nodes). As a result, most of the technologies that use this kind of topology offer low and predictable latency. From the

reliability point of view, this topology is an excellent option due to the ease of isolating certain nodes when faults occur.

Disadvantages of this topology are similar to point-to-point. The overall range of the network is limited to the range of a single device (the hub). In addition to this in a star topology there is a single point of failure. As a result, if the central node loses connectivity, the network is cut off and no exchanges can be made between each node. This topology is usually found in long range technologies such as classical 3GPP networks, NB-IoT, LoRa and Sigfox.

1.3.5.3 Mesh

A mesh topology consist of three different type of nodes. First of all, one or several central nodes. Second, some sensor/router nodes and finally simple sensor nodes. In this way, the central nodes serve as connectors to the outside world, the sensor/router nodes are capable of capture their own data, disseminate it and also serve as relays for other nodes, and the sensor nodes are only capable of capture and send their own data to higher level nodes.

Consequently, mesh network nodes are deployed in a way that every node is within the transmission range of at least one sensor/router node. Data packets pass through multiple sensor/router nodes to reach the central node. For this reason, the overall coverage of this topology is not limited to the coverage area of a single node. Thus, the network can be scaled up to hundred or even thousand of nodes, but nevertheless, the complexity of the network is higher resulting in, for example, higher delays because the number of hops to reach the central node is not the same for all sensor nodes. Typically, this topology is used to extend the range of technologies such as Bluetooth and Wi-Fi forming the WSN.

1.4 LPWAN: a new IoT paradigm

In the early 2000s, what we know now as smart home was more commonly called as home automation. It was offered by making use of the first standards of wireless technologies such as WiFi and Bluetooth. Thus, during this decade, these technologies began to be used for different tasks that require a lot of coverage and we started to hear therms such as multi-hop networks, Mobile ad hoc networks (MANET) or WSN. The idea was to extend their coverage through the implementation of mesh-type network topologies. [4].

At that time, there was also access to cell phone networks such as 2G and 3G, they can provide provide larger coverage, but they consume excessive device energy. Therefore, new IoT applications requirements have driven to the emergence of new wireless communications technologies: LPWAN. It was not until 2013 that the term was coined [5]. It refers to a network with low power consumption, long range, and low-cost communication characteristics. It provides long-range communication up to 40 km in rural zones and 1-5 km in urban zones [6,7]. Additionally, the nodes are meant to be highly energy efficient (promising up to 10 years battery lifetime).

These outstanding promises led to an increasing interest from industry and the academic community [6, 53, 54]. Many LPWAN technologies have arisen in the licensed as well as unli-

censed frequency bands. As presented before, Sigfox, LoRa, and NB-IoT are the current leading emergent technologies. In the following we will provide a brief description of these three technologies. A more detailed description of LoRa and LoRaWAN, the technology chosen for our work will be presented in chapter 2.

1.4.1 NB-IoT

Along with eMTC and EC-GSM-IoT (Extended Coverage - GSM - Internet of Things), NB-IoT is one of the three technologies defined by the 3GPP to support IoT applications. These technologies have been standardized by the 3GPP in its Release 13, and are the result of a series of modifications of current standards to overcome different limitations of traditional cellular networks, allowing better indoor coverage, power consumption and massive number of low-throughput devices.

NB-IoT is designed for those Mobile Network Operators that already have LTE deployments and want to implement IoT solutions that involve low throughput (ranging from 20 to 50kb/s) and wide coverage (of the order of 10 km). A battery life of more than 10 years is expected to be supported for a wide range of IoT use-cases such as Smart Metering and Smart Building [55].

As expected, NB-IoT works on licensed spectrum that is already at the disposal of the operators. It supports three operation modes:

(a) Stand-Alone operation, designed to utilise any available spectrum. This can be done by re-farming spectrum currently being used by 2G/3G/4G systems.

(b) Guard band operation, utilising the resource blocks within a LTE carrier's guard-band.

(c) In-band operation, by using resource blocks within a normal LTE carrier.

In order to overcome limitations of LTE, different characteristics were incorporated by the 3GPP TR 45.820 Standard. At the physical level, Low Peak to Average Power Ratio (PAPR) transmission technology is supported, UE power consumption is optimised by using Power Saving Mode (PSM) and Extended idle-mode DRX cycle (eDRX).

PSM is a new low-power mode that allows a User Equipment (UE) to enter in sleep mode while at the same time remaining registered with the network, so that no further re-attach procedures are needed to be done. The maximal dormant time is about 413 days and the UE is not reachable when PSM is being activated. The procedure is done by using different counters, in a way that the network is always aware of the moment the node will be active.

eDRX is another mechanism that reduces power consumption by extending the sleeping cycle in idle mode between data reception from the network in connected mode, passing from 2.56 seconds to about 40 minutes. During the extended idle mode, the UE is not listening for paging or downlink control channels. Consequently, the UE is not reachable.

Selecting one mode or another largely depends on the use-case. PSM is intended for use-cases where the device initiates communication with the network, and eDRX is optimised for device-terminated applications. The former for use-cases such as smart metering or any IoT

devices that periodically send data, and the letter for use-cases such as asset tracking and smart public transport [56]. A diagram of both modes is depicted in Figure 1.4.

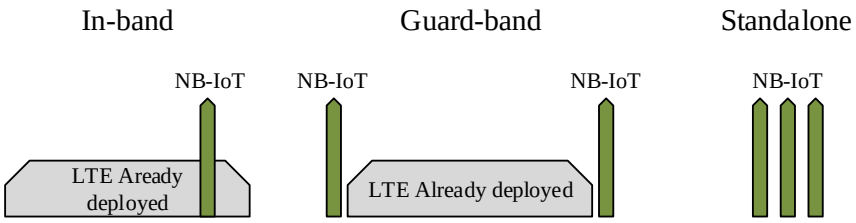


Figure 1.4: NB-IoT deployment modes.

1.4.2 SigFox

Sigfox is another leading LPWAN technology. It is a network operator that offers IoT solutions based on its proprietary base stations. A SigFox base station is equipped with Software Define Radio (SDR). The connection between the End Device (ED)s and base stations is done through the of binary phase-shift keying (BPSK) modulation in an ultra-narrow band (100 Hz). It employs the sub 1 GHz ISM band. Using this type of modulation techniques allow good spectral efficiency and at the same time very low noise levels. This leads to very low power consumption, high receiver sensitivity, and low-cost antenna design. However, the data rate is only 100 bps.

Most of the use-cases are designed for unconfirmed traffic, but bidirectional technology is supported. However, downlink transmissions are very limited (up to four eighth-bytes packets per day) and can only occur following an uplink communication. Another limitation of the network is that the number of messages over the uplink is limited to 140 messages per day and that the maximum payload length for each uplink message is 12 bytes.

Additionally, the uplink communication uses time and frequency diversity as well as transmission duplication. Therefore, an ED can randomly choose a frequency channel to transmit their messages. Thus, each uplink message is transmitted multiple times (three by default) over different frequency channels. As an example, in Europe the band between 868.180 and 868.220 MHz is divided into about 400 frequency channels (100 Hz). This simplifies the ED design and reduces its cost. This simplicity on the ED side is balanced at the base stations. They are built in a way that they can decode multiple messages simultaneously over all channels [57].

1.4.3 LoRa

LoRa (Long Range) is a physical layer technology that, as SigFox, modulates the signals in sub-GHz ISM band using a proprietary spread spectrum technique [58, 59]. It provides bidirectional communication by using chirp spread spectrum (CSS) modulation that spreads a narrow-band signal over a wider channel bandwidth. The resulting signal has low noise levels, enabling high interference resilience [60]. LoRa uses six spreading factors (7 to 12) to adapt the data rate and

range tradeoff. Higher spreading factor allows longer range at the expense of lower data rate, and vice versa. The LoRa data rate is between 300 bps and 50 kbps depending on spreading factor and channel bandwidth. Further, messages transmitted using different spreading factors can be received simultaneously by LoRa base stations. The maximum payload length for each message is 243 bytes. On top of this technology, LoRaWAN is a specification that defines an Aloha-type MAC (Medium Access Control) protocol [61]. A more detailed description of LoRa and LoRaWAN is presented in chapter 2.

1.4.4 LPWAN technologies comparison

As we have seen, various aspects need to be considered to choose an LPWAN technology. First of all we have the different application domains and use-cases presented before and secondly we also have the different aspects of the technology like those shown in the previous section: data-rate, whether it is proprietary or not, its topology and the ED power consumption.

In summary, Sigfox, LoRa, and NB-IoT each has their respective advantages in terms of different characteristics . A comparative kiviat diagram is presented in figure 1.5.

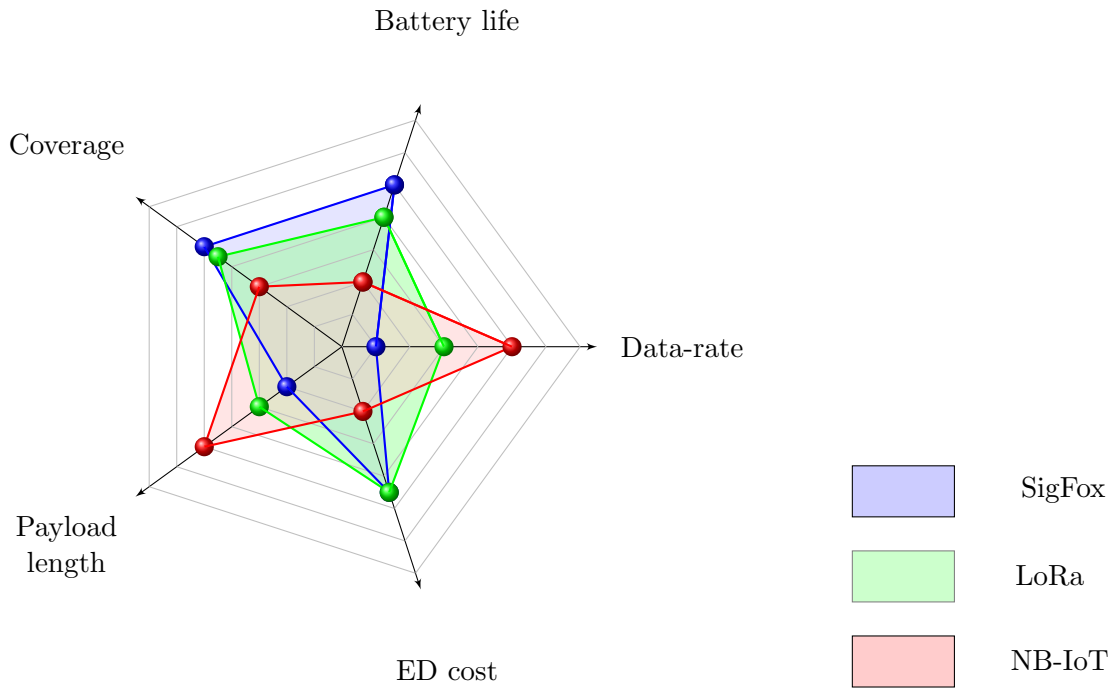


Figure 1.5: NB-IoT leading technologies comparison [2]

1.5 Security in LPWAN

Different authors have already identified numerous challenges that LPWAN networks are facing under the current state of the art [31, 62–66]. As any other communication technology, the main objective is to guaranty information assurance, which is classically defined as the act of

maintaining the confidentiality, integrity and availability (CIA) [63, 67–69]. Hence, ensuring that information is not compromised in any way when critical issues arise:

- Confidentiality: is the property, that information is not made available or disclosed to unauthorised individuals, entities, or processes.
- Integrity: is the process of maintaining and assuring the accuracy and completeness of data over its entire life-cycle. This means that data cannot be modified in an unauthorised or undetected manner.
- Availability: The process of ensuring that the service needed is available anywhere and anytime for the intended users. In the context of IoT, it includes the objects

1.5.1 Security requirements and potential Issues

These three security concerns cited before are mapped into different network security requirements:

- **Authentication**: the network and the user can verify the identity of their counterpart.
- **Data Integrity**: the system offers a way that the data exchanged is not modified by third parties.
- **Encryption**: all messages, or the most sensitive ones, exchanged between the user and the network are encrypted by using well-standardised algorithms
- **Key Management**: there is a way of creating cryptographic session keys.
- **User Identity**: EDs are uniquely identified by using Public Key Infrastructure like the UIT-T.509 Certificates
- **Untraceability**: the network provides a way of protect the location of the end-device.
- **Secure Elements**: EDs are provided with tamper-resistant platform capable of securely hosting confidential data like cryptographic keys. By doing it storage and computing that are taken please inside the node are protected from side-channel attacks.
- **Over The Air provisioning of security patches**: there is a way to update automatically end-devices with security patches after having been deployed.
- **Intrusion Detection**: security Rules relies not only on cryptographic algorithms but also in a system that monitors the network for malicious activity or policy violations.

Taking into consideration the above-mentioned requirements, in the following we will review several attacks that have taken advantage of security failings that arise from not respecting these requirements.

1. Man in the Middle (MITM): in this attack the attacker is able to secretly relay and possibly alter the communications between the two network entities. This attack mostly compromises the confidentiality and it can even break the integrity. An example of this attack will be presented in detail in Section 3.1.

2. Denial of Service (DoS): according to McAfee, this is the most common attack. It occurs when an attacker is able to disrupt the transmission of nodes by using jamming techniques. It mostly affects the availability of the network. An example of this attack is the well known Mirai attack, which is a distributed DoS. According to KrebsOnSecurity, the victim of the attack, Mirai was able to generate 620 Gbps of traffic coming from all sort of connected things in different countries around the world [70]. The attack was able to turn down several sites including Twitter, the Guardian, Netflix, Reddit, CNN and many others in Europe and the US [71].

3. Eavesdropping/sniffing: it is a passive attack that consist in listen to the private communications. It mainly attacks the confidentiality of the networks and is usually employed as a first step for bigger attacks.

4. Routing attacks: it affects the way data is routed from one point to another. The attacker is able to intrude the network. Then, it spoofs, redirects, misdirects or even drops packets at the network layer [72, 73].

1.6 Summary and Discussion

In this chapter, we presented an overview of IoT, its evolution, its current applications and more specifically the current use-cases present in the market. We saw that it is a growing sector that is continue evolving, where the needs are always much ahead of the technologies. From there new technologies emerge quickly.

Then, we focused on the IoT enabling technologies, we have provided a generic architecture that allows to give an order to the different technological factors involved in the vast world of the IoT. Therefore, we specified that in this work we were going to focus exclusively on the so-called network layer, which is what allows communication between the different actors of an IoT network.

We thus presented an exhaustive comparison of several technologies present in this network layer to date. For that we select several characteristics that allows to map the previously reviewed uses cases into specific network technologies. Among them, we cited LPWAN which has emerged recently to respond to the demands of low-power and long-range transmission applications. Then, we saw that, they are three leading technologies: LoRa, NB-IoT and SigFox and we conclude that though the three of them are meant to the same group of application domains, they carry several differences that can be complementary.

Finally, we provide an overall description of the main security requirements and issues of LPWAN. We mentioned that the main objective is to guaranty information assurance in all levels, and that for that three aspects must be guaranteed: confidentiality, integrity and

availability. Particularly, we mentioned that DoS attacks are the ones that occur more regularly, that they affect the availability of the networks and that they can have a greater negative impact on networks.

In the next section, we will go deeper into the description of LoRa and LoRAWAN, which as we have seen is one of the leading LPWAN technologies. Special emphasis will be placed on it security rules and vulnerabilities.

Chapter 2

LoRaWAN Networks

As discussed previously in Chapter 1, the IoT paradigm encompasses a great multiplicity of network technologies. In this thesis we analyse LoRaWAN which is one of the LPWAN networking technologies, the following sections provide a detailed description of the LoRaWAN Networks. We start by describing the LoRaWAN specification, the LoRa modulation, and then we address different particularities of this type of network: (i) ADR, (ii) duty cycle restrictions and (iii) band regulations. The security features of the specification will be addressed in chapter 3.

We will start by describing the technology stack. As shown in figure 2.1, LoRa is the physical (PHY) layer, i.e., the wireless modulation used to create the communication link, while LoRaWAN is an open networking specification that delivers secure bi-directional communication, mobility, and localisation services standardised and maintained by the LoRa Alliance [74].

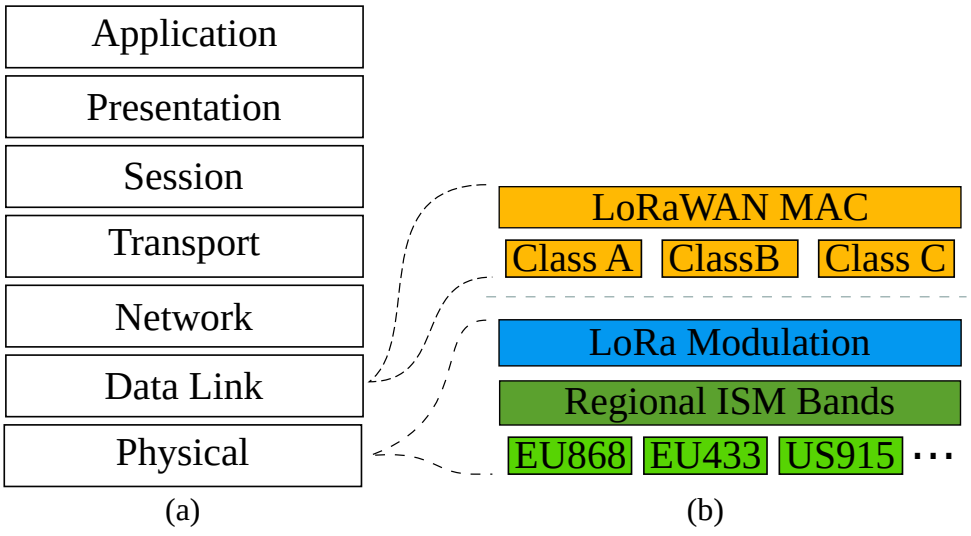


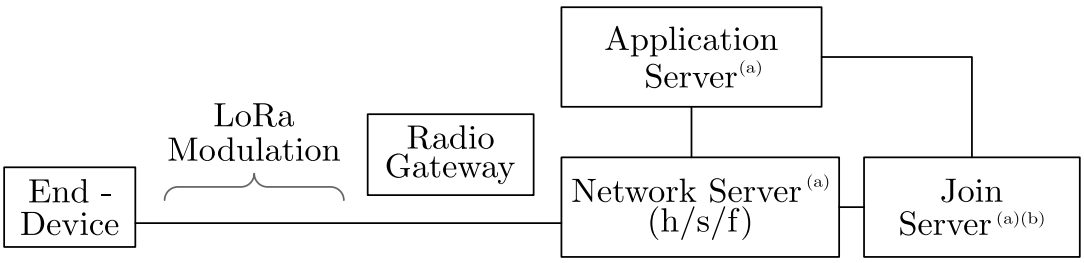
Figure 2.1: LoRaWAN technology stack: (a) OSI model, (b) LoRaWAN Technology

2.1 LoRaWAN Specification

Contrary to LoRa, the LoRaWAN specification designates the upper networking layers. It is a MAC-layer protocol for managing communication between Gateways and EDs. It is maintained

by the LoRa Alliance. It was first released in January 2015: 1.0 [75]. Later updated in February and July 2016 and then in October 2017, the v1.1 was released [61,76]. It is designed for allowing wireless connectivity for battery-based ED that can be mobile or fixed. It operates in the Sub 1 GHz band and are typically deployed in a star-of-stars topology. A LoRaWAN network is composed by ED, Gateways, a Network Server, an Application Server and (for LoRaWAN 1.1) a Join Server. A top level diagram of the architecture is depicted in Figure 2.2.

It is worth nothing that in real world deployments the different versions coexist with each other. That is why there is backward compatibility between versions.



(a) Depending on the deployment these three servers can be merged into one
(b) The Join Server was formally introduced in LoRaWAN Backend interfaces v 1.0
(c) h - home, s - serving, f - forwarding

Figure 2.2: Top level LoRaWAN Architecture

All the exchanges made between EDs and Gateways are spread out on different frequency channels and data-rates (ranging from 0.3 to 50 Kb/s). The selection of the data-rate is a trade-off between communication range and message duration following an ADR algorithm. EDs may transmit on any channel available at any time, using any available data rate. Channel selection is done by following a pseudo-random approach. The number of channels varies depending on the region in which it is deployed (see table 2.2 for more details).

Uplink transmissions are subject to duty-cycle limitations, this parameter depends on the band limitations, as it will be shown later in Section 2.4 it is typically set at 1% for the European case. At the MAC Layer the LoRaWAN specification defines three classes: Class A (baseline), Class B (beacon) and Class C (continuous). Class A is supported by all nodes while B and C are optional.

2.1.1 Class A operational mode

This is the default operational mode, transmissions are done by following an Aloha-like protocol. As depicted in figure 2.3, the ED has its own transmission schedule and each time a transmission is done, the GW may send information using two short downlink receive windows (RX1 and RX2). If additional information is required to be sent by the Gateway, it should be delayed until the next transmission event.

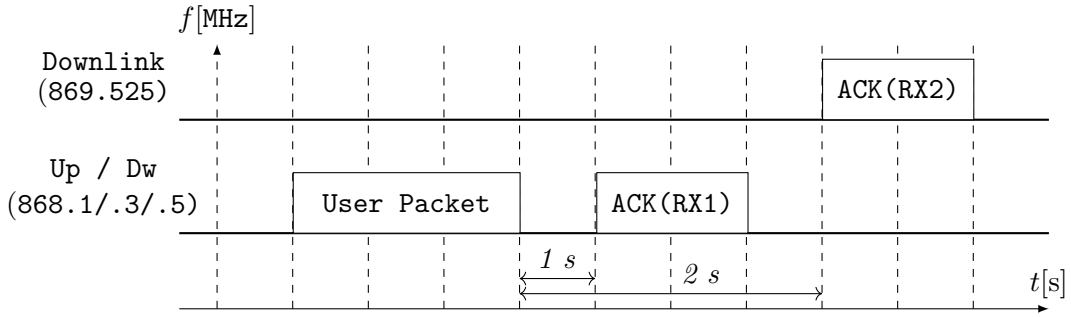


Figure 2.3: LoRaWAN Class A Transmission Schedule

2.1.2 Class B operational mode

The class B operational mode is designed for use-cases that require opening server-initiated receive windows at fixed intervals. As depicted in 2.4, in addition to the two reception windows, in Class B the Gateway send a beacon (BCN) on a regular basis to synchronise all EDs in the network so that it can open a short additional reception window (called ping slot) [61, 77]. Then, these ping slots are used by the network server to start a downlink transmission (PN) when required. A summary of this procedure is presented below:

i) The application server queues a downlink message into the network server.

ii) The network server computes the next ping slot schedule

iii) The network server computes the best gateway to use based on the last uplink received from the device and the current gateway's transmission schedule

iv) The network server queues the downlink into the selected Gateway

v) When the selected ping slot start time is reached, the gateway transmits the downlink message. At the same time, the device turns its receiver on and receives the downlink message.

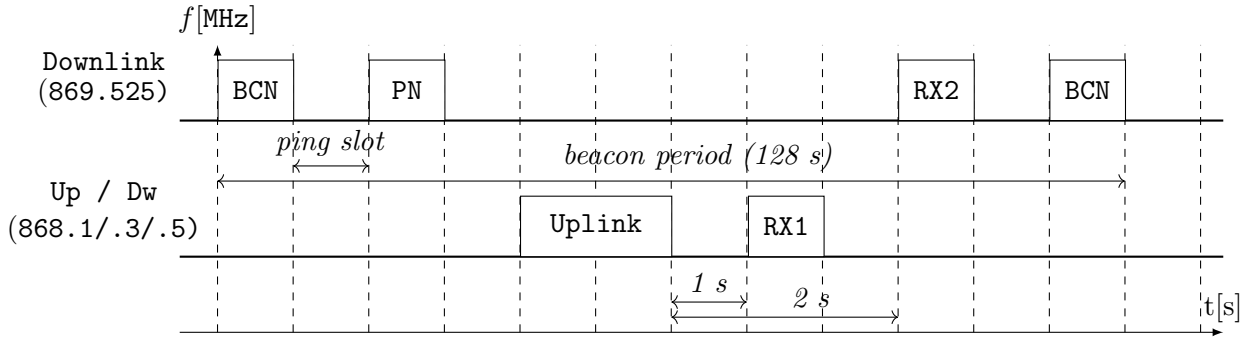


Figure 2.4: LoRaWAN Class B Transmission Schedule

It should be noted that when there is no downlink transmission to be done, the Network Server does not transmit anything. However, the ED still opens the ping slot. When it does not detect anything, it goes back to sleep as quickly as possible to conserve power.

2.1.3 Class C operational mode

Class C mode is implemented for use-cases requiring continuous listening EDs. In this mode, EDs does not have power consumption restrictions. As depicted in 2.5, after each Uplink transmission, the ED is required to open a receive window with the RX2 parameters. Upon the end of RX2, it opens a short Reception window with the same parameters as RX1. Once, the RX1 is finished, it opens again a RX2 window until the next uplink transmission.

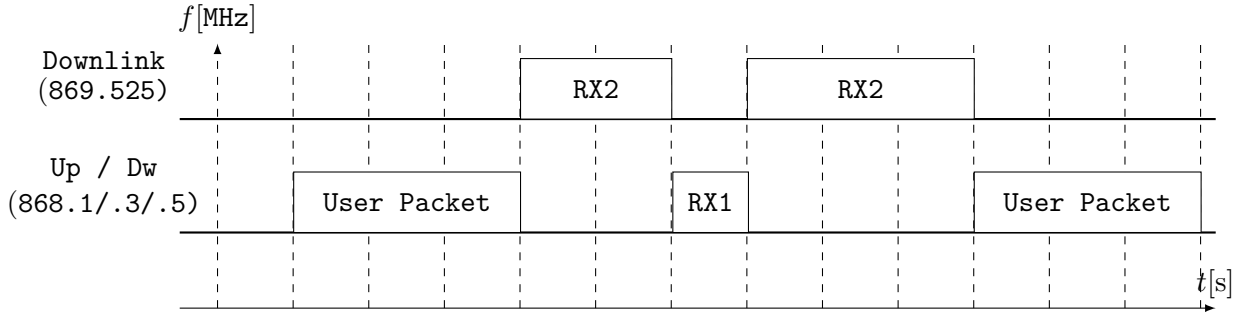


Figure 2.5: LoRaWAN Class C Transmission Schedule

2.1.4 Changes added by the LoRaWAN v1.1 specification

The LoRaWAN 1.1 specification adds features to the previous one. A new network component called Join Server was officially added, it is in charge of managing the join procedure for new end-devices including the derivation of session keys [61]. It also incorporates new features specifically designed to handle roaming procedures. To do that, a Network Server can play three different roles depending on whether the end-device is willing to execute roaming or not and the type of roaming involved (*namely, Serving NS (sNS), Home NS (hNS), and forwarding NS (fNS)*) and finally, Class B operational mode is fully supported by the specification.

2.2 LoRa Modulation

LoRa is a proprietary low-power wide-area modulation based on spread spectrum techniques derived from chirp spread spectrum (CSS). It was developed by Cycleo of Grenoble, France and acquired by Semtech, the founding member of the LoRa Alliance. Because this technology is patented, no official information is available. However, there are some technical documents from Semtech and the LoRa alliance that allow inferring some of its characteristics like the Time-on-Air (ToA) [59, 75, 78, 79]. However, some researchers have carefully analysed and reverse engineered the modulation. An example of that is the open source Matthew Knight's work available in [80].

2.2.1 LoRa - Chirp Spread Spectrum

The LoRa's CSS modulation represents symbols as linear changes in the frequency of a chirp. A chirp is then a signal whose frequency changes linearly. Thus, an instantaneous change in its frequency, or lack thereof, constitutes a symbol. [9]. The fact of distributing a symbol over the spectrum bandwidth makes the signal more resistant to frequency-selective noise, and Doppler effect. However, this leads to a decrease in the spectral efficiency. An Spreading Factor (SF) denoted as s is then a parameter that allows to adjust the number of bits in a chirp. Therefore, the number of possible symbols is equal to 2^s. The bit-rate (R_b) can be expressed as follows [81]:

$$R_b = s \cdot \left[\frac{4}{4+CR}\right] / \left[\frac{2^s}{BW}\right] b/s \tag{2.1}$$

where s ranges from 7 to 12, CR is the Coding Rate (ranging from 1 to 4) and BW is the bandwidth in Hz (it could be 125 , 250 or 500 KHz depending on the band and region).

An example of a LoRa packet transmission from a spectrogram can be seen in Figure 2.6. It is a representation of an experiment performed by Liando et. al in [60] where a SDR environment is used to study the LoRa modulation. The figure illustrates different types of chirps of a regular LoRa packet:

1. Preamble, the chirps follow a predictable pattern: the frequency increases over time and restarts from the minimal frequency (f_{min}) toward max frequency (f_{max}).
2. Start Data Frame (SFD): shortdown chirps that goes from f_{max} to f_{min}.
3. Modulated chirps: chirps that contain actual data bits.

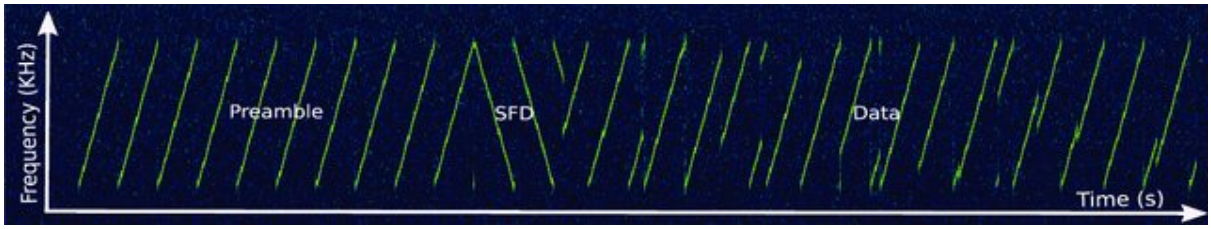


Figure 2.6: A snapshot of LoRa transmission that shows up, down, and data chirps as seen on spectrogram [60].

2.2.2 LoRa time on air

The spreading factor can be also used to compute the duration of a symbol, according to the following expression [81]:

$$T_{sym} = \frac{2^s}{BW} \tag{2.2}$$

The necessary time to send a packet is given by the sum of the preamble and payload duration:

$$T_{packet} = T_{pre} + T_{pay} \tag{2.3}$$

Table 2.1: Time on air (ToA) and SF

s	T_{sym} $[ms]$	T_{pre} $[ms]$	$T_{CAD}[ms]$	T_{packet} $(50\ B)[ms]$	T_{packet} $(250\ B)[ms]$
7	1.024	12.544	0.128	97.536	389.376
8	2.048	25.088	0.2304	174.592	686.592
9	4.096	50.176	0.4352	328.704	1229.824
10	8.192	100.352	0.8448	616.448	2254.848
11	16.384	200.704	1.664	1150.917	4100.096
12	32.768	401.408	3.3024	2138.112	7544.832
Equation.	(2.2)	(2.4)	(2.7)	(2.3)	(2.3)

where T_{pre} and T_{pay}, are given by Equations (2.4) and (2.5):

$$T_{pre} = T_{sym} \cdot (n_{bytes} + 4.25) \tag{2.4}$$

$$\begin{aligned} T_{pay} &= T_{sym} \cdot (8 + \max\left(ceil(Sym_{Nb}) \cdot (CR + 4), 0\right)) \\ Sym_{Nb} &= \frac{8 \cdot PL - 4 \cdot s + 28 + 16 \cdot CRC - 20 \cdot H}{4 \cdot (s - 2 \cdot DE)} \end{aligned} \tag{2.5}$$

where PL is the number of payload bytes, $H = 1$ when the header is enabled, $DE = 1$ when the low data rate optimisation is enabled, $CRC = 1$ when CRC is enabled and CR the coding rate (ranging from 1 to 4).

The time required to detect a preamble is given by Equation (2.6):

$$T_{dpre} = T_{rx1} = N_{dsym} \cdot T_{sym} \tag{2.6}$$

where $N_{dsym} = 8$ for SFs 11 and 12, and $N_{dsym} = 12$ for the others [82].

Another ToA related parameter is the Channel Activity Detection (CAD), which is the minimum time required to detect an activity on the channel; it is normally used to implement the time the ED is in receive mode during RX2 [83]. It is denoted as T_{CAD}, and is given by Equation (2.7).

$$T_{rx2} = T_{CAD} = \frac{2^s + 32}{BW} \tag{2.7}$$

Table 2.1 presents different time parameters based on the ToA per symbol and the ToA per packet according to the SF. We consider 8 bytes as preamble length, a bandwidth of 125 kHz, $CR = 1$ and a payload of 50 and 250 `bytes` respectively. These LoRaWAN parameters derived from the ToA are also fundamental to simulate jamming attacks since they offer the possibility to jam communications easily as we will show in Section 3.2.2.2.

2.2.3 duty-cycle restrictions

According to the ETSI [84], the duty-cycle is defined as the ratio, expressed as a percentage, of the maximum transmitter "on" time over one hour, relative to a one hour period. These limitations apply to every receiver, excluding those with LBT capabilities. Since LoRaWAN

has defined no Listen-Before-Talk (LBT) mechanism as of now, these limits must be respected by all LoRa devices.

Table 2.2 presents the band restrictions that apply for the European case. It shows the frequency band, the Effective Radiated Power (ERP) restriction, the number of channels per band, and finally the duty-cycle restriction per band. It should be noted that for our research we work with the g1 and g3 bands with a configuration of 125 kHz per channel.

Table 2.2: Duty-cycle and power Restrictions in LoRaWAN Bands

Band	Frequency [MHz]	Max ERP [dBm]	# of channels 125 kHz	duty-cycle [%]
g	865 - 868	6.2	15	1
g1	868 - 868.6	14	3	1
g2	868.7 - 869.2	14	2	0.1
g3	869.4 - 869.65	20	1	10
g4	869.7 - 870	14	1	1

2.3 Adaptive Data Rate (ADR)

The ADR mechanism aims at providing reliable connectivity of ED and GW by adapting SF and Transmission Power (TP) to change in the link conditions. It also seeks to minimise the power consumption of EDs.

The algorithm uses downlink messages to handle the selection of SFs and TP. If the EDs detects that a large number of consecutive uplink transmissions are not acknowledged, it assumes that the connectivity is lost. Then it starts to gradually increase its TP, if there is still no answer, it start doing the same for SF.

The specification defines two parameters namely `ADR_ACK_LIMIT` and `ADR_ACK_DELAY` that control the number of uplink messages, after which if the uplink transmission is not acknowledged, an ED increases either TP or SF. The tuning of these two parameters may affect the time to converge to a steady state.

The study of this mechanism has gain great attention amongst the research community [85–94]. It has been proven that the current ADR mechanism proposed in the LoRaWAN v1.1 specification lacks the agility to adapt to the changing link conditions [85], and can even do more harm than good since it does not take into consideration the distribution of SFs amongst users [94].

2.4 Band regulations and duty-cycle restrictions

As pointed out before, LoRaWAN Networks implement an Aloha-type MAC protocol, this means that contrary to other wireless networks, there is no congestion avoidance protocols (like CSMA-CA in 802.11) implemented. As a result, the traffic generated by EDs is regulated by imposing restrictions on the duty-cycle and on the maximum power emitted [84].

These restrictions are contained in specific European Telecommunications Standards Institute (ETSI) and Electronic Communications Committee (ECC) recommendations. LoRaWAN operates on Industrial, Scientific, and Medical (ISM) bands, in particular, in the 902 MHz band in the US and in the 863-870 MHz band in Europe [76, 84, 95]. Thus radios operating in these license-exempt bands are subject to regulations on radio emissions [84] and are required to either adopt a LBT policy or duty-cycle restrictions.

2.5 Conclusion

In this chapter, we have introduced LoRa and LoRaWAN, which is one of the leading LPWAN technologies. We describe how LoRa allows to have long-range transmissions through the use of CSS modulation technique. However, we also stressed the fact that using this type of technology makes the ToA much longer than in other technologies.

Then, we give a description of the LoRaWAN specification, which defines the way in which the medium is shared between Gateways and End Devices. We saw that it defines an Aloha-based protocol, in which packet transmission times are selected randomly. Additionally we have also described that there are three modes in which End Devices can be configured: Class A, for infrequent uplink transmissions, Class B for use cases requiring more downlink transmissions and, Class C for continuous listening EDs.

In the next section, we will review in detail how some aspects of the LoRa and LoRaWAN can be exploited to perform different of attacks. Special attention will be paid to Jamming attacks, which as indicated before is the main problem addressed in this thesis.

Chapter 3

Security and vulnerability analysis for LoRaWAN

In spite of the fact that the provision of security functionalities in network architectures have been studied for years and that the security requirements of IoT Networks are mainly the same as traditional networks, IoT networks are still facing some challenges due to its nature. As we have seen in Chapter 1 the main characteristic of IoT technologies is to use very constrained devices at the access network. As a result all the added functionalities, including the security ones, should be selected carefully, which means that already proven secure solutions for traditional networks might not be applied directly.

Therefore, in this chapter we present an analysis of the LoRaWAN security. First, we will describe the security rules of the two LoRaWAN versions. Then, an analysis of its vulnerabilities will be presented, and finally, we will define the threat model and attacker profiles that will be considered throughout this thesis.

3.1 Security features in LoRaWAN

Security procedures in LoRaWAN are implemented by using the IEEE 802.15.4-2006 Annex B. Hence it provides end-to-end data confidentiality (Payload encryption) and network level data integrity (Data authentication). Encryption of data is done by following the Advanced Encryption Standard (AES) 128-bit key in CTR (counter) mode and data authentication is provided by using the AES standard in block-Cipher-based MAC (CMAC) mode.

3.1.1 Security in LoRaWAN v1.0

Each LoRaWAN v1.0 node is personalised with a unique 128-bit AES key (AppKey) and a globally unique identifier (EUI-64 based DevEUI). Networks are also identified by a 24-bit globally unique identifier (NwkID) assigned by the LoRa Alliance. These parameters are used to handle security procedures.

The specification defines two ways of joining nodes to the network; Over the Air Activation (OTAA) and Activation by Personalization (ABP). The former defines a procedure to derive session keys based on a pre-stored key shared between the ED and the Application Server, and the latter assumes that session keys are stored on both sides before starting the communication,

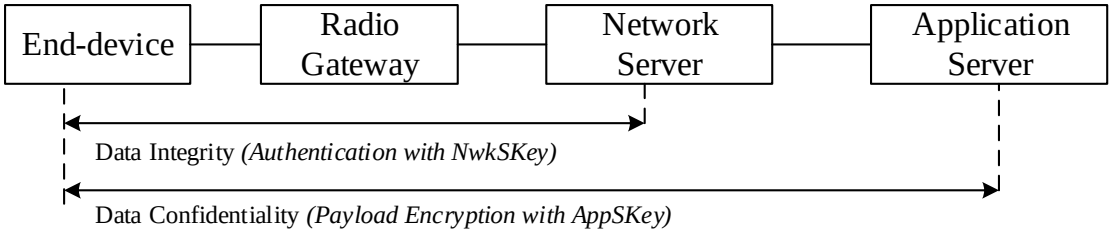


Figure 3.1: LoRaWAN 1.0 Security Approach

ABP devices use the same session keys throughout their lifetime.

EDs following the LoRaWAN v1.0 specification are personalised with a root key (Appkey) only known by themselves and the application server. Two session keys are derived, the AppSKey and the NwkSKey. The AppSkey is used to guarantee confidentiality between the end-device and the application server, and NwkSKey guarantees data integrity between the ED and the Network Server.

In this manner, when a frame is exchanged exclusively between an end-device and the Network Server, data confidentiality and data integrity are provided by the NwkSKey (no end-to-end data integrity is offered). Additionally, if an application payload is present, it is encrypted by using the AppSKey. This security approach is depicted in figure 3.1.

3.1.1.1 Join Procedure in LoRaWAN 1.0

When Over-The-Air-Activation is being used, EDs are required to follow a join procedure prior executing data exchanges with the NS. This procedure is performed each time a node joins the network, it is always performed by using Class A operational mode and may be performed several times if certain node lost session context information. EDs following the LoRaWAN v1.0 specification are personalised with a globally unique identifier (DevEUI), an application identifier (AppEUI), and an AES-128 key (AppKey).

From the ED's point of view, the join procedure consists of two MAC-level messages, a Join Request sent by the node to the Network Server and a join accept message sent by the NS to the ED. The join request message is a concatenation of the AppEUI, the DevEUI and a DevNonce, which is a 2-bytes pseudo-random number used only once [1]. The message is authenticated with a 4-byte Message Integrity Code (MIC) tag by using AES in CMAC mode with the AppKey and no encryption is performed.

Once the server has received the join-request message, it recalculates the MIC and, if valid, sends a join-accept message as depicted in figure 3.2. Thus, the join-accept message is a concatenation of the AppNonce, the NetID, the DevAddr, and some radio parameters. The join-accept message is authenticated with a 4-byte MIC tag by using AES in CMAC mode and encrypted

[1]The specifications (LoRaWAN v1.0 and 1.1) recommend that this nonce should be implemented as a 16-bit counter.

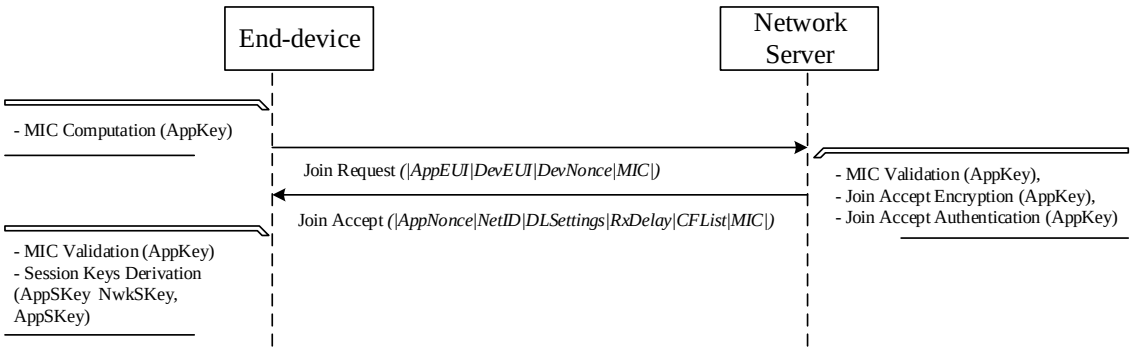


Figure 3.2: Join Procedure in LoRaWAN 1.0

by using AES in CTR mode, both by using the AppKey.

Upon receipt of the join-accept message, the ED calculates both, the NwkSKey and the AppSKey from the AppNonce, the NetID and the DevNonce by using the AppKey. From then on, if using Class A operation mode, all exchanges made between the end-device and the NS are encrypted and authenticated. Contrariwise, when Class B is being used, beacons are sent by Gateways which enable Nodes to synchronise their timing for downlink messages, and for Network Servers to know through which Gateway a particular Node can be contacted [96]. These Beacons are not encrypted nor authenticated. The join procedure is depicted in figure 3.2.

3.1.2 Security features in LoRaWAN v1.1

Some changes in security rules were incorporated in the LoRaWAN v1.1 specification. At the network architecture level, the specification formalises the possibility of having a Join Server, this server can be operated by a third trusted party and is in charge of deriving session keys. The Join server sends Network Session keys to the Network Server and the AppSKey to the Application Server.

EDs following the LoRaWAN v1.1 specification are personalised with two root keys (namely AppKey and NwkKey), a JoinEUI, that uniquely identifies the Join Server and a DevEUI, allowing to identify the end-device. Three session keys are derived from the NwkKey (namely, NwkSEncKey, SNwkSIntKey and FNwkSIntKey) and the AppSkey is derived from the AppKey.

Two additional lifetime keys are also derived from the NwkKey in LoRaWAN v1.1 end-devices. The JSIntKey and JSEncKey, used to authenticate and encrypt Rejoin-Request messages and Join-Accept answers triggered by a Rejoin-Request message.

Data integrity of MAC-level commands is provided by using two network session keys, the SNwkSIntKey and the FNwkSIntKey. Uplink frames are authenticated by using both keys (*each key is used to calculate 2 separate 16 bit MICs*) and downlink frames are authenticated by using SNwkIntKey.

Splitting the MIC in two parts is especially useful when roaming mode is activated. In such case, the end-device is physically attached to the fNS, which in turn serves as a relay to

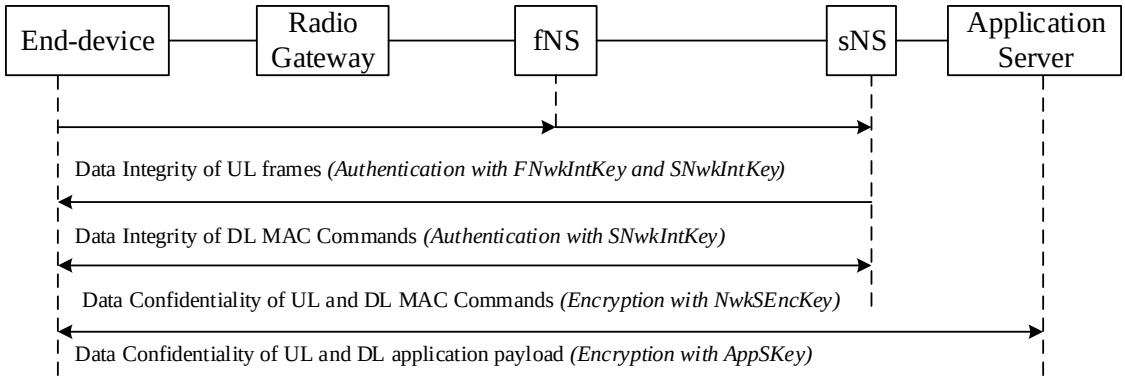


Figure 3.3: LoRaWAN 1.1 Security Approach

the serving Network Server (sNS). Thus, it is not desirable for the forwarding Network Server (fNS) to know the Session Key because it is a network element managed by a different operator, which can be compromised and therefore a source of replay attacks. As a result, data integrity is offered by FNwkSIntKey between the end-device and the fNS, and by the SNwkSIntKey between the end-device and the sNS.

