THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Informatique et Télécommunication

Présentée et soutenue par :

Mme SELMA ZAMOUM

le jeudi 28 novembre 2019

Titre :

Random Access Techniques for Satellite Communications

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (IRIT)

Directeur(s) de Thèse :

MME MARIE LAURE BOUCHERET
M. JÉRÔME LACAN

Rapporteurs :

M. JEAN-MARIE GORCE, INSA LYON
M. JEAN-PIERRE CANCES, UNIVERSITE DE LIMOGES

Membre(s) du jury :

M. DIDIER LE RUYET, CNAM PARIS, Président
M. JEAN-BAPTISTE DUPE, CENTRE NATIONAL D'ETUDES SPATIALES CNES, Membre
M. JÉRÔME LACAN, ISAE-SUPAERO, Membre
M. MATHIEU GINESTE, THALES ALENIA SPACE, Membre
Mme ANA ISABEL PEREZ-NEIRA, UNIV POLITECNICA DE CATALUNYA BARCELONA, Membre
Mme MARIE LAURE BOUCHERET, TOULOUSE INP, Membre

Remerciements

L'homme ne se construit qu'en poursuivant ce qui le dépasse. Cette citation d'André Malraux m'a toujours interpellée dans les situations où je devais entreprendre de nouvelles choses. La peur de l'échec nous mène parfois à nous remettre en question, parce que nous sommes dépassés par l'inconnu. Toutefois, savoir transformer cette appréhension en force et en insistance, est ce qui nous permet d'avancer en tant qu'être, communauté et civilisation. Cette thèse de doctorat était une belle aventure remplie avec toute sorte d'émotions et de sentiments: hésitation, peur de l'inconnu, déception, mais aussi curiosité, enthousiasme, passion, motivation et satisfaction. Beaucoup de personnes et d'entités que je voudrais remercier, ont rendu ce travail de recherche possible, tant sur le plan scientifique et technique mais aussi humain et moral.

Je tiens tout d'abord à adresser mes plus sincères remerciements à mes directeurs de thèse Jérôme Lacan et Marie-Laure Boucheret pour l'intérêt qu'il ont porté à ce sujet de thèse et leurs contributions dans le domaine. Je remercie Marie-Laure pour son regard expert, enrichissant, et complémentaire envers mon travail de recherche mais aussi pour tous les échanges intéressants que nous avons eus sur divers sujets. Je remercie également Jérôme pour la qualité de son suivi, son enthousiasme et son orientation tout au long des trois années qui ont permis d'alimenter mes réflexions et mener mon travail à bon port. Merci Jérôme de m'avoir appris à ne jamais abandonner une idée, pas avant de l'avoir méticuleusement étudiée et analysée, merci aussi pour l'optimisme et les précieux encouragements. Ensuite, je souhaite remercier le CNES et Thales Alenia Space d'avoir financé cette thèse et d'avoir apporté un regard curieux et expert dans le domaine grâce à mes encadrants Jean-Baptiste Dupé et Mathieu Gineste. Merci Mathieu pour tous les conseils et questions qui m'ont permis d'améliorer la qualité de mon travail. Merci Jean-Baptiste pour les précieux encouragements, explications pertinentes, conseils et disponibilité. Je souhaite notamment remercier Patrice Raveneau qui s'est beaucoup intéressé à ma thèse et m'a suivie dès son arrivée au CNES; merci Patrice pour ta perspicacité et pour tes judicieuses remarques et questions. Vous avez tous contribué à l'enrichissement de ce travail de recherche et je vous témoigne sincèrement ma profonde gratitude.

Je remercie Jean-Marie Gorce et Jean-Pierre Cances d'avoir rapporté ma thèse, d'y avoir consacré du temps, d'avoir su lire entre les lignes et pour tous les retours constructifs. Je tiens aussi à remercier les examinateurs Ana Isabel Pérez-Neira et Didier Le Ruyet pour leurs

questions et remarques pertinentes. Je souhaite notamment remercier Didier Le Ruyet d'avoir présidé la soutenance de ma thèse et l'ensemble des membres du jury d'avoir évalué et valorisé mon travail de recherche. J'ai aussi beaucoup apprécié les divers échanges que nous avons eus durant le repas de soutenance, j'en garde un très bon souvenir et j'espère sincèrement vous revoir dans le futur pour d'éventuelles collaborations.