Data confidentiality at the network level is provided by encrypting MAC-level Commands with the NwkEncKey, and at the application level by encrypting the Payload with the AppSkey. This security approach is depicted in figure 3.3.

3.1.2.1 Join Procedure in LoRaWAN 1.1

In the same manner as in LoRaWAN 1.0, end-devices are required to follow a join procedure prior to exchanging data with the network server. This procedure is performed each time an ED joins the network and is always executed by using Class A operational mode. It may be performed several times if some nodes lost session context information. From the ED's point of view, the join procedure consists of two MAC-layer messages, either a join or rejoin-request and a join-accept message.

The join-request message is a concatenation of the JoinEUI, the DevEUI and the DevNonce. The message is authenticated with a 4-bytes MIC tag by using AES in CMAC mode with the AppKey and not encryption is performed. If the end-device is allowed to join the network, the network server answers back with a join-accept message containing a server nonce (JoinNonce), a network identifier (NetID), an end-device address (DevAddr), a (DLSettings) field providing downlink parameters, and some radio parameters. The JoinNonce is a device specific counter value (that never repeats itself) provided by the Join Server and used by the end-device to derive the session keys FNwkSIntKey, SNwkSIntKey, NwkSEncKey and AppSKey. This procedure is depicted in detail in figure 3.4 (a).

Within the DLSettings, there is a field called OptNeg indicating whether the ED is LoRaWAN v1.1 compliant or not. If the OptNeg is unset, Network session keys are set equal

and derived from the NwkKey *(which acts as the former AppKey)*. The AppSKey is also derived from the NkwKey and the Join-accept message is authenticated and encrypted with the NwkKey. Else, if the OptNeg is set, the AppSKey is derived from AppKey and Network session keys are derived from the NwkKey. In this case, the join-accept message is authenticated with the JSIntKey, and encrypted either with the NwkKey or the JSEncKey. The NwkKey is used when the Join-accept message is triggered by a Join-Request message, and JSIntKey is used when the Join-accept message is triggered by a Rejoin-request message.

LoRaWAN 1.1 also includes the possibility of having an additional MAC-level message called Rejoin-request message. This message can be sent periodically by the ED on top of its normal application traffic. It is intended to be used by the back-end to initialise a new session context for the ED, which can be especially useful when the node is no longer reachable and there is a need of changing RX windows and for hand-over procedures.

There are three types of Rejoin-Request messages. Each one differs from the previous one depending on the information contained in the message. Rejoin-request message type 0 contains NetID+DevEUI and are used to reset a device context. Rejoin-request message type 1 and 2 contains JoinEUI+DevEUI. The former is used to restore a lost session context and includes the same information as a regular Join Request message but can be sent on top of a regular application payload, and the letter is used to rekey a device or change its DevAddr. The join procedure for re-join request messages is depicted in detail in figure 3.4 (b).

3.2 LoRaWAN Vulnerabilities

As we have seen in the previous section, the LoRaWAN specification includes certain security level. However, several works in the literature have revealed vulnerabilities and risks regarding the security of LoRaWAN v1.0 [97–104]. As a result the LoRaWAN Alliance introduced several improvements in the LoRaWAN v1.1. These changes deal with the inclusion of a new Join Server that is in charge of managing the OTAA procedure so that security keys do not have to be shared when roaming. These changes makes the LoRaWAN networks reasonably resistant against its integrity and confidentiality.

In spite of these changes, there is no way a network is fully shielded from all type attacks. Hence, in this section, the main security challenges, which can lead to security gaps in LoRaWAN will be discussed. We will also present several attacks in LoRaWAN networks that have been reported in the literature.

In section 1.5 we analysed several security threats that apply to IoT networks as a hole. There, we saw that there are several surveys and Proof-of-Concept papers [31,62,63,63–69,105] that have discussed the different security threats of IoT and more specifically LPWAN Networks. In this section we will be interested in the particularities of LoRaWAN and how they make it more vulnerable to certain types of attacks than other IoT technologies. Therefore, we can classify these vulnerabilities according to the following classification:

i) Attacks that exploit the weaknesses of the specification.

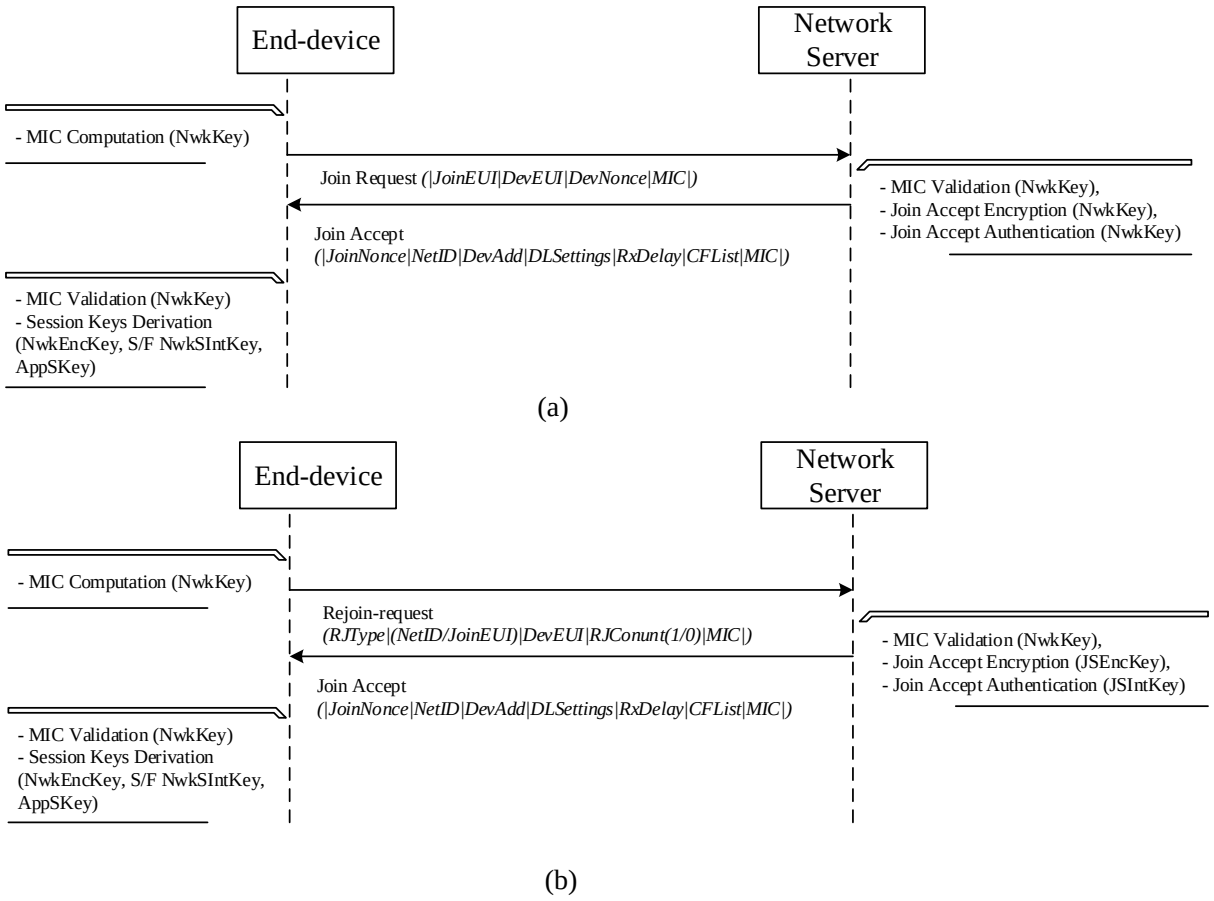


Figure 3.4: Join Procedure in LoRaWAN 1.1, (a) Join procedure from scratch, (b) Join procedure triggered by a rejoin-request message.

ii) Attacks that exploit the weakness of the nature of wireless communication

3.2.1 Attacks that exploit the weaknesses of the specification

Although the LoRaWAN v1.1 introduces several changes that makes the authentication procedure more secure, some authors suggest that there is still a possibility of breaking the implementation of the Encryption algorithm [98, 101] if it is not well implemented.

As we have seen, LoRaWAN implements the AES algorithm in counter mode. The security of this algorithm relies on the no repetition of the nonce. Therefore, values must be encrypted with a different keystream[2]; if this condition is not assured, the confidentiality is no longer guaranteed [106]. Thus, in LoRaWAN when the packet or frame counter is reset (counter overflow, or when the physical device is reset in bad implementations), while the session key is not changed, the block cipher recreates the same key values. Then, if an attacker is able to capture several packets encrypted with the same key values, it is possible to compromise the encryption [101,106]. In such a case, as explained in detail in [106], the attacker only needs two packets to obtain the encrypted plaintext (application payload in the LoRaWAN case).

[2]A *keystream* is a stream of random or pseudo random characters that are combined with a plaintext message to produce an encrypted message (the ciphertext) [106].

Additionally, as we have seen before, join request messages are not encrypted when sent in the OTAA procedure. This is because, at that moment, sessions keys haven't been exchanged. Therefore, all the information carried by a Joint-request message are sent in clear. These two security breaches have been used to perform MITM attacks (replay) and confidentiality attacks.

3.2.1.1 Replay attack

As we have seen in Section 1.5, in a replay attack valid data transmission is maliciously or fraudulently repeated or delayed [102]. In LoRaWAN, an attacker takes advantages of non encrypted Join request messages. This MITM attack works as follows:

1. The attacker stores several packets from the GWs (ACK) or EDs (Join request).
2. It selectively jams given ED.
3. It replays previously recorded ACK to disrupt uplink messages.

3.2.1.2 Confidentiality attack

A confidentiality attack consists in having access to data that is supposed to be hidden. It uses the counter overflow security breach. This attack works similarly to the replay attack [102,105]:

1. The attacker stores several packets from the EDs (regular uplink messages).
2. It analyses the recorded packets, and as suggested in [106] it performs XOR operations of consecutive packets.
3. If lucky, the attacker is able to have access to intelligible information, which corresponds to the payload the packet carries.

Another Confidentiality attack can be performed in Class B EDs. As we have seen in the previous section, Beacons are not encrypted nor signed, and therefore represent both a source of information and a route to inject malicious data in to a system. An attacker can go even further by creating and transmitting their own Beacon packets. And, since EDs have no way of knowing whether the beacons it receives are malicious or genuine, they can be easily manipulated by jammers, what may result in messages from EDs not being received by the Network Servers as EDs lost synchronisation [96].

Therefore, to avoid these kind of attacks, different thinks should be considered when implementing security procedures at the ED side:

- Use a reliable random number generators in order not to reuse root keys.
- Do not use the ABP join method.
- Use advanced storage techniques to store root keys (*secure elements*).
- Use persistent memory in order to avoid repeating nonce values (Application, Device, Network).
- Use trusted third party Join Servers in order not to expose root key in the clear.

3.2.2 Attacks that exploit the weakness of the nature of wireless communication

In this category fall all the attacks that exploit the broadcast nature of radio frequency. Therefore, the attacker uses modulation techniques to either make a passive attack in which it obtains information from the network, or occupies the same spectrum of the attacked network with the sole objective of decreasing its availability.

In the context of LoRaWAN, these attacks are particularly harmful due to the nature of the LoRa modulation. As we have seen LoRa uses SFs to spread the information across the frequency band. Hence, the higher the SF, the lower the data-rates and in turn, the longer the communication range. However, it also increases the ToA which as shown in table 2.1 it can be in the order of 7 seconds, something that is huge when compared to other IoT technologies [107]. This single feature makes the LoRaWAN networks specially susceptible to Denial of Service (DoS) and MITM attacks.

3.2.2.1 Jamming attacks

Although jamming attacks are not unique to LoRaWAN, the challenge is to distinguish packet losses due to network congestion or deliberate injection of packets to create interference. This is because according to the specification, a LoRaWAN compliant ED must use an Aloha-type access protocol. Consequently, transmissions times are done at random times. This randomness makes it hard to detect these attacks.

In addition to the detection, another challenge in LoRaWAN is that there is no mechanism to sense the channel before transmitting. This makes the mitigation even harder specially in dense deployment scenarios. Another problem with this kind of attacks is that, as we will see in the following chapters, they have a huge impact on the overall network performance. Jamming attacks can be classified into two categories according to their capability of sensing the medium: *channel-aware* and *channel-oblivious*.

3.2.2.2 Channel-aware

This kind of jammer has the ability to sense the medium. Then, based on that it launches an attack based on previously defined criteria [108]. As depicted in figure 3.5 and explained in detail in Section 3.1, encryption is only performed for the application payload. Hence, an attacker is able to read the preamble, the physical the header (PHDR), the MAC header (MHDR), and even the application-level header (FHDR). Thus, it has access to specific fields: `MType`, which contains information related to the type of message the packet is carrying on (Join Request, Join Accept, Unconfirmed data UP, etc), and the device address `DevAddr`, which identifies each node (EDs and GWs) in the network. This security breach has been used to perform three channel-aware attacks: (i) triggered jamming, (ii) selective jamming due to long ToA, and (iii) selective jamming along with a wormhole attack.

The **triggered jamming** attack consists in detecting any activity in the channel and then launches an attack without taking the time to read the information on the packets. For that,

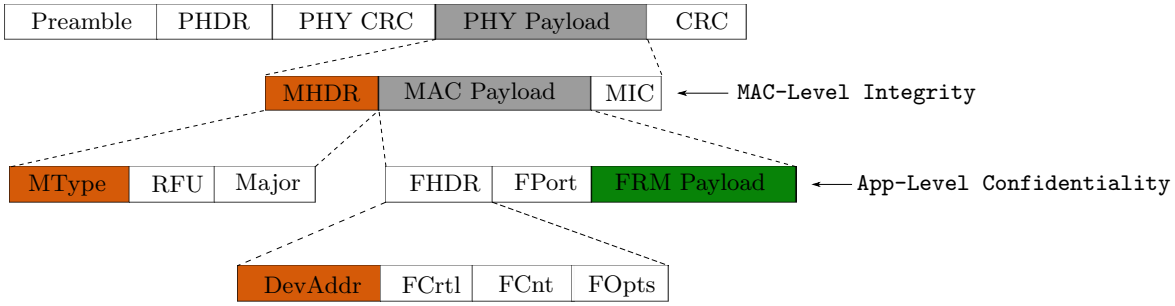


Figure 3.5: LoRaWAN Frame.

it uses the the CAD functionality. A graphical representation of how it works is presented in figure 3.6 (a). It basically consists in a regular LoRa module that has been tampered to perform an always listening routine. Hence, each time CAD detects something, it transmits an unauthenticated packet on the medium by using the same transmission parameters (SF an channel).

Contrary to the triggered, in a **selective jamming attack due to long ToA**, the attacker does not use the CAD procedure, instead it listens to the channel and decodes the packet until the first bytes of the frame. Once it gets there it verifies if the content match the jamming policy, and then starts the jamming transmission. The policy is either a set of `DevAddrs` or an specific type of message `MType`. A typical example of this attack is to jam only the Join-accept messages, in that way, EDs are denied access to the network.

A more elaborated version of the selective jamming, is the **selective jamming along with a wormhole attack**. In this attack, two devices are involved and they act as sniffer and jammer separately. The sniffer is in charge of receive and store messages from the network. Then, it decides whether or not to jam, then it signals to the jammer using a low latency network (usually a wired one) to jam the message immediately. However, this attack is limited to the reaction time of each specific jammer hardware. For example, in [104] the authors were able to successfully perform this kind of attacks only for $SF \geq 9$ and with packets length greater than 37 `bytes`.

These three jamming attacks were first reported and implemented in [103, 104]. They used the RFM95 radio module together with an Arduino based micro-controller. This selection of hardware is made because, as we will see in detail in Section 6.4.4.2, it allows communications via SPI interfaces, which makes it possible to stop receiving data before finishing the packet and start jamming immediately.

3.2.2.3 Channel-oblivious

The channel-oblivious jammer, also known in the literature as *collision attack* is the simplest attack to do, but in some cases the most difficult to detect. The rationale of the attack is to create interference in the RF medium in a premeditated way. Interference occurs then when signals simultaneously collide in time, frequency, and SF.

Thus, as in the other three cases, the jammer is usually a regular LoRa module, but in

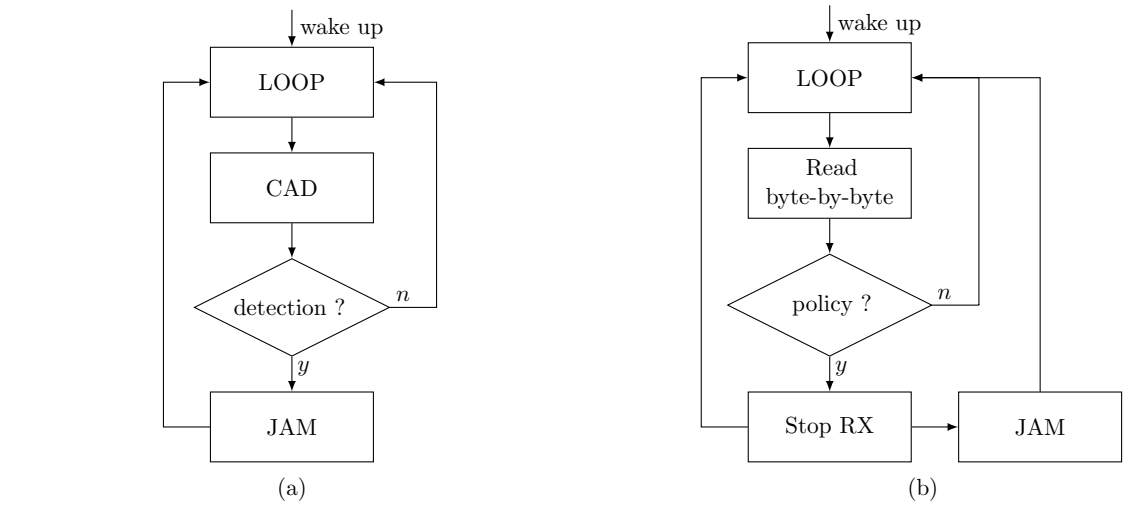


Figure 3.6: Channel-aware jamming in LoRaWAN: (a) triggered jamming, (b) selective jamming due to long ToA.

this case it acts similarly to a regular ED. It transmits by following a random approach, where transmission times, SF and channel are selected randomly.

As we will see later, in this thesis we developed an analytical model that models an steady-state network that is being under the attack of channel-oblivious jammers, and we also present a simulation environment and a test-bed that allows to simulate and evaluate the selective jamming attacks here mentioned.

3.3 Threat model and attacker profiles considered in this thesis

For our threat model, we consider a network architecture as the one presented in figure 3.7. As we can see, the network is composed of: multiple EDs, which are considered to be legitimate nodes since they meet the LoRaWAN specification and band restrictions, a legitimate Gateway that handles the radio channel with EDs, a network server that is in charge of handling the joint procedure and an Application Server.

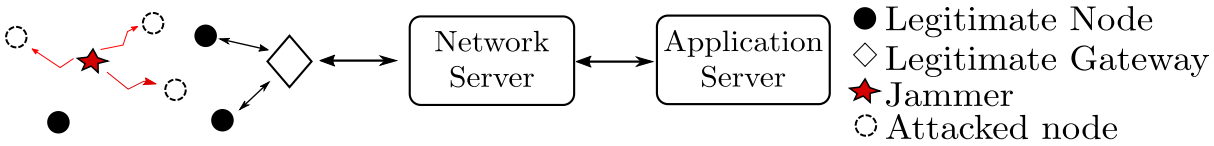


Figure 3.7: Threat Model

We assume that the attack takes place at the access network. More specifically, we will only address attacks that exploit the weakness of the nature of wireless communication. In this regard, we will consider two types of attack: (i) channel-oblivious jammer and (ii) channel aware jammer. Hence, we consider that jammers do not a belong to the network, and that from

the hardware point of view they have the same characteristics as regular EDs. This includes the same transmission power and the same hardware limitations.

Therefore, as detailed in 3.2.2.3 **channel-oblivious jammers** do not listen to the channel, and they transmit randomly on the same bands, channels and SFs as legitimate nodes. Therefore, we assume that each channel-oblivious jammer has a packet arrival times that follows a Poisson distribution.

On the contrary, for **channel-aware jammers** we model them as detailed in section 3.2.2.2. More specifically, we only consider the **triggered jammer**, which is able to detect ongoing transmissions and to transmit unauthenticated packets by using the same transmission parameters (SF and channel).

In regards to the energy source of EDs and jammers we consider two approaches depending on the type of evaluation we perform:

1. Network-level performance metrics: this correspond to the performance evaluation presented from section 5.2 to section 5.6 where we consider a steady state network. Therefore, EDs and Jammers are considered to have a continuous energy source.

2. Current consumption and battery lifetime: for this scenario presented in section 5.7, we consider that EDs are powered with a 2400 mAh battery, while jammers have a continuous current supply.

3.4 Conclusion

In this chapter an analysis of the LoRaWAN and its security have been provided. First, in section 3.1 we have reviewed the way LoRaWAN handles security by comparing the two versions of the LoRAWAN specification. Hence, we have detailed the authentication and security rules of both versions.

Then, in section 3.2 an analysis of the LoRaWAN vulnerabilities was provided. We have seen that various works have shown that these networks have various vulnerabilities related to the definition of the specification itself and its wireless nature. First, we saw that despite offering end-to-end confidentiality and network level authentication, there is still a possibility of performing replay attacks and even confidentiality attacks. Then, we have seen that the LoRa modulation makes the ToA to be longer, which can be used by attackers to perform different jamming attacks seeking to compromise the availability of the network.

And finally, in section 3.3 we specify the type of attacks that we will consider throughout the thesis and give a first definition of their capabilities in the context of the different types of attacks that exist.

In the next chapter, we will introduce a novel analytical model and a simulation environment that allows us to evaluate the performance impact of those attacks.

Chapter 4

Network Modelling

As we have seen in the previous chapter, LoRaWAN is particularly vulnerable to jamming DoS attacks. Despite this, this problem has been little studied in literature. To contribute filling this gap, in this chapter we provide two tools that can serve to establish the impact on the performance of this type of attacks. Therefore, in this chapter we will provide a mathematical model and a simulation environment to estimate the performance of a LoRaWAN network under those attacks.

Additionally, as previously described in chapter 2 the specification also considers confirmed traffic, which enables control applications and reliable services. Although this feature is not widely used in commercial deployments, we decided to include it in both the mathematical model and in the simulation environment.

For the mathematical model we consider several simplifications and hypothesis that allow to have accurate estimations for low traffic loads. Thus, we only consider the case of channel-oblivious jammers and a steady state network deployment. On the contrary, the simulation approach provides a more complete environment which considers the dimension of time, channel-aware jammers and different particularities of an actual deployment of a LoRaWAN network such as hardware limitations and the capture effect.

4.1 Analytical model

In the following, we introduce an analytical model that aims at characterising the behaviour of a LoRaWAN-like network with confirmed traffic under jamming. We have decided call it LoRaWAN-like since, as we will see in detail, we do not incorporate different features such as the use of two reception windows or the ADR algorithm.

We will first present a review of previous works on LoRaWAN analytical models. Then, the reference scenario and assumptions are described. Latter, we will explain the packet scheduling and re-transmission scheme considered. Then, we introduce the performance metrics used and finally the model is described.

Table 4.1: LoRaWAN mathematical models comparison

Ref.	Model	ISF	CE	ET	CT	PR	2RX	1RX	JM
[109]	Sorensen	✓	✗	✓	✗	✗	✗	✗	✗
[110]	Bankov et al.	✓	✓	✓	✓	✓	✓	✗	✗
[112]	Paul	✓	✗	✓	✓	✓	✗	✗	✗
[11]	Capuzzo et al.	✓	✓	✓	✓	✓	✓	✓	✗
–	Ours	✓	✗	✓	✓	✓	✗	✓	✓

4.1.1 State of the art

Several analytical models for LoRaWAN have been proposed in the literature. In [109] the performance of LoRaWAN uplink transmissions is investigated. They derived expressions for the latency, collision rate, and throughput. Exponential packet inter-arrival times and SFs orthogonality were assumed, no capture effect was considered, and duty-cycle limitations were taken into consideration.

In [110], the authors provide a LoRaWAN model considering re-transmissions and capture effect. They assumed SFs orthogonality, Poisson packet generation, and systematic transmissions of two ACK packets, one on each reception window. They also provide an Modulation and Coding Scheme (MCS) selection approach that satisfies user Quality of Service (QoS) requirements.

In [111], the authors provide a model for bi-directional LoRaWAN networks for a single GW, that considers multiple reception paths at the GW, duty-cycle restrictions at both EDs and GW, and priority ACK transmissions. As well as the previous models, exponential packet inter-arrival times and SFs orthogonality were assumed, and no capture effect was considered.

In [112], the author provides a mathematical model to evaluate the energy consumption impact of packet re-transmissions in LoRaWAN. For that, mathematical formulas for the packet loss probability as well as the power consumption as a function of the SF and the number of re-transmissions are developed.

Table 4.1 reports the most relevant work in LoRaWAN modeling. We classify these works depending on the assumptions considered. We consider 8 different characteristics: Imperfect SF Orthogonality (ISF), Capture Effect (CE), Exponential traffic (ET), Confirmed traffic (CT), Packet re-transmissions (PR), two reception windows (2RX), only one reception window (1RX) and jamming (JM).

As for mathematical modeling of LoRaWAN under jamming, this is to the best of our acknowledge the first study on that matter. Hence, we expand the previous studies by proposing an analytical model that, besides the capture effect, and the inclusion of two reception windows, accurately models a LoRaWAN-like network with confirmed traffic under channel-oblivious jamming.

4.1.2 Scenario and assumptions

For the analytical model, we consider a LoRaWAN-like network composed of one GW located at the center of a disc of radius R, and N_u users. The network operates with F frequency uplink channels and one downlink channel, each channel has a bandwidth of 125 kHz (it uses the g3 band as described before in 2.1).

As in LoRaWAN, the LoRa modulation is used. Hence, we assume that the network operates with orthogonal SF $s \in \mathcal{S} = \{7, ..., 12\}$. Thus, packets sent with different SF can be distinguished and received simultaneously. Each packet is characterised by its ToA T_u^s, which is a function of its SF s and packet length l_u [`bytes`] (see [78]).

We assume that each user has a packet generation rate of λ_u packets per second, and that all user packets have the same length of l_u bytes. Transmission attempts occur according to a Poisson distribution. Then, given a duty-cycle limitation d_u, the packet generation rate for each user verifies $d_u = \lambda_u \cdot T_u^s$ with $d_u = 0.01$ in practice.

The LoRaWAN-like network is put under attack of $\{N_{ju}, N_{ja}\}$ jammers that jam either user or ACK packets by transmitting on the same band and SF. Similarly to users, they transmit following a Poisson distribution. Thus, we assume that each jammer transmits packets with length $\{l_{ju}, l_{ja}\}$, ToA $\{T_{ju}^s, T_{ja}^s\}$ and packet generation rate of $\{\lambda_{ju}^s, \lambda_{ja}^s\}$. Yet, unlike users the duty-cycle $\{d_{ju}, d_{ja}\}$ is not limited to 0.01.

4.1.3 Packet Scheduling

As depicted in figure. 4.1, in the LoRaWAN-like network, each uplink transmission is followed by a short reception window on a dedicated downlink channel. Therefore, we denote by T_{ack}^s the ToA of an ACK packet. The SF used is the same as that for the uplink transmissions[1].

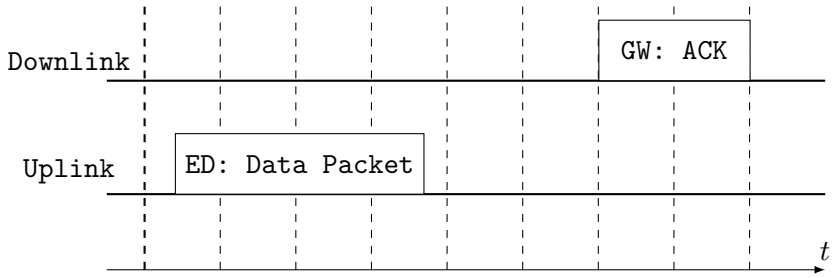


Figure 4.1: Packet Scheduling with ACK and user data packets on a different channel but identical SF

4.1.4 Re-transmissions

The acknowledgment scheme of the LoRaWAN-like network is presented in figure 4.2, we assume that each successful uplink transmission is acknowledged systematically by the GW. This scheme aims at mitigating information lost due to packet collisions. It works as follows: the user sends a

[1]Note that these two simplifying assumptions are taken for tractability, yet our model can be used to assess the performance impact of a jamming in LoRaWAN compliant networks. A comparison with numeric simulations of LoRaWAN in ns3 will be presented later.

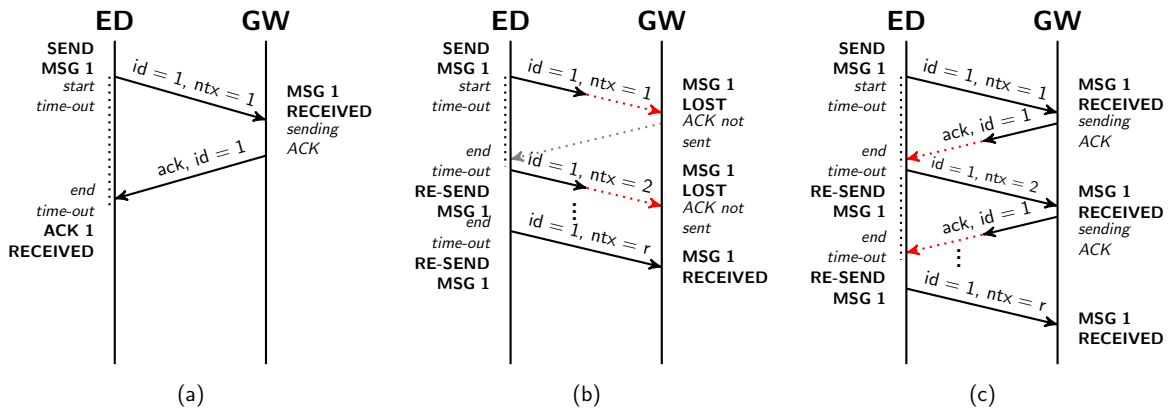


Figure 4.2: Acknowledgment Scheme: (a) message and ACK are well received, (b) message is lost, (c) message is well received but ACK is lost.

message (a packet identified with a unique id), then if it reaches the gateway, it is acknowledged by the means of an ACK transmission.

If the ACK does not reach the user before a certain time-out T_o, the ED makes a re-transmission ($ntx + 1$). Transmission times of message re-transmissions are set according to the same Poisson point process as messages. The timeout countdown is reset after uplink transmission. It is set as the time required by the GW to detect a user packet plus the time required by the ED to detect an ACK. The message is re-transmitted a maximum of r times. Under this scheme, three cases may arise:

- the message and ACK packets are well received,
- the message is lost due to collisions with another user/jamming packet. Then, the message is resent,
- the message is well received but ACK is lost due to jamming. Hence, the message is re-transmitted.

It should be noted that a message can lead to multiple packet transmissions depending on network congestion. Additionally, since we only consider one GW, there is no possibility of collisions between ACK packets. However, as discussed later, since F channels are used for uplink and only one for downlink, ACK transmissions are subject to gateway saturation.

The parameter (λ_u) and duty-cycle (d_u) restriction are respected regardless the number of re-transmissions, and no exponential back-off mechanism is considered.

Finally, it is worth mentioning that, contrary to LoRaWAN there is no possibility of having collisions between user packets and ACKs since both are sent in different frequency channels.

4.1.5 Performance metrics

Performance is evaluated in terms of four metrics:

- User throughput ($\overline{\mathcal{G}_u}$), defined as the sum of the user packet data rates that are effectively delivered in the network.
- User goodput ($\mathcal{O}_u$), defined as the sum of the user message data rates that are effectively delivered.
- Expected number of re-transmissions per message ($\mathcal{E}(r)$).
- Message Success Probability ($\mathcal{P}_m$), defined as the probability of successfully delivering a given message.

In the following, we develop our model for the uplink and downlink channels, as well as expressions for the user goodput ($\mathcal{O}_u$) and the expected number of re-transmissions $\mathcal{E}(r)$.

4.1.6 Uplink model

We assume F channels for uplink transmissions. Let ρ_u^s be the ratio of users employing SF s. We can write the packet arrival rate on a given channel for SF s per user as follows:

$$r_u^s = \frac{\rho_u^s \cdot \lambda_u}{F} \tag{4.1}$$

Let G_u^s be the normalised user traffic load, which is the aggregated packet generation rate on a given SF s:

$$G_u^s = r_u^s \cdot N_u \cdot T_u^s \tag{4.2}$$

Similarly, for jammers transmitting on the uplink channels, we can define the normalised traffic load as

$$G_{ju}^s = r_{ju}^s \cdot N_{ju} \cdot T_{ju}^s \tag{4.3}$$

where r_{ju}^s is the packet arrival rate per uplink jammer on a given channel, and SF s defined as $\rho_{ju}^s \cdot \lambda_{ju}/F$.

Hence, as both packet arrival times (user and jammer) occur according to a Poisson distribution process, we can consider that the resulting process is the superposition of the processes which is again a Poisson process but with a packet arrival rate of $G_u^s + G_{ju}^s$. Thus, we can write the normalised traffic load G_{up}^s per uplink channel and SF s as follows:

$$G_{up}^s = G_u^s + G_{ju}^s \tag{4.4}$$

Let $\mathcal{P}_u^s$ be the user packet success probability per SF s, which is the probability of no user packet being collided by any other packet transmitted on the same SF. This probability is the joint probability of two independent events: no user packet being collided by another user packet (P_{uu}^s) and no user packet being collided by any jammer packet (P_{uj}^s):

$$\mathcal{P}_u^s = P_{uu}^s \cdot P_{uj}^s \tag{4.5}$$

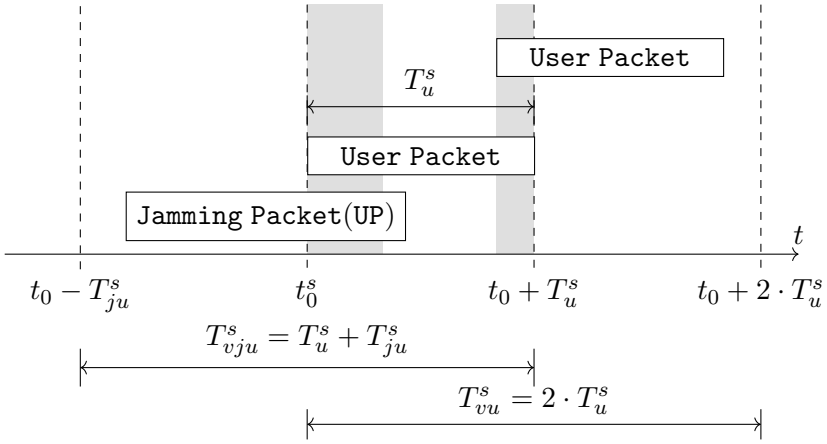


Figure 4.3: Vulnerability periods for an user packet

Let $T_{vu}^s = 2 \cdot T_u^s$ and $T_{vju}^s = T_u^s + T_{ju}^s$ be the vulnerability periods in which an arbitrary user packet can be overlapped by any other (user and/or jammer) packet during its transmission as depicted in figure. 4.3. Thus, as derived in [113] for classical Pure Aloha networks, P_{uu}^s and P_{uj}^s can be defined as:

$$\begin{aligned} P_{uu}^s &= \exp(-r_u^s \cdot N_u \cdot T_{vu}^s) \\ P_{uj}^s &= \exp(-r_{ju}^s \cdot N_{ju} \cdot T_{vju}^s) \end{aligned} \tag{4.6}$$

Finally, the user throughput[2] can be written as:

$$\overline{\mathcal{G}_u} = \sum_{s=1}^{S} G_u^s \cdot \mathcal{P}_u^s \tag{4.7}$$

4.1.7 Downlink channel model

As described in Section 4.1.4, the generation of ACK are subject to the successful reception of uplink packets by the GW, on any frequency channel. Therefore, the rate of uplink packets that are successfully received at the GW per SF s is:

$$r_{succ}^s = F \cdot r_u^s \cdot P_u^s \cdot N_u = \rho_u^s \cdot \lambda_u \cdot P_u^s \cdot N_u \tag{4.8}$$

Now, let G_{ack}^s be traffic load corresponding to ACK packets effectively transmitted in response to successfully uplink transmissions with SF s[3] :

$$G_{ack}^s = r_{succ}^s \cdot T_{ack}^s \cdot Q_{TX}^s \tag{4.9}$$

where Q_{TX}^s is the probability that the GW is not already transmitting an ACK packet on a

[2]Note that for the case where $T_u^s = T_{ju}^s$, the user throughput becomes $\sum_{s=1}^{S} G_u^s \cdot \exp(-2 \cdot G_{up}^s)$.

[3]As done in [114] for analytical tractability, we approximate all transmission processes as Poisson processes. This assumption may not hold in general since the memory-less assumption is not guaranteed, but as presented later in the validation Section, it provides a good approximation of Q_{TX}^s.

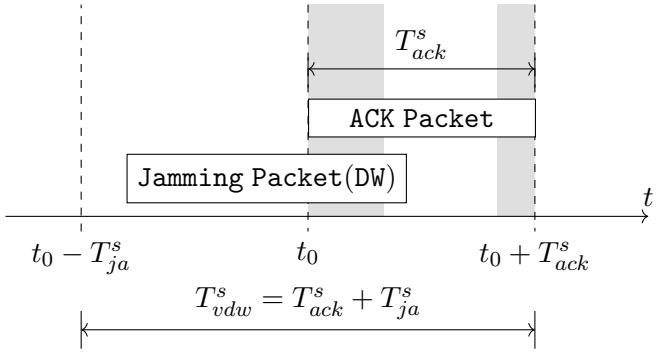


Figure 4.4: Vulnerability period for an ACK packet

given SF and is developed in Section 4.1.8.

Let $\mathcal{P}^s_A$ be probability that a user packet is correctly acknowledged given that it was successfully received by the GW on a given SF s. This probability can be defined as:

$$\mathcal{P}^s_A = Q^s_{TX} \cdot Q^s_{RX} \tag{4.10}$$

where Q^s_{RX} is the probability of successfully reception of an ACK packet given that it was transmitted. Since our model only considers one GW, we can assume that ACK packets do not collide with one another. Hence, this probability is the probability of no ACKs packet collisions with jammer packets.

Thus, considering there is a set of N_{ja} jammers that jam ACK packets with transmission attempts occurring according to a Poisson distribution with packet arrival rate $r_{ja} = \rho_{ja} \cdot \lambda_{ja}$, we can define the traffic load of jammers transmitting on the downlink channel (G_{ja}) as:

$$G_{ja} = \lambda_{ja} \cdot N_{ja} \cdot T_{ja} \tag{4.11}$$

Then, as previously done, we can consider that the resulting process is the superposition of the two Poisson processes, which is also a Poisson process. Thus, we can define the normalised downlink channel traffic (G_{dw}) as:

$$G^s_{dw} = G^s_{ack} + G^s_{ja} \tag{4.12}$$

Therefore, as presented in figure. 4.4 and similarly to the uplink case the vulnerability period for an ACK packet can be defined as $T^s_{vdw} = T^s_{ack} + T^s_{ja}$. Thus, Q^s_{RX} is:

$$Q^s_{RX} = \exp\left(-N_{ja} \cdot r^s_{ja} \cdot T^s_{vdw}\right) \tag{4.13}$$

4.1.8 Gateway transmission model

In the following we develop an equation for Q^s_{TX}, since we consider that ACK packets are sent systematically after each successful uplink transmission. This probability can be expressed

as a function of the rate of successfully received traffic r^s_{succ}, the ToA of ACK packets T^s_{ack} and the number of uplink channels F.

We assume that the GW is able to transmit S orthogonal packets in parallel on a sole downlink frequency channel without duty-cycle limitation. Consequently, there is a probability that the GW may be locked in the instant it is required to send another ACK, which results in the non-transmission of the ACK. Hence, Q^s_{TX} can be defined as the probability of having at least one ACK packets generated in the last T^s_{ack} seconds, which is the probability of no ACK packet generated plus the probability that the GW is asked to generate i ACK packets $\forall i \in [2, ..., F]$.

$$\begin{aligned} Q^s_{TX} &= \Pr[\texttt{no ACK generated}] \\ &+ \Pr[\texttt{GW asked to generate 2 ACK}] \\ &\vdots \\ &+ \Pr[\texttt{GW asked to generate F ACK}] \\ &= \exp\left(-r^s_{succ} \cdot T^s_{ack}\right) \\ &+ \exp\left(-r^s_{succ} \cdot T^s_{ack}\right) \cdot \sum_{i=2}^{F} \frac{(r^s_{succ} \cdot T^s_{ack})^i}{i\,!} \end{aligned} \tag{4.14}$$

4.1.9 Goodput and re-transmissions

We denote by P^s_x the probability that a user packet sent with SF s is successfully received and acknowledged at the x-th transmission attempt as:

$$P^s_x = \sum_{a=0}^{x} (\mathcal{P}^s_u)^{x-a+1} \cdot \mathcal{P}^s_A (1 - \mathcal{P}^s_u)^a \cdot (1 - \mathcal{P}^s_A)^{x-a} \tag{4.15}$$

The average number of times that a message is re-transmitted per SF s can be written as:

$$\mathcal{E}^s(r) = \sum_{x=1}^{r} x \cdot P^s_x + r \cdot \left(1 - \sum_{x=1}^{r} P^s_x\right) \tag{4.16}$$

Thus, the expected number of re-transmissions of any message sent by users in the network is:

$$\mathcal{E}(r) = \sum_{x=1}^{S} \rho^s_u \cdot \mathcal{E}^s(r) \tag{4.17}$$

Let $\mathcal{P}^s_m$ be the message success probability, defined as the probability of having a packet well received after, at most r re-transmissions:

$$\mathcal{P}^s_m = 1 - (1 - \mathcal{P}_u)^{r+1} \tag{4.18}$$

Thus, the message success probability of any message is given by:

$$\mathcal{P}_m = \sum_{s=1}^{S} \rho_u^s \cdot \mathcal{P}_m^s \tag{4.19}$$

This allows us to define the user goodput ($\mathcal{O}_u$) as:

$$\mathcal{O}_u = \sum_{s=1}^{S} \frac{G_u^s}{\mathcal{E}^s(r)+1} \cdot \mathcal{P}_m^s \tag{4.20}$$

4.1.10 Model Validation

In order to provide a first validation of the model, we launch a simulation on ns-3[4]. Therefore, as in the analytical model, the simulator considers systematic acknowledgment of all uplink transmissions with a maximum number of re-transmissions r. For downlink transmissions, we model a bank of FIFO (First In First Out) buffers (one for each SF) with only one packet storage capacity. Hence, if the GW does not handle to transmit given ACK under the time constraint (TRX1 expired), the next ACK will overwrite the data.

We model a network consisting of one GW, and we vary different parameters in order to evaluate its performance. The network operates in the 868 MHz frequency band with three uplink and one downlink channel for transmission ($F = 3$). Users employs six SF ranging from 7 to 12, with a uniform distribution ($\rho_u^s = 1/6\ \forall s$). The user duty-cycle is $d_u = 0.01$, all user packets have a length of $l_u = 50$ `bytes`, while ACK are sent with a length of $l_{ack} = 10$ `bytes`.

The results obtained are presented in figure 4.5. The dotted line represents the analytical results[5], while the firm line the ns-3 simulation results. The model follows closely simulation results for lower user traffic loads, yet it is slightly pessimistic for higher traffic loads. The figure shows the total ACK success probability. We can see that P_A is inversely proportional to the throughput achieved in the network. Indeed, it decreases rapidly as the number of successfully received packet increases. This is because the GW gets saturated and cannot transmit ACK at the required rate. Once, the peak of successfully received user packets is reached, the number of collisions on the uplink channels grows, and consequently the probability of ACK packet effectively transmitted increases.

[4] a detailed description of the simulation environment is presented in Section 4.2

[5] The results presented consider the total ACK success probability ($\mathcal{P}_A$), which correspond to $\mathcal{P}_A^s$ weighted by the proportion of users transmitting in a specific SF (ρ_u^s). Since we consider the case of no jamming, this probability is : $\mathcal{P}_A = \sum_{s=1}^{S} Q_{TX}^s \cdot \rho_u^s$.

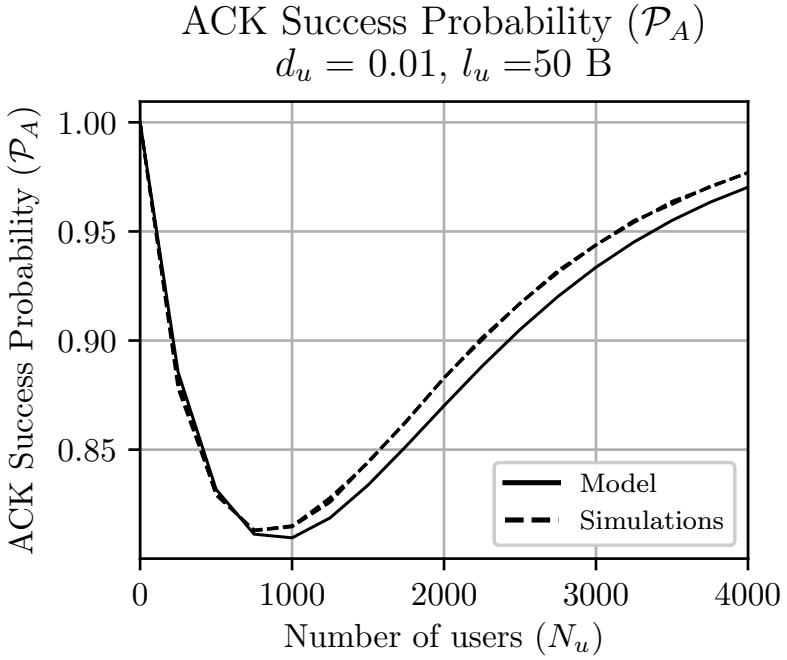


Figure 4.5: ACK success probability ($\mathcal{P}_\mathcal{A}$) of a LoRaWAN-like network without jamming

4.2 Event driven simulator

In the following we introduce an open source event driven simulator that aims at characterising the behavior of a LoRaWAN network at the physical, MAC and Application levels with both channel-oblivious and channel-aware jammers [17]. The module allows modelling several characteristics of the LoRaWAN Protocol and has the flexibility of changing some parameters in order to validate the mathematical model described in Section 4.1.

In the previous section, we have introduced an analytical model that allows modeling Channel-oblivious jammers. This model is very helpful for developers and network operators aiming at having faster and accurate results for lower traffic loads. In this section, we take a step forward and propose a simulation environment that allows to evaluate the network performance of LoRaWAN under jamming. Therefore, unlike the analytical model, we consider different particularities that are present in a regular network deployment. Thus, for our model we consider:

i) An accurate radio propagation model based on previous works.

ii) The capture effect.

iii) Imperfect spreading factor orthogonality.

iv) An application model that allows having random and deterministic traffic.

v) A current consumption model for EDs.

vi) A re-transmission system.

vii) Multiple Gateways.

viii) Selective jamming.

ix) Channel-oblivious jamming.

In the following we will describe in detail the simulation environment in ns3 we use. Section 4.2.1 presents a revision of the previous works, then, Section 4.2 describes the different assumptions we took for modelling LoRaWAN under jamming, and finally in 4.2.3 an overall description of the ns3 module implementation is presented.

4.2.1 Previous works on simulation tools for LoRaWAN

Several simulation tools have been proposed for LoRaWAN in the literature. Table 4.2 compares the different simulators available based on the different LoRaWAN characteristics described in Section 2.1 they allow to simulate. We consider 12 different characteristics: Programming language, OS (Open source), Imperfect SF Orthogonality (ISF), Capture Effect (CE), Exponential traffic (ET), Periodic Traffic (PT), Confirmed traffic (CT), Duty-cycle (DC), Energy Consumption (EC), Adaptive Data Rate (ADR), Multi Gateway (MG), Jamming (JM). We also compare the tools based on the way the work has been validated (Val): comparison with a real implementation (RE), comparison with a mathematical model (MM) or comparison with previous simulation tools (ST).

Table 4.2: Comparison of LoRaWAN simulators

Ref.	Model	Language	OS	ISF	CE	ET	PT	CT	DC	EC	ADR	MG	JM	Val.
[115, 116]	LoRaSim	Python	✓	✗	✓	✓	✗	✗	✗	✓	✗	✓	✗	RE
[117]	LoRaWANSim	Python	✓	✗	✓	✓	✗	✓	✗	✓	✓	✓	✗	ST
[94, 118]	Ta	Python	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	–
[119]	FloRa	Omnet++	✓	✗	✓	✓	✗	✓	✓	✓	✓	✓	✗	–
[120, 121]	Haxhibeqiri	ns-3	✓	✓	✓	✓	✗	✓	✓	✗	✗	✓	✗	RE
[122]	Croce	USRP / Matlab	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	RE
[123]	Centenaro	Matlab	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓	✗	MM
[124]	Callebaut	Python	✓	✗	✓	✓	✓	✓	✓	✓	✓	✗	✗	RE
[125]	Yousuf	Java	✓	✓	✓	✓	✓	✗	✓	✗	✗	✓	✗	RE
[126]	Kouvelas	ns-3	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	ST
[127]	Reynders	ns-3	✓	✗	✓	✗	✓	✓	✓	✓	✓	✓	✗	–
[128]	To	ns-3	✓	✗	✓	✓	✗	✗	✓	✓	✗	✗	✗	RE
[129]	Khan	ns-3	✓	✗	✓	✗	✓	✓	✓	✗	✗	✗	✗	–
[10–13]	Magrin	ns-3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	MM
[130, 131]	Martinez (Ours)	ns-3	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	MM

One of the first and widely used is the LoRaSim coded with Python and presented in [115, 116]. It is open source and provides useful framework to assess the performance of LoRaWAN However, it does not implement message acknowledgment.

In [117], an extension of the LoRaSim is presented. It adds support for downlink transmissions and studies its impact on the network scalability. They concluded that the achievable network capacity reported in earlier studies is quite optimistic. Another extension of LoRasim is presented in [132]. It adds support for multiple concurrent IoT applications.

Another Python simulator for LoRaWAN is the one presented in [94, 118]. It is a an event-based simulator for LoRaWAN that takes into account the capture effect and inter-spreading factor interference. The authors also provide a resource allocation algorithm aiming at fairly distributing the radio resources amongst EDs.

Similarly, FLoRa ,an Omnet++ implementation, has been proposed in [119]. It implements the physical and medium access control layers of LoRa, supports bi-directional communications and allows end-to-end simulations including the backhaul network. It also incorporates an Adaptive Data Rate (ADR) scheme that dynamically manages link parameters for scalable and efficient network operations.

In [120, 121] the impact of interference on LoRaWAN scalability is studied. The authors developed an ns3 module based on real transmission measurements showing that the impact of the interference relies on the section of the frame that is affected.

In [122] the impact of LoRa Imperfect Orthogonality is studied. The authors developed an open source simulator in MATLAB that quantifies the co-channel rejection, including the impact of PHY coding. The results were validated with experimental measurements with LoRa transceivers and USRP experiments. As it will be presented after, we use the results of this research to model the capture-effect in our simulation approach in ns-3.

Another MATLAB simulator is the one proposed in [123] where the LoRaWAN MAC layer is addressed. The authors studied confirmed traffic for a scenario with multiple GW and multiple SFs. The simulation results showed that the performance of LoRa is severely impacted if the fraction of end devices of confirmed-data type grows.

In [124] a simulator framework for the energy consumption of LoRaWAN nodes is presented. The authors proposed a physical and MAC-layer open-source simulator coded in Python. It allows to assess the energy budget of LoRaWAN nodes considering actual measurements of the power states and some particularities of the LoRaWAN specification such as the ADR and downlink messages.

In [125] an open source simulator coded in Java is proposed. It allows to study the LoRaWAN Specification for the North American bands. It is suitable to study the scalability of LoRaWAN since it allows to estimates the capacity of a single LoRaWAN GW. The work is part of a measurement campaign that considered a city-wide LoRa deployment aimed at characterising the throughput and coverage of LoRaWAN.

In [127] a module for ns-3 is proposed. It models the LoRaWAN v1.0 class A specification, It is built to study the effect of different parameters on the network performance, including a flexible backbone architecture allowing easy integration of new protocols and distributed gateways.

Another ns-3 module is presented in [128, 133]. There, the authors evaluate Carrier sense multiple access (CSMA) and compared to the Aloha-type LoRaWAN channel access. The work was validated with actual measurements and by comparing it with previous works. The authors concluded that for higher loads, CSMA shows an improvement with respect to LoRaWAN in terms of the packet data rate achieved in the network.

Similarly in [126] the authors propose to use a persistent CSMA (p-CSMA) protocol on

the MAC layer of LoRaWAN. The proposal is implemented in ns3 by extending the module proposed in [10–13]. The authors concluded that impementing such a mechanism improves the packet reception ratio achieved in the network.

In [129] an extension of the LR-WPAN (Low-Rate Wireless Personal Area Network - IEEE 802.15.4) ns-3 module [134] is presented. It extends PHY/MAC layers to support LoRaWAN. The author evaluated the network performance for different scenarios including device mobility and Poisson traffic pattern traffics.

Finally, in [10–13] a LoRaWAN simulator is presented, they developed a detailed ns-3 module that allows to accurately assess the performance of LoRaWAN considering multiple parameters such as bi-directional traffic, duty-cycle restriction for gateways and End-Devices and confirmed traffic. As it will be presented after, we extended this ns-3 module for LoRaWAN networks in the presence of jamming.

4.2.2 Simulation assumptions

This section presents the different assumptions and simplifications made during the construction of the simulation environment. As pointed out before, we extended this ns-3 module presented in [10–13]. Hence, the general structure of the model was preserved, however several implementations details were changed in order to make simulation of attacks possible.

4.2.2.1 Radio propagation model

For the radio propagation model, we follow the results obtained in [7], where extensive measurement campaigns were performed across the city of Beirut. They concluded that for indoor deployments, the model that best fits the radio propagation is a mixture of the ITU-R P.1238 [135] and the Cost-231 MWF [136], while for outdoor deployments a linear polynomial fit was derived. Therefore, both models where considered in our simulation environment. The path loss (PL) model developed for outdoors deployments is:

$$PL = 10n \log_{10}(d/d_0) + PL_0 + L_h \log_{10}(h_{ED}) + X_\sigma \tag{4.21}$$

where, d is the distance between the GW and the ED, PL_0 is the PL at a reference distance d_0, L_h is the additional loss due to the ED antenna height h_{ED}, and shadow fading is represented by a zero-mean Gaussian random variable X_σ with standard deviation σ (in dB).

As for indoor deployments, the PL is:

$$PL = 10n \log_{10}(d/d_0) + PL_0 + n_w L_w + n_f^{\left[\frac{n_f+2}{nf+1}-b\right]} L_f \tag{4.22}$$

where n_w and n_f are the number of walls and floors, respectively. b is taken equal to 0.47 to obtain a suitable fit. L_f and L_w represent the loss factor of floors and walls, respectively. All floors and walls are assumed identical in this model.

The fitting values are presented in Table 4.3

Table 4.3: Fitting parameters for the PL models

Metric	Indoor	Outdoor	Urban	Rural
PL exponent(n)	2.851	3.12	4.179	3.033
PL intercept(PL_0)	120.4	140.7	102.86	111.75
Shadow fading (σ)	8 dB	9.7 dB	7.2 dB	6.4 dB
Wall/Floor Loss (Lw/Lf)	10/1.412	–	–	–
ED height loss (L_h)	–	-4.7	-6.3	-6.65

4.2.2.2 Interference model

As done in the original version of the simulator, we consider a standalone LoRaWAN network. Hence, interference is done only between LoRa transmissions. Thus, interference is modeled as done in [1, 116] by using the following Signal-to-interference-plus-noise ratio (SINR) threshold matrix:

$$\mathbf{T} = \begin{bmatrix} 6 & -16 & -18 & -19 & -19 & -20 \\ -26 & 6 & -20 & -22 & -22 & -22 \\ -27 & -27 & 6 & -23 & -25 & -25 \\ -30 & -30 & -30 & 6 & -26 & -28 \\ -33 & -33 & -33 & -33 & 6 & -29 \\ -36 & -36 & -36 & -36 & -36 & 6 \end{bmatrix} \tag{4.23}$$

The element $T_{i,j}$ is then the SINR threshold (in dB) needed to decode a packet transmitted with SF i if the interfering packet has an SF j. Therefore, the $SINR_{i,j}$ is defined as:

$$SINR_{i,j} = \frac{P_{x,i}}{\sum_{l \in \mathcal{I}_j} P_{x,l} + \sigma^2}$$

where $P_{x,i}$ is the power of the packet under consideration, $P_{x,l}$ is the power of the $l-th$ interfering packet, $\mathcal{I}_j$ is the set of interfering packets using the spreading factor j, $\sum_{l \in \mathcal{I}_j} P_{x,l}$ is the aggregated interference power (from EDs and/or jammers) on a given Spreading Factor j and, σ^2 is the white channel power. Therefore, we consider that a particular packet with $s = i$ is correctly decoded if the following inequality holds:

$$SINR_{i,j}^{dB} > T_{i,j} \tag{4.24}$$

It shall be noted that the elements in matrix $\mathbf{T}$ are calculated assuming that the two packets are perfectly overlapping [1]. However, in the general case, packets are not perfectly synchronised. Because of this, we must equalise the interfering power value for the computation of the SINR. Therefore, as done in the original version of the simulator: *"we assume that the interfering energy for any reciprocal position of the signal and an interferer can be spread out*

on the signal in order to then compute the SINR using Equation 4.24" [10]. Then, the power of the interfering packet l can be expressed as:

$$P_{x,l}^{int} = \frac{P_{x,l} * t_{ol}}{T_x} \tag{4.25}$$

where, t_{ol} is the period of time during which the interferer is overlapping and T_x is the transmission time of the packet that is under consideration.

This assumption is justified by the fact that there is an underlying channel code employed by the modulation. We can assume that a good interleaver will spread it out and allow the channel code to eventually correct the errors caused by the interferer. Moreover, thanks to the channel coding technique used by the LoRa modulation, we also assume that we will always correctly receive a packet that is above sensitivity and survived interference [10].

It shall be noted that, as described in detail in Appendix A, the previously presented approximation of the SINR could be improved if one considers that:

- Goursaud's matrix [1] is made for the case in which the signal has not been de-spreaded (i.e. before decoding with the corresponding spreading factor). Moreover, it is built in such a way that an arbitrary threshold of at least 6dB is guaranteed after decoding the signal at the receiver side.

- Magrin's [10] and our approach estimate the interference based on the SINR level for each spreading factor without considering the de-spreading phase. A more precise computation of the SINR shall consider the power levels before and after the de-spreading phase taking into account the 6dB threshold.

- More recent research on this matter have concluded that there is no such thing as *pseudo-orthogonal* spreading factors on LoRa. Indeed, in [16] a SIR matrix was built considering pairwise comparisons from actual measurements showing that the impact of interfering transmissions from different spreading factors is not negligible.

Additionally, the module also supports simulation of the network as in an pure Aloha mode. Under this situation, all transmissions that overlap in time, frequency and SF are destroyed. We included this in order to validate our analytical model as described in Section 4.1.

4.2.2.3 End Device Model

For the ED, we implement the state machine presented in figure 4.6. The Physical layer switch from one state to another based on application and MAC-level instructions. Hence, we consider that EDs are only capable of either transmitting or receiving at a certain moment of time. Additionally for bi-directional simulations, the ED sensitivity is modelled as the Semtech SX1272 and summarised in Table 4.4.

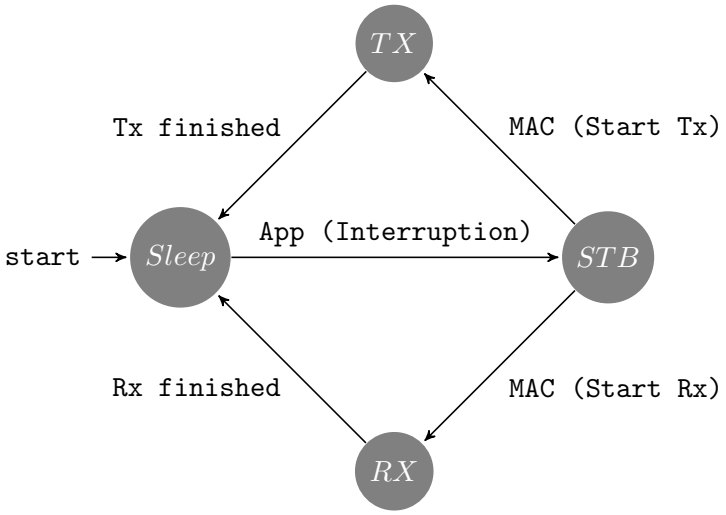


Figure 4.6: Finite state machine for the End Device physical layer

Table 4.4: LoRaWAN devices Sensitivities for 125 Khz

Device	SF7 [dBm]	SF8 [dBm]	SF9 [dBm]	SF10 [dBm]	SF11 [dBm]	SF12 [dBm]	Ref.
GW	-130.0	-132.5	-135.0	-137.5	-140.0	-142.5	[79]
ED	-124	-127	-130	-133	-135	-137	[81]

4.2.2.4 Current consumption

Generally, the consumed energy at the ED stems from two sources: RF signals generation at the transceiver, which mostly depends on the number of generated bits in a packet. The second source is due to electronic components, i.e filters, amplifiers, sensors *etc.* Therefore, in order to estimate the current consumption of an ED, and therefore the potential impact of a jamming attack on it, a current consumption model of a LoRaWAN transceiver is proposed. The model is based on the finite state machine depicted in figure 4.6 and takes into account the current consumption per state.

Table 4.5 present typical consumption levels for different LoRa nodes as stateted in it respective data-sheets. For our simulator, the current consumption levels associated to each state are presented in Table 4.6, and are based on real measurements. We consider the Semtech SX1272 transceiver with a transmission power of 14 dBm and a modulation scheme a coding rate of 4/5.

Based on these current consumption levels, we compute the average current consumption per sent packet (I_{avg}) during a given session:

$$I_{avg} = \frac{1}{T_{app}} \sum_{i=1}^{N_{states}} T_i \cdot I_i \tag{4.26}$$

where T_{app} is the application time in seconds, defined as the time between two consecutive

Table 4.5: Current Consumption of LoRaWAN nodes.

Transceiver	Device Name	Current consumption			Ref.
		Sleep	Transmit	Receive	
Semtech SX1272	MultiConnect® mDot™	40 μA	26 - 42 mA	-	[137]
Semtech SX1272	NetBlocks XRange	70 μA	109 mA	20 mA	[14]
Microchip RN2483	Microchip RN2483 LoRa mote	3.4 mA	47.5 - 47.9 mA	17.2 mA	[138]
Semtech SX1276	LoRaBug	3.7 mA	120.7 mA	16.6 mA	[139]
HopeRF HM-TRLR-LF/HFS	iLoad	7.66 μA	133.3 mA	16.3 mA	[140]

Table 4.6: Current Consumption for different device states

State	Current consumption (I_i) [mA]
Tx	83
Rx	32
Standby (STB)	32
Sleep	45×10^{-3}

packet transmissions, N_{states} is the number of states through which an ED passes to transmit a packet, T_i is time duration spent in state i and I_i is the current consumption in mA/s in state i.

To estimate the lifetime of a battery-operated ED, we considered a linear function of the battery capacity ($C_{battery}$) in mAh:

$$T_{lifetime} = \frac{C_{battery}}{I_{Avg}} \tag{4.27}$$

4.2.2.5 Gateway

For the Gateway, we leave the same methodology followed in the original simulator [10]. Hence, we simulated the SX1301 LoRa Gateway [79] that is capable of having 8 parallel receivers (reception path), have the sensitivities as in Table 4.4 and have the following characteristics:

- Each reception path is centered in a unique Frequency channel: three reception paths in 868.1, three in 868.3 and two in 868.5 MHz
- All reception paths are capable or receiving transmissions from any SF, even if multiple packets have the same SF.
- If a given packet arrives and all reception paths are occupied, it is then considered to be dropped.
- The gateway sensitivity varies according to the SF as states in the data-sheet and summarised in Table 4.4.

Additionally, as in the case of the interference model, we add support to Aloha-pure operational mode. In such case, the Gateway is capable of decoding up to 18 packets in parallel (6 spreading factors and three channels).

4.2.2.6 Channel-oblivious jammer

The channel oblivious jammer, is simulated as a regular ED. It can be configured to select packet transmission times randomly or deterministically. Hence, different parameters can be configured:

- Traffic load per jammer (λ_{ju} or λ_{ja}), if it is configured to have random transmission times.
- Packet length.
- Packet transmission period, if it is configured to have deterministic transmission times.
- Random initial delay, if it is configured to have deterministic transmission times.
- Frequency channel and SF.

4.2.2.7 Channel-aware jammer

The channel-aware jammer implemented in the simulator is the triggered jamming presented in Section 3.2.2.2. The jammer listens to one of the sub-bands and once it decodes the first bits of the preamble, it sends back a packet on the same channel and SF. The attack timing, as shown in figure. 4.7, is defined in Equation (4.28):

$$T_{jam} = T_{dpre} + T_{sw} + T_{tx} \tag{4.28}$$

where T_{dpre} is the time necessary to detect a frame (firsts bytes of the preamble), T_{sw} the switching time, which is the time required to switch from reception to transmission state, and T_{tx} the duration of jamming. The response time of the attack is the sum of T_{dpre} and T_{sw}. T_{dpre} is modeled by using Equation (2.6) as presented previously in Section 2.2.

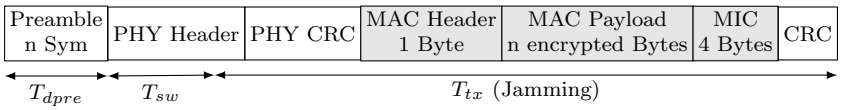


Figure 4.7: Channel-aware jammer timing

4.2.2.8 Packet re-transmissions

In order to evaluate the impact that including re-transmissions have on LoRaWAN when it is being attacked by jammers, and to validate our mathematical model, we added support for such a mechanism in the simulator.

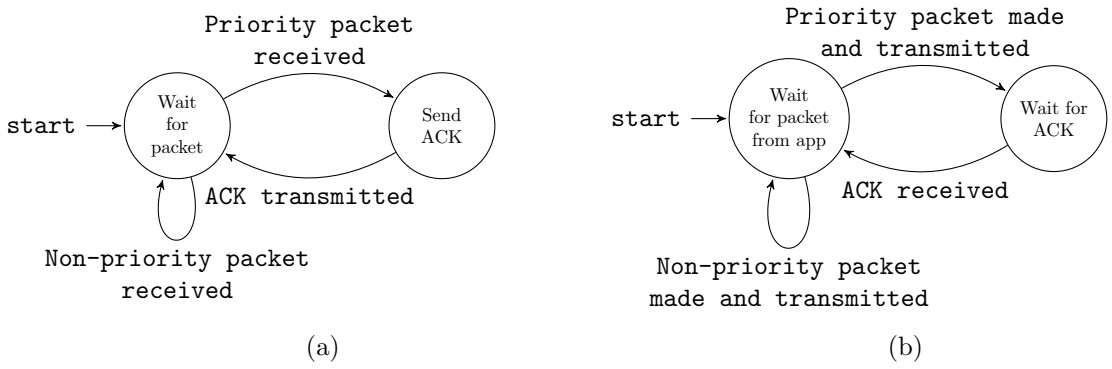


Figure 4.8: Finite state machine for the re-transmissions scheme: (a) NS side, (b) ED side.

Contrary to the mathematical model, in a simulation we can put aside the hypothesis in which all the packet are acknowledge systematically. Thus, in the simulator we add the possibility of having priority packets. In this way, there is a proportion of the traffic that is acknowledged and may be re-transmitted. The scheme is implemented at the MAC-level, for that we use the `MType` field. Therefore, each uplink transmission of a priority packet is set as `Confirmed Data Up`. Figure 4.8 depicts the way the system works at the NS and ED side. Under this scheme, different cases may arise:

- a non-priority packet is well received at the NS. Then, no ACK transmission is done, nor re-transmission,
- a non-priority packet is lost due to collisions with another user / jamming packet. Then, no ACK transmission is done, nor re-transmission,
- a priority packet is well received at the NS, then, an ACK transmission is done and the ACK is well received by the user. Hence, no re-transmission is performed,
- a priority packet is lost due to collisions with another user / jamming packet. Then, no ACK transmission is done. Hence, the packet is re-transmitted,
- a priority packet is well received, but ACK is lost due to jamming. Hence, the packet is re-transmitted.

Additionally, downlink ACK packets may be configured to be handled as in the LoRaWAN specification as shown in figure 2.3, where two reception windows are used or as in the mathematical model shown in figure 4.2, with only one reception window.

It should be noted that, we consider that each time a priority packet is received by the NS, it selects only one GW to acknowledges it. Hence, there is no possibility of collisions between ACK packets coming from the GWs, but collisions may occur between ACK and packets from ED / jammers.

Then, as done in the mathematical model, we assume that ED's transmission times follow a Poisson process. Thus, we consider that each packet re-transmission falls within the same

process. In other words, re-transmissions are handled in a way that no extra user traffic is re-injected into the network due to the re-transmission system.

4.2.3 Implementation details

In this Section, we present a detailed description of the LoRaWAN ns-3 module. We start by giving an overall description of the suimulation environment used, which is ns3 (network simulation 3), then we present a unified modeling language (UML) schema where we present all the relevant classes for the LoRaWAN implementation. And finally, we give a description of the main classes and helpers.

4.2.3.1 Network simulator 3

The network simulator 3 (ns-3) is a C++ open source simulator used primarily by researchers. It is licensed under the GNU General Public License (GPL). The vast majority of modules added to the simulator are the result of peer review from researchers from all over the world. The main goal of ns-3 is to develop an open simulation environment for networking research. It includes modules that are well documented, easy to use and debug for the majority of network standards and specifications such as `LTE`, `WiFi`, and `6lowpan`. The ns-3 simulator also includes a strong environment that facilitates realtime network data collection, simulation configuration, trace collection and data analysis [141].

Furthermore, based on the literature review presented above, we conclude the most complete and accurate simulator is the one proposed by Magrin et al. in [10–13]. Thus, we reuse a significant amount of the code within the ns-3 framework already done and validated. We extended this module in order to add support for jammer nodes that do not meet the LoRaWAN standard.

4.2.3.2 Overall description

The ns-3 module presented in [10–13] models LoRaWAN by creating a collection of classes that work together to model LoRa EDs, GW and Network server at various levels. It includes the PHY, the MAC and the Application level for each device type.

Figure 4.9 depicts the main classes and connections of ns3 module. Some classes model the protocol stack on a given device while others are used to model other aspects like the propagation model or the electric current consumption. In the following, the implementation details of the module are presented. We start from the application layer, then the MAC layer, the physical layer and finally some additional modules.

4.2.3.3 Application Layer

The application layer is modeled by means of three classes: (i) `AppJammer`, (ii) `PeriodicSender` and (iii) `SimpleNetworkServer`. The three of them inheriting methods from `ns3::Application`. The main role of these classes is to generate the application-level packets and select its transmission times for the three types of devices: ED, GW and Network Server.

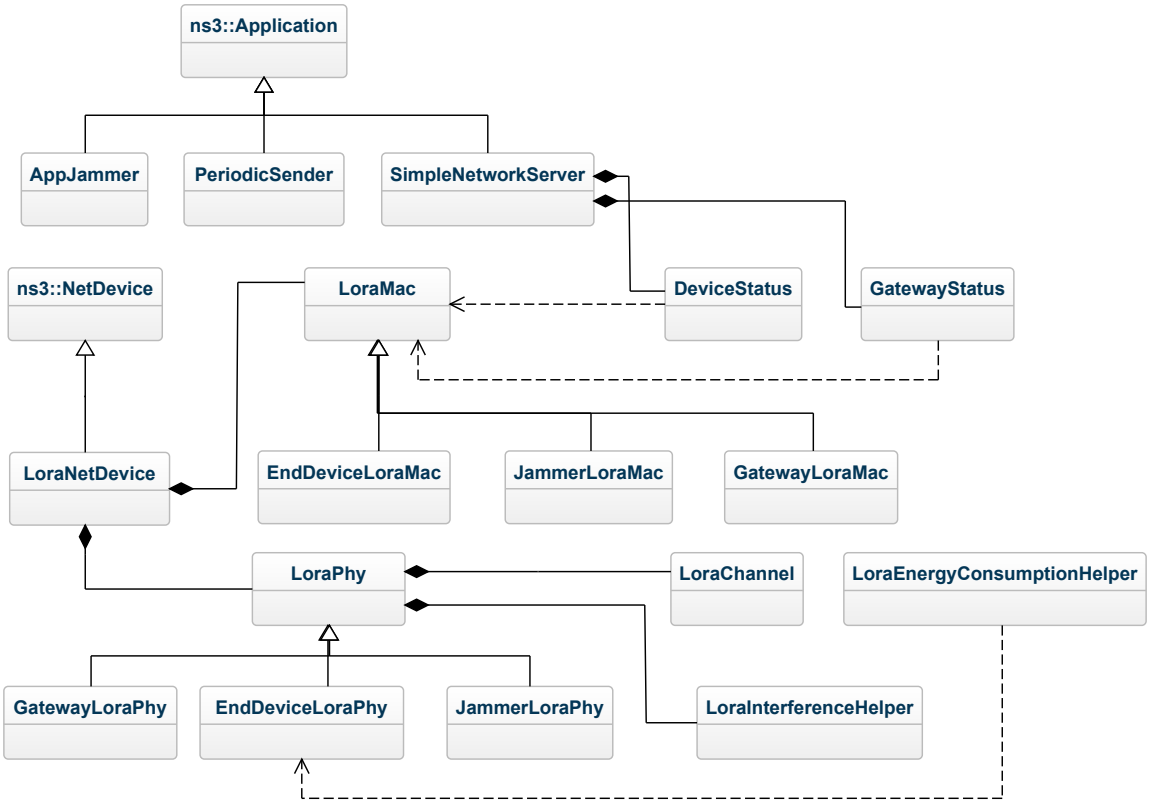


Figure 4.9: UML Schema of the ns-3 LoRaWAN module.

PeriodicSender This class models the behaviour of the ED application. It can be configured for three different operational modes: (i) periodic generation, (ii) exponential distribution and (iii) confirmed traffic:

1. Periodic generation: it creates zero-filled packets with a predefined payload size. Transmission times are selected deterministically with a given time period. A random period of time is defined before the transmission begins.
2. Exponential distribution: it creates zero-filled packets with a predefined payload size. Transmission times are selected randomly from an exponential distribution with a given mean defined as a multiple of the time on air (as previously defined in Section 4.1).
3. Confirmed traffic: it creates zero-filled packets with a predefined payload size. Transmission times are selected identically as in the exponential distribution mode. Packets are labelled as first transmission or re-transmitted based on information from the MAC level (as previously defined in Section 4.1 for *messages and packets*)

AppJammer This class is in charge of creating the packet sent by the channel-oblivious jammer. It reuses most of the code developed previously in the `PeriodicSender` class. It can be configured either in the exponential distribution mode or deterministically. The channel-aware jammer is modeled only on the MAC and PHY level as it will be presented latter.

SimpleNetworkServer This class is in charge of defining the transmission times of ACK packet based on GW occupancy and the transmission policy of the simulation. It can either follow the LoRaWAN specification as it is conceived or it can be configured to select some transmission parameters. Two modes can be used:

1. LoRaWAN : it sends two ACK packets, the first one on the same frequency and SF as the uplink transmission and the second one in a different frequency band on the SF 12.
2. OneRX : it only sends one ACK to acknowledge uplink transmissions, the transmission band and frequency can be freely configured.

In addition to handling ACK transmission the Simple Network Server class keeps track of different performance metrics that can be used to extract performance indicators from each packet arrival: arrival time, SINR, IAT, SF and the frequency band. This data is stored in traces for post treatment and to implement the real-time EWMA jamming detection algorithm presented latter in chapter 6.4.

4.2.3.4 MAC layer

For the MAC layer, the `LoraMac` class implements the LoRaWAN protocol. It is responsible of handling the available network channels of each device. Hence, it receives the messages from the application layer and then decides whether or not to send the packet according to the policy implemented. There are three objects in this class: `EndDeviceLoraMac`, `GatewayLoraMac` and `JammerLoraMac`.

LoraMac This class implements the basic methods used by all device types at the MAC layer. It is responsible of enforcing the duty-cycle restrictions of the band. It uses the `LogicalLoraChannelHelper` object, it keeps track of the available waiting time for each band and ED. Hence, it prevents packets from being transmitted if it breaks the duty-cycle restriction. The behaviour of each type of device is handled in three sub-classes: `EndDeviceLoraMac`, `GatewayLoraMac` and `JammerLoraMac`.

EndDeviceLoraMac This object receives the interruption aiming at sending a new packet from the application layer, and acts according to the transmission policy. Hence, each time a packet is needed to be transmitted, it checks whether or not the packet is allowed to be transmitted based on the duty-cycle and packet length limitations. Then, it wakes up the radio and opens the receptions window in the required channel and spreading factor.

This class also receives information from the application layer related to re-transmissions. Hence, it keeps track of all packet statuses and decides whether or not a packet is labeled as a packet re-transmission. Four main procedures are defined in this class:

1. `Send`: As presented in Algorithm 1 it defines the procedure of taking a packet from the application layer and pass it to the PHY layer.

2. `GetChannelForTx`: It randomly selects a free channel to transmit a given packet and is presented in 2.
3. `TxFinished`: As presented in Algorithm 3 it opens the receive window(s) based on the transmission policy and on the transmission outcome once the transmission is finished at the PHY Layer.
4. `PacketTrack`: It keeps track of the packet IDs and labels each packet transmission.

Algorithm 1 `Send` in `EndDeviceLoraMac`

```
1: procedure SEND(packet)        ▷Sending a packet (from the App layer) to the PHY layer
   global: m_state                                                  ▷PHY State of the device
2:     sent ← true                                  ▷Boolean variable that indicates if a packet is sent
3:     size ← GETSIZE(packet)                                                ▷Get packet size
4:     t ← GETMAXWAITINGTIME()        ▷Get channel waiting time from DutyCycleHelper
5:     if t != 0 or m_state = RX or size < MaxSize then
6:         sent ← false
7:         return sent                                  ▷The packet cannot be sent in this try
8:     end if
9:     txChannel ← GETCHANNELFORTX(packet)                                ▷Ask for a free channel
10:    if txChannel then                                  ▷Check if a free channel was granted
11:        id ← GETPACKETID(packet)                                          ▷Get the packet ID
12:        ntx ← GETNTX(packet)                                ▷Get the number of re-transmissions
13:        retx ← GETRETX(packet)                            ▷Ask if the packet is a re-transmission
14:        TxParams ← GETTXPARAMS()                          ▷Tx parameters <SF, CR, BW,...>
15:        packet ←ADDMACHEADER ()
16:        SWITCHTOSTANDBY ()                          ▷Switch to standby at EndDeviceLoraPhy
17:        SEND(packet, TxParams)                        ▷Send Procedure at EndDeviceLoraPhy
18:        PACKETTRACK(id, ntx, retx)                       ▷Adding packet to the packet tracker
19:        sent ← true
20:    end if
21:    return sent                                                        ▷End of the procedure
22: end procedure
```

JammerLoraMac This class defines the MAC layer of a jammer. It is based on the `EndDeviceLoRaMac` but the duty-cycle limitation was removed. This class also supports the channel-oblivious mode. In this mode a given jammer is able to listen the medium and then jam accordingly. Hence, contrary to the `EndDeviceLoRaMac` a jammer is capable of choosing whether or not to:

1. Follow the regulated parameters of the corresponding sub-band (DC, packet length and Tx power)
2. Open receive windows.
3. Sense the medium in order to detect packets coming from other EDs and GWs.
4. Send back unauthenticated packets if a regular packet is detected.

Algorithm 2 `GetChannelForTx` in `EndDeviceLoraMac`

```
1: procedure GetChannelForTx(packet)          ▷Procedure to get a free channel
2:     C ← [1, ..., N]                        ▷Array of all Channels
3:     Shuffle(C)                             ▷Randomly Shuffling the channel vector
4:     for i ← 1 to N do
5:         t ← GetWaitingTime(C[i])           ▷Waiting time from DutyCycleHelper
6:         if t = 0 then
7:             return C[i]                    ▷Returning the available channel
8:         end if
9:     end for
10:    return ∅                               ▷No channel available
11: end procedure
```

Algorithm 3 `TxFinished` in `EndDeviceLoraMac`

```
1: procedure GetChannelForTx(packet)          ▷Procedure trigged from the PHY Level
2:     global: DR_1, DR_2, F_1, F_2           ▷Data Rate and Frequency of both receive windows
3:     global: TWO_RX                         ▷Boolean variable for two receive windows
4:     OpenFirstReceiveWindow(DR_1, F_1)
5:     if TWO_RX = false then
6:         OpenSecondReceiveWindow(DR_2, F_2)
7:     end if
8: end procedure
```

Algorithm 4 `StartReceive` in `EndDeviceLoraMac`

```
1: procedure GetChannelForTx(packet)          ▷Procedure trigged from the PHY Level
2:     global: DR_1, DR_2, F_1, F_2           ▷Data Rate and Frequency of both receive windows
3:     global: TWO_RX                         ▷Boolean variable for two receive windows
4:     OpenFirstReceiveWindow(DR_1, F_1)
5:     if TWO_RX = false then
6:         OpenSecondReceiveWindow(DR_2, F_2)
7:     end if
8: end procedure
```

5. Deterministically select the channel to transmit.

4.2.3.5 PHY layer

Similarly to the MAC Layer, this layer is modeled by means of a `LoRaPhy` Class that implements the basic methods and three sub-classes: `GatewayLoRaPhy, EndDeviceLoRaPhy`, and `JammerLoRaPhy` specific for each device type. In addition to these three classes, the PHY Layer uses the `LoRaChannel` class to handle some transmission parameters like the transmission power and the propagation losses, and the `LoRaInterferenceHelper` class for the interference and capture effect.

LoRaPhy This class implements the basic methods used by all device types at the MAC layer. It is responsible of computing the time-on-air based on the transmission parameters and the packet length based on the SX1272 and SX1272 LoRa transceivers as described before in chapter 2.1. It also serves as an interface to send and receive packets and implements various trace-sources and callbacks used by upper layers. The main procedures implemented in this class are:

1. `GetOnAirTime`: it computes the time on air of each packet as defined in equation (2.3)
2. `GetPreambleTime`: it computes the time on air of a given preamble as defined in equation (2.6), it is used by the MAC layer to open the receive windows.
3. `GetCAD`: it computes the Channel Activity Detection (CAD) as defined in equation (2.7), which is the minimum time needed to detect the channel occupancy.

EndDeviceLoraPhy This class defines the basic procedures needed to send and receive a packet. It is triggered from the MAC Layer and communicates with the helper to make an estimate of the energy consumption of the node. The object controls the states of the ED's PHY Layer based on the finite state machine presented in figure 4.6 using an object called `m_state` with the following states:

1. `TX`: when there is an ongoing transmission
2. `RX`: when the ED is receiving a packet
3. `STB`: when the ED is ready either to receive or to transmit a packet
4. `Sleep`: in low power consumption
5. `Dead`: when the battery is empty

The main procedures defined in this class are :

1. `Send`: it receives the MAC interruption with the MAC-level packet and transmission parameters. Then, it verifies the transmission parameters and starts the transmission procedure.

2. `StartReceive`: it stars the reception of a packet if the packet has the correct transmission parameters; i.e it is received at the same channel and SF the ED is listening to and it is intended to the specific ED.

3. `EndReceive`: this procedure calls the `LoRaInterferenceHelper` to verify if the packet was destroyed or not, and based on that it determines if the reception was successful.

4. `StateDuration`: it keeps track of the duration of each state in order to compute the power consumption in the `LoRaConsumptionHelper`.

Algorithm 5 `Send` in `EndDeviceLoraPhy`

procedure SEND($packet$,$TxParams$) ▷*Procedure trigged from the MAC Level*
  global: m_state ▷*PHY State of device*
  if $m\_state = STB$ **then** ▷*Verify that the ED is on standby*
    SWITCHTOTX (); ▷*Changing the the ED's state to TX*
    $ToA \leftarrow$ GETONAIRTIME($packet$, $txParams$) ▷*Compute the ToA in* `LoRaPhy`
    SEND ($packet, toa, TxParams$); ▷*Call the send procedure in* `LoRaChannel`
  end if
end procedure

Algorithm 6 `StartReceive` in `EndDeviceLoraPhy`

procedure STARTRECEIVE($packet$,toa,$Freq_{pkt}$,SF_{pkt},P_{rx}) ▷*Trigged from* `LoRaChannel`
  global: m_state ▷*PHY State of device*
  local: t ▷*Moment in which the function is called*
  local: SF_{ed}, $Freq_{ed}$ ▷*SF and Frequency this ED is listening to*
  local: $sensitivity$ ▷*Sensitivity for the SF and frequency*
  local: $event$ ▷*Event with information about the received packet*
  $event \leftarrow$ ADD ($packet$,P_{rx},toa,t) ▷*Add event to* `LoRaInterferenceHelper`
  if $m\_state = STB$ **then** ▷*Verify that the ED is on standby*
    if $SF_{ed} = SF_{pkt}$ **and** $Freq_{ed} = Freq_{pkt}$ **then** ▷*Verify the SF and Frequency*
      if $P_{rx} > sensitivity$ **then** ▷*Check the reception power*
        SWITCHTORX () ▷*Changing the the ED's state to RX*
        ENDRECEIVE ($packet$,$event$) ▷*Trigged at the end of the reception*
      end if
    end if
  end if
end procedure

JammerLoraPhy This class defines the PHY layer of the jammer. It implements the two types of jammers: channel-oblivious and channel-aware. The former sending packets randomly on the channel and the latter being aware of the channel activity. Hence, the channel-oblivious jammer is modeled similarly as a regular ED.

Then, as in the case of ED and presented in the PHY layer is implemented by means of three procedures:

Algorithm 7 `EndReceive` in `EndDeviceLoraPhy`

```
1: procedure EndReceive(packet,event)                ▷Trigged by StartReceive
2:     local: PktDestroy          ▷Boolean variable to staying if the packet is destroyed
3:     PktDestroy ← CheckInterf (event)              ▷Check at LoRaInterferenceHelper
4:     if PacketDestroyed then
5:         PktInterfered (packet)                    ▷Call the Packet Interference callback
6:     end if
7:     PktSuccess (packet)                           ▷Call the Packet Success callback
8:     SwitchToSleep ()                              ▷Changing the the ED's state to Sleep
9: end procedure
```

1. `Send`: for the channel-aware jammer it is triggered by the `EndReceive` procedure. In the case of channel-oblivious it is trigged by a MAC-level interruption. It verifies the transmission parameters and starts the transmission.
2. `StartReceive`: it is implemented by the channel-aware jammer, it stars the reception of a packet if the packet has the correct transmission parameters; i.e it is received at the same channel and SF the jammer is listening to.
3. `EndReceive`: this procedure calls the `LoRaInterferenceHelper` to verify if the preamble of the packet was destroyed or not, and based on that it determines if the reception was successful and starts the `Send` procedure.

Algorithm 8 `Send` in `JammerLoraPhy`

```
1: procedure Send(packet,TxParams)          ▷Trigged from EndReceive or AppJammer
2:     local: m_state                                  ▷PHY State of the jammer
3:     local: oblivious, aware                         ▷Type of the jammer
4:     local: SF_j, Freq_j                             ▷SF and Frequency of the jammer
5:     local: T_sw                                     ▷Reception to transmission switching time
6:     if m_state != TX and aware = True then          ▷Verify that the jammer is not on TX
7:         Wait (T_sw);                                ▷Wait T_sw before sending
8:         SwitchToTx ();                              ▷Changing the state to TX
9:         T_tx ← GetOnAirTime(packet, SF)             ▷Compute the ToA in LoRaPhy
10:        Send (packet, T_tx, TxParams);              ▷Call the send procedure in LoRaChannel
11:    end if
12:    if m_state != TX and oblivious = True then      ▷Channel-oblivious jammer
13:        SwitchToTx ();                              ▷Changing the state to TX
14:        T_tx ← GetOnAirTime(packet, SF)             ▷Compute the ToA in LoRaPhy
15:        Send (packet, T_tx, TxParams);              ▷Call the send procedure in LoRaChannel
16:    end if
17: end procedure
```

LoraChannel This class is in charge of modeling the RF channel shared by all devices in the network (ED, Gateway and Jammers).

During the configuration phase, all LoraPhy objects are added to the channel object. Then, throughout the simulation this object is called each time a particular device (Jammer, ED or

Algorithm 9 `StartReceive` in `JammerLoraPhy`

procedure StartReceive($packet$,T_{pre},$Freq_{pkt}$,SF_{pkt},P_{rx}) ▷*Trigged from* `LoRaChannel`
  local: m_state ▷*PHY State of the jammer*
  local: t ▷*Moment in which the function is called*
  local: $SF_j, Freq_j$ ▷*SF and Frequency of the jammer*
  local: $sensitivity$ ▷*Sensitivity for the SF and frequency*
  local: $event$ ▷*Event with information about the received packet*
  $event \leftarrow$ Add ($packet$,P_{rx},T_{pre},t) ▷*Add event to* `LoRaInterferenceHelper`
  if $m\_state != TX$ **or** RX **then** ▷*Verify that the jammer is not in TX nor RX*
    if $SF_{pkt} = SF_j$ **and** $Freq_{pkt} = Freq_j$ **then** ▷*Verify SF and Frequency*
      if $P_{rx} > sensitivity$ **then** ▷*Check the reception power*
        SwitchToRx () ▷*Changing the the jammer's state to RX*
        EndReceive ($packet$,$event$) ▷*Trigged at the end of the preamble*
      end if
    end if
  end if
end procedure

Algorithm 10 `EndReceive` in `JammerLoraPhy`

procedure EndReceive($packet$,$event$) ▷*Trigged by* `StartReceive`
  local: $PreDestroy$ ▷*Boolean variable to staying if the preamble is destroyed*
  local: $length$ ▷*Packet length of the jammer packet*
  local: $TxParams$ ▷*Transmissions parameters of the jammer*
  $PreDestroy \leftarrow$ CheckInterf ($event$) ▷*Check at* `LoRaInterferenceHelper`
  if $PacketDestroyed$ **then**
    PreInterfered ($packet$) ▷*Call the preamble interference callback*
  end if
  PreSuccess ($packet$) ▷*Call the preamble success callback*
  $packet_1 \leftarrow$ CreatePacket ($length$) ▷*Check at* `LoRaInterferenceHelper`
  Send ($packet_1, TxParams$); ▷*Call the* `Send` *procedure in* `LoraChannel`
end procedure

Gateway) wants to start a transmission. It computes then the reception power based on the propagation model (it can be Log-distance propagation losses or Building Penetration Loss as described previously in Chapter2.1) used and the transmission parameters (SF, duration, Transmission Power, localisation). Once the reception power has been calculated, this objects schedules a Receive event to all previously added. Hence, four main procedures are defined in this class:

1. `Add`: It connects a `LoraPhy` (GW, ED or Jammer) object to the `LoraChannel`
2. `Send`: It sends a packet in the channel. It is invoked by a PHY (ED, GW or jammer) that needs to send a packet. Then, it calls the `StartReceive` procedure in all the `LoraPhy` objects connected.
3. `GetRxPower`: It computes the reception power based on the transmission parameters, the path-loss model presented in Section 4.2.2.1 and the physical location of each device.
4. `Receive`: It notifies each connected LoRaPhy object by calling their `StartReceive` methods with the corresponding reception parameters.