Tout au long de ma thèse, j'ai eu le plaisir et la chance de travailler au sein du laboratoire TéSA, très cher à mon cœur. Tout d'abord, je m'adresse à la maman TéSA: Corinne, merci infiniment pour l'accueil et l'hospitalité et surtout d'avoir créé cette ambiance de famille à TéSA (même si tu ne te rends pas compte) très favorable à la vie doctorale. Merci pour les beaux échanges, pour ton enthousiasme et ta positivité. Isabelle, merci pour ta bonne humeur, les blagues très "spéciales" et ton esprit riche. Tu ajoutes une bonne touche harmonieuse à la famille TéSA. Raoul, merci de m'avoir rassurée souvent ("tout le temps"), d'avoir été un bon collègue de bureau et un bon ami. Merci de répondre à toutes mes questions, merci pour les discussions infinies en science, technologie, politique, actualité, et merci pour ton humour très "spécial". J'ai aussi partagé mon bureau avec Sylvain, Oumaima, et Karine, merci pour tous les échanges scientifiques et autres que nous avons eus. J'entends toujours le rire de Karine (you are missed girl). Je voudrais aussi remercier tous les doctorants TéSA pour leur belle amitié. Lorenzo, j'aime nos discussions et nos querelles, merci d'avoir bien pris le temps de répondre à mes questions, je suis contente que nous ayons fini cette belle aventure de doctorat que nous avons commencée exactement le même jour, il y a trois ans. Adrien, le meneur des blagues en douce durant les pauses, tu t'en sors bien (pour le moment), merci pour ta bonne humeur et ton réalisme. Babouchette, surtout ne change pas, tu es géniale, tu as apporté une belle valeur ajoutée humaine au laboratoire, et tu mets une bonne ambiance avec tes fameux chants (déclenchés souvent par une phrase dite par quelqu'un), merci pour tout. Quentin, camarade occasionnel de bureau, merci de m'avoir fait découvrir le Taekwondo, ça défoule bien après une longue journée, et merci pour toutes les répliques rigolottes, tu es cool, cool cool cool. Amal, merci pour les riches discussions théologiques, littéraires et parfois politiques, j'ai apprécié chaque livre échangé et j'ai hâte pour les prochains, tu es une bonne personne et je te souhaite bon courage pour la dernière année. Merci aussi à tous ceux qui sont devenus aujourd'hui docteurs: Charles-Ugo, Romain, Simone (et sa douce Agathe), Fabio (et sa mignonne petite famille), Antoine, Julien (merci pour la bonne musique Juju) nous avons passé de bons moments au labo et en dehors. Merci aux ingénieurs de recherche: Philippe pour les bonnes recettes et pour tout ce que j'ai pu appendre de toi à travers tes opinions et ta culture, Serge pour les discussions occasionnelles mais j'attends toujours une invitation pour venir t'écouter jouer. J'aimerais aussi remercier toutes les autres personnes à TéSA: Jean-Yves pour la belle aventure d'enseignement que j'ai savourée jusqu'à la fin et l'esprit sympa, Patrice pour les innombrables blagues (pas faites exprès) et les interminables discussions sur l'histoire, la politique et les plantes, Bernard et Jacques pour leurs encouragements et les quelques échanges. Je n'oublie pas bien sûr les stagiaires et externes qui sont passés par TéSA: Tatiana, Nelson, Kareth ...etc

Au cours de ma thèse, j'ai aussi eu l'opportunité de travailler à l'ISAE-SUPAERO. Je remercie les collègues de bureau et de département Dorianne, Luca, Bastien, Henrich et Frédérique. J'ai aussi rencontré des personnes formidables pleines de vie et de diversité. Franco, j'étais très ravie de partager ma passion pour les langues étrangères avec toi, notamment l'italien, merci pour tes sujets de discussion très divers et ton éternel désaccord (on t'accepte comme tu es) et tes fameuses Paellas. Erwan, tu as toujours su tout expliquer à travers des histoires fabuleuses, parfois même crédibles, merci pour le Kouign-amann cher Breton. Zoe et Narjes, merci pour votre militantisme environnemental et féministe, j'apprécie chaque échange aussi extrême qu'il soit, je vous aime. Luca, Paolo, Andrea, Mina et Francesco, mes italiens préférés à l'ISAE. Merci de m'avoir appris des expressions, de m'avoir corrigée et de m'avoir raconté tout et n'importe quoi sur votre pays et sur d'autres sujets. Clément, An-Toann, Eyal, Alexandre, François, Guillaume Valentin et Léa, merci pour les mots croisés, la pétanque (avec les autres), la pelote basque, les questions existentielles et les cool discussions. Merci aussi à Juan-Jose, Vatsal, Sophie, Sophia, Ilya et Ann pour leur bonne humeur et les divers échanges.

Je souhaite aussi remercier les professeurs que j'ai eu l'occasion de côtoyer durant mes trois ans de thèse à travers les divers évènements scientifiques, de laboratoire, et d'enseignements ...etc, Emmanuel Rachelson, Emmanuel Lochin, Emmanuel Chaput, Julien Fasson, Nathalie Thomas, Charly Poulliat, André-Luc Beylot, Stéphanie Bidon, Damien Roques et Riadh Dhaou.

Depuis mon arrivée à Toulouse, j'ai rencontré des personnes qui sont aujourd'hui de très bons amis et avec lesquels je passe toujours d'excellents moments. Les salseros: Ryan (Aguaaa), Kristina (la forever africaine d'esprit), Hajer (merci pour ton aide précieuse pour le pot de thèse), Adrien, Oyono et Cristina (you are so missed), Clément, Cédric, Florence et Camille merci pour les soirées danse, les soirées raclette et fondue et pratiquement tout ce qui est mangeable, je vous aime. Je n'oublie pas tous les autres: Svetlana, l'autre Kristina, Jonathan, et tous les africains qui respirent et te font respirer la danse (ils se reconnaitront). Les chicas du F207: Oana, Sonia, Elena et Simona, j'ai adoré les voyages avec vous, merci pour votre esprit aventurier et surtout pour les super cool et typiques mariages Roumains (je n'oublierai jamais l'ambiance de folie). Merci encore une fois à tous mes collègues de laboratoire pour les sorties, le basket et petits voyages de Ski, de stage et de découverte et ainsi à Laetitia avec ses très bonne idées et pour être toujours partante pour tout (ping Quentin). Merci aussi à tous les gens du Taekwondo et du rock pour les bons moments de détente, de souffrance (on s'entraine dur quand même), pour les soirées au Trac ...etc. Je voudrais remercier tous mes camarades du cours de kabyle à Toulouse: Saleha, Rabah, Fatima, Mohamed, Leila et Slimane, merci pour les bons moments linguistiques et culturels, merci pour votre amitié et de m'avoir transmis l'odeur de la kabylie. Je voudrais aussi remercier Henri (petit frère prifti sans filtre) et Davide il fratello pour tous les bons moments, pour leur joie de vivre et la bonne humeur, pour la découverte du moto GP (go Rossi) et pour les cool matchs de foot au stade (viva l'Italia). Bien sûr, je consacre un grand merci particulier à ma Enrica adorée, la personne la plus honnête et la plus pure que j'ai jamais rencontrée dans ma vie (oui oui je pense ça aussi de toi). Notre amitié s'est créée instantanément et naturellement et j'en apprécie chaque pan. Je suis très reconnaissante de t'avoir, merci pour tous nos voyages, pour nos discussions interminables, nos joyeuses rigolades, merci pour tout ce que tu m'as appris en Italien, tu es ma première prof, merci à toi, à Simone et, à toute ta famille pour les très bon moments. Je n'oublie pas mes deux bonnes copines Meriem (ma Marie) et Sihem, même si la distance nous sépare, notre amitié reste forte et chaleureuse. Merci aussi aux cool kids, Laura et Alessia, vous êtes des personnes fantastiques, merci pour votre amitié et joie de vivre. Shivank, merci à toi et à ta famille pour votre hospitalité, mon voyage en Inde était l'un des plus beaux.