Algorithm 11 `Send` in `LoraChannel`

procedure SEND(`LoraPhy`,$packet$,P_{tx},$TxParams$,ToA,F_{tx}) ▷*Trigged from* `LoraPhy`
  local: $phy\_list \leftarrow [1, \ldots, N]$ ▷*The array containing all the* `LoraPhy` *objects added*
  $M_{send} \leftarrow$ GETMOBILITY(LoraPhy) ▷*Mobility model of the sender*
  $SF \leftarrow TxParams.SF$
  for $i \leftarrow 1$ **to** N **do**
    $M_{rec} \leftarrow$ GETMOBILITY($phy\_list[i]$) ▷*Mobility model of the receiver*
    $delay \leftarrow$ GETDELAY(M_{send}, M_{rec}) ▷*Compute the delay*
    $P_{rx} \leftarrow$ GETRXPOWER($P_{tx}, M_{send}, M_{rec}$) ▷*Compute the reception power*
    WAIT ($delay$); ▷*Wait before starting the reception*
    STARTRECEIVE($packet$,ToA,F_{tx},SF,P_{rx}) ▷*Start reception in* `LoraPhy`
  end for
end procedure

LoRaInterferenceHelper Our module considers a standalone LoRaWAN network, and we assume that interference can only come from other LoRa-type transmissions that coincide in the same time-frequency slot. By making this assumption, we implement the classes necessary to model the capture effect.

Then, as described in detail in Chapter 4.2 a packet collision occurs when two or more radio signals are overlapped at the receiver. In an Aloha Pure system (as the one used for the analytical model), this collision results in all packets being destroyed. However, in case of capture-effect, a collision might not result in packet loss. The capture effect occurs then when the receiver stays synchronised to the strongest signal even though a collision has occurred. To model this phenomenon we use two approaches:

1. Aloha-Pure: all packets that overlap in time/frequency/SF are destroyed.

2. Co-channel interference: A packet may be demodulated at the GW if it has the required SINR margin by following the SINR threshold matrix as described in [1] and presented in Equation (4.24).

4.2.4 Model Validation

In order to provide a first validation of the the extension we did to the ns3-module, we evaluated the performance of a LoRaWAN cell composed entirely by channel-oblivious jammers that do not respect the duty-cycle and the packet length limitations imposed by the standard in order to verify if the throughput presents a similar behaviour as the mathematical model of LoRaWAN presented in Section 4.1. Thus, a LoRaWAN Class A network without confirmed traffic is modelled as a pure Aloha model, which is a Poisson point process with parameter λ_{ju} [packets/s]. The normalized throughput of jammers is given by:

$$\overline{\mathcal{G}_{ju}} = N_{ju} \cdot \lambda_{ju} \cdot T^s_{ju} \exp(-2 \cdot N_{ju} \cdot \lambda_{ju} \cdot T^s_{ju}) \tag{4.29}$$

where N_{ju} is the number of jammers and T^s_{ju} is the ToA given by Equation (2.3)

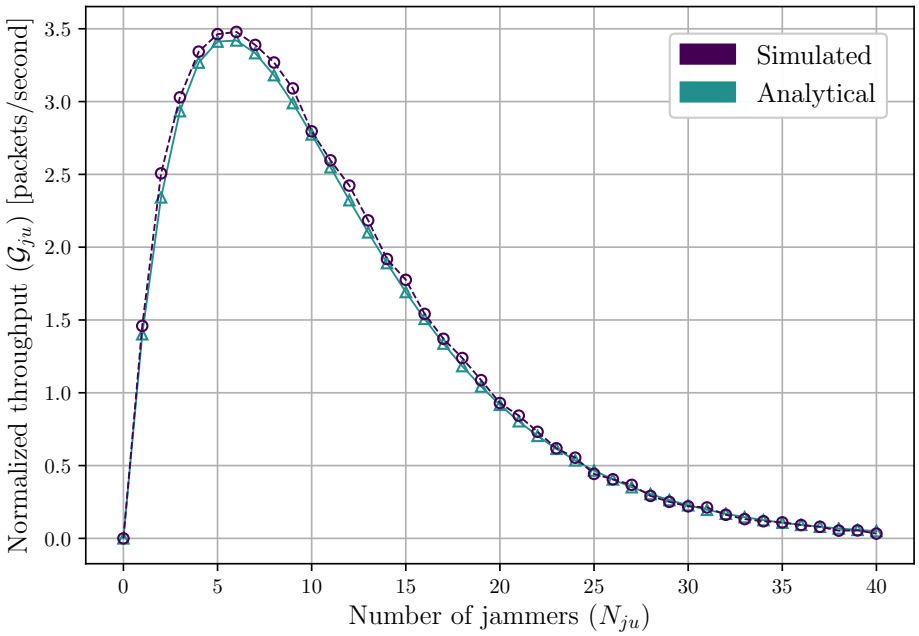


Figure 4.10: Normalized network throughput of a LoRaWAN network composed entirely with jammers.

Figure. 4.10 shows the network throughput of a LoRaWAN network composed entirely by jammers with $duty - cycle = 0.1$, $SF = 7$, a packet length of 500 *bytes*, a preamble of 8 *bytes*, a GW with only one reception path. For this simulation, we do not consider the capture effect, nor the co-channel rejection matrix. Thus, we can observe that the simulated throughput fits well the analytical one.

It is worth noting that, in the following chapter, we will provide an extensive performance

evaluation where we carried out several simulation campaigns to assess the performance of LoRaWAN with and without jammers. In these simulations, the mathematical model is compared with simulation results. Hence, these comparisons should be considered also as an integral part of the model validation.

4.3 Conclusion

In this chapter, we have presented the main contribution of this thesis. We proposed two tools allowing to assess the impact of jamming attacks on LoRaWAN networks.

We first introduced a mathematical model that considers a LoRaWAN with an ideal channel. The model allows dimensioning a network with a single gateway and several bidirectional EDs under the attack of channel-oblivious jammers. It also includes a re-transmission mechanism that seeks to improve the message success probability.

Then, we go an step forward and extend an existent simulation model, which allows not only to model channel-oblivious jammers but also jammers that have the ability to sense the network and to jam accordingly. Additionally, this model also allows to model a realistic channel and to consider several particularities of an actual LoRaWAN deployment.

In the next Section, we will use both models by providing a extensive performance evaluation of LoRaWAN under jamming.

Chapter 5

Performance Evaluation

In this chapter we will present an extensive performance evaluation analysis of LoRaWAN under jamming. For that we consider several scenarios and performance evaluation metrics: (i) a LoRaWAN network with ideal channel, (ii) a LoRaWAN network with realistic channel and hardware limitations, (iii) a LoRaWAN Network with gateway diversity, (iv) jammer strategy, (v) geographical impact of jammers and (vi) energy consumption and battery life. In the following we will introduce the different performance metrics used and then we will explain in detail each scenario.

5.1 Performance evaluation metrics

In order to evaluate the scenarios cited before we compute the following metrics:

Network throughput: defined as the average number of packets per second [packets/s] well received by the gateway.

Network goodput: defined as the average number of messages per second [messages/s] well received by the gateway.

Message Success Probability: defined as the probability of having a message well received after, at most, r re-transmissions.

$$\frac{\texttt{Messages Sent}}{\texttt{Messages successfully received}} \tag{5.1}$$

Average number of re-transmissions: defined as the average number of packets a given ED needs to send a single message.

$$\frac{\texttt{Packets sent per message}}{\texttt{Messages successfully received}} \tag{5.2}$$

ACK Success probability: defined as the proportion of ACK packets lost at the EDs' side. For the LoRaWAN case it considers both, ACK sent on RX1 and RX2.

$$\frac{\texttt{ACKs Received}}{\texttt{ACKs Sent}} \tag{5.3}$$

It is worth noting that, as assumed in the analytical nodel, for the simulation we consider that message transmissions and re-transmissions are the result of the same Poisson point process with parameter (λ_u) and duty-cycle restriction (d_u), the user traffic load is the same regardless r. Therefore, the results reported in the following sections consider the same network throughput regardless the number of re-transmissions considered. Hence, a goodput with $r = 0$ is equivalent to the throughput reached in the network.

5.2 Performance with ideal channel

In this section, we quantify the LoRaWAN-like system performance with ideal channel conditions. We use the developed model and a simulation approach using ns3. The simulation considers most of the assumptions taken for the development of the mathematical model: (i) Poisson packet arrivals, (ii) static nodes uniformly distributed, (iii) a sole GW, (iii) an Aloha-like channel access protocol, where the capture effect is neglected, (iv) SF orthogonality, and (v) perfect coverage of the GW.

As in the analytical model, the simulator considers systematic acknowledgment of all uplink transmissions with a maximum number of re-transmissions r. For downlink transmissions, we model a bank of FIFO (First In First Out) buffers (one for each SF) with only one packet storage capacity. Hence, if the GW does not handle to transmit given ACK under the time constraint (TRX1 expired), the next ACK will overwrite the data.

We model a network consisting of one GW, and we vary different parameters in order to evaluate its performance. The network operates in the 868 MHz frequency band with three uplink and one downlink channel for transmission ($F = 3$). Users employ upto six SF ranging from 7 to 12, with a uniform distribution ($\rho_u^s = 1/6\ \forall s$). The user duty-cycle is $d_u = 0.01$, all user packets have a length of $l_u = 50$ `bytes`, while ACK are sent with a length of $l_{ack} = 10$ `bytes`. The maximum number of re-transmissions is $r = \{0, 4, 8\}$.

5.2.1 LoRaWAN-like without jamming

In the following we present numerical and analytical results for a standalone LoRaWAN-like deployment. Therefore, we exploit the theoretical expressions to find out the maximum capacity of the network. We evaluate three performance metrics: User goodput ($\mathcal{O}_u$), Message Success Probability ($\mathcal{P}_m$) and Expected number of re-transmissions ($\mathcal{E}(r)$. For that we considered different scenarios varying the number of users $N_u = \{1, 250, 500, ..., 4000\}$ and the maximum number of re-transmissions per message (r). Results are presented in Figure. 5.1.

Figure. 5.1 (a), presents the user goodput. For $r = 0$, all packets are sent only once, this means that the number of packets and messages is the same. Hence, the goodput is equivalent

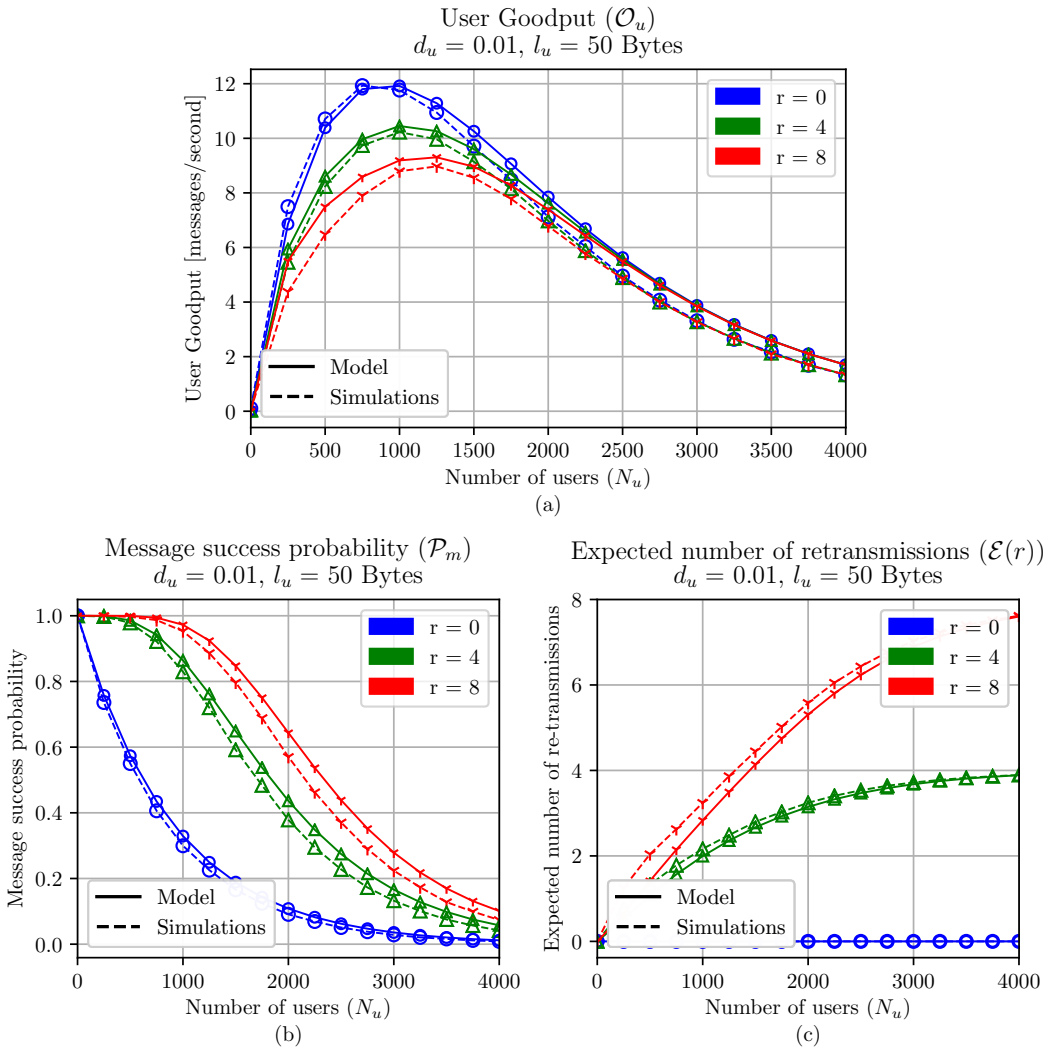


Figure 5.1: Performance evaluation without jamming: (a) User goodput ($\mathcal{O}_u$), (b) Message Success Probability ($\mathcal{P}_m$) and (c) Expected number of re-transmissions ($\mathcal{E}(r)$).

to the throughput reached in the network. Then, a classic behavior of an Aloha-type network can be observed, the peak is reached at `[11.91 messages/s]` for $N_u = 1000$. On the contrary, for $r > 0$ we note that, as r increases the goodput obtained decreases. This decrease is due to gateway saturation; as ACK are sent in a unique frequency band, its responsiveness decreases as it becomes saturated. Then, users begin to make unnecessary re-transmissions even when the messages arrive well (a more detailed explanation of this is presented in the validation in Section 4.1.10).

From Figure. 5.1 (b), we can see that $\mathcal{P}_m$ degrades rapidly by increasing the number of users for $r = 0$. For example, the message success probability falls to $\backsimeq 0.43$ when $N_u = 750$. This reduction becomes even more important for networks with $N_u \geq 2000$ where this number falls to only 0.1 or less. On the other hand, for $r > 0$, it can be seen that the higher r is, the higher the message success probability will be, thus increasing the network reliability.

It should be noted that, having a $\mathcal{P}_m = 1$ does not necessarily means that the network performs better. In fact, having a better reliability has the cost of having more re-transmissions per message, and in fact a lower throughput.

Figure. 5.1 (c) presents the average number of re-transmissions per message as a function of r and N_u. We can see that this number increases in proportion to the number of users served in the network. Hence, there is a compromise between the user goodput and the average number of re-transmissions. As a result, if an operator is willing to have an adequate message success probability, it needs to carefully decide of the number of re-transmission the application needs to support as it has a direct impact on the nodes' current consumption.

Therefore, we conclude that acknowledging all messages systematically can significantly improve network reliability, however, it comes at the cost of reducing the goodput obtained in the network. Then, as we have seen from Figure 5.1, a sole gateway reaches the maximum goodput at $N_u = 1000$ with $r = 0$. This point correspond to $G_u = 36.20$ `[messages/s]` and $\mathcal{O}_u = 11.91$`[messages/s]`. This means that, only about 32.9% of messages sent are correctly received at the gateway.

It is worth nothing that, for this performance evaluation, we considered a uniform distribution of users among SFs. Previous investigations on this matter have concluded that this is not the optimal distribution (the distribution that reaches the maximum throughput) [94]. Therefore a more in-depth analysis of it shall be considered.

5.2.2 LoRaWAN-like and LoRaWAN without jamming

So far we have studied the performance of the system considered for the analytical model presented in Section 4.1. As we have seen, in that model we considered various simplification assumptions such as the use of a unique reception window in a separate frequency band, spreading factor orthogonality and a gateway that is able to decode up to 18 packets in parallel. Consequently, a comparison of this system with a more accurate LoRaWAN model becomes necessary to verify whether the results obtained are similar to what would be obtained in LoRaWAN.

Therefore, we carried out a simulation campaign in ns3 to compare both systems. To do a fair comparison, we still consider a gateway that is able to decode up to 18 packets in parallel and a uniform distribution of SFs ($\rho_u^s = 1/6$). However, we add the possibility of having two reception windows (RX1 and RX2) as stated in the specification and depicted in figure 2.3, the outdoor path-loss described in Equation 4.21 and the capture effect. The simulation parameters are summarised in Table 5.1, and the results obtained are depicted in Figure 5.2.

As we can see from Figure 5.2, the LoRaWAN cell presents a similar behaviour to that obtained in the results with the analytical model. Hence, we have an increasing curb as the number of users increases to a maximum point where the performance starts to decrease. However, this maximum goodput point is obtained for a much higher number of users in LoRaWAN. Indeed, the maximum goodput achieved in LoRaWAN is 13.46 `[messages/s]` at $N_u = 2000$ with $r = 0$. This point correspond to $G_u = 72.41$ `[messages/s]`. This means that, only about 18.48% of messages sent are correctly received at the gateway.

To explain this behaviour we shall refer to the two hypotheses taken into account for the

Table 5.1: Simulation Parameters: LoRaWAN and LoRaWAN-like comparison

	Parameter	Values
Network	N_u	$1, 250, \ldots, 4000$
	l_u	50 bytes
	l_a	10 bytes
	d_u	0.01
	s	7 - 12
	ρ_u^s	$1/6\ \forall s$
	T_u	82.17 ms - 1.81 s
	T_{ack}^s	30,976 ms - 0.82 s
	r	0 - 8
	radius	5 km

	Parameter	Values
Path-loss	d_0	40 m
	PL_0	140.7
	n	3.12
	L_h	-4.7
	h_{ED}	3 m
	σ	9.7 dB
Band	Up / Dw	868.1/.3/.5 MHz
	Downlink	869.525 MHz
Simulation time		10 h

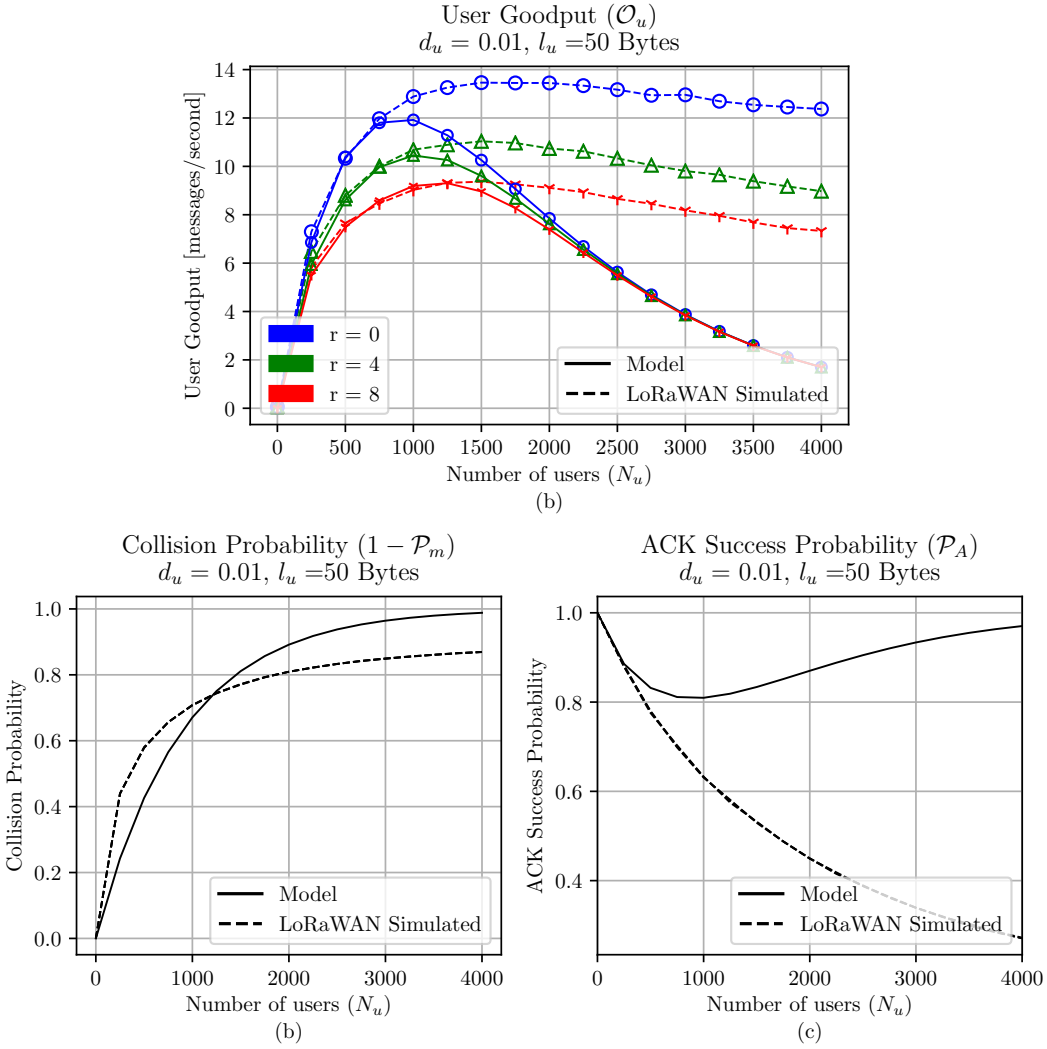


Figure 5.2: LoRaWAN and LoRaWAN-like comparison: (a) Network goodput, (b) Collision probability and (c) ACK success probability (RX1 in LoRaWAN).

mathematical model: the inclusion of only one reception window (RX1) a with a separate frequency channel and neglecting the capture effect.

Firstly, including a first reception window in the same band as uplink transmissions should cause the goodput to decrease as ACK packets generate collisions with user packets. This can be seen clearly in Figure 5.2 (c). As we can see, when there is a low number of users (and hence a lower user traffic load) LoRaWAN-like and LoRaWAN have the same ACK success probability, however as the number of users increases, this probability drops fast in LoRaWAN.

Secondly, the capture effect increases the probability of having a packet well decoded at the gateway even if there is a collision. This effect is clearly presented in Figure 5.2 (b). Indeed, for $N_u > 1250$, the probability of collision is lower in the case of LoRaWAN making the goodput obtained higher for higher user traffic loads.

Then, as a consequence of these two factors, the first pushing down the performance of the network and the second doing the opposite; the LoRaWAN cell simulated finds a medium equilibrium point, where for fewer users ($N_u < 1000$) it follows nearly the same behaviour as the LoRaWAN-like model. On the contrary, for $N_u > 1000$, the fact of considering the capture effect becomes much more preponderant than the probability of collision due to ACKs sent in the first reception window. This means that the goodput obtained does not decay as quickly as in a Pure Aloha network.

The analysis mentioned above allows us to conclude that the analytical model works well for lower traffic loads where the two strong assumptions taken into consideration are valid. From this point on, the LoRaWAN system outperforms our model.

5.2.3 LoRaWAN-like with jamming on uplink channels

In this scenario, we evaluate the performance of a LoRaWAN-like cell under the attack of channel-oblivious jammers transmitting on the uplink channels only. As in the case of no jamming, we evaluate three performance metrics: User goodput ($\mathcal{O}_u$), Message Success Probability ($\mathcal{P}_m$) and Expected number of re-transmissions ($\mathcal{E}(r)$).

As we have seen, the point at which a LoRaWAN-like cell reaches the maximum throughput is $N_u = 1000$. Therefore, we chose a midpoint that would allow us to see the impact of jammers on a network that is not congested, and that is also inside the part of the curb where LoRaWAN and LoRAWAN like converge. Thus, we consider a cell with a sole gateway and we set $N_u = 500$.

This cell is under the attack of 60 (10 on each SF) jammers whose aggregated traffic load $\overline{G_{ju}}$ [1] varies between 0 to 1 and with a packet length $l_{ju} = 50$ [`bytes`]. As in the case of users, they are deployed by following a uniform distribution around the gateway. Figure. 5.3 reports the results obtained.

From Figure 5.3 (a), we can observe that the performance degrades rapidly by increasing $\overline{G_{ju}}$. For instance, when $r = 0$ the goodputs goes from 10.38 for the case of no jamming, to only 1.96 [`messages/s`] with $\overline{G_{ju}} = 0.25$.

However, when the re-transmission mechanism is activated, the $\mathcal{P}_m$ achieved in the network increases. Indeed, for $r = 0$ and $\overline{G_{ju}} = 0.17$. It is only about 0.2 while for $r = 4$ it goes up to nearly 0.4.

[1] $\overline{G_{ju}}$ designates the aggregated jammers' traffic load on the uplink channels and is defined as $\sum_s \frac{G^s_{ju}}{T^s_{ju}}$.

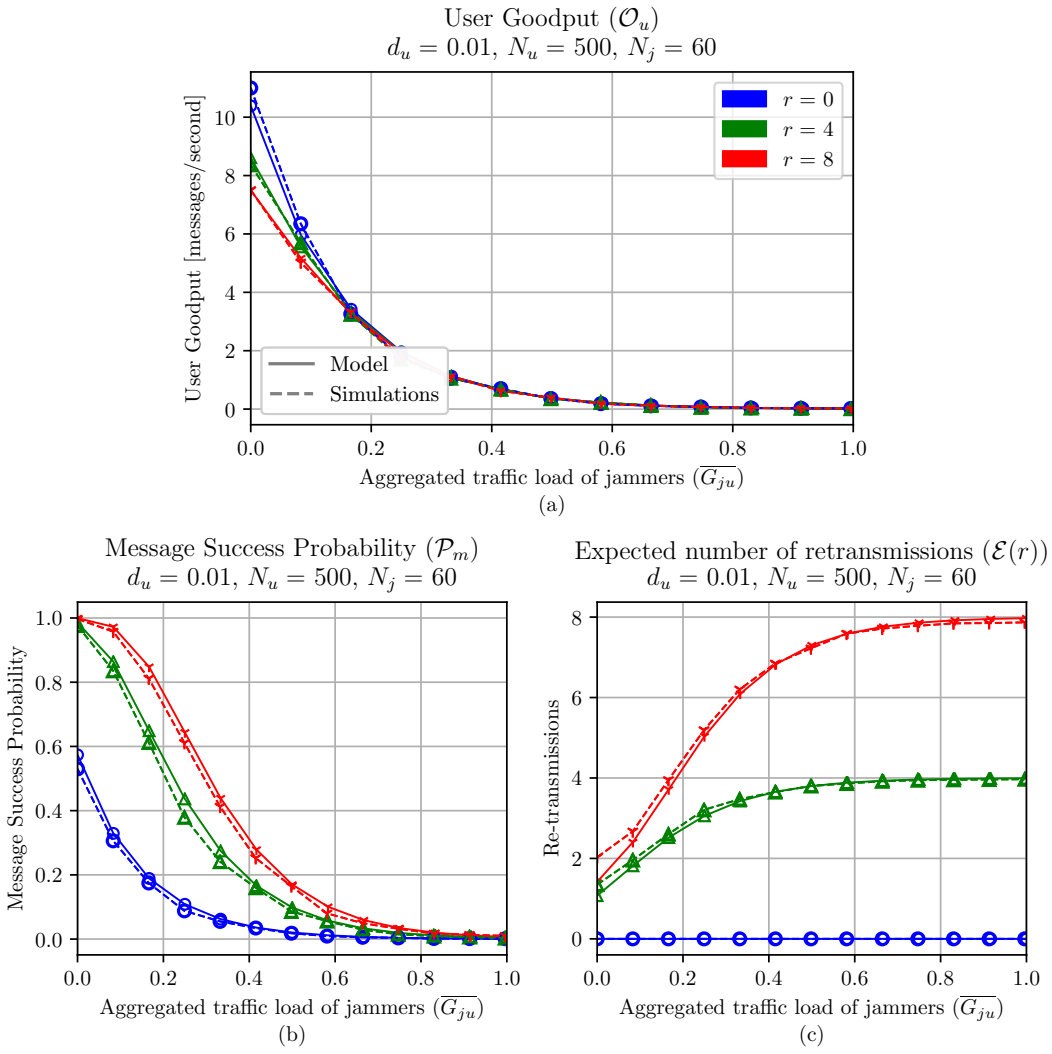


Figure 5.3: Performance evaluation under jamming on the uplink channels: (a) User goodput ($\mathcal{O}_u$), (b) Message Success Probability ($\mathcal{P}_m$) and (c) Expected number of re-transmissions ($\mathcal{E}(r)$).

Similarly, from Figure 5.3(a), we note that for a cell with 500 nodes and $r = 8$, the network can alleviate the problem of jammers when they jam the uplink channels moderately. In fact, for $\overline{G_{ja}} = 0.25$, $\mathcal{P}_m$ goes from roughly 0.10 for $r = 0$ to 0.43 for $r = 4$. It can go up to 0.64 for $r = 8$.

Finally, from Figure. 5.3 (c) we can observe that the expected number of re-transmissions per message increases as the jammers' traffic load grows, which can lead to an increase in the current consumption.

5.2.4 LoRaWAN-like with jamming on downlink

Similarly to the case of jamming on the uplink channels, in this scenario the LoRaWAN-like cell is put under the attack of channel-oblivious jammers. This time transmitting on the downlink band. For that, 30 jammers (10 on each SF) are deployed uniformly around the gateway. As

in uplink, they are characterised by the the aggregated traffic load $\overline{G_{ja}}$ [2] that varies between 0 and 2 and jammers' packet length $l_{ja} = 50$ [bytes].

Figure 5.4 reports the network performance for the case the case where jammers transmit on the downlink channel only. Contrary to the case where only the uplink transmissions are jammed, the goodput does not drop to zero when the $\overline{G_{ja}}$ increases, instead it stabilizes at 3.5 [messages/s]. It represents the point where $\mathcal{P}_A = 0$, and all messages are re-transmitted r times systematically. Hence, as the goodput measures the rate of *new-messages* actually received by the GW, having a lower goodput level, and at the same time a message success probability of nearly 1, means that the vast majority of packets are repeated versions of the same messages due to re-transmissions.

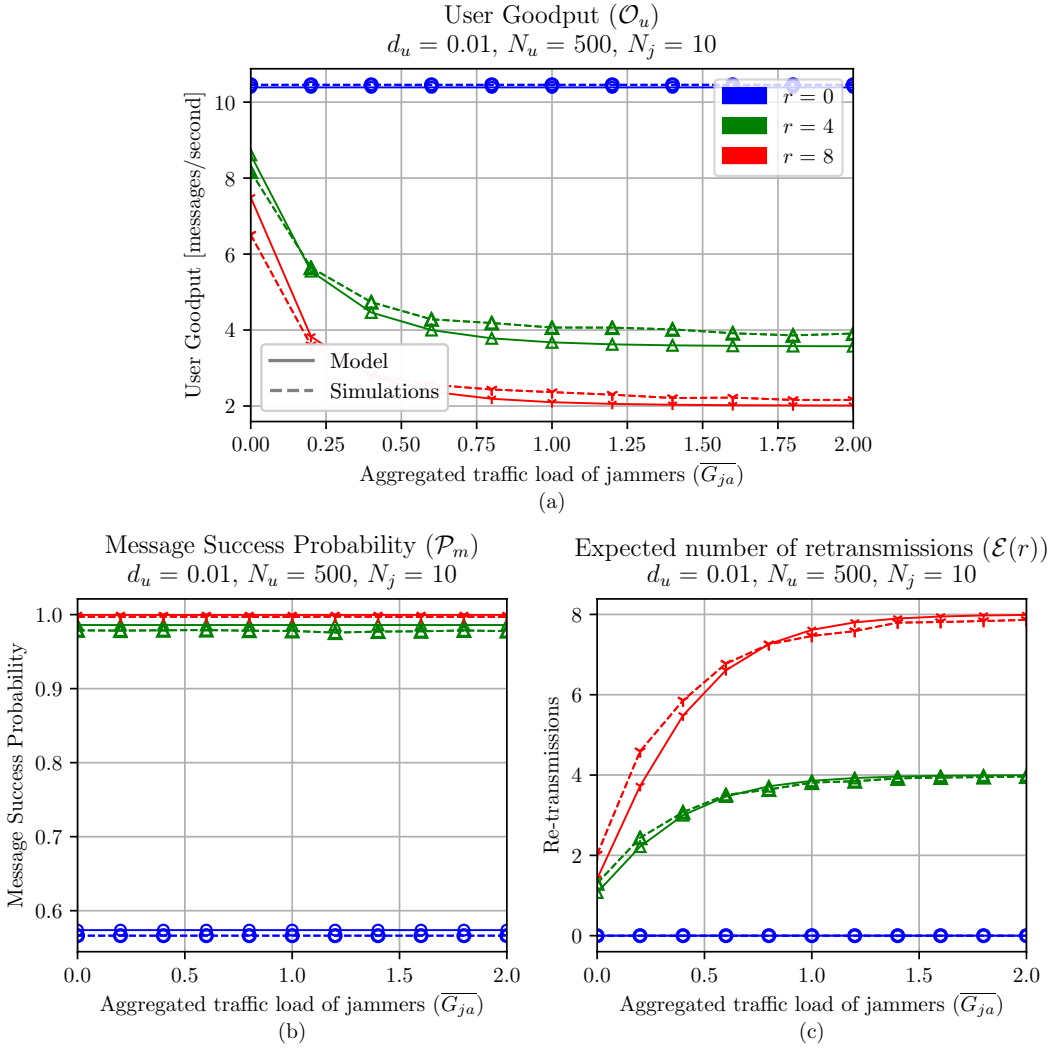


Figure 5.4: Performance evaluation of a LoRaWAN-like network under jamming on the downlink channel: (a) User goodput ($\mathcal{O}_u$), (b) Message Success Probability ($\mathcal{P}_m$) and (c) Expected number of re-transmissions ($\mathcal{E}(r)$).

[2] $\overline{G_{ja}}$ designates the aggregated jammers' traffic load on the downlink channel and is defined as $\sum_s \frac{G^s_{ja}}{T^s_{ju}}$.

In regards to the expected number of re-transmissions per message we see a similar behaviour as the one obtained for jammers transmitting on the uplink channels. It increases as the jammers' traffic load increases. However, contrary to the uplink case, this increase in re-transmissions does not make the network more resilient since there is no gain in the message success probability.

In summary, we have seen that for the case where jammers transmit on the downlink channel only and the re-transmission system is activated, the goodput decrease faster but does not drop to zero. Additionally, the fact of acknowledging all uplink transmissions systematically can do more harm than good when the downlink channel is saturated. Indeed, the number-of re-transmissions (and the energy consumption) increases without this implying a gain in the probability of success of messages.

5.3 Performance of LoRaWAN with realistic channel and hardware limitations

In Section 5.2 we have evaluated the performance of the LoRaWAN-like network considered for our mathematical model with and without jamming. In this section we will consider a more realistic scenario where the different particularities of a real deployment can be considered. For that, we will use the ns3 module described in Section 4.2.

Therefore, as done in the case of LoRaWAN-like, we will first present an standalone LoRaWAN without jamming performance evaluation and then we will evaluate the performance of the network under jamming. As shown in figure 5.5 we consider a LoRaWAN cell consisting of several users and one gateway. Nodes are uniformly distributed around the gateway within a radius of 5 km. Nodes are static and configured to use the best SF possible as a function of their position and the gateway's sensitivity. We consider the outdoor path-loss presented in Section 4.2.2.1.

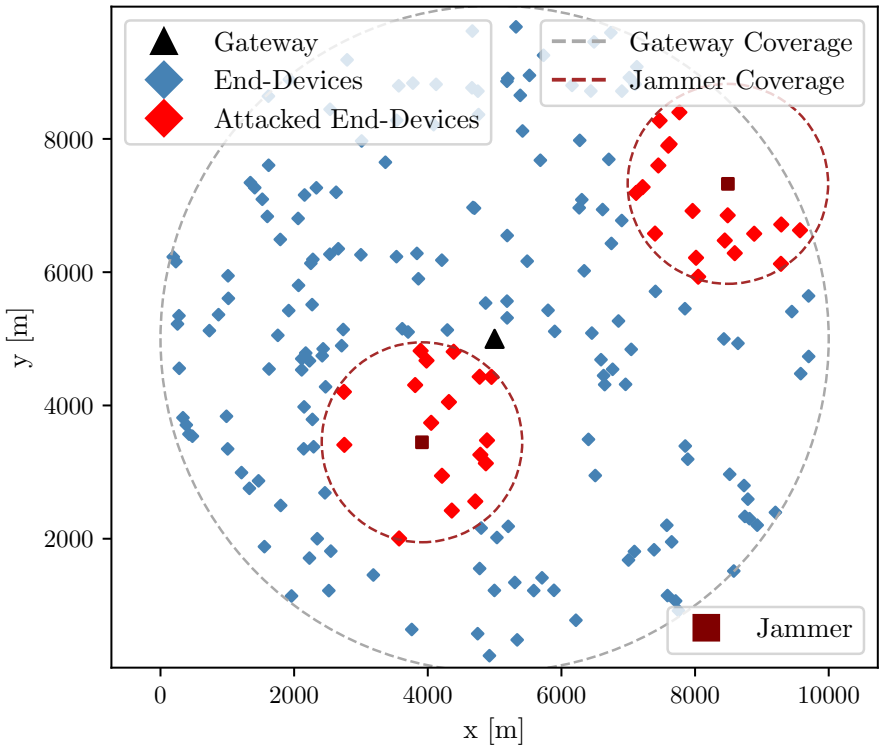


Figure 5.5: Simulation Scenario

Hence, acording to the path-loss model and the uniform distribution of nodes, SF are distributed as follows: $\rho_u = \{0.33, 0.22, 0.1, 0.09, 0.19, 0.07\}$. As regards the nodes' application profile, we considered a packet length l_u of 50 bytes, that all users, are configured to use a duty-cycle d_u of 0.01 with an exponential inter-arrival time distribution. Thus, depending on the SF, T_u varies from 82.17 ms to 1.81 s.

For uplink and first ACK transmissions, the cell operates in the 868 MHz band, three sub-

bands are considered: 868.1, 868.3 and 868.5 MHz, each one with a bandwidth of 125 kHz, all EDs belong to Class A with confirmed traffic. For the second ACK, a separate channel operating in the 869.525 MHz band is used, an SF of 12 is considered. The ACK packet length (l_a) is set to 25 bytes, and the ACK transmission time T^s_{ack} varies from 46, 33 ms to 1.15 s as a function of the SF s.

Contrary to the scenarios considered in Section 5.2, in this simulation we consider a gateway with hardware limitations. Hence, we consider the Semtech SX1301 that cannot support decoding 18 packets in parallel but 8.

It should be noted that, as packet transmissions and re-transmissions are the result of the same Poisson point process with parameter (λ_u) and duty-cycle restriction (d_u), the user traffic load is the same regardless r. For this simulation scenario and SF distribution, the average traffic load per user is 2.83 [packets/min].

We define three scenarios considering the LoRaWAN cell described before with parameters reported in Table 5.2: (i) LoRaWAN without jamming, (ii) LoRaWAN under jamming with lower user traffic loads and (iii) LoRaWAN under jamming with higher traffic loads.

For the LoRaWAN without jamming we simulated a cell with N_u varying from 1 to 2000 and r varying from 0 to 64. As for the jamming scenarios we simulated a LoRaWAN cell under the attack of jammers transmitting either on the 868 MHz (three channels) band only or on the 869.525 MHz. The number of users is set to 100 and 500 respectively.

The network is under the attack of 10 jammers per channel whose aggregate traffic load ($\overline{G_{ja}}$) varies from 0 to 2. The jammer's packet length l_{ju}/l_{ja} is set to 50 bytes with a transmission time T_j varying from 82.17 ms to 1.81 s (as a function of the SF).

5.3.1 LoRaWAN without jamming

Figure. 5.6 presents the performance evaluation of a LoRaWAN cell without jammers considering a realistic channel and hardware limitations. Different network configurations varying N_u and r were considered.

Figure. 5.6 (a), presents the network goodput. For $r = 0$, all messages are sent only once, this means that the number of packets and messages is the same. Hence, a classic behavior of an Aloha-type network can be observed, reaching the maximum goodput at 11 `[messages/s]` for $N_u = 600$.

On the contrary, for $r > 0$ we note that, as r increases the goodput obtained decreases. This decrease is caused by two reasons: (i) ACKs sent in the same band and SF same as user data packets cause collisions with user packets, and (ii) ACKs sent in the 869.525 MHz band are sent with $SF = 12$. Hence, the GW is locked a considerable amount of time sending a single ACK packet (1.15 s). Consequently, as the number of ED increases, its responsiveness decreases as it becomes saturated. Then, EDs begin to make unnecessary re-transmissions even when the original packet arrived well.

From Figure. 5.6 (b), we can see that the Message Success Pobability degrades rapidly by increasing the number of EDs for $r = 0$. For example, it falls to 0.37 when $N_u = 600$. This

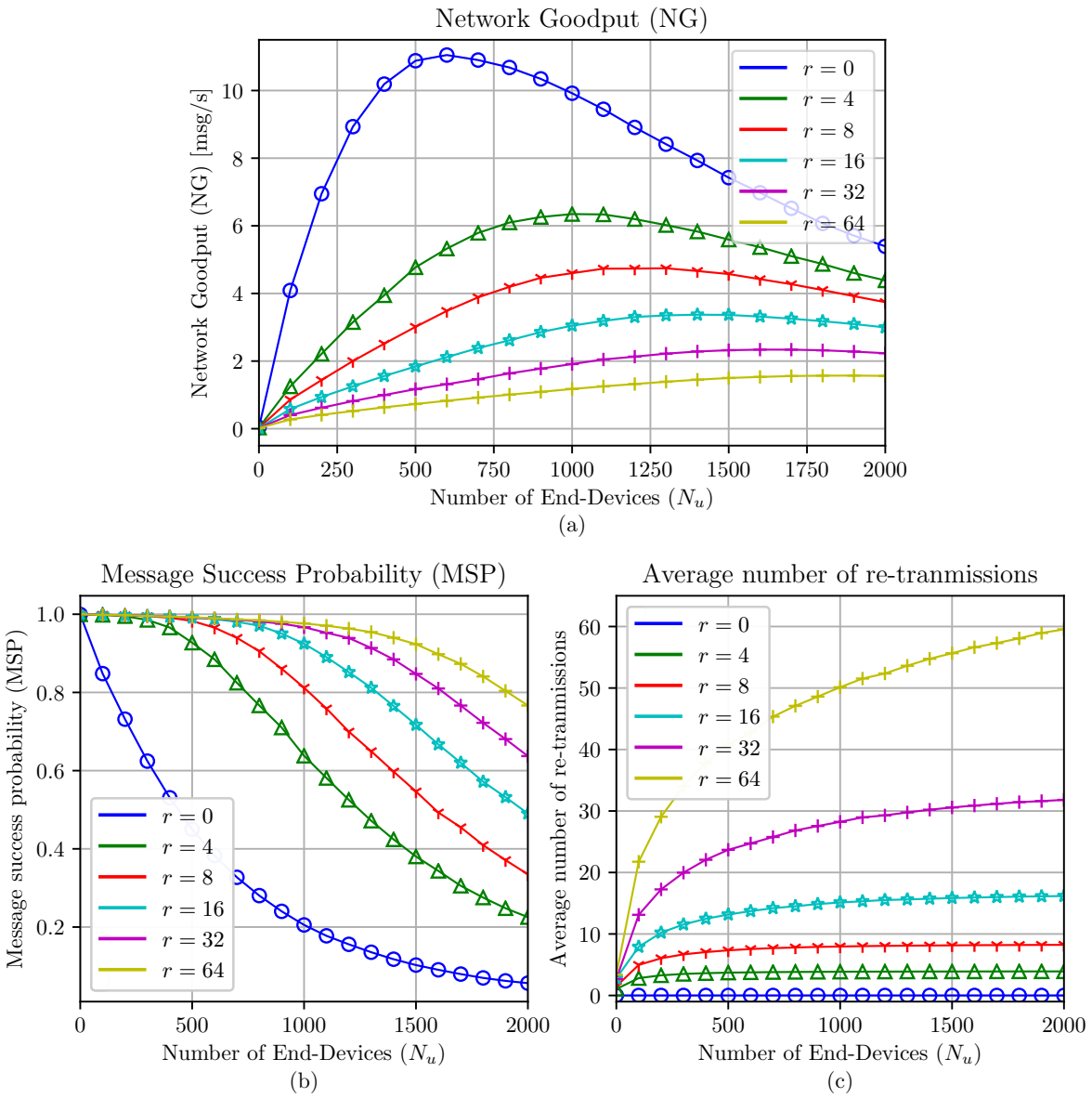


Figure 5.6: Network Performance of a realistic LoRaWAN cell considering re-transmissions: (a) Network Goodput, (b) Message Success Probability and (c) Average Number of Re-transmissions.

Table 5.2: Simulation Parameters: LoRaWAN with realistic channel

	Parameter	No jamming	Jamming on the uplink/downlink
Network	N_u	$1, 100, \ldots, 2000$	$100, 500$
	l_u	50 bytes	50 bytes
	l_a	25 bytes	25 bytes
	d_u	0.01	0.01
	s	7 - 12	7 - 12
	T_u	82.17 ms - 1.81 s	82.17 ms - 1.81 s
	T^s_{ack}	46, 33 ms - 1.15 s	46, 33 ms - 1.15 s
	r	0 - 64	0 - 64
	radius	5 km	5 km
Gateway	Chipset	SX1301	SX1301
	# of paths	8	8
	$RX1$	868.1/.3/.5	868.1/.3/.5
	$RX2$	869.525	869.525
	Sensitivity	-130.0 to -142.5 dBm	-130.0 to -142.5 dBm
Jammers	l_j	-	50 bytes
	T_{ju}/T_{ja}	-	82.17 ms - 1.81 s
	$\overline{G_{ju}}/\overline{G_{ja}}$	-	$0 - 2$
	N_{ju}/N_{ja}	-	60/10
Path-loss	d_0	40 m	40 m
	PL_0	140.7	140.7
	n	3.12	3.12
	L_h	-4.7	-4.7
	h_{ED}	3 m	3 m
	σ	9.7 dB	9.7 dB
Band	Up / Dw	868.1/.3/.5 MHz	868.1/.3/.5 MHz
	Downlink	869.525 MHz	869.525 MHz
	Bandwith	125 kHz	125 kHz
Simulation time		10 h	10 h

reduction becomes even more important for networks with $N_u \geq 1500$ where this number falls to only 0.1 or less.