J'aimerais remercier mes deux grandes familles maternelle et paternelle pour leur amour et encouragements, particulièrement à tonton Rabah, Tata Luisa et ma cousine Wissem, je vous aime. Je voudrais évidemment témoigner ma plus profonde gratitude aux personnes sans lesquels je n'aurais pas écrit ces mots, mes parents qui m'ont toujours encouragée, soutenue, et poussée à réaliser mes rêves et mes ambitions. Je n'aurais jamais assez de mots pour vous en remercier. Merci mama de m'avoir appris à assumer mes choix, merci pour ton amour et ta bienveillance, et merci pour la chaleur de la maison, pour ta bonne éducation, pour les principes que tu m'as inculqués et pour ton éternelle gentillesse. Papa, tu ne peux pas imaginer à quel point tu as participé à forger ma personne, j'ai toujours voulu te ressembler car j'admirais ta personnalité, tes idées, tes ambitions, ta soif pour la connaissance et ton coeur. Merci d'avoir fait de moi une féministe naturelle, merci de croire en le pouvoir de la femme, même dans un pays ou le combat est loin d'être fini. Merci d'avoir réveillé ma conscience politique et humaine ainsi que l'amour pour la littérature, merci pour ton éducation et d'avoir été mon meilleur ami depuis mon enfance. Merci d'écouter toutes mes histoires et tout ce que je raconte (Dieu sait combien je peux être bavarde) avec tant d'enthousiasme, merci pour tes encouragements et le fait de croire en moi. Tu es ma force. Tu m'as tant appris (tu m'as appris la vie et l'amour) et tu m'apprends toujours, je ne trouve point à rajouter que de te dédier tous les travaux de ma thèse. Les padres je vous aime. Merci aussi à mes deux frères Yanis (combien on a roulé ensemble!) et Nassim (mon éternel petit bibiche) pour votre amour, vos encouragements et votre esprit rigolo, chacun de vous apporte un charme à la famille, je vous aime. Grand merci à ma petite soeur adorée tellement "spéciale" pour tes joyeux encouragements en Coréen. Merci d'avoir cru en moi et merci pour ton esprit léger et chatouilleux et merci de gronder tout le monde à la maison !!! Je te souhaite un brillant avenir car moi aussi je crois fort en toi. Comme on dit: On se bagarre, on se chamaille, mais pour faire des bêtises on est toujours d'accord ! fâchés pour un jour... frère et soeur pour toujours ! je vous aime les gars.

Mes derniers remerciements vont à Einar min, my partner in crime and love, et qui est devenu aujourd'hui ma famille à Toulouse. Pour me réconforter, ta présence seule suffisait, même si tu faisais beaucoup plus. Tu es la personne qui a tout vécu, partagé, avec moi durant ces trois ans de thèse: les bons et mauvais moments. Grâce à ta joie et ton enthousiasme, j'ai pu mieux apprécier les bon moments; grâce à ton réconfort, au fait d'avoir toujours cru en moi et ton amour, j'ai pu surmonter plus facilement les moments difficiles. J'ai tant appris et découvert à tes côté à travers tes passions pour la science, le monde, la vie, la technologie, nos innombrables voyages, nos fou-rires, notre curiosité, nos diverses discussions et activités. Tu es sincèrement une personne formidable, pleine de vie et de trésors cachés. Nous sommes au coeur d'une belle aventure et j'ai hâte de savourer chaque moment à venir, elsker deg. Je n'oublie pas évidemment tous les membres de ta famille qui m'ont accueillie d'une façon très koselig et avec amour: Elin, Torunn, Karl-Petter, Marit, Halvor et Sigurd, j'ai hâte de vous rencontrer à nouveau.

A mon très cher papa

Contents

List of Figures

List of Tables

Résumé

Les nombreux avantages introduits par l'utilisation des satellites tels que la couverture à grande échelle, notamment dans les zones difficilement accessibles ou pauvres en infrastructure terrestres, a incité différentes communautés à développer des communications efficaces pour l'accès à Internet, la télévision et la téléphonie. Pendant longtemps, les techniques d'accès multiple basées sur la réservation de ressources (DAMA) ont largement été déployées sur la liaison retour, occupant ainsi une grande partie de la bande passante. Cependant, outre le temps aller-retour (RTT) additionnel dû à la demande d'allocation, qui est à la base important lors d'une communications par satellite, les ressources peuvent être sous-exploitées ou insuffisantes face à des applications entraînant un grand nombre d'utilisateurs telles que l'Internet des objets et les communications de machine à machine. Par conséquent, les techniques d'accès basées sur le protocole ALOHA ont largement pris place dans les études de recherche sur l'accès aléatoire (RA), et ont considérablement évolué ces derniers temps. La méthode CRDSA a particulièrement marqué ce domaine; elle a inspiré de nombreuses techniques d'accès aléatoire. Dans ce contexte, une méthode complémentaire, appelée MARSALA, permet de débloquer CRDSA lorsque celle-ci n'est plus en mesure de décoder de nouveaux paquets. Par contre, cela entraîne une complexité de corrélation liée à la localisation des paquets, qui est nécessaire pour combiner des répliques afin d'avoir une puissance de signal potentiellement plus élevée. C'est pourquoi, l'objectif principal de cette thèse est de proposer des alternatives efficientes et moins complexes. Nous nous intéressons plus précisément à la manière de gérer les transmissions multi-utilisateurs et de résoudre les interférences à la réception, avec la plus petite complexité. De plus, le phénomène de boucle qui se produit lorsque plusieurs utilisateurs transmettent leurs paquets dans les mêmes positions est traité, sachant qu'un plancher d'erreur au niveau des performances en taux de perte de paquets est par conséquent créé. Nous proposons donc des solutions synchrones et asynchrones, principalement basées sur un partage de données, au préalable, entre l'émetteur et le récepteur, dans le but de réduire la complexité de localisation, atténuer le phénomène de boucle et améliorer les performances du système. Ces techniques sont décrites et analysées en détails au cours de ce manuscrit.

Mots clés : Communications par satellite, accès aléatoire, localisation de paquets, canal multi-utilisateur, complexité du récepteur, protocol ALOHA, transmissions synchrones et

asynchrones, positions des répliques.

Abstract

The effective coverage of satellites and the technology behind have motivated many actors to develop efficient communications for Internet access, television and telephony. For a long time, reservation resources of Demand Assignment Multiple Access (DAMA) techniques have been largely deployed in the return link of satellite communications, occupying most of the frequency bandwidth. However, these resources cannot follow the technological growth with big users communities in applications like the Internet of Things and Machine to Machine communications. Especially because the Round Trip Time is significant in addition to a potential underuse of the resources. Thus, access protocols based on ALOHA took over a big part of the Random Access (RA) research area and have considerably evolved lately. CRDSA have particularly put its fingerprint in this domain, which inspired many different techniques. In this context, a complementary method, called MARSALA comes to unlock CRDSA when packets can no longer be retrieved. This actually involves a correlation complexity related to packet localization which is necessary for replicas combinations that results in a potentially higher signal power. Accordingly, the main goal of this PhD research is to seek for effective and less complex alternatives. More precisely, the core challenge focuses on the way to manage multi-user transmissions and solve interference at reception, with the smallest complexity. In addition, the loop phenomenon which occur when multiple users transmit their packets at the same positions is tackled as it creates an error floor at the packet loss ratio performance. Synchronous and asynchronous solutions are proposed in this thesis, mainly based on providing the transmitter and the receiver with a shared prior information that could help reduce the complexity, mitigate the loop phenomenon and enhance the system performance. An in-depth description and analysis of the proposed techniques are presented in this dissertation.