On the other hand, for $r > 0$, it can be seen that the higher r is, the higher the probability of success on the messages will be, thus increasing the network reliability. Indeed, the selection of r should consider the number of EDs an operator is willing to serve. For example, there is no interest in selecting $r = 8$ for $N_u \geq 1400$ since the Message Success Probability will decay to 0.6 or less.

Figure. 5.6 (c) presents the average number of re-transmissions per message as a function of r and N_u. We can see that this number increases in proportion to the number of EDs served in the network. Hence, there is a compromise between the goodput and the average number of re-transmissions. As a result, if an operator is willing to have an adequate goodput, it needs to carefully decide the number of re-transmission the application needs to support.

Finally, we can conclude that hardware limitations have a significant impact on the final performance achieved. Indeed, the maximum goodput obtained in this LoRaWAN cell is 11 `[messages/s]` for only $N_u = 600$, while in the LoRaWAN network simulated in Section 5.2.2

the maximum goodput reached was 13.46 [messages/s] at $N_u = 2000$. This difference is mainly due to the limited capacity of the gateway, since it goes from 18 packets in parallel to only 8.

In addition to that, we have also seen that the way in which reception windows are handled has a significant impact on network performance when there are re-transmissions. Indeed, a LoRaWAN cell where all ACK are sent on a separate channel with the biggest SF possible reduces its goodput significantly. For instance, for $r = 4$ in the LoRaWAN cell simulated in Section 5.2.2, the maximum goodput was 10.85 [messages/s] for $N_u = 1500$, while the goodput obtained in this LoRaWAN cell is only 6.25 [messages/s] for $N_u = 1000$.

5.3.2 Jamming with a less congested network

In this scenario we consider a network with only 100 nodes. In such case, the network is not congested. Consequently, it might be less vulnerable to jamming. As in the case of no jamming we also evaluated three performance metrics: (i) Network Goodput, Message Success probability and (iii) the average number of re-transmissions. Then, we variate the number of maximum re-transmissions allowed and the aggregated jammer's traffic load ($\overline{G_{ju}}/\overline{G_{ja}}$) between 0 and 2.

Figure. 5.7 presents the results obtained. Clearly, when jammers transmit on the 868 MHz Band (.1/.3/.5), the greater $\overline{G_{ju}}$ is, the lower the goodput is, and the higher the average number of re-transmissions is. Indeed, for all cases, the goodput droops to nearly zero when $\overline{G_{ju}} = 1.0$. However, the network can alleviate the problem of jammers when they jam moderately by allowing re-transmissions. For instance, for a $\overline{G_{ju}}$ of 0.2, the success probability goes from 0.44 for $r = 0$ to 0.91 for $r = 4$. This figure can be even better for $r = 32$, where a succes probability close to 1 is achieved.

On the contrary, for jammers transmitting on the downlink band (868.525 MHz), the impact on the network goodput is much less important. Indeed, for $r = 4$ the goodput downs from 1.19 to 1.01 [messages/s] in the worst case. As for the success probability, we can see that it stays constant regardless the value of $\overline{G_{ja}}$. Besides, the average number of re-transmissions increases although the performance of the network does not improve.

In summary, a LoRaWAN cell with $N_u = 100$ with re-transmissions is reasonably resilient against jamming on the uplink channels if the aggregated traffic load of jammers is less than 0.2 with an $r = 4$. As for jammers on downlink, we have seen that the goodput obtained is not affected regardless the traffic load of jammers, however if messages are systematically acknowledged the impact on the number of re-transmissions (and hence the energy consumption) is not negligible.

5.3.3 Jamming with a congested network

In this scenario we consider a network with 500 nodes. In such case, the network is congested. Consequently, it might be more vulnerable to jamming. As in the case of no jamming we also evaluated three performance metrics: (i) Network Goodput, Message Success probability and (iii) the average number of re-transmissions. Then, we variate the number of maximum

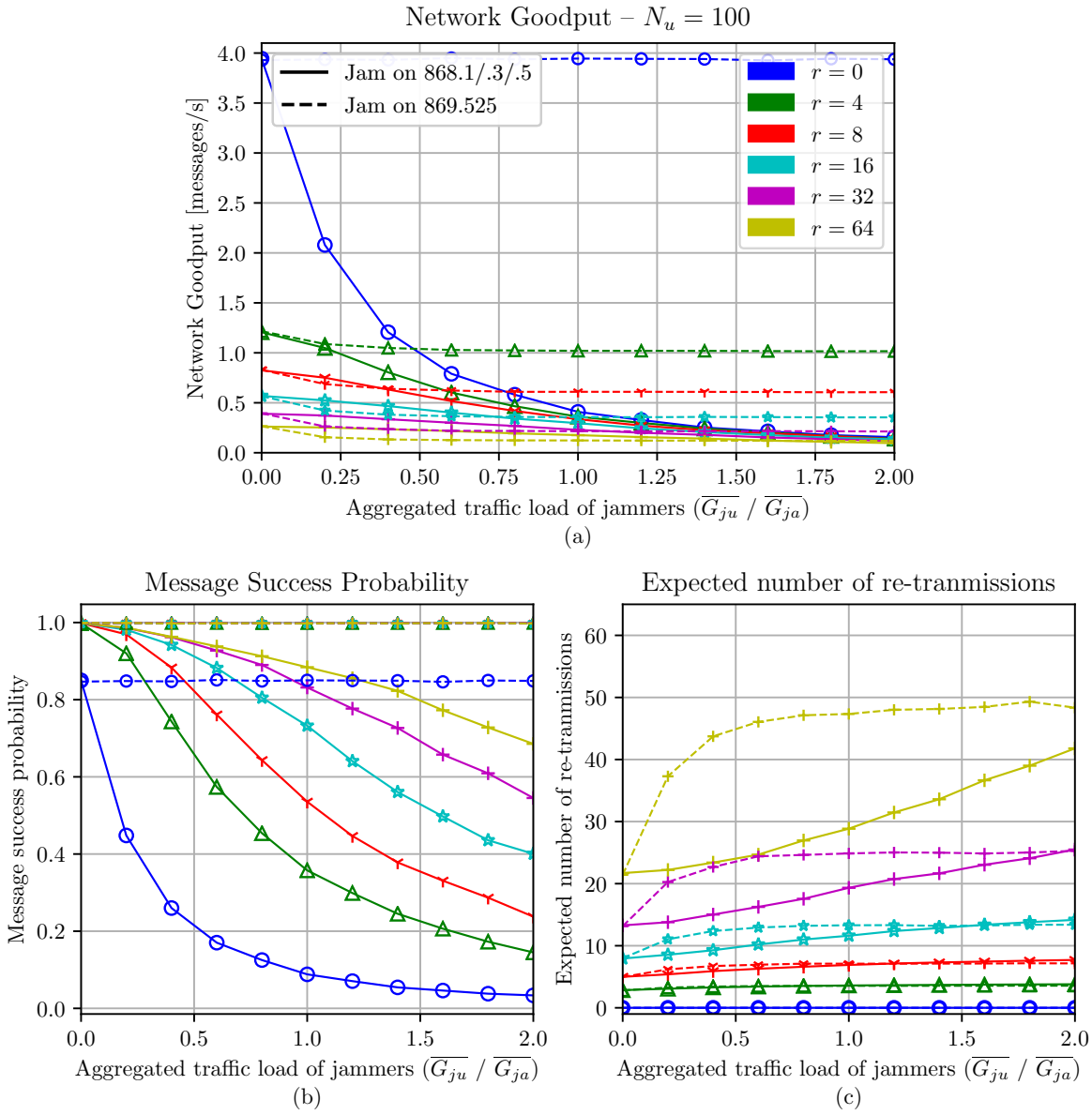


Figure 5.7: Realistic LoRaWAN cell under jamming with $N_u = 100$: (a) Network Goodput, (b) Message Success Probability and (c) Average Number of Re-transmissions.

re-transmission allowed and the aggregated jammer's traffic load ($\overline{G_{ju}}/\overline{G_{ja}}$) between 0 and 2.

In contrast with the less congested scenario, from Figure. 5.8 we note that for $N_u = 500$ the impact of jammers transmitting on the 868 MHz band have on the message success probability is much more important. This is due to the fact that the channel quality even without jammers is already very degraded. Indeed, the message success probability reached without re-transmissions and $\overline{G_{ju}} = 0$ is only 0.44. Consequently, to obtain a probability close to 1, a higher r is necessary. For instance, a value r of 8 is needed in order to get a message success probability of 0.98. As for the average number of re-transmissions, a behaviour similar to that with $N_u = 100$ is obtained. It increases as $\overline{G_{ju}}$ increases.

As for jammers transmitting on the downlink band, a similar behaviour as in the $N_u = 100$ scenario is obtained. Hence, the impact on the goodput is much less important and the message success probability stays constant regardless the $\overline{G_{ja}}$.

In summary, a LoRaWAN cell with $N_u = 500$ is much less resilient than a cell with 100 nodes. Indeed, as we have seen a much bigger r is needed to get a good message success probability when the uplink channels are jammed. As for the case of jamming on the downlink channel, a similar behaviour of that obtained in the less congested network is obtained. Thus, the impact on the number of re-transmissions (and hence the energy consumption) is very important and affects the energy consumption of nodes without increasing the network performance.

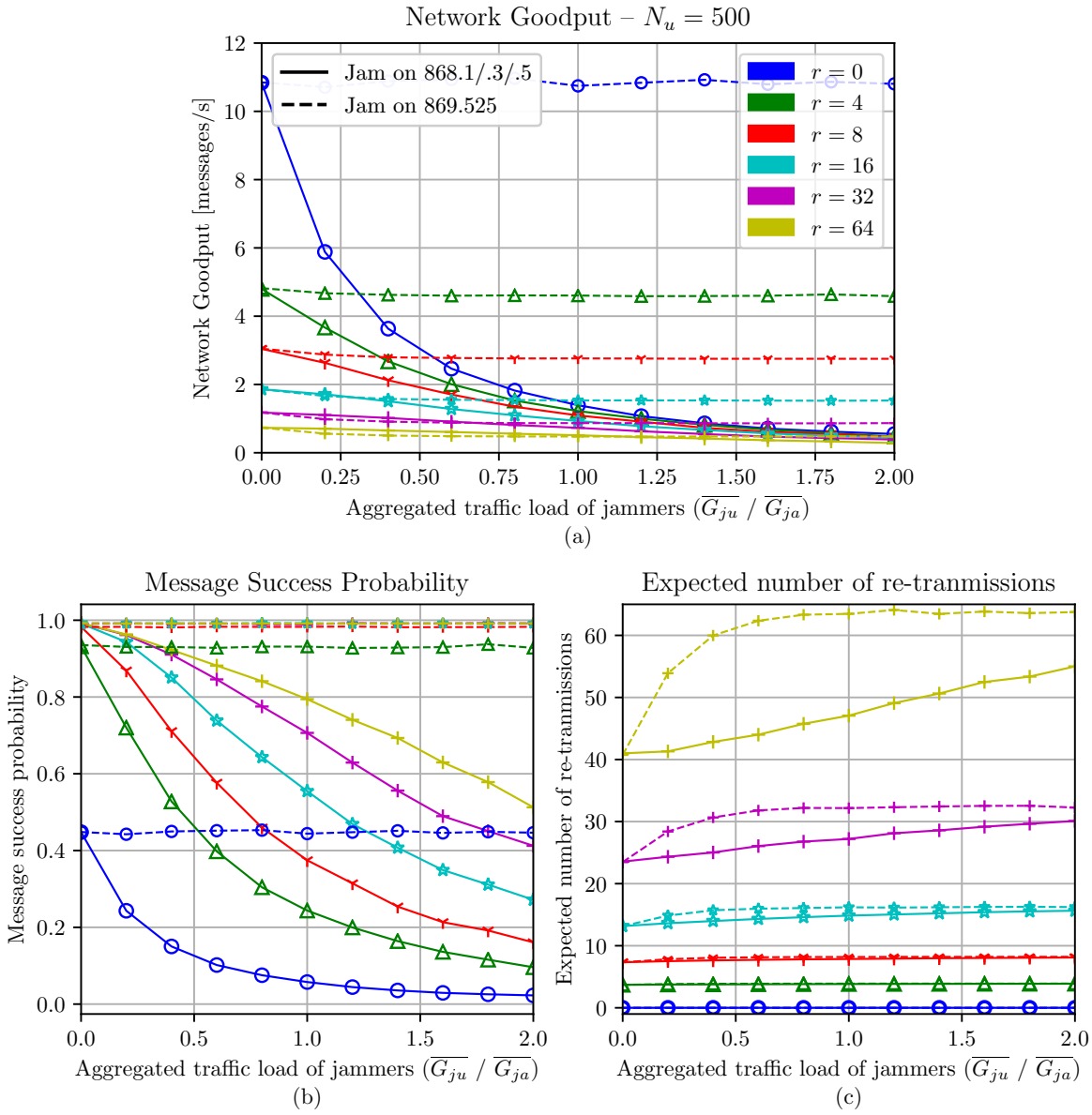


Figure 5.8: Realistic LoRaWAN cell under jamming with $N_u = 500$: (a) Network Goodput, (b) Message Success Probability and (c) Average Number of Re-transmissions.

5.4 Performance of LoRaWAN with gateway diversity

In the previous sections we have evaluated the performance of LoRaWAN with a single gateway by using the mathematical model and the ns3 module extension proposed in this thesis. In this section, we extend our work to scenarios with multiple gateways.

As depicted in figure 5.9 we consider a LoRaWAN deployment with two gateways and several nodes uniformly distributed around them. We evaluated three different performance metrics: (i) network throughput, (ii) message success probability and (iii) average number of re-transmissions[3].

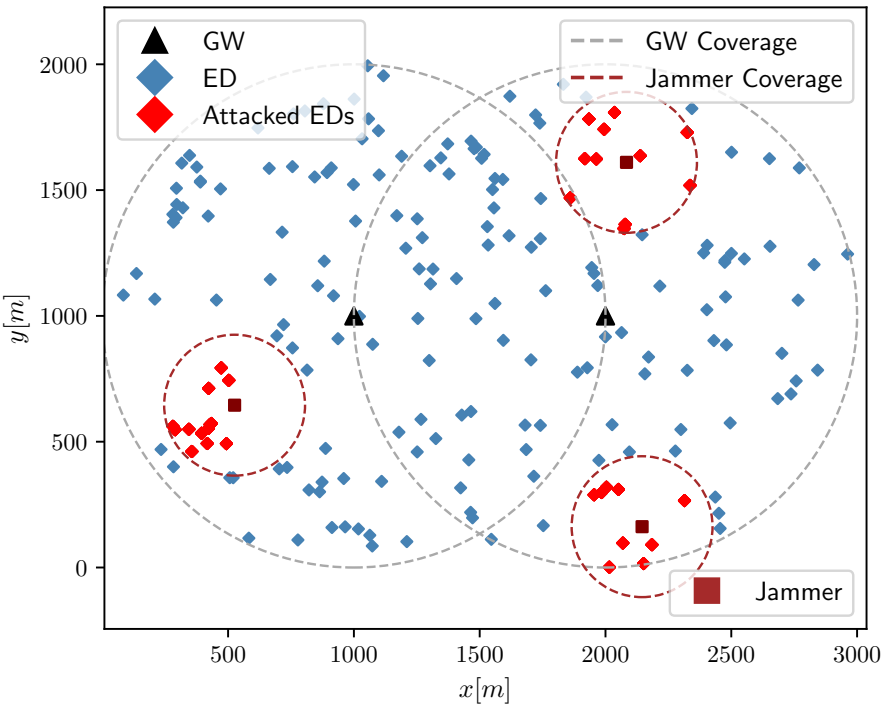


Figure 5.9: Simulation Scenario.

As in the other scenarios, we also included the possibility of having re-transmissions. However, since the previous results showed that in some cases, such a system causes more harm than good, we considered that only a portion of the traffic is acknowledged. Hence, some of the packets are treated as "priority packets" while others aren't. The specific details of this re-transmission system are described in Section 4.2.2.8. Therefore, two parameters define the way the system works: (i) the percentage of priority-packets (pr), which is the proportion of packets requiring an ACK and re-transmissions, and (ii) the maximum number of re-transmissions allowed for each priority packet (r).

Contrary to the previous scenarios, we consider that EDs transmit on a unique SF, which is set to 7. In that way the coverage is limited to a few kilometers, although it does not take advantage of LoRaWAN's capacity to have great coverage, it makes possible to evaluate in a delimited environment the effectiveness of using gateway diversity. Hence, as SFs are considered

[3]This scenario is also used for the ED current consumption evaluation presented in Section 5.7.

to be quasi-orthogonal, the results presented in this section can be easily extended to a more complex scenario that, for instance, considers multiple SFs, a high node density, and a bigger area deployment.

Additionally, for the case of jammers, in this simulation campaign jammers are capable of jamming both uplink and downlink channels. Thus, they are characterised by the aggregated traffic load ($\overline{G_j}$), which represents the traffic generated by jammers as a proportion of the total channel capacity (uplink and downlink included).

Then, as detailed in Table 5.3, two scenarios are considered: a baseline network aiming at evaluating a LoRaWAN network that is not under jamming, and a LoRaWAN network under the attack of jammers. Both of them implement gateway diversity ($N_{gw} = 2$) and packet re-transmissions. We consider a LoRaWAN network consisting of several EDs and multiple gateways. All EDs belong to Class A, the wireless channel connecting EDs and GWs uses the 868.1 MHz band with a bandwidth of 125 kHz for user packets and ACKs on RX1, and the 869.525 MHz band for ACK transmitted on RX2. The path-loss model considered is the outdoor path-loss as in the previous simulations.

EDs are uniformly distributed around GWs within an area of 2×2 km for the case of one gateway and 3×2 km for two gateways. As regards the ED's application profile we considered a packet length (l_u) of 50 bytes, and that all EDs, are configured to use a duty-cycle (d_u) of 0.01. The packet length l_{ack} of all ACK is set to 25 bytes. As for message re-transmissions, we consider that the maximum number of re-transmissions r is set to 16, and the percentage of priority packets pr is set to 10 % and 100 % respectively.

5.4.1 LoRaWAN without jamming and gateway diversity

In this scenario, we evaluated the performance of a LoRaWAN network with re-transmissions and gateway diversity that is not being attacked by jammers. Simulation results are depicted in 5.10.

Figure 5.10 (a) presents the Network throughput as a function of N_u. All nodes in the network have to contend for sending their data in an uncoordinated fashion. Hence, the goodput achieved is proportional to the injected traffic load. It can be seen that there is an inflection point for $N_u = 75$ and $N_u = 125$ respectively, where the curves stop growing. Hence, the network with gateway diversity is able to handle a much larger number of nodes.

Additionally, from Figure 5.10 (a) we can see that adding an extra gateway can significantly increase the network throughput without the need of changing the MAC protocol owing to the capture effect. In fact, the maximum throughput goes from 3 to 4 `[packets/s]`, for an N_u of 75 and 125 respectively. Therefore, a network with gateway diversity outperforms the single-gateway deployment.

As for the message success probability, presented in Figure 5.10 (b), we note that, when packets are not acknowledged systematically ($pr = 10\%$), it decreases rapidly by increasing N_u. Indeed, for $N_u = 25$, it goes from 0.7 and 0.8 respectively to less than 0.3 for $N_u \geq 200$. Conversely, when packets are re-transmitted systematically ($pr = 100\%$), the network can support a larger number of users, thus increasing their resilience. Indeed, an success probability

Table 5.3: LoRaWAN with Gateway diversity: simulation Parameters

	Parameter	No jamming	Jamming on uplink and downlink
Network	N_u	$1, 25, \ldots, 400$	50
	l_u	50 bytes	50 bytes
	l_a	25 bytes	25 bytes
	T_u	82.17 ms	82.17 ms
	T_{ack}	46, 33 ms	46, 33 ms
	d_u	0.01	0.01
	s	7	7
	Area	$2/3 \times 2$ km	3×2 km
	N_{gw}	$1, 2$	$1, 2$
	r	16	16
	pr	10 %, 100 %	10 %, 100 %
Jammers	l_j	-	50 bytes
	T_j	-	82.17 ms
	$\overline{G_j}$	-	$0.1 - 1$
	N_j	-	25
Path-loss	d_0	40 m	40 m
	PL_0	140.7	140.7
	n	3.12	3.12
	L_h	-4.7	-4.7
	h_{ED}	3 m	3 m
	σ	9.7 dB	9.7 dB
Band	Uplink and RX1	868.1 MHz	868.1 MHz
	Downlink (RX2)	869.525 MHz	869.525 MHz
Simulation time		10 h	10 h

of nearly 1 can be achieved for $N_u \leq 100$ with and without gateway diversity.

Figure 5.10 (c) presents the average number of re-transmissions. For all cases, we note that this number increases as the number of users rises. However, we can see that the transmission process is more efficient when multiple GWs are deployed in the network, this is because EDs do not need to re-transmit messages as often as it would be with a single gateway. Indeed, for the worst case scenario ($N_u = 400$, $N_{gw} = 1$) and considering $pr = 100\%$, the number of re-transmissions needed can be decreased by roughly 40% if two gateways are present in the network. This is because each message is re-transmitted 9.7 times on average instead of 16.

Finally, we note that having a portion of the traffic that is not acknowledged nor re-transmitted decreases the average number of re-transmissions. Indeed, for $pr = 10\%$ and $N_u = 400$, the average number of re-transmissions is 3.55 and 7.57 for two and one gateway respectively.

5.4.2 LoRaWAN with gateway diversity under jamming

In this scenario, we evaluate the network performance of LoRaWAN network with gateway diversity under jamming. As in the case of no jamming, we consider the possibility of having a portion of the messages that are not acknowledged nor re-transmitted. Similarly to the previous scenario, we consider the same three performance evaluation metrics: (i) network throughput, (ii) message success probability and (iii) average number of re-transmissions.

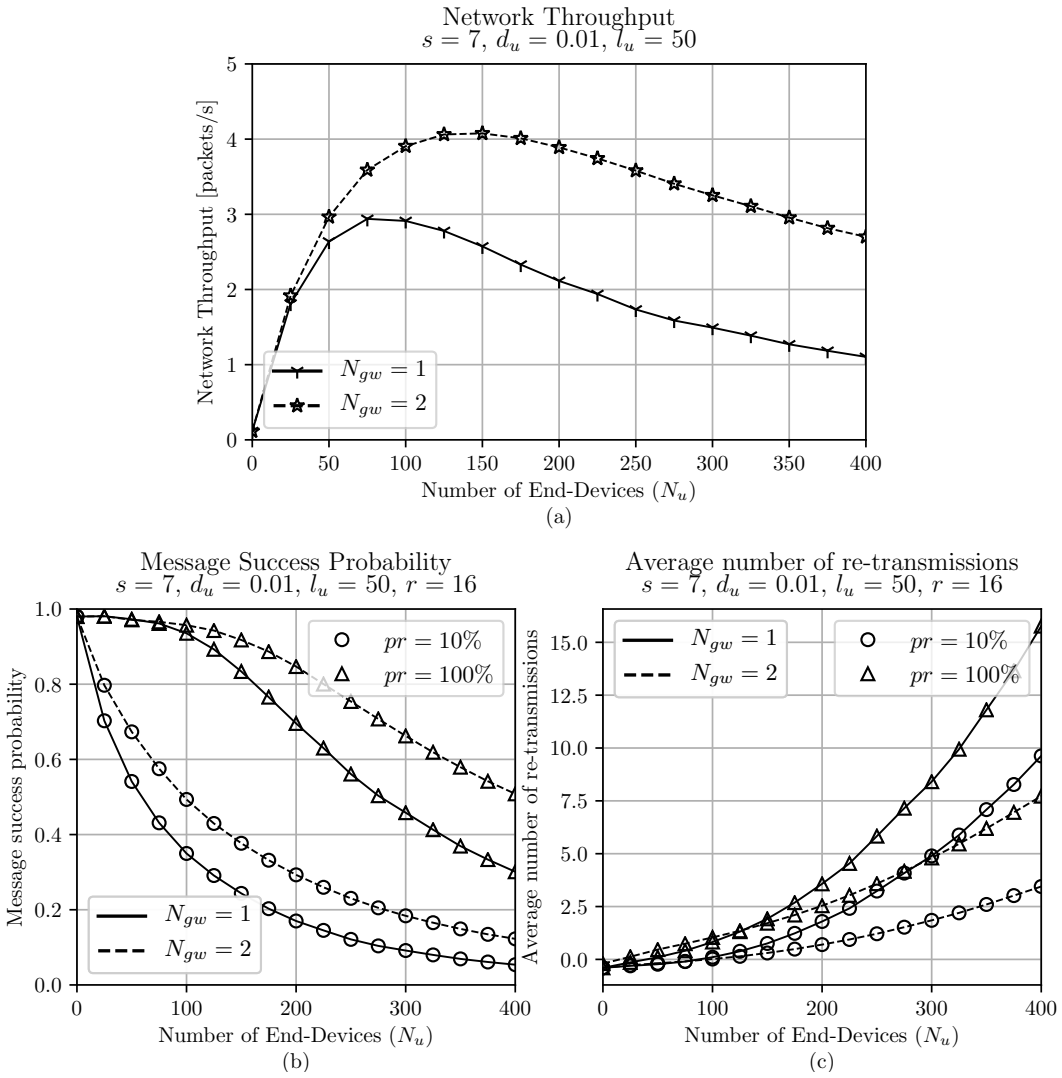


Figure 5.10: Performance evaluation of a LoRaWAN network with re-transmissions and GD: (a) Network Throughput, (b) Message Success Probability, (c) Average number of re-transmissions.

Figure 5.11 presents the achieved network throughput for $N_u = 50$ as a function of the aggregated jammers' traffic load ($\overline{G_j}$). Clearly, the greater $\overline{G_j}$ is, the lower the throughput is. However, this behavior can be attenuated by considering gateway diversity. For instance, for a $\overline{G_j} \leq 0.3$ we note that an enhancement of roughly 0.8 `[packets/s]`. Therefore, a network with gateway diversity is more resilient than a single cell.

Figure 5.12 presents the message success probability and the average number of re-transmissions with and without gateway diversity for $pr = 10\%$ and $pr = 100\%$ respectively. From Figure 5.12 (a) we note that for $pr = 10\%$ a success probability of 0.66 is achieved in the best case scenario ($r = 16, N_{gw} = 2$, $\overline{G_j} = 0$), which represents 31% less than in the case where all packets are systematically re-transmitted. This reduction is even more more explicit for all cases in which $\overline{G_j} \geq 0.5$, since only a success probability of less than 0.2 is reached. Better success probability is obtained if $pr = 100\%$ is employed. However, From Figure 5.12 (c) we

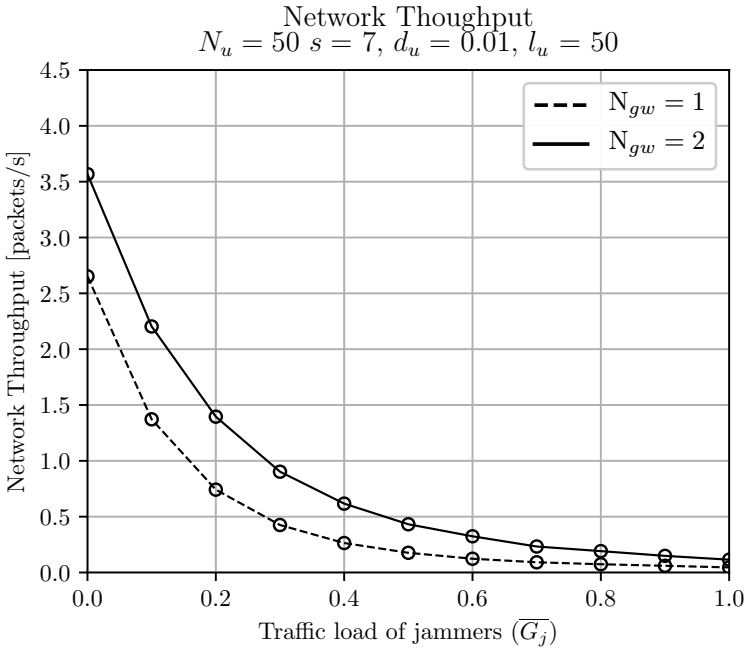


Figure 5.11: Network Throughput of a LoRaWAN network with Gateway diversity under jamming.

can see that it degrades rapidly when no re-transmissions, nor gateway diversity are considered. Indeed, for $r = 0$ and $N_{gw} = 1$, this probability falls from roughly 0.41 to about 0.2 at $\overline{G_j} = 0.1$. This reduction becomes even more important for $\overline{G_j} \geq 0.5$ where this number drops to nearly zero. In contrast, for $N_{gw} = 2$, we note that the success probability increases significantly in all cases. For instance, for $r = 4$ and $\overline{G_j} \leq 0.4$, this number increases by 100% percent. As for re-transmissions, we note that it is always beneficial to the system, if only the success probability is considered. Indeed, the higher r is, the higher the success probability will be. For instance, a success probability of about 0.7 can be achieved even with $\overline{G_j} = 0.4$, if $r = 16$ and $N_{gw} = 2$.

As for the the average number of re-transmissions, from Figure 5.12 (d) similarly to the case without jammers, we note that it is inversely proportional to the throughput achieved in the network. The average number of re-transmissions reaches its maximum when the channel quality is very poor (throughput ≤ 0.2 `[packets/s]`) and the ED is forced to make r re-transmissions. We also note that having multiple GWs can, owing to the capture effect, reduce the average number of re-transmissions. In fact, for the worst case scenario ($r = 16, \overline{G_j} = 1$), it can be reduced by 11% in comparison with the single gateway case.

On the contrary, when less packets are re-transmitted ($pr = 10\%$) as presented in Figure 5.12 (b), we note that the average number of re-transmissions is much lower. Indeed, for the worst case scenario ($\overline{G_j} = 1, r = 16, N_{gw} = 1$), EDs make 5.18 re-transmissions in average, which almost halves the case where all messages are systematically acknowledged. Additionally, contrary to the case where $pr = 100\%$ we note that adding an extra gateway reduces drastically the number of re-transmissions. Indeed, if we look at the case where $r = 4$ and $\overline{G_j} = 1$ the average number of re transmissions downs from 3.69 to 0.63 when there are two gateways.

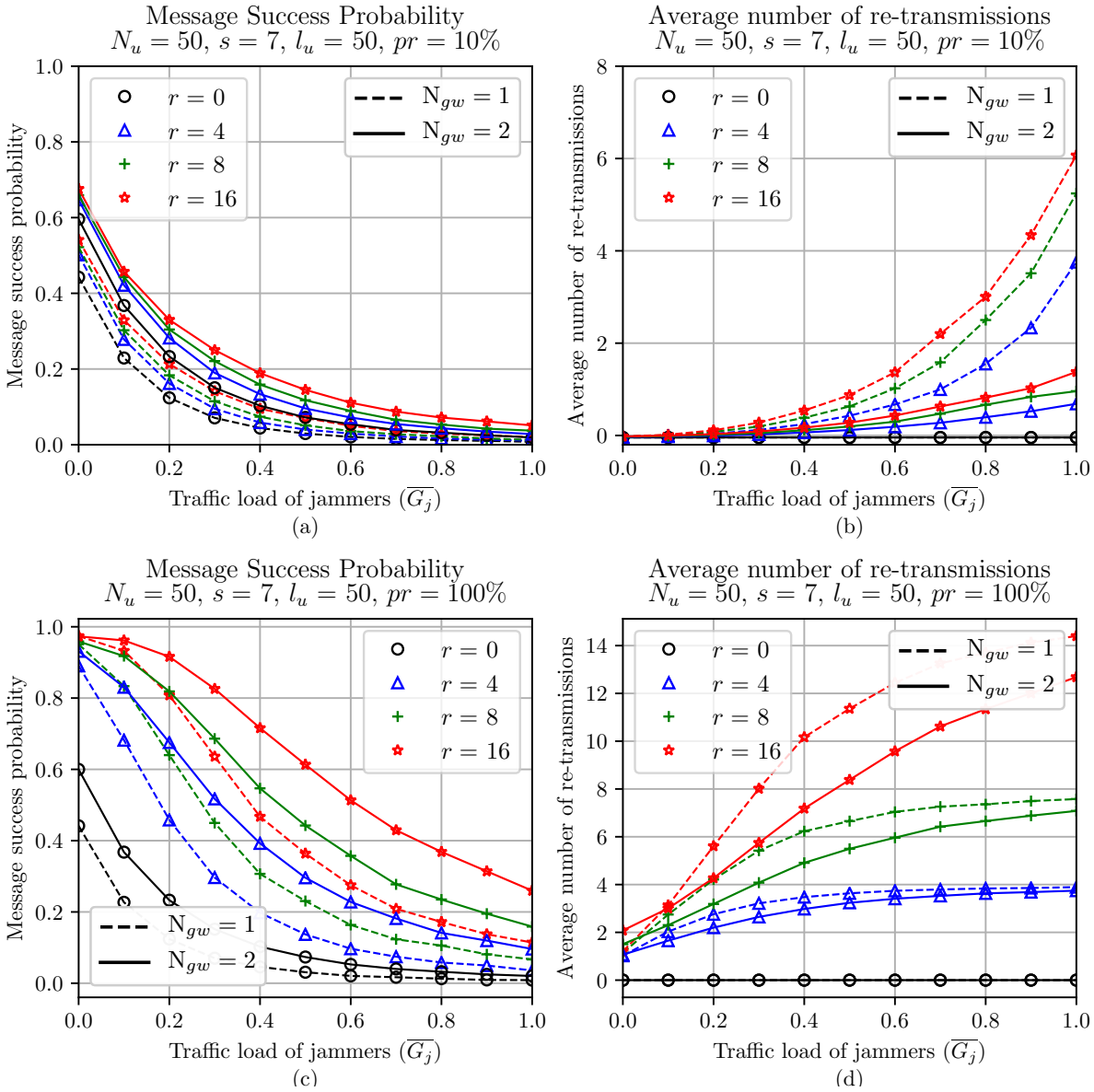


Figure 5.12: Performance evaluation of a LoRaWAN network with re-transmissions and Gateway diversity under jamming (a) Message Success Probability with $pr = 10\%$, (b) Average number of re-transmissions with $pr = 10\%$, (c) Message Success Probability with $pr = 100\%$, and (d) Average number of re-transmissions with $pr = 100\%$

5.5 Jammer strategy

In this section, we will explore the standpoint of a channel-oblivious jammer, and how it can maximise the damage it can do to the network with minimal effort. As we have seen from previous sections, there is a compromise between the jammers' traffic load and the network performance achieved in the network. This is an expected result because the network gets saturated. However, that does not tell the whole story of what happens in the system.

Performing a channel-aware jamming attack is quite simple, yet if we assume that jammers have a limitation on the amount of energy they can spend to jam the network, we can define a strategy in which transmission attempts can be selected intelligently. The strategy is then to select the packet length and the rate at which packets are transmitted that impact the most the network performance while keeping the jamming traffic load constant.

We call $\{\lambda_{ju} \cdot T_{ju}, \lambda_{ja} \cdot T_{ja}\}$ the traffic load per uplink/downlink jammer[4]. Therefore, we use our mathematical model to evaluate the performance of different networks configurations that are being attacked with a certain jammers' traffic load.

Then, we use the simulation scenario used for the ideal channel performance evaluation in 5.2. We recall that this scenario considers a users' duty cycle of 0.01, a uniform distribution of users amongst spreading factors ($7 < s < 12$), three uplink channels are (868.1/.3/.5 MHz) and only one reception window in the 868.525 MHz channel.

For this evaluation we consider a cell with $N_u = 500$. The cell is put under the attack of jammers with a given traffic load, while varying the packet length and number of re-transmissions. Results are reported in Figure 5.13.

As we can see, for the same traffic load, a different goodput is obtained. Indeed, for the case in which an uplink jammer transmits $8 - \texttt{byte}$ packets, the impact it can make is 8 times greater than that made with a size of 200 `bytes` for a traffic load of 0.2 and $r = 0$. Hence, from a jammer point of view, it is much better to transmit small packets more recurrently rather than large packets spaced out over time.

To explain this behaviour, we shall return briefly to the definition of the jammer traffic load G^s_{ju}, G^s_{ja} given in Eqs. (5.4) and (4.11). As we can see, it is directly proportional to the time on air $\{T_{ja}, T_{ju}\}$, which in turn depends on the packet length l_{ju}, l_{ja}.

$$G^s_{ju} = r^s_{ju} \cdot N_{ju} \cdot T^s_{ju} \tag{5.4}$$

$$G^s_{up} = G^s_u + G^s_{ju} \tag{5.5}$$

Then, selecting a lower l_{ju}, l_{ja}, while keeping the jammer traffic load constant, makes $\{\lambda_{ju}, \lambda_{ja}\}$ higher. Hence, this faster decrease for lower packet lengths stems from the effect

[4] T_{ju} and T_{ja} are defined as the jammer time-on-air weighted by the proportion of jammers transmitting in each SF ($T_{ja} \cdot \rho^s_{ja}$ and $T_{ju} \cdot \rho^s_{ju}$)

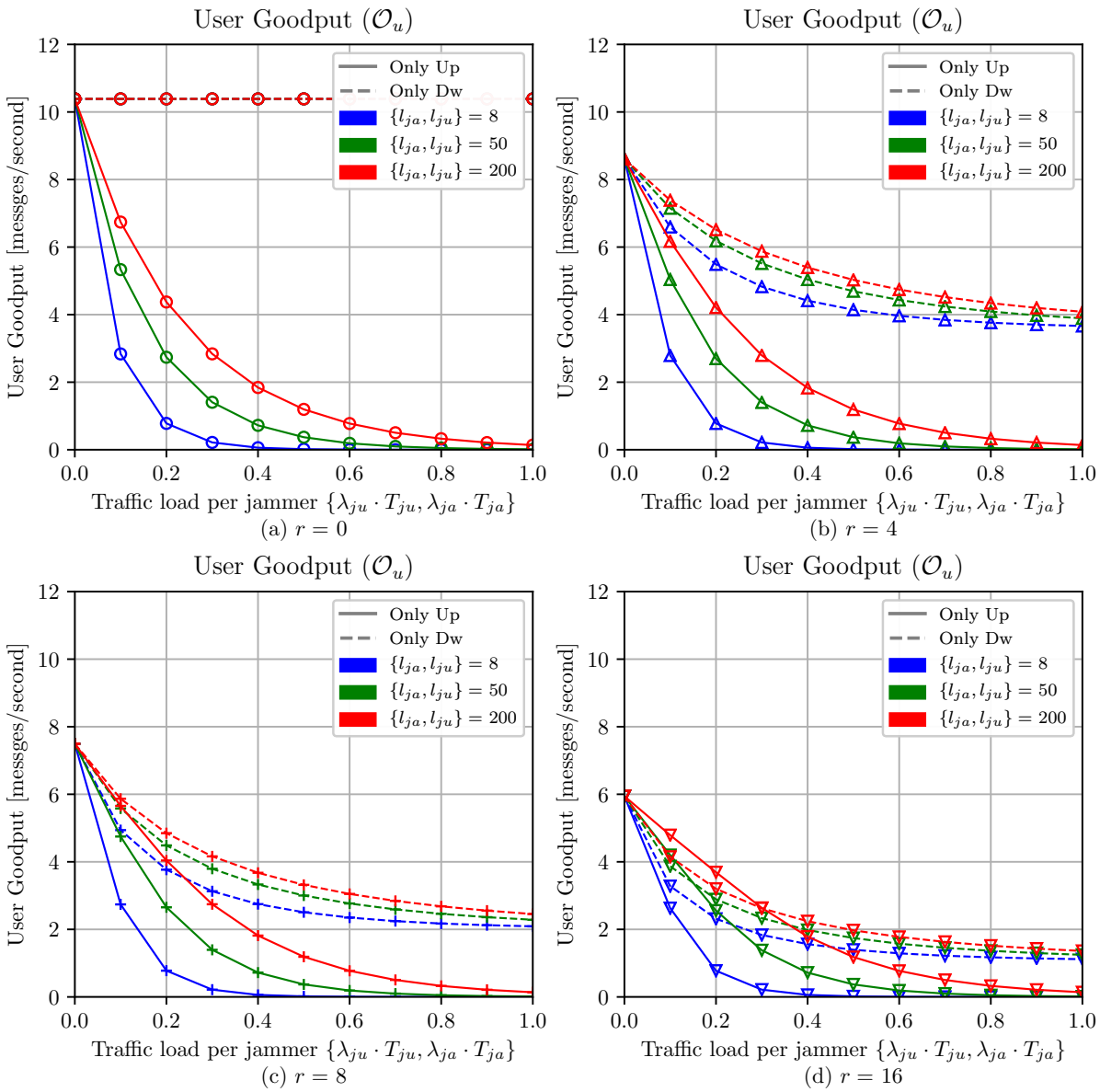


Figure 5.13: Performance impact of jammers when varying the jammer strategy: (a) $r = 0$, (b) $r = 4$, (c) $r = 8$ and (d) $r = 16$

of the jammers' packet generation rate on the packet success probability $\mathcal{P}_u$[5].

$$\mathcal{P}_u^s = P_{uu}^s \cdot P_{uj}^s \tag{5.6}$$

$$\begin{aligned} P_{uu}^s &= \exp(-r_u^s \cdot N_u \cdot T_{vu}^s) \\ P_{uj}^s &= \exp(-r_{ju}^s \cdot N_{ju} \cdot T_{vju}^s) \end{aligned} \tag{5.7}$$

Likewise, from Fig. 5.13 (b) and (c), it can be observed that the impact made by the jammers

[5]we recall that r_{ju}^s is the packet arrival rate per uplink jammer on a given channel, and SF s and is defined as $\rho_{ju}^s \cdot \lambda_{ju}/F$.

on the goodput is much more important when the network is allowed to have a greater r. This is because, as discussed above, the number of messages re-transmitted is much higher which makes the rate of new messages received less.

Another important point to note is the difference on the impact made by jammers transmitting on uplink or downlink. Indeed, as described earlier and contrary to the uplink case, the goodput does not drop to zero. Hence, the minimum point to which it reaches depends on r. Therefore, for $r = 0$, there is no impact, while for $r = 4$ and $r = 8$ the value reached is ≈ 4 [messages/s] and ≈ 2 [messages/s] respectively.

5.6 Geographical impact of jammers

Something that has not been sufficiently relayed in the results presented so far is the impact of the spreading factor distribution among users. In the ideal channel scenario we have assumed a uniform distribution, while in the subsequent scenarios EDs were configured to use the lower SF possible as a function of its link quality. Then, EDs deployed near the gateway will transmit on an SF 7, while those on the periphery will use an SF 12.

Therefore, it is worth studying the performance of each particular node. To do that, we well reuse the realistic channel scenario presented in 5.3, and we will study the performance impact of both channel-aware and channel-oblivious jammers from a geographical perspective. Hence, instead of looking at the average goodput achieved in the network, we will be interested in the goodput achieved per user.

We recall that this scenario considers a users' duty cycle of 0.01, three uplink channels (868.1/.3/.5 MHz) and two reception windows, the first one is opened in the same channel as the uplink transmission and the second in the 868.525 MHz channel with an SF 12. We set the number of users to 1000.

The SF distribution is then dependent on the Path-loss considered. As in the other cases we use the outdoor path-loss model with parameters reported in table 5.2. Hence, according to the path-loss model and the uniform distribution of nodes, SF are distributed as follows: $\rho_u = \{0.33, 0.22, 0.1, 0.09, 0.19, 0.07\}$ (from 7 to 12).

In regards to jammers, we consider three cases: (i) 10 channel-aware jammers (ii) 10 channel-oblivious jammers jamming the uplink channels with ($\overline{G_{ju}} = 1$), and(iii) 10 channel-oblivious jammers with ($\overline{G_{ju}} = 0.5$).

The channel-aware simulated is the triggered jammer. As described in detail in Chapter 3, this jammer detects any activity in the channel and then launches an attack without taking the time to read the information on the packets. For that it uses the the CAD functionality, which allows it to detect any activity without the need of reading a whole preamble. Then, the triggered jammer simulated is able to sense the medium on every SF and channel in parallel and to transmit unauthenticated packets by using the same transmission parameters (SF an channel) as uplink transmissions.

Results are presented in figure. 5.14, it depicts a linear interpolation of the throughput (measured in $packets/h$) achieved by each EDs in a LoRaWAN cell from a geographical point of view. We present this metric for the baseline case with no jammer and for the cases in which the cell is under attack.

In figure 5.14 (a), we have the baseline cell with no jammers. As expected, we can see that EDs that are close to the GW achieve the highest throughput. Inversely, EDs deployed in the periphery presents a much lower performance. This is due to the fact that EDs are deployed so that the best possible SF is selected based on the reception sensitivity of the GW.