Keywords: Satellite communications, random access, packet localization, multiuser channel, receiver's complexity, ALOHA protocol, synchronous and asynchronous transmissions, replicas' positions.

List of acronyms

3GPP *3rd Generation Partnership Project.*

ACRDA *Asynchronous Contention Resolution Diversity ALOHA.*

AR-SPOTiT *Asynchronous Random SPOTiT.*

AVBDC *Absolute VBDC.*

AWGN *Additive White Gaussian Noise.*

BPSK *Binary Phase Shift Keying.*

BTU *Bandwidth-Time Unit.*

CDMA *Code Division Multiple Access.*

CFDAMA-PB *Combined Free DAMA- Piggy-Backing.*

CPODA *Contention-based Priority Oriented Demand Assignment.*

CRA *Constant Rate Assignment.*

CRDSA *Contention Resolution Diversity Slotted Aloha.*

CSA *Coded Clotted ALOHA.*

DAMA *Demand Assignment Multiple Access.*

DS *Direct Spreading.*

DSA *Diversity Slotted Aloha.*

DVB *Digital Video Broadcasting.*

DVB-S *DVB-Satellite.*

DVB-S2 *DVB-S/Second Generation.*

DVB-S2X *Extension of the DVB-S2.*

DVB-SH *DVB-Satellite services to Handheld.*

DVB-RCS *DVB-Return Channel Link via Satellite.*

DVB-RCS2 *DVB-RCS/Second Generation.*

ECRA *Enhanced Contention Resolution Aloha.*

EGC *Equal Gain Combining.*

EM *Expectation Maximization Algorithm.*

E-SSA *Enhanced Spread Spectrum ALOHA.*

FCA *Free Capacity Assignment.*

FDMA *Frequency Division Multiple Access.*

FEC *Forward Error Correction.*

FH *Frequency Hopping.*

HARQ *Hybrid Automatic Repeat request.*

HID *Hardware IDentifier.*

HTTP *HyperText Transfer Protocol.*

IC *Interference Cancellation.*

ID *Identifier/ Identification information.*

IoT *Internet of Things.*

IP *Internet Protocol.*

IRSA *Irregular Repetition Slotted ALOHA.*

ISM *Industrial, Scientific and Medical (frequency bandwidth).*

LAN *Local Area Network.*

LDPC *Low-Density Parity-Check.*

LMS *Land Mobile Satellite (channel).*

LTE *Long Term Evolution.*

MA	*Multiple Access.*
MAC	*Media Access Control.*
MARSALA	*Multi-replicA decoding using corRelation baSed LocALisAtion.*
MF	*Multi Frequency.*
MI	*Mutual Information.*
MIMO	*Multiple Input Multiple Output.*
MISO	*Multiple Input Single Output.*
MODCOD	*Modulation and Coding.*
MRC	*Maximum Ratio Combining.*
m-sequences	*maximum length pseudo noise sequences.*
MUD	*Multi-User Detection.*
MuSA	*Multi-Slots Coded ALOHA.*
NOMA	*Non-Orthogonal Multiple Access.*
NC	*Network Coding.*
OFDMA	*Orthogonal Frequency Multiple Access.*
OMA	*Orthogonal Multiple Access.*
PER	*Packet Error Rate.*
PLR	*Packet Loss Ratio.*
PLNC	*Physical-Layer Network Coding.*
PRDAMA	*PRedective DAMA.*
PRNG	*PseudoRandom Number Generator.*
QPSK	*Quadrature Phase Shift Keying.*
RA	*Random Access.*