For the reactive jammers presented in figure 5.14 (b), it is interesting to note that the performance impact caused by this class of jammer is highly correlated with its geographical position. Indeed, EDs that are close enough to jammers experienced a throughput reduction

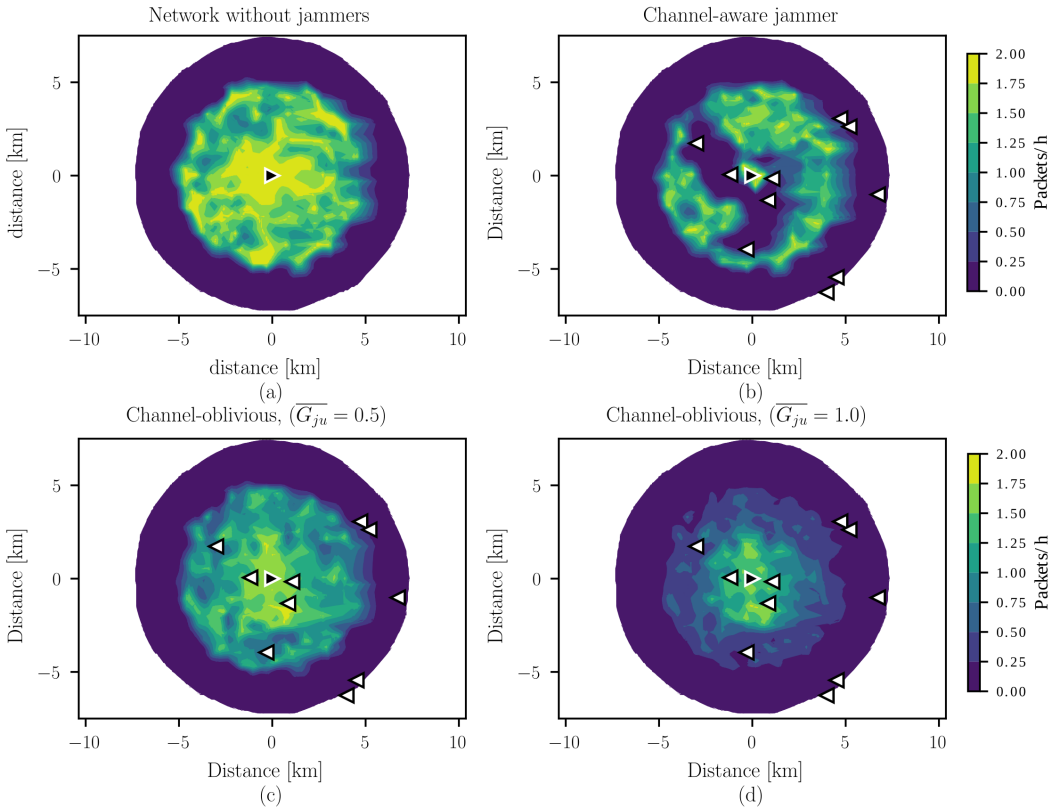


Figure 5.14: Throughput per user ($packets/h$) of a LoRaWAN network under jamming attacks, ▶ represents the GW and ◁ represents a jammer.

of 99.8%, which is consistent with experimental implementations as presented in [104].

Finally, for the jammer-oblivious cases presented in figure 5.14 (c) and (d), the network performance decreases widely. In the case of periodic jammer, we can see that the throughput achieved by all EDs is decreased by 16.6% when adding 10 periodic jammers with $\overline{G_{ju}} = 0.5$. Similarly, for jammers jamming 100% of the time, this reduction is even bigger as the performance is reduced by 37% in average.

5.7 Current consumption and battery lifetime

Until now, we have focused our attention on the evaluation of network-level performance metrics. In this section we will evaluate another aspect, which is of special importance in LoRaWAN networks, the current consumption of end devices. For that, we will use the ns3 module described in Section 4.2, and more specifically the current consumption class described in detail in Section 4.2.2.4, we will also give an estimation of the battery life of a battery based end device.

As discussed before, our current consumption model is based on the PHY state machine presented in Figure 5.15. Hence, total current consumed by a given EDs depends on the time it spends on each state. For our simulation we consider the current consumption values presented in table 5.4. We consider the SX1272 [14] module with a transmission power of 14 dBm and a modulation scheme a coding rate of 4/5.

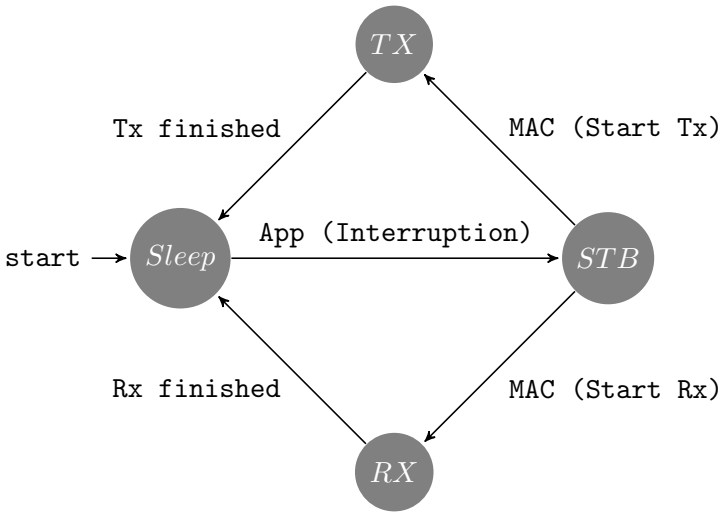


Figure 5.15: Finite state machine for the End Device physical layer

Table 5.4: Current Consumption for different device-states

State	Current consumption (I_i) [mA]
Tx	83
Rx	32
Standby (STB)	32
Sleep	45×10^{-3}

Based on these current consumption levels, we compute the average current consumption

per sent packet (I_{avg}) during a given session:

$$I_{avg} = \frac{1}{T_{app}} \sum_{i=1}^{N_{states}} T_i \cdot I_i \tag{5.8}$$

where T_{app} is the application time in seconds, defined as the time between two consecutive packet transmissions, N_{states} is the number of states an ED passes to transmit a packet, T_i is time duration spent in state i and I_i is the current consumption in mA/s in state i.

Therefore, if we only consider the transmission state, we can easily see that the current consumption of a given packet depends on its ToA, which in turn depends on the length and SF of the packet. Therefore, a comparison of this current consumption is presented in Figure 5.16.

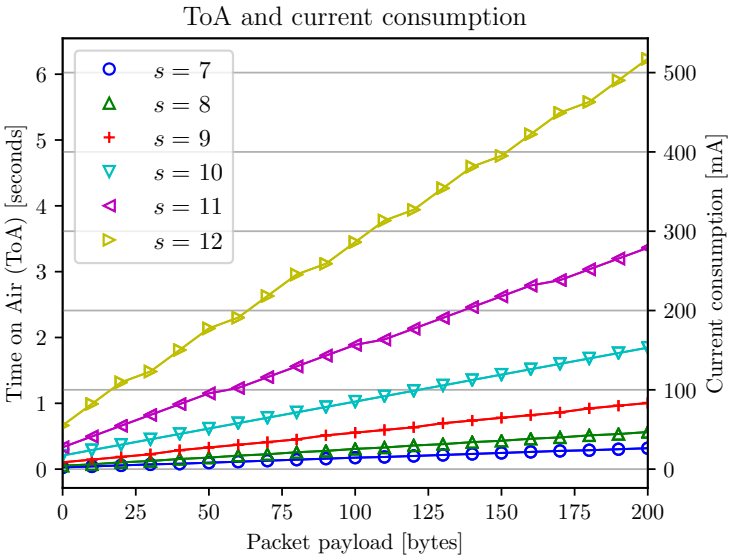


Figure 5.16: Time on Air and current consumption as function of the payload

However, as we have seen in the previous sections, to send a single message, some times EDs are required to re-send it if the network is congested or jammed. Thus, in the following we will consider the LoRaWAN with realistic channel scenario presented in Section 5.3 in order to estimate the current consumption per message. As in the previous section we will first present a standalone deployment without jammers and then we compare it with a network under attack. Then, this results will be extrapolated to estimate the battery lifetime.

5.7.1 Current consumption

In the following we well present the results obtained in the simulation campaign presented in Section 5.3. In addition to the performance metrics presented before, we also extract the mean current consumption per message. It includes all the states (STB, TX, RX, Sleep) through which an ED had to pass to send each package, and also includes the number of times it had to send that message.

Figure 5.17 depicts the results obtained for both scenarios, a LoRaWAN network without

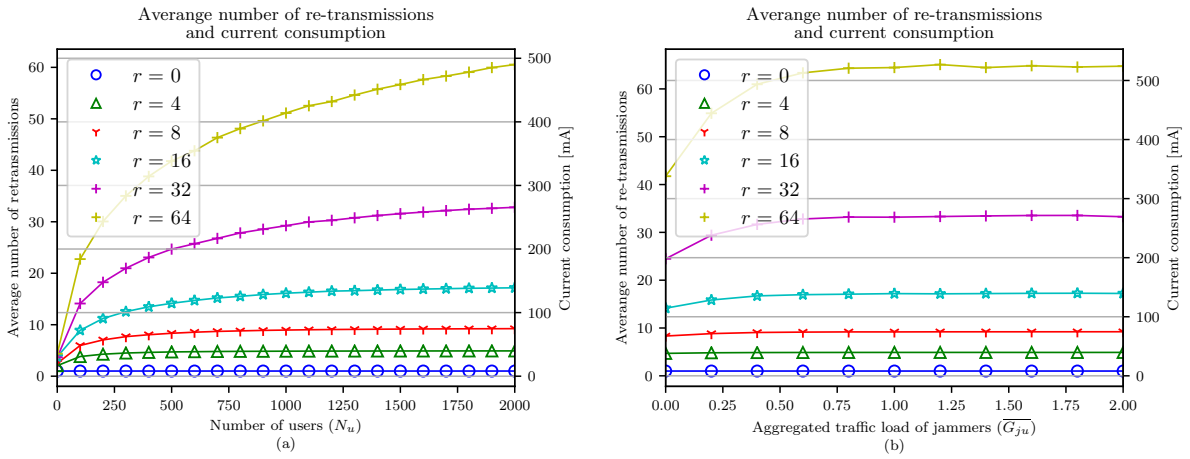


Figure 5.17: Average current consumption and number of re-transmissions: (a) no-jamming, (b) jamming on uplink.

jamming and a LoRaWAN network with channel-oblivious jammers jamming on the uplink channels. It represents the average current consumption of a given ED as a function of the maximum number of re-transmissions (r) allowed in the network.

From Figure 5.17 (a) we can see that, as expected, the current consumption is inversely proportional to the channel quality of the network, and that adding packet re-transmission increments it drastically. In fact, for the worst case scenario ($r = 64, \overline{G_{ju}} = 2$), the current consumption per message is 518 [mA], when each message is re-transmitted 60 times in average.

Similarly, when the network is under the attack of jammers as in 5.17 (b), we note that the average current consumption increases as the network gets saturated and that it is also directly proportional to the maximum number of re-transmissions allowed in the network. In fact, for the worst case scenario ($r = 64, G_j = 1$), the current consumption per message is 486 [mA], when each message is re-transmitted 60 times in average.

It should be noted that, the values presented in Figure 5.17 where computed based on a packet length of $l_u = 50$ `[bytes]` and that they are mean values. Hence, a single ED will have more or less current consumption according to its SF. Therefore, according to the path-loss model and the uniform distribution of nodes, SF are distributed as follows: $\rho_u = \{0.33, 0.22, 0.1, 0.09, 0.19, 0.07\}$. Therefore as we can see in a regular deployment most of the EDs are configured to use an SF 7, and will eventually have less current consumption than that reported in figure 5.17, whereas EDs deployed in on the periphery (with bigger SFs) will have a much bigger current consumption. Therefore, the above presented graphics can be only seen as reference values.

5.7.2 Battery lifetime

This section presents an estimation of the battery lifetime of ED considering the model presented in Section 4.2.2.4. As pointed out before, we first model the current consumption of the communication unit, which only considers the LoRaWAN transceiver. Then, the battery

lifetime is obtained from Equation (5.9) for $C_{battery} = 2400$ mAh and I_{avg} extrapolated from the previous simulation scenarios.

To estimate the lifetime of a battery-operated ED, we considered a linear function of the battery capacity ($C_{battery}$) in mAh:

$$T_{lifetime} = \frac{C_{battery}}{I_{Avg}} \tag{5.9}$$

We consider different network configurations: multiple gateways ($N_{gw} = \{1, 2\}$), two SF ($S = \{7, 12\}$), different re-transmissions settings ($r = \{0, 4, 8\}$, $pr = 100\%$), application profiles ($T_{app} = \{30\text{ s}, 10\text{ min}, 1\text{ h}\}, l_u = 50$ `bytes`). As for the jammers' traffic load we set it as $G_j = \{0, 0.2\}$.

As presented in Table 5.5, we first consider a LoRaWAN network with and without packet re-transmissions and gateway diversity that is not being attacked by jammers. For this case, a battery lifetime of 14.97 months can be achieved while sending messages very frequently ($T_{app} =$ 10 min) with an $s = 7$, and up to 76.94 months for infrequent communications ($T_{app} = 1$ day). Having a higher spreading factor reduces drastically the battery life. Indeed, these figure can be reduced to to 0.84 and 72.57 months respectively.

Table 5.5: Battery Lifetime

Network parameters	Battery Lifetime [months]					
$\{r, \overline{G_j}, N_{gw}\}$	$T_{app} = 10$ min		$T_{app} = 1$ h		$T_{app} = 1$ day	
	$s = 7$	$s = 12$	$s = 7$	$s = 12$	$s = 7$	$s = 12$
$\{0, 0, 1\}$	14,97	0,84	72,17	30,65	76,94	72,57
$\{0, 0.2, 1\}$	12,90	0,70	71,24	27,36	76,89	71,72
$\{0, 0, 2\}$	22,10	1,39	74,08	40,39	77,03	74,34
$\{0, 0.2, 2\}$	17,85	1,05	73,11	34,85	76,98	73,45
$\{8, 0, 1\}$	2,25	0,11	49,66	5,87	75,42	51,24
$\{8, 0.2, 1\}$	2,15	0,10	48,79	5,61	75,34	50,40
$\{8, 0, 2\}$	2,76	0,13	53,22	7,10	75,74	54,69
$\{8, 0.2, 2\}$	2,50	0,12	51,50	6,47	75,59	53,03
$\{16, 0, 1\}$	1,14	0,05	36,61	3,05	73,76	38,36
$\{16, 0.2, 1\}$	0,92	0,04	32,36	2,46	72,95	34,08
$\{16, 0, 2\}$	1,02	0,05	34,36	2,73	73,35	36,09
$\{16, 0.2, 2\}$	1,30	0,06	39,19	3,47	74,17	40,93

Regarding the scenarios where the network is under jamming, it can be clearly seen that the battery life is reduced as ED are forced to make more re-transmission in average. For instance, the battery lifetime of a network with $r = 16$ and $T_{app} = 1$ h, is reduced from 36.61 to 32.36 months when the jammers' traffic load (G_j) is set to 0.2 for an ED using an SF of 7. This reduction becomes even more important when the ED uses an SF 12, in which case the battery life time is only 3.05 months in the case of no jamming and 2.46 when there is jamming.

An enhancement of the battery lifetime can be achieved if gateway diversity is implemented. A network with $r = 8$, $T_{app} = 1$h $G_j = 0.2$ is a case point. In such case, having gateway diversity increases the battery lifetime of 5.55% in average, which for an application profile with $T_{app} = 1$ h

represents a gain of 2.71 months.

In summary, we have shown that a battery of 2400 mAh can achieve a battery life time up to 73.45 months if EDs make very infrequent transmissions, if gateway diversity and if an SF 7 is used. On the contrary, if a network with a re-transmission scheme ($r = 16$) is moderately jammed, and an SF 12 is employed instead a battery lifetime of a few days is achieved.

5.8 Summary and Conclusion

In this chapter we have presented an extensive performance evaluation analysis of LoRaWAN under jamming, we have evaluated several scenarios and performance evaluation metrics: (i) a LoRaWAN network with ideal channel, (ii) a LoRaWAN network with realistic channel and hardware limitations, (iii) a LoRaWAN Network with gateway diversity, (iv) jammer strategy, (v) geographical impact of jammers and (vi) energy consumption and battery life.

For the ideal channel evaluation in section 5.2 we have presented an evaluation that considers both our analytical model and an ns3 simulation campaign. From that performance evaluation, we have concluded that our mathematical model is valid and that it closely follows the results obtained with the simulator if the two simplifying hypotheses are considered in the simulation. However, we noted that if we compare the results with a model considering the capture effect, and two reception windows, as stated in the LoRaWAN specification, it is only valid for lower user traffic loads.

Then, in section 5.3 we focused on the performance of LoRaWAN considering a real channel and gateway hardware limitations. Contrary to the previous evaluation, we consider that the Gateway does not support transmitting 18 packets in parallel (3 channels and 6 SFs), but 8 reception paths instead. For this evaluation we also considered a non uniform distribution of EDs. Therefore, we note that, for a LoRaWAN standalone network, the fact of using only one SF factor on the downlink band has a huge impact on network performance when considering re-transmissions. This is due to the fact that the Gateway expends too much time processing ACKs and it gets saturated very fast. As for the network performance considering channel-oblivious jamming, we have proven that it has a significant impact on the goodput as well as on the probability of success per message. This impact is directly related to the number of nodes present in the network and the traffic load injected by jammers. We also identified that if jammers transmit on the downlink channel and there is a re-transmission mechanism, EDs are forced to re-transmit messages much more effectively resulting in a decrease in goodput.

Later, in section 5.4 we present an evaluation of a LoRaWAN Network with gateway diversity under jamming. For this simulation campaign it is considered a simpler environment where only one SF is supported and with a smaller number of nodes. Additionally, we also consider the possibility of dividing the EDs into two groups, some considered as a priority where all messages are systematically acknowledged and a second group that does not require it. The results of this scenario showed that, adding an additional gateway makes the performance of the network to improve considerably, and also allows the network to be much more resilient, which results in EDs with lower energy consumption.

Then, in section 5.5, we focus on the impact that channel-oblivious jammers can have if they choose intelligently their transmission parameters and not only on the traffic load it injects into the network. We have proven that, they can maximize their impact on the network if they sends small packets oftenly instead of long packets spaced in time.

Later in section 5.6, we focus on the performance of a network under the attack of channel-aware jammers. For that, we evaluated the goodput from a geographical point of view. Hence, instead of plotting the average goodput of the network we have focused on the goodput achieved by each ED. The results of this evaluation lead us to conclude that in a regular LoRaWAN deployment most of the nodes are configured to transmit on lower spreading factors. This causes the nodes deployed on the periphery of the network to get a much lower goodput, this when considering jamming causes these nodes to be excluded from the network.

Finally, in section 5.7 we evaluated the current consumption and battery lifetime of battery-based LoRaWAN nodes. For that used the ns3 module described in Section 4.2 and the previous simulation scenarios. The results of this evaluation lead to the conclusion that the current consumption is inversely proportional to the channel quality of the network when considering a re-transmission system. We have also noted that adding gateway diversity is a very efficient way to reduce current consumption as it allows ED to transmit less packets per message. Additionally, in regards to the battery life we saw that this varies a lot depending on the SF the ED uses, it can go from 4 months for a SF 12 to 72 months for a SF 7 depending on the application profile, the network conditions and the type of jamming considered.

From this performance evaluation we conclude that LoRaWAN networks have several security flaws that makes them vulnerable against jamming attacks:

- Gateways do not make any type of filter and send all packets to the network server as they arrive.
  - The gateway spends a lot of time reading messages that do not belong to the network, this makes legitimate packets to be dropped, which decreases network performance.
- The SF selected by the ED is made according to its link quality with the Gateway. Hence, most of the EDs are configured to use lower SFs.
  - The goodput obtained by the nodes of the periphery is much lower than that of those near the gateway.
  - If the network is attacked by jammers, the nodes on the periphery may be excluded from the network as they are the most vulnerable due their long ToA.
- The use of re-transmissions can be more harmful than beneficial if it is done systematically.
  - If the network is attacked in the downlink, EDs are forced to re-transmit many messages which causes the battery lifetime to be significantly reduced.
- Use of a single SF in the second reception window.

  - Using a single SF 12 in the download band can be a big bottleneck that decreases network performance and its resilience.

- Gateway Hardware limitations.

  - The maximum number of transmission paths supported by a gateway is a factor to consider when designing a LoRaWAN network, as the performance achieved and resilience against jamming is largely determined by this parameter.

Therefore, in the next section we will explore different countermeasures aiming at mitigating the above mentioned limitations in order to improve the resilience of LoRaWAN.

Chapter 6

Jamming countermeasures

Until now we have presented an extensive analysis of the impact of jamming attacks to the network performance. It has been done by using a novel mathematical model and a ns-3 simulation approach. These analyses have derived in several conclusions regarding different security flows in the LoRaWAN specification.

In this chapter we present countermeasures that deal with possible radio jamming scenarios aiming at informing and familiarising the reader with the most effective countermeasures against jamming. Hence, in the following subsections we review security schemes proposed in the literature to address this issue. Similarly as done in [142] for the WSN case we classify the countermeasures into three categories:

i) Detection techniques.

ii) Proactive countermeasures.

iii) Reactive countermeasures.

The relevant advantages and disadvantages of each method are highlighted in the following sections. Then, in Section 6.4 we apply and evaluate two jamming detection mechanisms based on statistical analysis and Machine Learning (ML). The counter measure is developed based on data obtained from the simulation scenarios presented in the performance evaluation in chapter 5 and from actual measurements from the test-bed described in 6.4.4.2.

6.1 Detection techniques

The main purpose of these techniques is to detect jamming activity as soon as possible. Hence, it should be implemented in the lower stack levels. Along with other countermeasures, they enhance the network resilience against jamming. These techniques can be classified based on the type of algorithm they use:

- Statistical approach and control theory
  - moving average, skyline, Statistical Process Control (SPC) analysis.

- Machine Learning.
  - classification algorithms: decision tree, k-nearest neighbors, random forest.
  - reinforcement learning
- Nature inspired techniques
  - artificial bee colony
- External decision agents

In order to detect jamming, all algorithms use information gathered from the network. It can be obtained by the nodes itself (i.e EDs, Gateways, Network Servers) or by external nodes specifically deployed for that purpose. Previous works have used the following metrics as presented in table 6.1:

1. IAT: it refers two the time between to subsequent received frames, it is normally recorded at the gateway or network/application server side.
2. Received Strength Signal (RSS): it is dependent on the link quality and is collected at the gateway.
3. Packet Data Rate (PDR): it measures the number of packets well received at the gateway side.
4. Bad Packet Ratio (BPR): it measures the number of packets unsuccessfully received at the gateway side, normally it refers to bad checksum or broken preambles at the gateway.
5. Data frame: it refers to the bits contained in the frame itself, it is to detect statistical differences between frames.
6. Packet Losses (PL): it refers to the difference between the number of packets sent and received.
7. Energy Consumption Amount (ECA): it refers to the energy consumption in a given window time. It is useful to determine whether or not a given ED consumes more energy than usual.
8. Carrier Sense Time (CST): is the detection of activity time in networks implementing CSMA algorithms.
9. Channel Busy Ratio (CBR): it represents the percentage of the time the RF channel is occupied.
10. Inactivity Time (IT): it represents the percentage of the time the RF channel is free.

Previous works proposing different jamming detection algorithms can be found in the literature, table 6.1 summarises some of them. As we can see, there is only one work on jamming detection for LoRaWAN to date. It was proposed by Danish in [143]. It is a jamming detection algorithm for join-accept packets that utilises the Hamming distance of consecutive packets.

Table 6.1: Detection of jamming attacks - relevant previous works

Ref.	Algorithm used	Metric employed	Type of jamming	Technology	Validation
[143]	Hamming distance detection	Data-frame	CA	LoRaWAN	Real implementation
[144]	Statistical analysis (EWMA)	IAT	CA and CO	WSN	Real measurement database
[145]	ML (decision three, random forest, Gaussian SVM)	RSSI	CA and CO	WSN (802.11)	Real measurement database
[146]	ML (random forest)	CBR, PDR and IT	CO	VANET (802.11)	Real implementation
	Error simple acquisition	RSSI	CA	WSN	Real implementation

They developed a real test-bed to validate their proposal. The algorithm is able to detect those attacks with a 98% detection rate with 5% false alarm rate.

In contrast, several works have been done on jamming detection WSN. In [144] an statistical approach that uses the Exponentially Weighted Moving Average (EWMA) algorithm to detect jamming attacks in WSN networks is presented. The algorithm was tested by using a previous jamming attack implementation, which results are public available through an online data-set [147]. They use as evaluation metric the IAT. The algorithm can be used for both channel-aware and channel-oblivious jammers. According to the authors, the algorithm is able to detect jamming attacks with a 100% detection rate.

Another work on jamming detection for WSN network is the one presented in [145]. The authors present a comparison of several ML algorithms: decision tree, random forest and Gaussian Support Vector Machine (SVM). The authors use a data-set of a real indoor deployment [148]. The evaluation metric used was the RSSI. According to the authors the proposed detection algorithms report accuracy between 89.7 and 99.06% with a false positive percentage ranging from 0.61% to 5.89%.

In [146] a jamming detection algorithm for Vehicular Ad-Hoc Network (VANET)s based on ML is presented. The authors have conducted static indoor and mobile outdoor experiments. Then, they collect the data and run their algorithm in a post-processing phase. The algorithm employed is the random forest and they used the CBR, PDR and IT metrics. Their proposal can be applied to both channel-aware and channel-oblivious jammers. Therefore, they conclude that the proposed algorithm provides an accuracy of 97% in the best cases.

6.2 Proactive countermeasures

A proactive countermeasure aims at making the network less vulnerable to jamming attacks instead of reactively respond to the attack one it has already happened. Hence, they can be implemented at the software or hardware level.

Most of the networks have several proactive countermeasures implemented by default on its design. In the case of LoRaWAN, as we have seen, it implements application-level encryption and network level authentication. However, this is not enough, because as we have seen the Aloha-type protocol and the fact that gateways acts are pure relays make the network vulnerable. This makes it possible to say that there is a lack of proactive countermeasures that could improve security within the network. The following are some proactive countermeasures ideas

that can be implemented in the network and that should improve its resilience against jamming attacks:

i) Authenticated preambles.

ii) Frequency Hopping.

iii) CSMA MAC protocol.

6.2.1 Authenticated preambles

As we have seen, one of the main weaknesses of LoRaWAN are gateways acting as pure relays. This is because, the gateway expends lots of time processing packets that might not come from the nodes it serves. A good way of preventing that is to filter those packet at the gateway level. However, a part from computing the cyclic redundancy check (CRC) at the end of each packet gateways do not perform any filtering of packets arriving from EDs. Hence, they relay all the packets as they arrive. Therefore, an easy way of detecting malicious packets before even finishing the transmission could be adding a verification method at the physical layer.

This mechanism has already been implemented in a LoRaWAN deployment in a previous work in [149]. It consists in using authenticated preambles so that malicious packets can be rejected sooner. Thus, saving energy and so guaranteeing network's long-term availability. Figure 6.1 (a) shows the packet structure as used in LoRaWAN, while 6.1 (b) shows the authors' proposed packet structure.

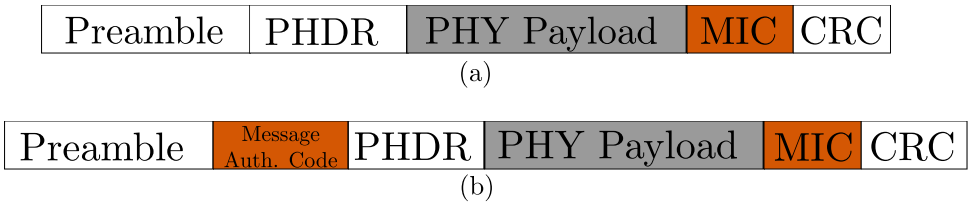


Figure 6.1: LoRaWAN Frame with authenticated preambles.

As we can see from figure 6.1 the message integrity code (MIC) is generated using the payload of the message it accompanies. Hence, in a regular LoRaWAN frame it comes right after the PHY payload. Consequently, it is too late in the case of a very long jamming packet to do a successful detection. Therefore, the authors propose to generate an authentication code that is known at the start of the reception frame and that is different at each frame.

Adding such a system requires to create a sort of extra authentication procedure aiming at exchanging the symmetric keys used to perform this second authentication procedure. The authors propose a token exchange scheme that is performed each time the ED is rebooted.

The authors have concluded that implementing preamble authentication at the gateway side prevents malicious attackers to significantly impact the operation of a network. However, adding such a system may impact the current consumption of nodes, this is because EDs have to perform cryptographic computations more often than in a regular LoRaWAN deployment. According to the authors this systems only adds a 4% energy overhead.

6.2.2 Frequency Hopping

As we have seen, in a standard LoRaWAN deployment each ED could be co figured to transmit on a single channel throughout given session. Hence, its behaviour could be easily predicted if a sniffer stores several packets from the medium. To reinforce the network robustness, a possibility is to have multiple channels so that EDs can randomly select an uplink channel each time an uplink transmission is required. In this way, the jammer's task of detecting and intercepting a transmission becomes more difficult.

However, the selection of the channel must be random so that transmission attempts cannot be predicted. The authors in [150] propose to select each frequency hop from a random sequence generator that uses the AES-128 algorithm in CTR mode. This algorithm, as described in Section 3.1 is a symmetric encryption algorithm that is already supported by compliant LoRaWAN EDs.

The above mentioned work was implemented in a controlled test-bed that considers several LoRa EDs, a gateway and a set of jammers. The results show that such a system can improve network performance in two cases: (i) when selective jammers are present in the network and (ii) when there is a relatively high number of nodes in the network ($N_u = 200$).

6.2.3 CSMA-type channel access

One of the main characteristics of LoRaWAN is to use an Aloha-type access protocol. This allows the complexity of the nodes to be minimal and to save energy. However, as we have seen, this is a protocol known to be very inefficient since, for high traffic loads the total capacity of the channel is not exploited as many collisions between packets are generated.

Additionally, the fact of transmitting in an uncoordinated fashion makes the network vulnerable to jamming attacks. Therefore, to improve the utilisation of the channel, several works have proposed to change the MAC protocol and to use a CSMA-type access protocol instead [126,128,133,151]. As depicted in Figure 6.2 the CSMA protocol works as follows: each ED continuously senses channel to check its state i.e. idle or busy so that it stars a transmission. Then, in case when channel is busy, the ED waits for channel to become idle. Each time the ED finds an idle channel, it transmits a frame to the channel.

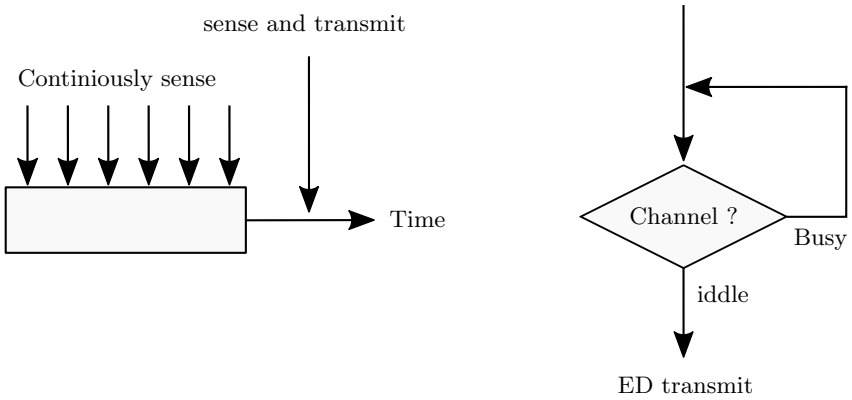


Figure 6.2: Carrier Sense Multiple Access.

Therefore, several versions of this protocol have been proposed in the literature. The difference between one another are: the sensing times, the probability with which the ED transmits once it has found an idle channel, and some other aspects as extra functions like collision avoidance.

Thus, a persistent-CSMA (p-CSMA) as the one proposed in [126] for instance, continuously senses the channel and transmits with a probability p each time the channel is idle. Then, with a probability $(1-p)$, the channel waits for next time slot. If the next time slot is idle, it again transmits with a probability p and waits with a probability $(1-p)$.

More specifically, for the LoRaWAN case, the sensing phase can be done by using the CAD functionality of LoRa nodes. This is the case of the CSMA mechanism proposed in [151], which is an open source Arduino library that extends the widely used lmic library for LoRa nodes.

Therefore, the specification should be modified in order to add support for this functionality. The results of these works suggest that the network performance can be significantly improved, However, as in the case authenticated preambles the energy consumption, and therefore the battery life of EDs is affected.

6.3 Reactive countermeasures

A reactive countermeasure takes place after a jamming attack has been detected. As in the proactive case they can be classified into software and combined software-hardware. The following are some proactive countermeasures ideas that can be implemented in LoRaWAN and that should improve its resilience against jamming attacks:

i) Jammer-Aware ADR

ii) Mapping jamming area

iii) External sensing nodes

6.3.1 Jammer-aware ADR

Something that has not been reviewed in detail in this thesis is the Adaptive Data Rate proposed by the LoRaWAN specification [75]. The idea of this mechanism is to provide reliable connectivity of ED and GW by adapting SF and the transmission power to change in the link conditions. It also seeks to minimise the power consumption of EDs.

The algorithm uses downlink messages to handle the selection of SFs and the transmission power. If the EDs detects that a large number of consecutive uplink transmissions are not acknowledged, it assumes that the connectivity is lost. Then, it starts to gradually increase its transmission power, if there is still no answer, it starts doing the same for SF.

The specification defines two parameters, namely `ADR_ACK_LIMIT` and `ADR_ACK_DELAY` that control the number of uplink messages, after which if the uplink transmission is not acknowledged, an ED increases either its transmission power or SF. The tuning of these two parameters may affect the time to converge to a steady state.

The study of this mechanism has gain great attention amongst the research community [85–94]. It has been proven that the current ADR mechanism proposed in the LoRaWAN v1.1 specification lacks the agility to adapt to the changing link conditions [85], and can even do more harm than good since it does not take into consideration the distribution of SFs amongst users [94].

Consequently, a novel jammer-aware ADR algorithm can also consider input alert signals that may come from different sources: (i) from an intrusion detection mechanism as the one proposed in Section 6.4, (ii) external decision agents, or (iii) from the node itself. Therefore, the network can tell the ED to act in order to avoid a jamming attack. The action taken may be changing its SFs, channel, frequency, transmission power, or even using an alternative modulation.

6.3.2 Mapping jamming area

As we have seen in Section 5.6, the impact area of channel-aware jammers is restricted to the jammer coverage area. Consequently, the nodes deployed closer to the jammers are much more affected than those outside their area of coverage. Then, in cases that jamming attacks are restricted to small portions of the network, specially in high density deployments, a mapping area detection algorithm can be useful to detect the actual localisation of jammers.

Therefore, an algorithm as the one proposed in 6.4 can be extended to take into consideration the physical location of EDs, in that way each time a jamming attack is detected, the algorithm can search for the location of each EDs affected and then derive a map of the jammed area. This map can be used to trigger an alert to the upper layers in order to start other countermeasures. Several works on this subject have been proposed for WSN [152–154], however for LoRaWAN there are no works to the best of our knowledge.

6.3.3 External sensing nodes

Implementing countermeasures such as jammer-aware ADR, CSMA or preamble authentication requires more energy consumption from nodes. This is something that, as we have seen in Section 1.2, cannot be supported by some use-cases. Therefore, a workaround could be to add external nodes in charge of sensing and detecting jamming attacks by using the LoRa modulation. Additionally, these external nodes may be wired so that they have no problem with sensing the medium permanently nor with battery constraints.

The system may work as follows: the external nodes can be be trained to learn the regular behaviour of the network. Then, they will record all the packets it senses; once they detect something, an alert is triggered either directly to ED or to upper layer in order to take further actions.

Several works on that matter have been proposed in the literature at upper levels. Generally speaking a system like this falls into the category of the so-called Anomaly-based intrusion detection system (AIDS), which are designed to detect any significant deviation between the observed behavior and a model of a regular behaviour of the network. Therefore, a normal

model of the behavior of the network is typically created using ML. In the LoRaWAN case for example, the training data might be previous recorded network traces containing information like the packet arrival times, SF, RSSI, packet length and so on.

The implementation of this countermeasure, as in the other cases has its disadvantages. The most important of them is the monetary cost, that because the external nodes that sense the network must have similar characteristics to a gateway, since they must have the capacity to sense several channels and SF in parallel. Consequently, this countermeasure is only relevant to use cases requiring enhanced security, such as IIoT applications. Moreover, it may be difficult to know all the localisation of the private EDs.

6.4 Proposed jamming detection mechanisms

The analytical model developed in this thesis and presented in 4.1, is useful to estimate the impact of jammers on the network performance. However, it is only valid for a steady state network. This means, that the performance evaluation presented previously is only valid for a network, which behavior has been stabilised in time.

Another way of defining jamming attacks is then to see them as malicious nodes in the network that are active for a finite period of time, with the aim to disrupt or interfere with the transmission and receipt of legitimate wireless signals among sensor nodes. Therefore, they affect the statistical features (for instance mean and variance) of a packet flow with temporal fluctuations. Consequently, our mathematical model does not allow to observe the performance of the network in the time domain, something that is vital if we want to detect attacks and take action accordingly.

Therefore, in this Section, we will explore the possibility to analyse the time domain in order to detect jamming attacks. This will be done by using network traces from our previous simulation and a small data-set made from the test-bed described in detail in Section 6.4.3.2. The algorithms considered are: (i) a statistical method, which is inspired from the SPC and a RNN ML algorithm.

6.4.1 Statistical approach

SPC is defined as the use of statistical techniques to control a process or production method. It can be used to detect anomalies by observing series of statistically homogeneous events. SCP is usually used in monitoring processes and controlling the quality of manufacturing processes [145].

Therefore, these techniques can detect changes in the process mean, process variance and the relationship between multiple variables, which can be either univariate or multivariate [155]. Cumulative Sum Control Chart (CUSUM), Shewhart and EWMA are three univariate control charts, which are often used in detecting shifts in mean values.

Then, for our detection algorithm we select EWMA, proposed by Roberts [156]. It is an efficient statistical technique used in detecting small shifts in time-series data. It functions by first defining a threshold that delimits a standard behaviour before periodically handling updates

of the observed metric's average Contrary to CUSUM and Shewhart, EWMA is characterised with its low complexity because the weighted average only needs to be updated for each newly observed data.

Another advantage of EWMA over other SPC techniques is that it combines current and historical data in a way that small shifts in time-series can be detected easily and quickly. On the other hand, other control charts such as Shewhart chart only consider the most current observations while neglecting the historical data [145, 157].

The EWMA statistic $z(t)$ is then computed as:

$$z(t) = \lambda \cdot x(t) + (1 - \lambda) \cdot z(t-1) \quad t = 1, 2, 3 \ldots n \tag{6.1}$$

where $x(t)$ is the observation at time t, n represents the number of observations to be monitored including $z(0)$ and λ is the smoothing constant ($0 < \lambda \leq 1$) that determines the depth of the EWMA.

The constant λ is used to determine the importance of both current and historical observations and to determine its sensitivity to small or gradual process drift. For instance, when λ has a value of 1, this means that only recent measurements influence the EWMA. On the contrary, a small value of λ gives more weight to older observations [158]. Several values for λ have been suggested by previous authors. In practice, λ is usually between 0.2 and 0.5. For jamming detection, as we will see, using a small value of λ results in the algorithm being insensitive to attacks characterised by moderate intensity or having short duration. The detection is done by comparing the statistic $z(t)$ with two values: Upper Control Limit (UCL) and the Lower Control Limit (LCL), which can be determined using the equation below:

$$UCL_z = z_0 + f \cdot \sigma_z \qquad LCL_z = z_0 - f \cdot \sigma_z \tag{6.2}$$

where z_0 is the target value and is set according to prior knowledge of the system, f is a tuning value usually set at 3-sigma control limits and σ_z is the standard deviation of $z(t)$. The estimated variance of the EWMA statistic can be approximated using:

$$\sigma_z^2 = \sigma_x^2 \cdot \left(\frac{\lambda}{2 - \lambda} \right) \tag{6.3}$$

where σ_x is the standard deviation of the observation series $x(t)$ obtained from the historical data.

6.4.1.1 Proposed detection mechanism

As we have seen, the LoRaWAN architecture considers a star deployment topology, where contrary to WSN, the sensing and monitoring function of the sensor nodes is not distributed into different levels. Then, all the data is centralised at central nodes. Therefore, we propose that the detection algorithm is implemented in the Network Server. As performance metric we use the IAT and the RSSI, which are taken from the previous simulation scenarios and from the test-bed presented in Section 6.4.4.2

Therefore, the detection is divided into two phases: (i) setting up phase, where the threshold values UCL and LCL are computed and (ii) the detection phase, where the EWMA is used to detect the anomalies by comparing it with the threshold values. We consider the RNN network architecture depicted in figure 6.5: the setting up is performed using the sequence(s) in s_N, which corresponds to a data-set with a normal behaviour. For the detection phase we have a test data-set t_N, with normal behaviour and another data-set t_a with anomalies.

6.4.2 Machine Learning approach

As we have seen in the previous Section, in statistical monitoring techniques such as EWMA the length of observation time window needs to be predetermined and the results greatly depend on this parameter. Hence, another option is to use the so-called Recurrent Neural Networks (RNN), which are a type of neural network where the output from previous step are fed as input to the current step. Therefore, this ability to persist information, or cell state, for use later in the network makes them particularly well suited for analysis of temporal data that evolves over time. Hence, our hypothesis is that a network performance metric, such as the IAT or the RSSI, that evolves in time can be used to train a RNN algorithm and eventually to detect a jamming attack.

In Section 6.4.2.1 we will explain in detail how RNN works, then Section 6.4.2.2 presents our detection algorithm proposal and finally 6.4.3 the result obtained.

6.4.2.1 Recurrent Neural Networks

RNN are a type of Neural Network where the output from previous step are fed as input to the current step. Figure 6.3 present a top level abstraction of a RNN. The function A looks at some input x_t and outputs a value h_t. Then, inside the loop, we have a chain-like architecture that passes previous learned data to the next step. By doing so, the network can learn from previous states. This capacity make them very useful for a variety of problems: speech recognition, language modelling, translation, image captioning and so on.

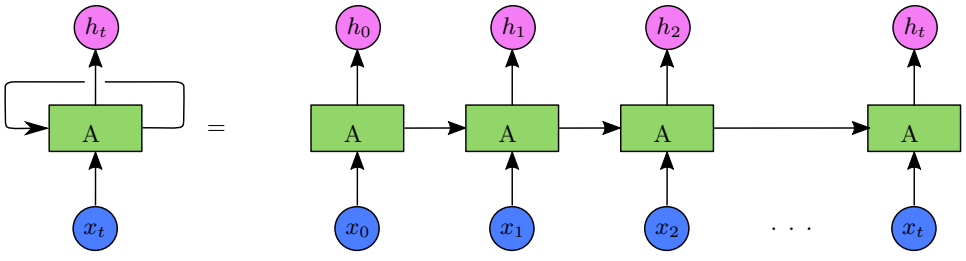


Figure 6.3: Top-level diagram of a RNN

More specifically, inside each block function (A), a set of mathematical operation is performed. The type of operation and its quantity varies from one type of RNN to another, in Figure 6.4 are shown the three most popular: (a) Vanilla, (b) LSTM and (c) GRU. Each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pair wise operations, like vector addition, while the yellow boxes are neural network

layers. Lines merging denote concatenation, while a line forking denotes its content being copied and the copies going to different locations.

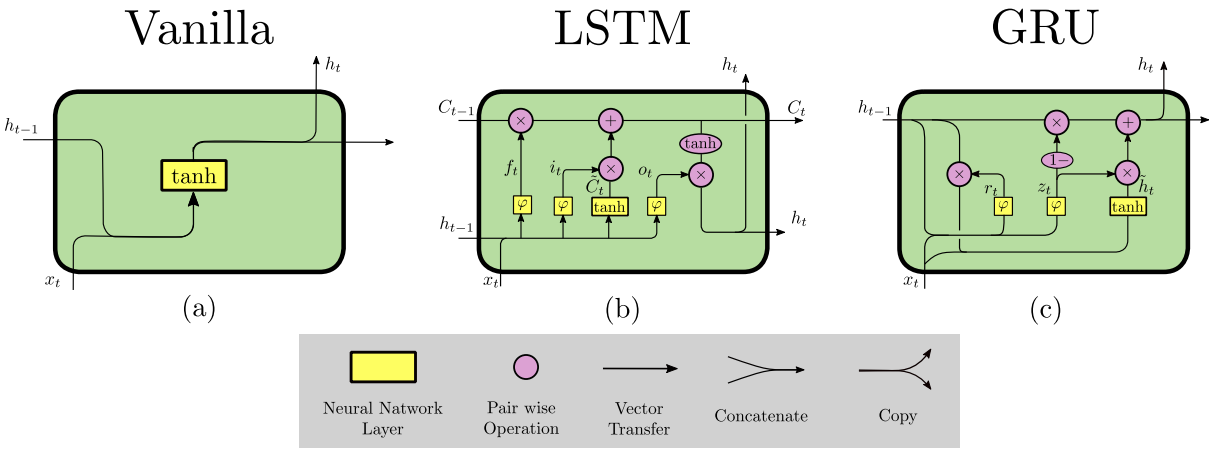


Figure 6.4: Different types of RNN: (a) Vanilla, (b) LSTM and (c) GRU. Figure adapted from [159]

There are four main concepts to consider if we look at the diagram:

1. Each time a chunk of information passes through a block, it is only modified by pair wise operations ($\otimes$, $\oplus$).
2. The Sigmoid (φ) operator is an scalar product that scales the input vector. Therefore, the output numbers describe how much of each component should be let through.
3. The tanh operator is the activation function, it takes the input value and maps it in a scale of $[-1, 1]$. However, this function can also be be linear or exponential.
4. The values W and b, are the weight and biases, and exist in every machine learning model. The weights control the signal (or the strength of the connection) between two neurons. In other words, a weight decides how much influence the input will have on the output. Biases, which are constant, are an additional input into the next layer that will always have the value of 1. Bias units are not influenced by the previous layer but they do have outgoing connections with their own weights. The bias guarantees that even when all the inputs are zeros there will still be an activation in the neuron. These two values are randomly initialised and are changed during the training step.

The simplest version (depicted in 6.4 (a)), is the Vanilla RNN. In these RNN there is only one layer that takes as input the output from previous layers (h_{t-1}) and outputs a value between -1 and 1:

$$h_t = \tanh\left(W_h[h_{t-1}, x_t]\right) + b_h) \tag{6.4}$$

The most popular RNN are the Long short-term memory (LSTM), they are designed to be capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhu-

ber (1997) [160]. The basic module is presented in figure 6.4 (a). There are four layers inside each LSTM block:

i) Forget layer: this layer take the output from previous layers (h_{t-1}) and outputs a value between 0 and 1, if the output is zero it means that the network forgets what was learnt in the previous step, while one replicates the value.

$$f_t = \varphi(W_f[h_{t-1}, x_t]) + b_f) \tag{6.5}$$

ii) New information layer: this layer decides what new information is going to be stored in the cell state. This has two parts. First, a sigmoid layer decides which values will be update. Next, a tanh layer creates a vector of new candidate values, $\tilde{C}_t$, that could be added to the state.

$$\begin{aligned} i_t &= \varphi(W_i[h_{t-1}, x_t]) + b_i) \\ \tilde{C}_t &= \tanh\left(W_C[h_{t-1}, x_t]\right) + b_c) \end{aligned} \tag{6.6}$$

iii) Update layer: this layer combines the two previous computed values (i_t and $\tilde{C}_t$) to create an update to the state. It also multiplies the old state by f_t, forgetting the things that where decided to forget earlier.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{6.7}$$

iv) Output layer: this layer decides the output. This output will be based on the cell state, but will be a filtered version. First, there is sigmoid layer which decides what parts of the cell state are going to be in the output. Then, it passes through a tanh gate and multiply it by the output of the sigmoid gate, so that we only output the parts we decided to.

$$\begin{aligned} o_t &= \varphi(W_o[h_{t-1}, x_t]) + b_o) \\ h_t &= o_t * \tanh\left(C_t\right) \end{aligned} \tag{6.8}$$

Finally, a slightly different version of the LSTM is the Gated Recurrent Unit (GRU) depicted in figure 6.4 (c). They were introduced by Cho, et al. (2014). It combines the forget and input gates into a single *update gate*. It also merges the cell state and hidden state, The resulting model is simpler than standard LSTM models, which makes them suitable to be executed in constrained devices.

$$\begin{aligned} z_t &= \varphi(W_z[h_{t-1}, x_t]) + b_z) \\ r_t &= \varphi(W_r[h_{t-1}, x_t]) + b_r) \\ \tilde{h}_t &= \tanh(W_h \cdot [r_t * h_{t-1}, x_t]) + b_h) \\ h_t &= (1 - z_t) * [h_{t-1} + z_t * \tilde{h}_t] \end{aligned} \tag{6.9}$$

6.4.2.2 Proposed detection mechanism

For the jamming detection we propose to adapt methodology previously proposed in [15]. As depicted in figure 6.5, there are two phases for the detection mechanism: (i) a training phase where the RNN learns the normal behaviour of the network. In our case, this correspond to previously recorded traces of a network that has not been attacked, and (ii) the detection phase, where the RNN compares the predicted normal behaviour with a trace that contains anomalies. Then, the output is the error between these two series. For the detection, the error is compared with a threshold value τ.

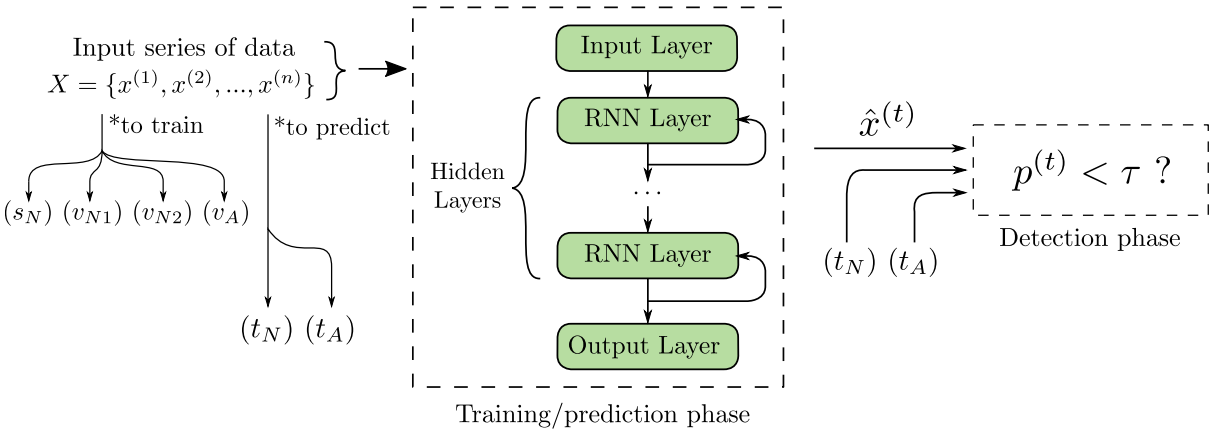


Figure 6.5: RNN jamming detection model

Thus, the input is a time series $X = \{x^{(1)}, x^{(2)}, ..., x^{(n)}\}$, where each point $x(t) \in R^m$ in the time series is an $m-$dimensional vector $\{x_1^{(t)}, x_2^{(t)}, ..., x_m^{(t)}\}$. The model learns to predict the next l values for d of the input variables s.t. $1 \leq d \leq m$. The normal sequence(s) are divided into four sets: normal train (s_N), normal validation-1 (v_{N1}), normal validation-2 (v_{N2}), and normal test (t_N). The anomalous sequence(s) are divided into two sets: anomalous validation (v_A), and anomalous test (t_A).

Therefore, as prediction model we use the three RNN networks previously described (Vanilla, LSTM and GRU), and then we compute the prediction error distribution to detect anomalies.

Training phase: We consider the RNN network architecture depicted in figure 6.5: We stack several RNN units (either, Vanilla, LSTM or GRU) such that each unit in a lower hidden layer is fully connected to each unit in the hidden layer above it through feed forward connections. The prediction model is learned using the sequence(s) in s_N. The set v_{N1} is used for early stopping while learning the network weights.

We take one unit in the input layer for each of the m dimensions, $d \times l$ units in the output layer such that there is one unit for each of the l future predictions for each of the d dimension. The RNN units in a hidden layer are fully connected through recurrent connections.

Detection phase: The trained RNN is used to perform a prediction of length of l, each of the selected d dimensions of $x^{(t)} \in X$ for $l < t \leq n - l$ is predicted l times. Then, we compute an

error vector $\epsilon(t)$ for point $x^{(t)}$ as $\epsilon(t) = [e_{11}^{(t)}, \ldots, \epsilon_{1l}^{(t)}, \ldots, \epsilon_{d1}^{(t)}, \ldots, \epsilon_{dl}^{(t)}]$, where $\epsilon_{ij}^{(t)}$ is the difference between $x_i^{(t)}$ and its value as predicted at time $t - j$.

The prediction model trained on s_N is used to compute the error vectors for each point in the validation and test sequences. The error vectors are modelled to fit a multivariate Gaussian distribution $\mathcal{N} = N(\mu, \Sigma)$. The likelihood $p(t)$ of observing an error vector $\epsilon(t)$ is given by the value of N at $\epsilon(t)$. The error vectors for the points from v_{N1} are used to estimate the parameters μ and Σ using Maximum Likelihood Estimation. An observation $x^{(t)}$ is classified as 'anomalous' if $p^{(t)} < \tau$, else the observation is classified as 'normal'. The sets v_{N2} and v_A are used to learn τ by maximising one of the evaluation metrics (precision, recall or F_1-score).

6.4.3 Algorithm evaluation and results

6.4.3.1 Evaluation methodology

In order to evaluate both, the statistical and machine learning approaches we use the precision and recall methodology. The results are compared by using the terms true positives (tp), true negatives (tn), false positives (fp), and false negatives (fn). The terms positive and negative refer to the classifier's prediction (sometimes known as the expectation), and the terms true and false refer to whether that prediction corresponds to the external judgement (sometimes known as the observation). Therefore, we compute three metrics: Precision, Recall, and $F_1 - \text{score}$, defined as follows:

Precision: is the fraction of relevant instances among the retrieved instances

$$\text{Precision} = \frac{tp}{tp + fp} \tag{6.10}$$

Recall: also known as sensitivity, is the fraction of the total amount of relevant instances that were actually retrieved.

$$\text{Recall} = \frac{tp}{tp + fn} \tag{6.11}$$

$F_1 - \textbf{score}$: is the harmonic mean of the precision and recall. The highest possible value of an F-score is 1, indicating perfect precision and recall, and the lowest possible value is 0, if either the precision or the recall is zero.

$$F_1 - \text{score} = \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})} \tag{6.12}$$

6.4.3.2 Data-sets description

Three data sets were used to evaluate our detection algorithms, they were taken from the different simulation campaigns previously described and from the test-bed described in Section 6.4.4.2. We select as evaluation metrics the IAT and the RSSI:

i) Congested Network from the realistic channel simulation scenario of Section 5.3: it is a LoRaWAN cell with $N_u = 1000$, and 5 channel oblivious jammers on the uplink channel. We consider two jammers profiles with $d_{ju} = 0.5$ and $d_{ju} = 1$ respectively. The simulation time is set to 1500 s and the attack lasts for 200 s.

ii) Network without congestion from the realistic channel simulation scenario of Section 5.3: it is a LoRaWAN cell with $N_u = 100$, and 5 channel-oblivious jammers on the uplink. We consider two jammers profiles with $d_{ju} = 0.5$ and $d_{ju} = 1$ respectively. The simulation time is set to 1500 s and the attack lasts for 200 s.

iii) Real network traces: It is a LoRaWAN cell with 5 EDs and one jammer. EDs are configured to use a unique SF, which is set to 12 and the application profile considered a $T_{app} = 1$ m. The jammer is also configured to use a SF of 12, and is send a packet each 30 s. The measurement campaign considered 3h, time during which the jammer is on for 1h.

Therefore, for each of the data-sets described above, we derive several realisations considering the cases with and without jamming in order to form the six time series needed to evaluate the algorithms:

- Normal behaviour (no jamming): normal train (s_N), normal validation-1 (v_{N1}), normal validation-2 (v_{N2}), and normal test (t_N).
- Anomaly behaviour (with jamming): anomalous validation (v_A), and anomalous test (t_A).

In addition to the IAT and the RSSI, the traces also carry additional information (instances), such as the SF, type of event (transmission, reception, jammer transmission) and the time-stamp. Each of one the traces also have a field indicating if there is an ongoing attack in order to compute the precision and recall metrics.

The data-sets extracted from ns3 simulations correspond to 6 different type of events : (i) GW receives, (i) ED receives, (iii) ED transmits, (iv) GW transmits, (v) Jammer transmits, and (vi) Collision. Each type of event contains 8 different features : (i) Type of event, (ii) Node ID, (iii) Sender ID, (iv) packet size, (v) Frequency band, (vi) Spreading Factor, (vii) RSSI, and (viii) Time Stamp. As for the data-sets from the test-bed, we have the same number of instances but there is only three types of events: (i) ED transmits and (ii) GW receives, and (iii) Jammer transmits. The size of the data-sets used is summarised in Table 6.2. An example of the congested network data-sets from simulations are depicted in figures 6.6 and 6.7.

6.4.3.3 Results

In the following a description of the results obtained is presented. First, we will present the results obtained for each of the two approaches. To do this, we will use the congested network data-sets, and apply each approach separately. This will make it possible to see clearly how each of the two algorithms works. Finally, we will compute the performance metrics presented above and make a comparison of the different algorithms.

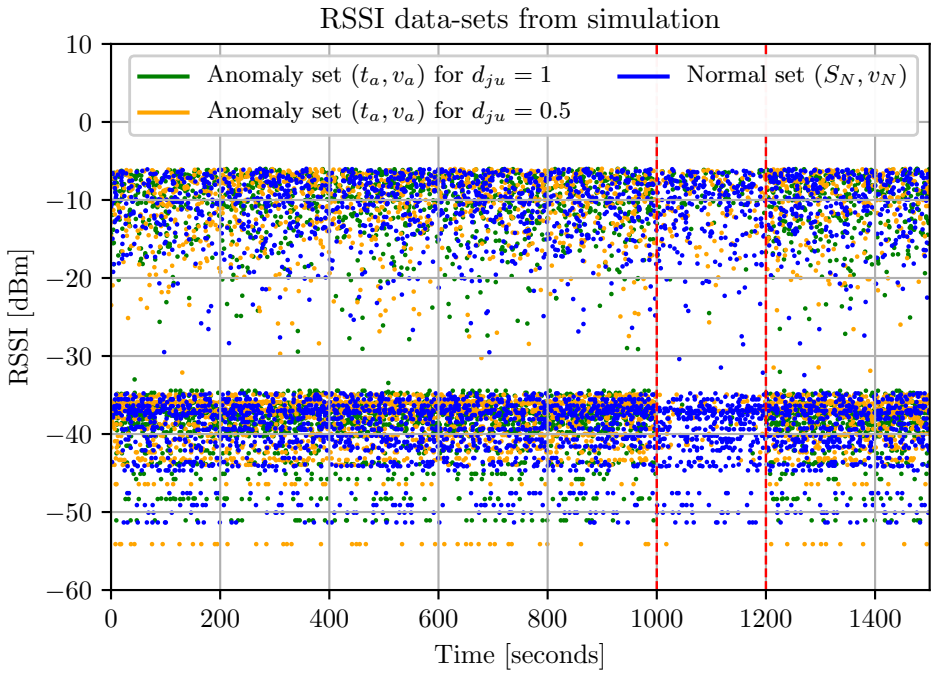


Figure 6.6: RSSI data-sets from simulation, the vertical red lines indicate the period during which jammers are active.

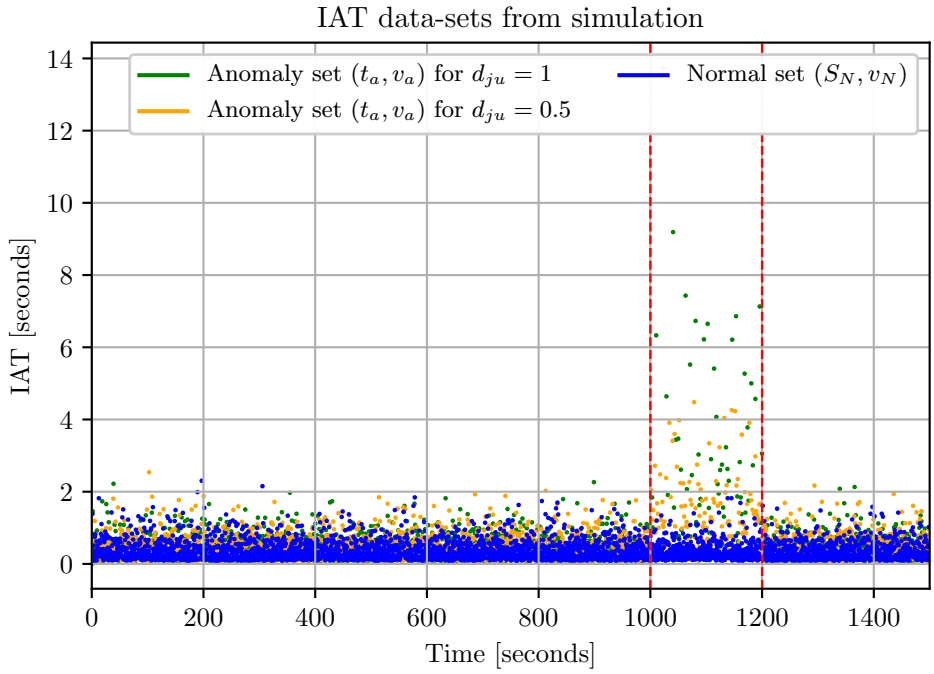


Figure 6.7: IAT data-sets from simulation, the vertical red lines indicate the period during which jammers are active

Table 6.2: Data-sets summary

Name	Number of features	Number of instances
Simulation [1000 ED / 100 ED] - Normal train	8	45256 / 5680
Simulation [1000 ED / 100 ED] - Normal test	8	56000 / 6550
Simulation [1000 ED / 100 ED] - Anomalous validation	8	34021 / 3108
Simulation [1000 ED / 100 ED] - Anomalous test	8	28401 / 3256
Testbed [5 ED] - Normal train	8	730
Testbed [5 ED] - Normal test	8	1010
Testbed [5 ED] - Anomalous validation	8	780
Testbed [5 ED] - Anomalous test	8	540

EWMA: Figures 6.8 and 6.9 depict an example of results obtained when applying the statistic EWMA for detecting jamming attacks to the congested network data-sets. As we have discussed earlier there are four tuning variables: (i) the number of samples to perform the statistic (n), the target value (μ), λ that represents the importance we give to current and historical observations, and f that establishes the detection point.

For the IAT case, depicted in 6.8, the reported results use $n = 10$, μ is set as the mean of the training data-set $\mu = E(s_N(t))$, $\lambda = 0.3$ and $f = 3$. As we can see, the most useful parameter for detecting anomalies is the UCL. For this case, the algorithm manages to detect reasonably well the jammers. It takes 15s to detect an attack with $d_{ju} = 1.0$, and 30 with $d_{ju} = 0.5$. This time could be improved if the size of the observation (n) is reduced with the risk of increasing the true negatives (tn) rate.

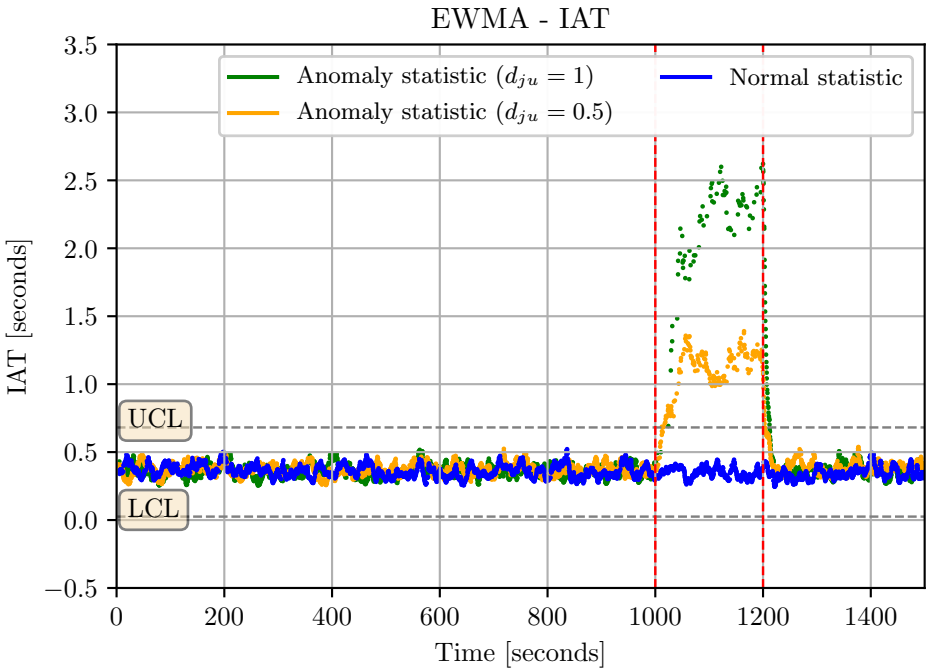


Figure 6.8: EWMA applied to the IAT, the vertical red lines indicate the period during which jammers are active

Similarly, for the RSSI case depicted in figure 6.9, the reported results use $n = 20$, μ is set as

the mean of the training data-set $\mu = E(s_N(t))$, $\lambda = 0.3$ and $f = 2$. As we can see, contrary to the IAT, this metric has much higher variance. Therefore, a higher value of n is needed. As in the IAT case, the most useful parameter for detecting anomalies is also the UCL. For this case, the algorithm manages to detect the jamming with $d_{ju} = 0.5$. However, it takes 70 s to detect the attack, this is due to the fact that an n of 20 was used instead of 10 as in the IAT case. Additionally, as we can see, the algorithm also triggers some false positives at $t = 242$ s and $t = 880$ s; and some false negatives at 1091 s and 1171 s. This could be changed if a different value of LCL is chosen with the risk of increasing the time to detection and the number of false positives.

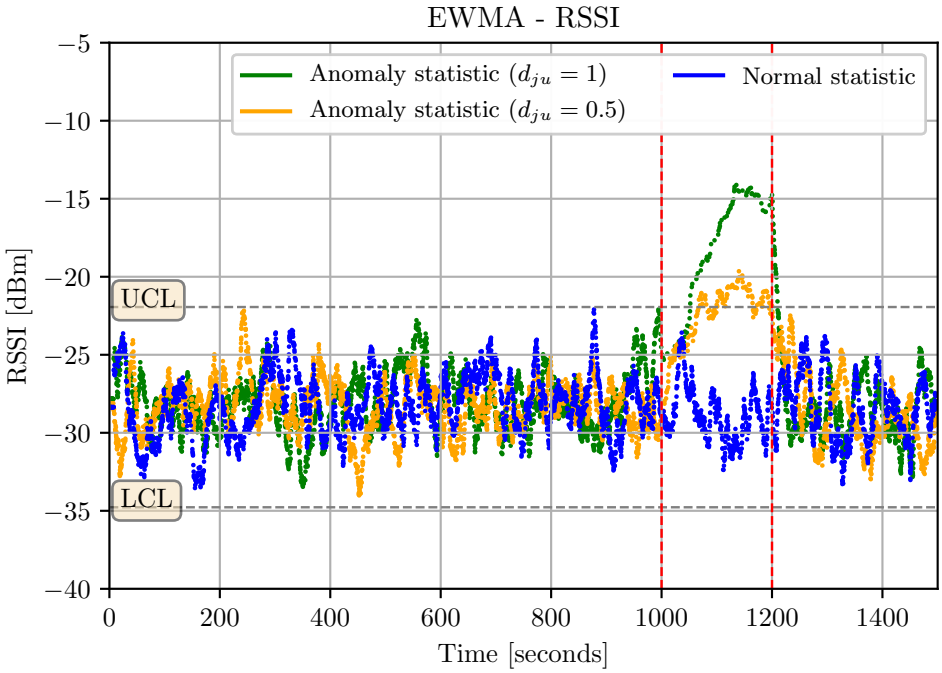


Figure 6.9: EWMA applied to the RSSI, the vertical red lines indicate the period during which jammers are active.

The algorithm performs better when the anomaly data-set for $d_{ju} = 1.0$ is used. In this case, the time to detection is reduced to 50 s, and there are no false negatives. This time could be improved if the size of the observation (n) is reduced with the risk of increasing the true negatives (tn) rate.

Finally, it should be noted that the results description presented above only reflects one iteration and its objective is to illustrate how the algorithm works. A comparison of the different algorithms considering the performance evaluation metrics will be presented later.

RNN case: Figure 6.10 depicts an example of results obtained when applying the RNN (LSTM) for detecting jamming attacks to the congested network data-sets. There are two tuning variables: (i) the number of samples used for each prediction step (l), and the (ii)

threshold to detect the anomaly (τ)[1].

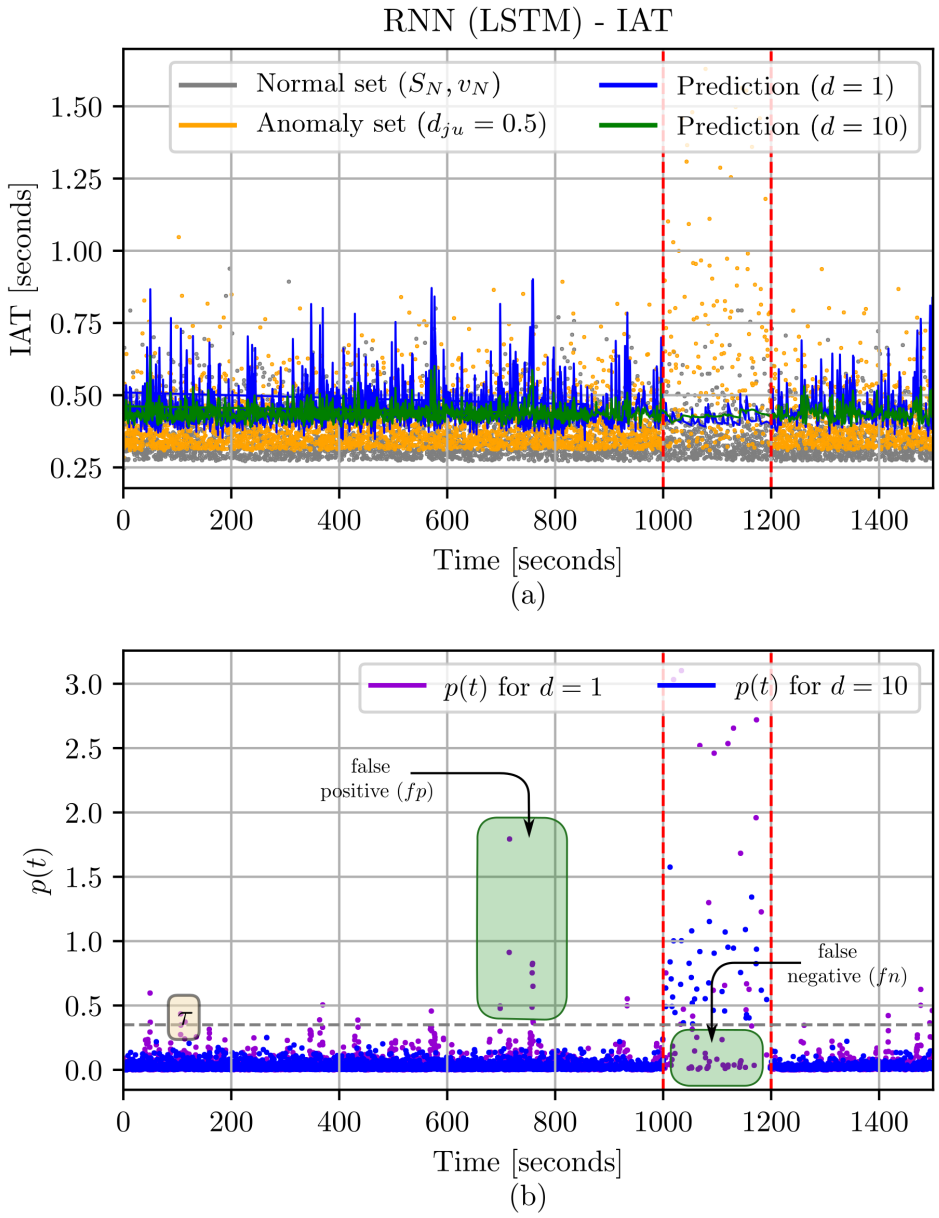


Figure 6.10: LSTM applied to the IAT, the vertical red lines indicate the period during which jammers are active.

Figure 6.10 depicts the results obtained for the IAT metric. On the top, we have the results of the prediction phase. The solid blue line presents the results for $d = 1$, while in green we have the prediction for $d = 10$. As, we can see, if the RNN is asked to perform 1-sample step predictions, we will get a series of time that varies very quickly in time ($\sigma^2 = 0.044$). On

[1]In addition to these tuning parameters, several parameters are needed to be set in the training phase: (i) the number of hidden layers, (ii) the number of epochs and (iii) the batch size. These parameters are fundamental since they determine if the curve will be well fitted. Therefore, for the training phase we perform follow an early stopping approach as suggested in [161].

the contrary, when the network is asked to make predictions considering 10 samples, the value obtained has a lower variance ($\sigma^2 = 0.003$).

Based on these predictions, the error vector and the likelihood $p(t)$ is computed. The latter is presented in figure 6.10 (b). In purple we have the results for $d = 1$, while in blue we have the likelihood for $d = 10$. We also present a gray dotted line that represents the decision threshold τ .

The key observations from our results are as follows: (i) figure 6.10 (b), the likelihood values $p(t)$ are significantly lower in the anomalous regions than the normal regions for the data-sets considered. (ii) the decision mechanism performs better when d is set to 10. This is because for the case when $d = 1$ the trained network is very sensitive to instantaneous changes in time, which makes the error of the predictions very high. This results in obtaining a higher rate of false positives and false negatives than in the case in which d = 10 is chosen. Indeed, if we look closely to the figure 6.10 (b), we can see that there are no false negatives when a $d = 10$ is selected. Opposite is the case when $d = 1$, where most points fall below the τ decision point. (iii) In terms of detection time, it can be observed that, contrary to EWMA, the RNN algorithm manages to detect attacks much faster. In both cases ($d = 1$ and $d = 10$), the RNN network requires only one sample for trigger the alert. This is because contrary to the EWMA case, the prediction does not depend on a sliding vector, since it uses the training done in the previous phase.

Finally, and as already mentioned in the description of the results for EWMA, it should be noted that the results description presented above only reflects one iteration and its objective is to illustrate how the algorithm works. A comparison of the different algorithms considering the performance evaluation metrics will be presented later.

Mechanisms Comparison: Table 6.3 reports the results of the EWMA and RNN detection mechanisms on the three data-sets described earlier. We report precision, recall, F1-score. The results are reported after choosing the threshold value (UCL and τ) with the maximum F_1-score using the validation sets as described in Section 6.4.3.2.

Table 6.3: Precision, recall and F_1 scores for different jamming detection algorithms

	Algorithm	Simulation ($d_{ju} = 1$) - IAT	Simulation ($d_{ju} = 1$) - RSSI	Simulation ($d_{ju} = 0.5$) - IAT	Simulation ($d_{ju} = 0.5$) - RSSI	Test-bed IAT	Test-bed RSSI
Precision	EWMA	0.90	0.85	0.82	0.69	0.95	0.93
	LSTM	0.97	0.88	0.94	0.88	0.98	0.96
	GRU	0.88	0.75	0.93	0.85	0.96	0.97
	Vanilla	0.87	0.86	0.82	0.69	0.91	0.93
Recall	EWMA	0.89	0.81	0.80	0.58	0.96	0.94
	LSTM	0.98	0.91	0.93	0.89	0.96	0,98
	GRU	0.87	0.64	0.88	0.77	0.98	0.97
	Vanilla	0.85	0.94	0.78	0.74	0.95	0.97
F_1-score	EWMA	0.90	0.83	0.81	0.63	0.97	0.94
	LSTM	0.98	0.90	0.94	0.88	0,97	0.97
	GRU	0.88	0.69	0.90	0.81	0.96	0.97
	Vanilla	0.86	0.90	0.80	0.71	0.93	0.95

The network architecture chosen was 50 hidden layers for the IAT and RSSI metrics respectively. The key observations from our results are as follows: For all cases, LSTM outperforms the other detection algorithms. In fact, when applied to the congested network data-set ($d_{ju} = 1$), a precision of 0.98 and a recall of 0.98 are obtained when the IAT metric is considered. The other two RNN models considered also get good results but, GRU shows a better performance than Vanilla in all cases.

On the other hand, we could also observe that EWMA offers very good results if we consider that the cost of implementation is much lower than a ML algorithm. In fact, it is only surpassed by LSTM. For instance, better results were obtained using EWMA than GRU for when applying less congested network data-set (d_{ju}) considering both the IAT and RSSI metrics.

Regarding the type of performance metric used, we can see that, for all the cases considered, it is better to use the IAT. This allows us to infer that the temporal dependency is much lower in RSSI, something that must be guaranteed both for the EWMA and for the RNNs considered. Additionally, we shall also consider the algorithms were applied to the whole data-sets. Therefore, we did not make any filter considering each of the EDs. In a real implementation this could be a very good option that could give better results even using the RSSI.

Another important aspect to consider is that, as we saw at the beginning of this chapter, there is a great variety of metrics that could be used to detect jamming attacks. In our work we only consider two because these are the ones that we can get easily from our simulation environment. However, in a real deployment, metrics such as the Bad Packet Ratio or the Energy consumption could also be useful. Our experimental evaluation is also limited by the scale of the test-bed, further analyses should be performed in order to have a deeper understanding of jamming in a real world application.

Hence, our approach should not be considered as a one-fits-all solution that detects all possible jamming attacks on LoRaWAN. It is rather a methodology aiming at improving the network resilience against jamming. We hence believe that training data that accounts for a representative amount of attacks and performance metrics in a wider deployment, has the potential to accurately detect a wider range of jammers.

In addition to that, another aspect that shout be taken into consideration is the specific use-case. Particularly regarding the periodicity of sent packets. Indeed, as we have seen, RNN algorithms require temporal dependencies in order to work well. Therefore, for a use case, where packets are not set periodically (or with a random distribution that does not have high variability), metrics such are the IAT might not be the most adequate.

6.4.4 Implementation details

The following is a description of the hardware platform used to implement the detection algorithms presented an evaluated above. We also present the test-bed used to get the real world data-sets used to train the algorithms.

6.4.4.1 Detection algorithms implementation

The RNN algorithms are implemented in Pytorch, using the public available git repository presented in [162]. The EWMA detection algorithm was also implemented in Python by using the NumPy library. The algorithms were run on a computer with Ubuntu 14.04, with a Intel(R) Core(TM) i5-2500K CPU @ 3.30 GHz processor, a NVIDIA GeForce GTX 970 GPU with 1664 cuda cores and 16 GB and four DIMM 4096 MB RAM modules @ MT/s. The code and data-sets are public available at [17].

6.4.4.2 Test-bed description

As pointed out before, the detection algorithms were also trained to detect jamming attacks from real jamming attack implementation. For that, we developed The LOrawan FRamework for SEcurity Audit (LOFRASEA), which is a test-bed to study jamming attacks on a LoRaWAN network. It allows to implement different types of attacks in a controlled environment [17]:

1. Eavesdropping
2. Channel-aware jamming: triggered jamming
3. Channel-oblivious jamming

The architecture of LOFRASEA is depicted in figure 6.11. It is composed by several elements: one or several legitimate nodes (`lofrasea-nl001`), one or several malicious nodes (`lofrasea-ml001`) a LoRaWAN gateway, a LoRaWAN network server, a LoRaWAN application server and a controller.

As depicted in 6.12 a legitimate/malicious is composed by: a Single Board Computer (SBC) that allows to manage a LoRaWAN node, a LoRaWAN node and an antenna. For the legitimate node we use the STM32 LoRaWAN node [163] and for the jamming node the Dragino Shield module [164]. However, our platform can also support any LoRaWAN module.

For the jamming module however, we specifically choose the Dragino module because it offers the possibility of reading of a message while it is being received. This is necessary to perform the channel-aware attack because, as we have seen in chapter 3, the jammers need to sniff the channel and then start the transmission based on the information they stored. This can be done in the Arduino Dragino module because it has a RFM95 radio modules, which has an SPI interface. Therefore, for the implementation of the jammer, we have the possibility to read the module's FIFO and stop the reception in order to perform the attack.

The LoRaWAN network is supported by only one provider, which is the ChirpStack project. It allows to deploy a LoRaWAN gateway, to manage a LoRaWAN network with a network server and to manage the application with the application server. These three elements are supported by a SBC module (Raspberry pi3 model b+) and the LoRaWAN WiMOD iC880A which contains the Semtech SX1301 LoRaWAN Gateway [165]. Chirpstack is an open-source LoRaWAN Network Server stack that provides different services that can be implemented on the same hardware architecture or separately [166]:

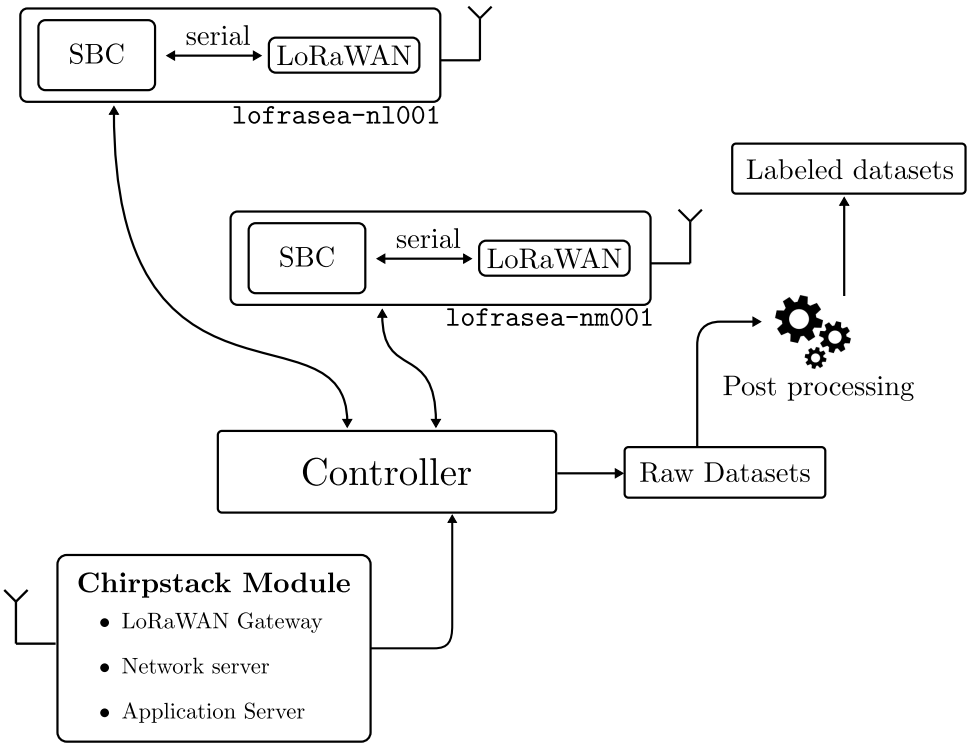


Figure 6.11: Top level architecture of the LOFRASEA test-bed.

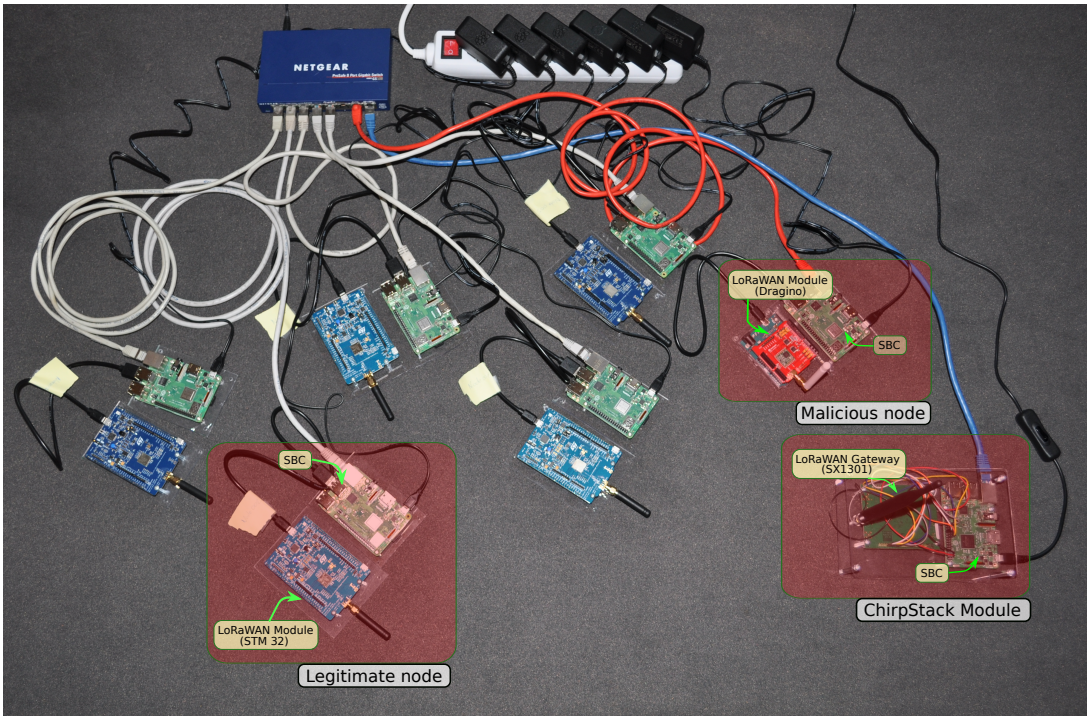


Figure 6.12: LOFRASEA test-bed hardware implementation.

- Packet Forwarder: it forwards the packet as they arrive from EDs.
- Gateway bridge: it is a link connecting the packet forwarder and the network server.
- Network server: it is in charge of the authentication process of all the nodes belonging to the network. It is also in charge of the signal processing needed to support the MAC protocol that transforms LoRa signal into packets, so that it can be sent to the Application server.
- Application server: it is responsible to handle all the aspects related to the application such as: inventory of all nodes, data base management, joint-request, and cryptographic operations.

The controller is a workstation that allows connecting the nodes to the LoRaWAN infrastructure. Then, after an experimentation raw data-sets can be post processed in order to build a labeled data-set, which is latter used to different purposes such as the detection algorithm presented in this chapter. The above described test-bed was used to get the different labelled data-set that we used to train the detection algorithms and to detect jamming attacks.

6.5 Summary and conclusions

In this chapter an analysis of several countermeasures that could be applied to LoRaWAN is presented. We have seen that they can be classified into three categories: (i) detection, that aims at providing early alerts in order to take further action, (ii) proactive counter-measures that are implemented before the attack actually happens and (iii) reactive-countermeasures that enhance the network resilience by taking specific actions during or after the attack.

We provide a literature review of the main detection mechanisms proposed for LoRAWAN and other IoT networks under jamming. We have also seen that there is a wide variety of performance metrics that can be used for such objective. Latter, we review several possible proactive countermeasures such as authenticated preambles, frequency hopping and CSMA. Then, we focus on reactive countermeasures and we briefly present three possible mechanisms than can be implemented: jammer-aware ADR, mapping jamming area and external sensing nodes.

We have proposed and evaluated two jamming detection approaches applied to different data-sets made from the previous simulation scenarios presented in chapter 5 and the test-bed described in section 6.4.4.2. First, we proposed to use EWMA, which is a statistical approach normally used in industrial control processes. It employs the moving average and a control limit to detect anomalies. Then, we used a family of neural networks called RNN that employ a recursive architecture that has been widely used for time series. Additionally, we have shown that EWMA gave similar results when compared with RNN. Hence, we have demonstrated that the proposed ML detection mechanism is able to learn the normal network behaviour, which can then be used to detect anomalies. Therefore, our RNN approach yields promising results for real-world data-sets. And finally, we have presented a description of the hardware platform used

for implementing our detection algorithms, as well as a description of our LOFRASEA test-bed, which is a free software environment that allows to implement and evaluate a LoRaWAN network under jamming.

Chapter 7

Conclusion and Perspectives

7.1 Conclusions

LoRaWAN Networks have been gaining popularity in the IoT environment because they offer wide coverage, low cost of implementation and an open specification. Such great success however comes with a lot of challenges to overcome. Within them jamming is presented as a major obstacle that should be considered for future versions of the specification. In this thesis this problem was addressed by following a holistic approach. Our contributions are briefly described as follows:

i) We first propose an analytical model that allows the evaluation of the performance of a LoRaWAN network under jamming attacks. For this model, we take several simplifying hypotheses, which although they make the obtained result to be below the performance of a real network, allows a good approximation for smaller traffic loads.

ii) We enhance a previously developed event-driven simulator build under ns3 for LoRaWAN. To this module we add several functions that allow us simulate and estimate the network performance of a LoRaWAN network under the attack of jammers.

iii) We use the two network modelling tools to perform a comprehensive performance evaluation of LoRaWAN under jamming. We consider six different scenarios: (i) a LoRaWAN network with ideal channel, (ii) a LoRaWAN network with realistic channel and hardware limitations, (iii) a LoRaWAN Network with gateway diversity,(iv) jammer strategy, (v) geographical impact of jammers and (vi) energy consumption and battery life.

iv) As can be seen in Appendix A, when developing the Simulation model in ns3, we found that the two existing interference models do not provide a complete estimate of collisions. The reason for this is that the most commonly used model in the literature (Goursaud's Matrix) [1] is designed for an ideal case in which there is a pseudo-orthogonality of spread factors. It has been used in an inaccurate manner. A more recent model [16] based on physical measurements indicates that this pseudo-orthogonal does not exist in actual deployments. This, in addition to the fact that the second model is not complete because it does not take into account all possible combinations of the different spreading factors

(where there are more than two spreading factors), opens up different research possibilities that will improve these interference models in the future.

v) We have proposed and evaluated two jamming detection approaches applied to different data-sets made from the previous simulation scenarios a test-bed. We first, proposed to use EWMA, which is a statistical approach normally used in industrial control processes. It employs the moving average and a control limit to detect anomalies. Then, we used a family of neural networks called RNN that employs a recursive architecture that has been widely used for time series. Additionally, we have shown that EWMA gave similar results when compared with RNN. Hence, we have demonstrated that the proposed ML detection mechanism is able to learn the normal network behaviour, which can then be used to detect anomalies. Therefore,our RNN approach yields promising results for real-world data-sets.

vi) Finally, we presented a description of the LOFRASEA test-bed, which is a free hardware/software environment that allows to implement and evaluate a LoRaWAN network under jamming.

7.2 Perspectives and Future Works

This work may normally be extended by considering the following research problems:

i) Add support for the capture effect and two reception windows for the mathematical model presented in the thesis.

ii) Studying the capture effect and its possible impact in the network performance when there is jamming. This can be done by considering a distribution of jammers different from a uniform distribution as what we consider in this thesis.

iii) Study the effect that the use of the ADR algorithm can have when considering jamming attacks. In fact as, other authors suggest, it does not offer the best performance because it does not consider the correct balance in the distribution of ED amongst SFs. Therefore, the network performance could be even worse in a network with jammers, which can cause energy consumption and life time to be affected.

iv) Consider a wider real deployment for the evaluation of the jamming detection algorithms proposed here, as well as extending the performance metrics used to detect jamming.

In addition to these open research problems, we have also identified several security features that can be added to the LoRaWAN specification:

i) Adding an extended version of the ADR that consider jamming.

ii) Adding additional nodes in charge of sensing the channel quality in order to improve the network resilience.

iii) Using a light version of CSMA in order to increase the network resilience and reliability.

iv) Increase the network and application level security of Class B EDs, specially regarding the broadcasting of Beacon messages.

Finally, we strongly believe that, if the use-case warrants it, developers should consider to include extra network security rules such as:

i) Adding a network-level jamming detection mechanism as the ones proposed in this thesis

ii) Adding additional nodes specifically deployed to sense the channel quality, this can be used to trigger early alerts and to take further action.

Appendix A

This appendix is aimed at providing a detail description on how the different LoRa interference models present in the literature are built and how they should be used. Then, we will point out the limitations of each of them and how these may affect network performance estimation. As we have seen in Section 4.2.1, we can either consider perfect spreading factor orthogonality, in which case we are in an Aloha pure model, or we can consider a model based on pair-wise spreading factor SINR thresholds comparison. For the second case, we can find in the literature two different approaches: (i) Goursaud's co-channel rejection matrix [1] and the Croce's SIR matrix [16].

7.3 Goursaud's co-channel rejection matrix

This model allows to identify if a LoRa signal can be decoded in the presence of another signal that has been modulated using LoRa. For that, they considered that one transmission over the two can be successful if, after the de-spreading process, the desired signal is at least 6dB over the other. Therefore, a pairwise comparison of all the interferer/desired spreading factor combinations was done. As a result, the co-channel rejection presented matrix presented in Equation 7.1 was computed.

$$\mathbf{T} = \begin{bmatrix} -6 & 16 & 18 & 19 & 19 & 20 \\ 26 & -6 & 20 & 22 & 22 & 22 \\ 27 & 27 & -6 & 23 & 25 & 25 \\ 30 & 30 & 30 & -6 & 26 & 28 \\ 33 & 33 & 33 & 33 & -6 & 29 \\ 36 & 36 & 36 & 36 & 36 & -6 \end{bmatrix} \tag{7.1}$$

This matrix indicates the threshold (before de-spreading in dB) between the desired signal (with SF i, $i \in \mathbb{Z}^+$, $7 \leq i \leq 12$, rows $\in \mathbf{T}$) and the interferer (with SF j, columns $\in \mathbf{T}$) necessary to ensure the correct reception of the desired signal. The SINR can be defined as :

$$SINR = \frac{P_{s,i}}{P_I + \sigma^2} \tag{7.2}$$

where $P_{s,i}$ is the power of the desired signal on a spreading factor i, P_I is the summation of the power of all interferers $\sum_{l=1}^{N_{int}} P_l$, N_{int} is the number of interfering signals and P_l is the power of the l^{th} interfering signal.

Therefore, to decide whether or not the desired signal is successfully received we shall:

1. Apply the co-channel rejection matrix to the interference term (after decoding):

$$\widetilde{P_I} = \sum_{l \in \mathcal{I}_j} P_{x,l} * 10^{T_{i,j}} \tag{7.3}$$

where $T_{i,j}$ is the corresponding threshold value from Eq.7.1, $\mathcal{I}_j$ is the set of interfering signals using and spreading factor j and i is the spreading factor of the desired signal.

2. Apply $\widetilde{P_I}$ to the SINR (after decoding):

$$\widetilde{SINR} = \frac{P_{s,i}}{\widetilde{P_I} + \sigma^2} \tag{7.4}$$

3. Decide whether or not the desired signal is correctly decoded. Thus, a signal with SF i is correctly decoded if the following inequality holds:

$$\widetilde{SINR} > \text{ 6 dB} \tag{7.5}$$

7.4 Croce's SIR matrix

In [122], and latter in [16] the authors quantified the impact of collisions and measure the channel rejection in order to evaluate the impact of collisions in LoRa. For that, they considered a real point-to-point LoRa link in presence of continuous packets collisions. The goal is then to identify a SIR (Signal-to-Interference Ratio) threshold under which the demodulation of the received frame is affected by errors.

The experiment carried out consisted in generating a combined traffic of two signals (reference signal and interfering signal) with the help of a LoRa synthesizer. This combined signal, is transmitted using a USRP, and received by a SX1272 node.

The traffic generated considered all possible combinations of spreading factors. For the reference spreading there are 500 packets with 20 bytes payload, and an adjustable payload length (with an equivalent ToA) for the interfering spreading factor. The offset of each interfering packet, was randomly selected within a window that guarantees that the two packets collide for at least one symbol. Finally, the SIR is measured at the receiver input (i.e. before de-spreading). The measurement results are presented in Eq.7.6

$$\mathbf{T_c} = \begin{bmatrix} 1 & -8 & -9 & -9 & -9 & -9 \\ -11 & 1 & -11 & -12 & -13 & -13 \\ -15 & -13 & 1 & -13 & -14 & -15 \\ -19 & -18 & -17 & 1 & -17 & -18 \\ -22 & -22 & -21 & -20 & 1 & -20 \\ -25 & -25 & -25 & -24 & -23 & 1 \end{bmatrix} \tag{7.6}$$

As we can see, the table show that the rejection is almost independent of the interfering spreading factor and is at least an order of magnitude higher than the values reported in the Goursaud's theoretical co-channel rejection matrix. Therefore, the widespreaded belief that spreading factor can be orthogonal does not holds in real deployments. In addition to this the authors also concluded that there is a high capture probability in LoRa transmissions. Indeed, they showed that a SIR of -3 dB is enough for the interfering signal to overcome the reference signal and capture the channel.

Therefore, similarly to the Goursaud's matrix we can also use the SIR matrix presented in Eq. 7.6 to model the interference in LoRa. To do this, we compute the SINR using equation 7.2 and compare the value obtained with the corresponding value of the matrix. It should be noted that this SIR matrix was done by comparing each possible combination of two spreading factors. As a result, it is not valid to determine if a packet is lost due to interference of the reference signal with several signals transmitted with more than one spreading factor. Indeed, if for example, the desired signal is transmitted with SF 7 and there are two interfering signals (one on SF 8 and the other on SF 11) this model cannot be applied.

7.5 Models comparison and limitations

As we could see in the description of the two interference models proposed in the literature for LoRa transmissions, we can identify the following limitations:

1. Goursaud's matrix [1] is made for the case in which the signal has not been de-spreaded (i.e. before decoding with the corresponding spreading factor). Therefore, contrary to Magrin's [10] and our approach a more precise computation of the SINR shall consider the power levels before and after the coding phase.

2. Goursaud's matrix [1] is built in such a way that an arbitrary threshold of at least 6dB is guaranteed after de-preading the signal at the receiver side. Therefore, a more precise computation of the SINR shall include this threshold.

3. More recent research on this matter have concluded that there is no such thing as *pseudo-orthogonal* spreading factors on LoRa. Indeed, in [16] a SIR matrix was built considering pairwise comparisons from actual measurements showing that the impact of interfering transmissions from different spreading factors is not negligible. Therefore, the results obtained in this thesis (and previous investigations using Goursaud's theoretical model) overestimate the performance that can actually be obtained with a LoRa network in a real deployment.

4. Even though Croce's SIR matrix was built based on actual measures from LoRa transmissions, it does not consider the case with multiple interfers on multiple SFs. Therefore, if one wanted to use it to estimate the interference in a very saturated environment, with several nodes; the performance of the network would be overestimated.

BIBLIOGRAPHY

[1] C. Goursaud and J. M. Gorce, "Dedicated networks for IoT: PHY / MAC state of the art and challenges," *EAI Endorsed Transactions on Internet of Things*, vol. 1, no. 1, p. 150597, Oct. 2015. [Online]. Available: https://doi.org/10.4108/eai.26-10-2015.150597

[2] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Express*, vol. 5, no. 1, pp. 1–7, Mar. 2019. [Online]. Available: https://doi.org/10.1016/j.icte.2017.12.005

[3] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *IEEE Network*, vol. 20, no. 3, pp. 41–47, May 2006. [Online]. Available: https://doi.org/10.1109/mnet.2006.1637931

[4] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, Mar. 2002. [Online]. Available: https://doi.org/10.1016/s1389-1286(01)00302-4

[5] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on LPWA technology: LoRa and NB-IoT," *ICT Express*, vol. 3, no. 1, pp. 14–21, Mar. 2017. [Online]. Available: https://doi.org/10.1016/j.icte.2017.03.004

[6] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-range communications in unlicensed bands: the rising stars in the IoT and smart city scenarios," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 60–67, Oct. 2016. [Online]. Available: https://doi.org/10.1109/mwc.2016.7721743

[7] R. E. Chall, S. Lahoud, and M. E. Helou, "LoRaWAN network: Radio propagation models and performance evaluation in various environments in lebanon," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2366–2378, Apr. 2019. [Online]. Available: https://doi.org/10.1109/jiot.2019.2906838

[8] Cisco, "The future of IoT miniguide: The burgeoning IoT market continues," 2019. [Online]. Available: https://www.cisco.com/c/en/us/solutions/internet-of-things/future-of-iot.html

[9] M. Knight and B. Seeber, "Decoding LoRa: Realizing a Modern LPWAN with SDR," in *Proceedings of the 6th GNU Radio Conference*. GNU Radio, 2016. [Online]. Available: https://pubs.gnuradio.org/index.php/grcon/article/view/8

[10] D. Magrin, "Network level performances of a lora system," Master's thesis, Università degli Studi di Padova, 12 2016. [Online]. Available: http://tesi.cab.unipd.it/53740/1/dissertation.pdf

[11] D. Magrin, M. Centenaro, and L. Vangelista, "Performance evaluation of LoRa networks in a smart city scenario," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, may 2017. [Online]. Available: https://doi.org/10.1109/icc.2017.7996384

[12] M. Capuzzo, D. Magrin, and A. Zanella, "Confirmed traffic in LoRaWAN: Pitfalls and countermeasures," in *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. IEEE, jun 2018. [Online]. Available: https://doi.org/10.23919/medhocnet.2018.8407095

[13] D. Magrin, M. Capuzzo, and A. Zanella, "A thorough study of LoRaWAN performance under different parameter settings," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 116–127, Jan. 2020. [Online]. Available: https://doi.org/10.1109/jiot.2019.2946487

[14] SEMTECH, "SX1272 LoRaWAN End Device - Datasheet," Tech. Rep., 2017. [Online]. Available: https://www.semtech.com/uploads/documents/sx1272.pdf

[15] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," 04 2015.

[16] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, "LoRa technology demystified: From link behavior to cell-level performance," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 822–834, Feb. 2020. [Online]. Available: https://doi.org/10.1109/twc.2019.2948872

[17] P. Tanguy and I. Martinez, "LoRaWAN & Jamming," https://sourcesup.renater.fr/www/lorawan-jamming/, 2016, accessed: [02/11/2020].

[18] E. Borgia, "The internet of things vision: Key features, applications and open issues," *Computer Communications*, vol. 54, pp. 1–31, Dec. 2014. [Online]. Available: https://doi.org/10.1016/j.comcom.2014.09.008

[19] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015. [Online]. Available: https://doi.org/10.1109/comst.2015.2444095

[20] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010. [Online]. Available: https://doi.org/10.1016/j.comnet.2010.05.010

[21] P. Ray, "A survey on internet of things architectures," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291–319, Jul. 2018. [Online]. Available: https://doi.org/10.1016/j.jksuci.2016.10.003

[22] V. Rozsa, M. Denisczwicz, M. Dutra, P. Ghodous, C. Ferreira Da Silva, N. Moayeri, F. Biennier, and N. Figay, "An Application Domain-Based Taxonomy for IoT Sensors," Curitiba, Brazil, Oct. 2016. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01581127

[23] F. L. Gall, S. V. Chevillard, A. Gluhak, and Z. Xueli, "Benchmarking internet of things deployments in smart cities," in *2013 27th International Conference on Advanced Information Networking and Applications Workshops*. IEEE, Mar. 2013. [Online]. Available: https://doi.org/10.1109/waina.2013.230

[24] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, "The Internet of Things for Health Care: A Comprehensive Survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[25] E. Poenaru and C. Poenaru, "A structured approach of the Internet-of-Things eHealth use cases," in *2013 E-Health and Bioengineering Conference (EHB)*, Nov. 2013, pp. 1–4.

[26] C. Perera, C. H. Liu, S. Jayawardena, and M. Chen, "A survey on internet of things from industrial market perspective," *IEEE Access*, vol. 2, pp. 1660–1679, 2014. [Online]. Available: https://doi.org/10.1109/access.2015.2389854

[27] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017. [Online]. Available: https://doi.org/10.1109/comst.2017.2652320

[28] L. Ruiz-Garcia and L. Lunadei, "The role of RFID in agriculture: Applications, limitations and challenges," *Computers and Electronics in Agriculture*, vol. 79, no. 1, pp. 42–50, Oct. 2011. [Online]. Available: https://doi.org/10.1016/j.compag.2011.08.010

[29] Z. Bi, L. D. Xu, and C. Wang, "Internet of things for enterprise systems of modern manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1537–1546, May 2014. [Online]. Available: https://doi.org/10.1109/tii.2014.2300338

[30] S. Krco, B. Pokric, and F. Carrez, "Designing IoT architecture(s): A european perspective," in *2014 IEEE World Forum on Internet of Things (WF-IoT)*. IEEE, Mar. 2014. [Online]. Available: https://doi.org/10.1109/wf-iot.2014.6803124

[31] S. Ankit, U. Raksha, and A. Jain, *Internet of Things and Wireless Physical Layer Security: A Survey*. Singapore: Springer Singapore, 2017, pp. 115–123.

[32] V. Baños-Gonzalez, M. Afaqui, E. Lopez-Aguilera, and E. Garcia-Villegas, "IEEE 802.11ah: A technology to face the IoT challenge," *Sensors*, vol. 16, no. 11, p. 1960, Nov. 2016. [Online]. Available: https://doi.org/10.3390/s16111960

[33] N. Ahmed, H. Rahman, and M. Hussain, "A comparison of 802.11ah and 802.15.4 for IoT," *ICT Express*, vol. 2, no. 3, pp. 100–102, Sep. 2016. [Online]. Available: https://doi.org/10.1016/j.icte.2016.07.003

[34] I. C. Society, "IEEE Standard for Low-Rate Wireless Networks Amendment 2: Ultra- Low Power Physical Layer," Tech. Rep., 2016. [Online]. Available: http://standards.ieee.org/getieee802/download/802.15.4q-2016.pdf

[35] U. Mehboob, Q. Zaib, and C. Usama, "Survey of IoT Communication Protocols Techniques, Applications, and Issues," *e-book, xFlow Research Inc*, 2016. [Online]. Available: http://xflowresearch.com/wp-content/uploads/2016/02/Survey-of-IoT-Communication-Protocols.pdf

[36] 3GPP, "3GPP TR 45.820 - Cellular system support for ultra-low complexity and low throughput Internet of Things (CIoT)," 3rd Generation Partnership Project, Tech. Rep., 2015.

[37] P. Reininger, "3gpp standards for the internet of things," Online, 2016. [Online]. Available: http://www.3gpp.org/images/presentations/2016_11_3gpp_Standards_for_IoT.pdf

[38] Ericsson, "3GPP Spectrum Bands," Online, 2017. [Online]. Available: https://www.ericsson.com/assets/local/policy-makers-and-regulators/170331-3gpp-spectrum-bands.pdf

[39] DASH7 Allicance, "DASH7 Alliance Wireless Sensor and Actuator Network Protocol VERSION 1.1," Tech. Rep., 2017.

[40] Maarten Weyn and Michael Andre, "DASH7, Passive RFID and LPWAN," Tech. Rep., 2017.

[41] L. Alliance, "LoRaWAN What is it? - A technical overview of LoRa and LoRaWAN," Tech. Rep., 2015. [Online]. Available: https://docs.wixstatic.com/ugd/eccc1a_ed71ea1cd969417493c74e4a13c55685.pdf

[42] WAVIoT, "WAVIoT NB-FI LPWAN Technology - Products and Tech Description," Tech. Rep., 2016. [Online]. Available: http://waviot.com/wp.pdf

[43] Weightless SIG, "LPWAN Technology Decisions: 17 critical features," Tech. Rep., 2016. [Online]. Available: http://www.weightless.org/membership/lpwan-technology-features-document-update/NTZjMC9MUFdBTiBUZWNobm9sb2d5IERlY2lzaW9uc192MS4xLnBkZg==

[44] Maarten Weyn and Michael Andre, "Wavenis Technology Plateform - Product summary," Tech. Rep., 2013. [Online]. Available: http://www-coronis-com.dyn.elster.com/downloads/Wavenis_Data_Sheet_A4_CS5.pdf

[45] ISO/IEC, "ISO/IEC 18000-3:2010 - Information technology – Radio frequency identification for item management – Part 3: Parameters for air interface communications at 13,56 MHz." [Online]. Available: https://www.iso.org/standard/53424.html

[46] M. LLC. (2017) Z - Wave public specification. [Online]. Available: http://z-wave.sigmadesigns.com/design-z-wave/z-wave-public-specification/

[47] EnOcean, "EnOcean Technology – Energy harvesting wireless," 2011. [Online]. Available: https://www.enocean.com/en/technology/radio-technology/

[48] V. Chawla and D. Ha, "An overview of passive RFID," *IEEE Communications Magazine*, vol. 45, no. 9, pp. 11–17, Sep. 2007. [Online]. Available: https://doi.org/10.1109/mcom.2007.4342873

[49] IrDA, "IrDA Physical Layer Specification v1.5." [Online]. Available: http://www.irdajp.org/irdajp.info/specifications.html

[50] J. Suhonen, M. Kohvakka, V. Kaseva, T. D. Hämäläinen, and M. Hännikäinen, "Key standards and industry specifications," in *Low-Power Wireless Sensor Networks*. Springer US, 2012, pp. 7–15. [Online]. Available: https://doi.org/10.1007/978-1-4614-2173-3_2

[51] M. S. Mahmoud and A. A. H. Mohamad, "A study of efficient power consumption wireless communication techniques/ modules for internet of things (IoT) applications," *Advances in Internet of Things*, vol. 06, no. 02, pp. 19–29, 2016. [Online]. Available: https://doi.org/10.4236/ait.2016.62002

[52] B. Martinez, M. Montón, I. Vilajosana, and J. D. Prades, "The Power of Models: Modeling Power Consumption for IoT Devices," *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5777–5789, Oct. 2015.

[53] A. M. Baharudin and W. Yan, "Long-range wireless sensor networks for geo-location tracking: Design and evaluation," in *2016 International Electronics Symposium (IES)*. IEEE, Sep. 2016. [Online]. Available: https://doi.org/10.1109/elecsym.2016.7860979

[54] J.-P. Bardyn, T. Melly, O. Seller, and N. Sornin, "IoT: The era of LPWAN is starting now," in *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*. IEEE, Sep. 2016. [Online]. Available: https://doi.org/10.1109/esscirc.2016.7598235

[55] A. Hoglund, X. Lin, O. Liberg, A. Behravan, E. A. Yavuz, M. Van Der Zee, Y. Sui, T. Tirronen, A. Ratilainen, and D. Eriksson, "Overview of 3gpp Release 14 Enhanced NB-IoT," *IEEE Network*, vol. 31, no. 6, pp. 16–22, Nov. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8120239/

[56] Qualcomm, "Paving the path to Narrowband 5G with LTE Internet of Things (IoT)," 2016.

[57] A. Lavric, A. I. Petrariu, and V. Popa, "Long range SigFox communication protocol scalability analysis under large-scale, high-density conditions," *IEEE Access*, vol. 7, pp. 35 816–35 825, 2019. [Online]. Available: https://doi.org/10.1109/access.2019.2903157

[58] F. Sforza., "Communications system," 2009, uS patent 8406275B2. [Online]. Available: https://patents.google.com/patent/US8406275B2/en

[59] L. Vangelista, "Frequency shift chirp modulation: The LoRa modulation," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1818–1821, Dec. 2017. [Online]. Available: https://doi.org/10.1109/lsp.2017.2762960

[60] J. C. Liando, A. Gamage, A. W. Tengourtius, and M. Li, "Known and unknown facts of LoRa," *ACM Transactions on Sensor Networks*, vol. 15, no. 2, pp. 1–35, Apr. 2019. [Online]. Available: https://doi.org/10.1145/3293534

[61] L. Alliance, "LoRaWAN 1.1 and Backend Interfaces 1.0 Specification," Tech. Rep., 2017.

[62] O. Garcia-Morchon, S. Kumar, and M. Sethi, "State-of-the-Art and Challenges for the Internet of Things Security," Internet Engineering Task Force, Internet-Draft draft-irtf-t2trg-iot-seccons-12, Feb. 2018, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-irtf-t2trg-iot-seccons-12

[63] Z. Li and T. Xin, "Threat Modeling and Countermeasures Study for the Internet of Things," *Journal of Convergence Information Technology*, vol. 8, no. 5, pp. 1163–1171, mar 2013.

[64] A. W. Atamli and A. Martin, "Threat-based security analysis for the internet of things," in *2014 International Workshop on Secure Internet of Things*. IEEE, Sep. 2014. [Online]. Available: https://doi.org/10.1109/siot.2014.10

[65] T. Pecorella, L. Brilli, and L. Mucchi, "The role of physical layer security in IoT: A novel perspective," *Information*, vol. 7, no. 3, p. 49, Aug. 2016. [Online]. Available: https://doi.org/10.3390/info7030049

[66] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: The internet of things architecture, possible applications and key challenges," in *2012 10th International Conference on Frontiers of Information Technology*. IEEE, Dec. 2012. [Online]. Available: https://doi.org/10.1109/fit.2012.53

[67] M. El-hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A survey of internet of things (IoT) authentication schemes," *Sensors*, vol. 19, no. 5, p. 1141, Mar. 2019. [Online]. Available: https://doi.org/10.3390/s19051141

[68] S. R. Oh and Y. G. Kim, "Security requirements analysis for the iot," in *International Conference on Platform Technology and Service (PlatCon)*, Feb 2017, pp. 1–6.

[69] M. Abomhara and G. M. Koien, "Security and privacy in the internet of things: Current status and open issues," in *2014 International Conference on Privacy and Security in Mobile Systems (PRISMS)*. IEEE, May 2014. [Online]. Available: https://doi.org/10.1109/prisms.2014.6970594

[70] B. Krebs, "KrebsOnSecurity Hit With Record DDoS," 2016, Accessed: 2020-09-30. [Online]. Available: https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/

[71] T. guardian, "DDoS attack that disrupted internet was largest of its kind in history, experts say," 2016, Accessed: 2020-09-30. [Online]. Available: https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet

[72] G. Glissa, A. Rachedi, and A. Meddeb, "A secure routing protocol based on RPL for internet of things," in *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, Dec. 2016. [Online]. Available: https://doi.org/10.1109/glocom.2016.7841543

[73] D. Airehrour, J. Gutierrez, and S. K. Ray, "Securing RPL routing protocol from blackhole attacks using a trust-based mechanism," in *2016 26th International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, Dec. 2016. [Online]. Available: https://doi.org/10.1109/atnac.2016.7878793

[74] SEMTECH, "What are LoRa® and LoRaWAN®?" https://lora-developers.semtech.com/library/tech-papers-and-guides/lora-and-lorawan/, 2019, accessed: [02/11/2020].

[75] L. Alliance, "LoRa Specification 1.0.1," Tech. Rep., 2015.

[76] ——, "LoRaWAN Backend Interfaces 1.0 Specification," Tech. Rep., 2017.

[77] SEMTECH, "An In-depth Look at LoRaWAN Class B Devices," Tech. Rep., 2019. [Online]. Available: https://lora-developers.semtech.com/library/tech-papers-and-guides/lorawan-class-b-devices/

[78] ——, "An1200.22 - LoRa Modulation Basics," Tech. Rep., 2015.

[79] ——, "SX1301 LoRaWAN Gateway - Datasheet," Tech. Rep., 2017. [Online]. Available: https://www.semtech.com/uploads/documents/sx1301.pdf

[80] M. Knight, "Reversing LoRa," Tech. Rep., 2016. [Online]. Available: https://github.com/matt-knight/research

[81] SEMTECH, "SX1272/3/6/7/8 LoRa Modem Designer's Guide," Tech. Rep., 2013. [Online]. Available: https://www.semtech.com/uploads/documents/LoraDesignGuide_STD.pdf

[82] L. Casals, B. Mir, R. Vidal, and C. Gomez, "Modeling the energy performance of LoRaWAN," *Sensors*, vol. 17, no. 10, p. 2364, oct 2017.

[83] SEMTECH, "An1200.22 - Reading Channel RSSI during a CAD," Tech. Rep., 2014.

[84] ETSI, "EN 300 220-1 v2.4.1 Electromagnetic compatibility and Radio spectrum Matters (erm) Short Range Devices (SRD); Radio equipment to be used in the 25 MHz to 1000 MHz frequency range with power levels ranging up to 500 mW," Tech. Rep., 2012.

[85] S. Li, U. Raza, and A. Khan, "How agile is the adaptive data rate mechanism of LoRaWAN?" in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, Dec. 2018. [Online]. Available: https://doi.org/10.1109/glocom.2018.8647469

[86] N. BENKAHLA, H. TOUNSI, Y.-Q. SONG, and M. FRIKHA, "Enhanced ADR for LoRaWAN networks with mobility," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, Jun. 2019. [Online]. Available: https://doi.org/10.1109/iwcmc.2019.8766738

[87] J. Park, K. Park, H. Bae, and C. kwon Kim, "EARN: Enhanced ADR with coding rate adaptation in LoRaWAN," *IEEE Internet of Things Journal*, pp. 1–1, 2020. [Online]. Available: https://doi.org/10.1109/jiot.2020.3005881

[88] J. Finnegan, R. Farrell, and S. Brown, "Analysis and enhancement of the LoRaWAN adaptive data rate scheme," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7171–7180, Aug. 2020. [Online]. Available: https://doi.org/10.1109/jiot.2020.2982745

[89] F. Cuomo, J. C. C. Gamez, A. Maurizio, L. Scipione, M. Campo, A. Caponi, G. Bianchi, G. Rossini, and P. Pisani, "Towards traffic-oriented spreading factor allocations in LoRaWAN systems," in *2018 17th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. IEEE, Jun. 2018. [Online]. Available: https://doi.org/10.23919/medhocnet.2018.8407091

[90] R. Marini, W. Cerroni, and C. Buratti, "A novel collision-aware adaptive data rate algorithm for LoRaWAN networks," *IEEE Internet of Things Journal*, pp. 1–1, 2020. [Online]. Available: https://doi.org/10.1109/jiot.2020.3020189

[91] J. M. Marais, R. Malekian, and A. M. Abu-Mahfouz, "Evaluating the LoRaWAN protocol using a permanent outdoor testbed," *IEEE Sensors Journal*, vol. 19, no. 12, pp. 4726–4733, Jun. 2019. [Online]. Available: https://doi.org/10.1109/jsen.2019.2900735

[92] A. Hoeller, R. D. Souza, S. Montejo-Sanchez, and H. Alves, "Performance analysis of single-cell adaptive data rate-enabled LoRaWAN," *IEEE Wireless Communications Letters*, vol. 9, no. 6, pp. 911–914, Jun. 2020. [Online]. Available: https://doi.org/10.1109/lwc.2020.2975604

[93] V. Hauser and T. Hegr, "Proposal of adaptive data rate algorithm for LoRaWAN-based infrastructure," in *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, Aug. 2017. [Online]. Available: https://doi.org/10.1109/ficloud.2017.47

[94] D.-T. Ta, K. Khawam, S. Lahoud, C. Adjih, and S. Martin, "LoRa-MAB: Toward an intelligent resource allocation approach for LoRaWAN," in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, Dec. 2019. [Online]. Available: https://doi.org/10.1109/globecom38437.2019.9013345

[95] CEPT, "ERC recommendation 7003 Relating to the use of Short Range Devices (SRD)," Tech. Rep., 2020.

[96] R. Miller, "LoRa Security Building a Secure LoRa Solution," https://labs.f-secure.com/assets/BlogFiles/mwri-LoRa-security-guide-1.2-2016-03-22.pdf, 2016, accessed: [02/11/2020].

[97] I. Butun, N. Pereira, and M. Gidlund, "Security risk analysis of LoRaWAN and future directions," *Future Internet*, vol. 11, no. 1, p. 3, Dec. 2018. [Online]. Available: https://doi.org/10.3390/fi11010003

[98] ——, "Analysis of LoRaWAN v1.1 security," in *Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects - SMARTOBJECTS '18*. ACM Press, 2018. [Online]. Available: https://doi.org/10.1145/3213299.3213304

[99] M. Eldefrawy, I. Butun, N. Pereira, and M. Gidlund, "Formal security analysis of LoRaWAN," *Computer Networks*, vol. 148, pp. 328–339, Jan. 2019. [Online]. Available: https://doi.org/10.1016/j.comnet.2018.11.017

[100] X. Yang, E. Karampatzakis, C. Doerr, and F. Kuipers, "Security vulnerabilities in LoRaWAN," in *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*. IEEE, Apr. 2018. [Online]. Available: https://doi.org/10.1109/iotdi.2018.00022

[101] X. Yang, "Lorawan: Vulnerability analysis and practical exploitation," Master's thesis, 2017. [Online]. Available: http://resolver.tudelft.nl/uuid:87730790-6166-4424-9d82-8fe815733f1e

[102] G. Avoine and L. Ferreira, "Rescuing lorawan 1.0," in *unpublished*, 2018. [Online]. Available: https://fc18.ifca.ai/preproceedings/13.pdf

[103] E. Aras, G. S. Ramachandran, P. Lawrence, and D. Hughes, "Exploring the security vulnerabilities of LoRa," in *2017 3rd IEEE International Conference on Cybernetics (CYBCONF)*. IEEE, jun 2017. [Online]. Available: https://doi.org/10.1109/cybconf.2017.7985777

[104] E. Aras, N. Small, G. S. Ramachandran, S. Delbruel, W. Joosen, and D. Hughes, "Selective jamming of LoRaWAN using commodity hardware," in *14th EAI International Conference on Mobile and Ubiquitous Systems Computing Networking and Services*. ACM Press, 2017. [Online]. Available: https://doi.org/10.1145/3144457.3144478

[105] D. Basu, T. Gu, and P. Mohapatra, "Security issues of low power wide area networks in the context of lora networks," 2020.

[106] R. Housley, "Using advanced encryption standard (AES) counter-mode with IPsec encapsulating security payload (ESP)," Internet Requests for Comments, VigilSecurity, RFC 3686, January 2004. [Online]. Available: https://tools.ietf.org/rfc/rfc3686.txt

[107] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of LoRaWAN," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017. [Online]. Available: https://doi.org/10.1109/mcom.2017.1600613

[108] A. Proano and L. Lazos, "Selective jamming attacks in wireless networks," in *2010 IEEE International Conference on Communications*. IEEE, May 2010. [Online]. Available: https://doi.org/10.1109/icc.2010.5502322

[109] R. B. Sorensen, D. M. Kim, J. J. Nielsen, and P. Popovski, "Analysis of latency and MAC-layer performance for class a LoRaWAN," *IEEE Wireless Communications Letters*, vol. 6, no. 5, pp. 566–569, Oct. 2017. [Online]. Available: https://doi.org/10.1109/lwc.2017.2716932

[110] D. Bankov, E. Khorov, and A. Lyakhov, "LoRaWAN modeling and MCS allocation to satisfy heterogeneous QoS requirements," *Sensors*, vol. 19, no. 19, p. 4204, Sep. 2019. [Online]. Available: https://doi.org/10.3390/s19194204

[111] M. Capuzzo, D. Magrin, and A. Zanella, "Mathematical modeling of LoRa WAN performance with bi-directional traffic," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, Dec. 2018. [Online]. Available: https://doi.org/10.1109/glocom.2018.8647351

[112] B. Paul, "A novel mathematical model to evaluate the impact of packet retransmissions in LoRaWAN," *IEEE Sensors Letters*, vol. 4, no. 5, pp. 1–4, May 2020. [Online]. Available: https://doi.org/10.1109/lsens.2020.2986794

[113] N. Abramson, "The aloha system: Another alternative for computer communications," in *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference*, ser. AFIPS '70 (Fall). New York, NY, USA: Association for Computing Machinery, 1970, p. 281–285. [Online]. Available: https://doi.org/10.1145/1478462.1478502

[114] D. Magrin, M. Capuzzo, A. Zanella, and M. Zorzi, "A complete lorawan model for single-gateway scenarios," 2019.

[115] T. Voigt, M. Bor, U. Roedig, and J. Alonso, "Mitigating inter-network interference in lora networks," in *Proceedings of the 2017 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '17. USA: Junction Publishing, 2017, p. 323–328. [Online]. Available: https://arxiv.org/pdf/1611.00688.pdf

[116] M. C. Bor, U. Roedig, T. Voigt, and J. 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*. ACM, Nov. 2016. [Online]. Available: https://doi.org/10.1145/2988287.2989163

[117] A.-I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, "Does bidirectional traffic do more harm than good in LoRaWAN based LPWA networks?" in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. IEEE, Dec. 2017. [Online]. Available: https://doi.org/10.1109/glocom.2017.8254509

[118] D.-T. Ta, K. Khawam, S. Lahoud, C. Adjih, and S. Martin, "LoRa-MAB: A flexible simulator for decentralized learning resource allocation in IoT networks," in *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, Sep. 2019. [Online]. Available: https://doi.org/10.23919/wmnc.2019.8881393

[119] M. Slabicki, G. Premsankar, and M. D. Francesco, "Adaptive configuration of lora networks for dense IoT deployments," in *NOMS 2018 - 2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, Apr. 2018. [Online]. Available: https://doi.org/10.1109/noms.2018.8406255

[120] J. Haxhibeqiri, F. Van den Abeele, I. Moerman, and J. Hoebeke, "LoRa scalability: A simulation model based on interference measurements," *Sensors*, vol. 17, no. 6, p. 1193, May 2017. [Online]. Available: https://doi.org/10.3390/s17061193

[121] F. V. den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, "Scalability analysis of large-scale LoRaWAN networks in ns-3," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2186–2198, Dec. 2017. [Online]. Available: https://doi.org/10.1109/jiot.2017.2768498

[122] D. Croce, M. Gucciardo, S. Mangione, G. Santaromita, and I. Tinnirello, "Impact of LoRa imperfect orthogonality: Analysis of link-level performance," *IEEE Communications Letters*, vol. 22, no. 4, pp. 796–799, Apr. 2018. [Online]. Available: https://doi.org/10.1109/lcomm.2018.2797057

[123] M. Centenaro, L. Vangelista, and R. Kohno, "On the impact of downlink feedback on LoRa performance," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, Oct. 2017. [Online]. Available: https://doi.org/10.1109/pimrc.2017.8292315

[124] G. Callebaut, G. Ottoy, and L. van der Perre, "Cross-layer framework and optimization for efficient use of the energy budget of IoT nodes," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, Apr. 2019. [Online]. Available: https://doi.org/10.1109/wcnc.2019.8885739

[125] A. M. Yousuf, E. M. Rochester, B. Ousat, and M. Ghaderi, "Throughput, coverage and scalability of LoRa LPWAN for internet of things," in *2018 IEEE/ACM 26th*

International Symposium on Quality of Service (IWQoS). IEEE, Jun. 2018. [Online]. Available: https://doi.org/10.1109/iwqos.2018.8624157

[126] N. Kouvelas, V. Rao, and R. R. V. Prasad, "Employing p-csma on a lora network simulator," 2018.

[127] B. Reynders, Q. Wang, and S. Pollin, "A LoRaWAN module for ns-3," in *Proceedings of the 10th Workshop on ns-3 - WNS3 '18*. ACM Press, 2018. [Online]. Available: https://doi.org/10.1145/3199902.3199913

[128] T.-H. To and A. Duda, "Simulation of LoRa in NS-3: Improving LoRa performance with CSMA," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, May 2018. [Online]. Available: https://doi.org/10.1109/icc.2018.8422800

[129] F. H. Khan and M. Portmann, "Experimental evaluation of LoRaWAN in NS-3," in *2018 28th International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, Nov. 2018. [Online]. Available: https://doi.org/10.1109/atnac.2018.8615313

[130] I. Martinez, P. Tanguy, and F. Nouvel, "On the performance evaluation of LoRaWAN under jamming," in *2019 12th IFIP Wireless and Mobile Networking Conference (WMNC)*. IEEE, Sep. 2019. [Online]. Available: https://doi.org/10.23919/wmnc.2019.8881830

[131] I. Martinez, F. Nouvel, S. Lahoud, P. Tanguy, and M. E. Helou, "On the performance evaluation of LoRaWAN with re-transmissions under jamming," in *2020 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, Jul. 2020. [Online]. Available: https://doi.org/10.1109/iscc50000.2020.9219644

[132] M. O. Farooq and D. Pesch, "Poster: Extended lorasim to simulate multiple iot applications in a lorawan," in *Proceedings of the 2018 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '18. USA: Junction Publishing, 2018, p. 175–176.

[133] C. Pham, "Robust CSMA for long-range LoRa transmissions with image sensing devices," in *2018 Wireless Days (WD)*. IEEE, Apr. 2018. [Online]. Available: https://doi.org/10.1109/wd.2018.8361706

[134] J. Zheng and M. Lee, "A comprehensive performance study of ieee 802.15.4," *Sensor Network Operations*, 01 2004.

[135] IUT, "P.1238 : Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 300 MHz to 450 ghz," Tech. Rep., 2001.

[136] E. C. A. 231, "Digital mobile radio towards future generation systems, final report," Tech. Rep., 1999.

[137] MULTITECH, "MultiTechmDot data-sheet," Tech. Rep., 2019. [Online]. Available: https://www.multitech.com/documents/publications/data-sheets/86002171.pdf

[138] Microchip, "Rn2483 Low-Power Long Range LoRa Technology Transceiver Module," Tech. Rep., 2015. [Online]. Available: http://ww1.microchip.com/downloads/en/devicedoc/50002346c.pdf

[139] C. Hesling, "LoRaBug - SX1276," Tech. Rep., 2019. [Online]. Available: https://github.com/OpenChirp/LoRaBug

[140] H. Electronics, "HM-TRLR-LF/HFS Series 100mW LoRa Transceiver," Tech. Rep., 2018. [Online]. Available: https://github.com/OpenChirp/LoRaBug

[141] ns-3 Consortium, "What is ns-3?" 2020, Accessed: 2020-09-30. [Online]. Available: https://www.nsnam.org/about/

[142] A. Mpitziopoulos, D. Gavalas, C. Konstantopoulos, and G. Pantziou, "A survey on jamming attacks and countermeasures in WSNs," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 42–56, 2009. [Online]. Available: https://doi.org/10.1109/surv.2009.090404

[143] S. M. Danish, A. Nasir, H. K. Qureshi, A. B. Ashfaq, S. Mumtaz, and J. Rodriguez, "Network intrusion detection system for jamming attack in LoRaWAN join procedure," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, May 2018. [Online]. Available: https://doi.org/10.1109/icc.2018.8422721

[144] O. Osanaiye, A. Alfa, and G. Hancke, "A statistical approach to detect jamming attacks in wireless sensor networks," *Sensors*, vol. 18, no. 6, p. 1691, May 2018. [Online]. Available: https://doi.org/10.3390/s18061691

[145] B. Upadhyaya, S. Sun, and B. Sikdar, "Machine learning-based jamming detection in wireless IoT networks," in *2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*. IEEE, Aug. 2019. [Online]. Available: https://doi.org/10.1109/vts-apwcs.2019.8851633

[146] O. Punal, I. Aktas, C.-J. Schnelke, G. Abidin, K. Wehrle, and J. Gross, "Machine learning-based jamming detection for IEEE 802.11: Design and experimental evaluation," in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014*. IEEE, Jun. 2014. [Online]. Available: https://doi.org/10.1109/wowmom.2014.6918964

[147] O. Puñal, C. Pereira, A. Aguiar, and J. Gross, "Crawdad dataset uportorwthaachen/vanetjamming2014 (v. 2014-05-12)," 2014. [Online]. Available: http://crawdad.org/uportorwthaachen/vanetjamming2014/20140512/

[148] K. Bauer, E. W. Anderson, D. McCoy, D. Grunwald, and D. C. Sicker, "Crawdad dataset cu/rssi (v. 2009-05-28)," 2009. [Online]. Available: http://crawdad.org/cu/rssi/20090528/

[149] I. Suciu, J. C. Pacho, A. Bartoli, and X. Vilajosana, "Authenticated preambles for denial of service mitigation in LPWANs," in *Ad-hoc, Mobile, and Wireless Networks*. Springer International Publishing, 2018, pp. 199–210. [Online]. Available: https://doi.org/10.1007/978-3-030-00247-3_19

[150] A.-U.-H. Ahmar, E. Aras, W. Joosen, and D. Hughes, "Towards more scalable and secure LPWAN networks using cryptographic frequency hopping," in *2019 Wireless Days (WD)*. IEEE, Apr. 2019. [Online]. Available: https://doi.org/10.1109/wd.2019.8734249

[151] A. Gamage, J. C. Liando, C. Gu, R. Tan, and M. Li, "LMAC: efficient carrier-sense multiple access for LoRa," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*. ACM, Sep. 2020. [Online]. Available: https://doi.org/10.1145/3372224.3419200

[152] E. Sasikala and N. Rengarajan, "An intelligent technique to detect jamming attack in wireless sensor networks (WSNs)," *International Journal of Fuzzy Systems*, vol. 17, no. 1, pp. 76–83, Mar. 2015. [Online]. Available: https://doi.org/10.1007/s40815-015-0009-4

[153] A. Wood, J. Stankovic, and S. Son, "JAM: a jammed-area mapping service for sensor networks," in *Proceedings. 2003 International Symposium on System-on-Chip (IEEE Cat. No.03EX748)*. IEEE Comput. Soc. [Online]. Available: https://doi.org/10.1109/real.2003.1253275

[154] T. Cheng, P. Li, and S. Zhu, "An algorithm for jammer localization in wireless sensor networks," in *2012 IEEE 26th International Conference on Advanced Information Networking and Applications*. IEEE, Mar. 2012. [Online]. Available: https://doi.org/10.1109/aina.2012.11

[155] Y. Chabchoub, R. Chiky, and B. Dogan, "How can sliding HyperLogLog and EWMA detect port scan attacks in IP traffic?" *EURASIP Journal on Information Security*, vol. 2014, no. 1, Feb. 2014. [Online]. Available: https://doi.org/10.1186/1687-417x-2014-5

[156] S. W. Roberts, "Control chart tests based on geometric moving averages," *Technometrics*, vol. 1, no. 3, pp. 239–250, Aug. 1959. [Online]. Available: https://doi.org/10.1080/00401706.1959.10489860

[157] H. Raza, G. Prasad, and Y. Li, "Dataset shift detection in non-stationary environments using EWMA charts," in *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, Oct. 2013. [Online]. Available: https://doi.org/10.1109/smc.2013.537

[158] P. Cisar, S. Bošnjak, and S. M. Cisar, "EWMA algorithm in network practice," *International Journal of Computers Communications & Control*, vol. 5, no. 2, p. 160, Jun. 2010. [Online]. Available: https://doi.org/10.15837/ijccc.2010.2.2471

[159] C. Olah, "Understanding LSTM Networks," https://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2016, accessed: [09/11/2020].

[160] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[161] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *CoRR*, vol. abs/1611.03530, 2016. [Online]. Available: http://arxiv.org/abs/1611.03530

[162] J. Park, "RNN based Time-series Anomaly Detector Model Implemented in Pytorch," https://github.com/chickenbestlover/RNN-Time-series-Anomaly-Detection, 2018, accessed: [29/10/2020].

[163] ST, "Multiprotocol LPWAN 32-bit Arm Cortex-M4 MCUs," https://www.st.com/resource/en/datasheet/stm32wle5c8.pdf, note = Accessed: [29/10/2020], 2016.

[164] dragino, "Lora Shield," https://wiki.dragino.com/index.php?title=Lora_Shield#Wireless_Specifications_of_RFM95W, 2016, accessed: [29/10/2020].

[165] IMST, "iC880A-SPI LoRa Concentrator," https://wireless-solutions.de/products/lora/radio-modules/ic880a-spi/, 2018, accessed: [29/10/2020].

[166] chirpstack, "ChirpStack open-source LoRaWAN Network Server," https://www.chirpstack.io/, 2018, accessed: [29/10/2020].

INSA de RENNES
Service des Formations

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse:
Jamming on LoRaWAN Networks : from modelling to detection

Nom Prénom de l'auteur : MARTINEZ BOLIVAR IVAN MARINO

Membres du jury :
- Madame GOURSAUD Claire
- Monsieur CLAVIER Laurent
- Monsieur CHAMOUN Maroun
- Monsieur TANGUY Philippe
- Monsieur BERDER Olivier
- Madame NOUVEL Fabienne

Président du jury : Olivier BERDER

Date de la soutenance : 12 Janvier 2021

Reproduction de la these soutenue

☒ Thèse pouvant être reproduite en l'état
☐ Thèse pouvant être reproduite après corrections suggérées

Fait à Rennes, le 12 Janvier 2021

Signature du président de jury

Le Directeur,

Abdellatif MIRAOUI

Titre : Brouillage sur les réseaux LoRaWAN : de la modélisation à la détection

Mots clés : LoRaWAN, LPWAN, Brouillage, ns3, Apprentissage Automatique, RNN.

Résumé : L'IdO (Internet des Objets) connaît une grande expansion. Parmi les technologies de l'IdO, les réseaux étendus basse consommation (LPWAN) sont des solutions rencontrant actuellement un fort engouement et semblent donc prometteuses. Elles sont adaptées aux appareils qui nécessitent d'envoyer de petites quantités de données sur une longue distance, à faible coût, tout en offrant une durée de vie importante de la batterie. Comme pour tout système de communication, la sécurité est une des plus grandes préoccupations. Pour LPWAN, la sécurité est actuellement assurée par des algorithmes à clé symétrique. Néanmoins, cela ne le protège pas contre les attaques de bas niveau telles que les attaques dites de brouillage.

Dans cette thèse, nous nous concentrons sur les réseaux LoRaWAN soumis à des attaques de brouillage de niveau PHY. Nous abordons ce problème à travers trois axes principaux :

- Modélisation réseau : un nouveau modèle mathématique et un module ns3 pour LoRaWAN en présence de brouilleurs.
- Évaluation des performances de LoRaWAN sous brouillage.
- Contre-mesures de brouillage : (i) un algorithme de détection d'attaque de brouillage par l'approche statistique et (ii) par l'apprentissage automatique .

Title : Jamming on LoRaWAN Networks : from modelling to detection

Keywords : LoRaWAN, LPWAN, Jamming, ns3, Machine Learning, RNN

Abstract : The IoT (Internet of Things) knows a great expansion with a predicted of 75 billions of IoT devices by 2025 [Cisco]. LPWAN have emerged as one of the most promising IoT wireless technologies. They are appropriate for devices that require small amounts of data over a long range, at a low cost, while maintaining long battery life. As for any communication systems, security is one of the biggest concerns. In LPWAN, security is currently provided by symmetric-key algorithms such as AES-128 at upper levels. In the case of LoRaWAN it offers application level payload encryption and network level integrity. Nevertheless, this does not shield it against attacks at a lower level such as jammer-type attacks. A jamming attack takes place at the PHY layer. It is an external node that sends unauthenticated packets to every wireless station in the network with the aim of disrupting communications.

In this thesis, we focus on LoRaWAN networks under jamming attacks. We address this problem through a holistic approach that encompasses three fundamental axes:

- Network modeling: A new mathematical model for LoRaWAN under jamming and an enhancement of a ns3 module for LoRaWAN under jamming.
- Performance evaluation of LoRaWAN under jamming: Extensive simulation campaigns.
- Jamming countermeasures: (i) a jamming attack detection algorithm using a statistical approach and (ii) a jamming attack detection algorithm using machine learning.