RBDC *Rate Based Dynamic Capacity.*

R-CRDSA *Reservation-CRDSA.*

R-SPOTiT *Random-SPOTiT.*

RTS *Reference Time Slot.*

RTT *Round Trip Time.*

SA *Slotted ALOHA.*

SC *Selective Combining.*

SDMA *Space Division Multiple Access.*

SIC *Successive IC.*

SNIR *Signal to Noise plus Interference Ratio.*

SNR *Signal to Noise Ratio.*

SPOTiT *Shared Position Technique for Interfered Random Transmissions.*

SS *Spread Spectrum.*

ST *Satellite Terminal.*

SW *Sliding Window.*

TCP *Transport Control Protocol.*

TDMA *Time Division Multiple Access.*

TTC *Tracking, Telemetry and Command.*

UNOOSA *United Nations Office for Outer Space Affairs.*

VBDC *Volume Based Dynamic Capacity.*

VCM *Variable Coding and Modulation.*

VF *Virtual Frame.*

VoIP *Voice over IP.*

VSAT *Very Small Aperture Terminals.*

List of notations

Common System parameters

M: The modulation order.

N_{b}: Number of bits of the packet's payload.

N_{R}: Number of replicas per packet.

N_{S}: Number of slots per frame/virtual frame.

N_{sym} Number of symbols of the packet's payload.

N_{P}: Number of preambles.

N_{U}: Total number of users.

r: The replica's index.

R: The coding rate.

u: The user's index.

Chapter 3

c: Index of one of the collided packets on the reference time slot.

F_{ID}: The frame ID.

k: It is equal to 0 if no packets have been decoded yet on the reference time slot, and it is equal to 1 if at least one packet have been decoded.

$N_{\mathrm{Coll}}^{\mathrm{Ref}}$: The total number of collided packets over the reference time slot before MARSALA's decoding.

$N_{\mathrm{Det}}^{\mathrm{P}}$: The number of detected preambles over the reference time slot.

$N_{\mathrm{MARSALA}}^{\mathrm{Corr}}$: The total number of data correlations, for MARSALA, over the reference time slot.

$N^{\text{Corr}(1),\text{c}}_{\text{MARSALA}}$: The number of data correlations which are necessary to decode only one of the collided packets c over a time slot for MARSALA.

$N^{\text{Corr}(1)}_{\text{MARSALA2}}$: The number of data correlations which are necessary to decode only one of the collided packets c over a time slot for MARSALA with two replicas per packet.

$N^{\text{Ref}}_{\text{PColl}}$: The number of potential packets in collision on the reference time slot.

$N^{\text{Ref}}_{\text{pot}}(p)$: The number of potential users with a detected preamble p that can transmit on the reference time slot.

$N^{\text{Corr}}_{\text{SPOTiT}}$: The total number of data correlations, for R-SPOTiT, over the reference time slot.

$N^{\text{Corr}(1),p,c}_{\text{SPOTiT}}$: The number of data localization correlations for one packet decoding (c), related to a specific detected preamble p, for R-SPOTiT.

$N^{\text{Corr}(1),p,c}_{\text{SPOTiT},2}$: The number of data localization correlations for one packet decoding (c), related to a specific detected preamble p, for R-SPOTiT with two replicas per packet.

P_u: The preamble used by user u.

$\text{Slot}(u, r)$: The time slot position of the replica r belonging to user u.

U_{ID}: The terminal HID.

Chapter 4

α: The number of times that a preamble of a given packet is detected on the reference time slot, and that its second replica's position exhibits a correlation peak for the same preamble (case A).

β: The number of times that only one of the two preambles is detected (case B). Thus $\rho = 0$.

C_{b}: The basic data correlation.

C_{bp}: The basic preamble correlation.

C_{D}: The total number of data localization correlations per frame.

C_{P}: The total number of preamble detection correlations per frame.

C_T: The total number of correlations per frame including the preamble detection operations and the data localization operations.

δ: The frame analysis index.

Δ: The maximum value of the frame analysis index δ that is reached when the whole system is blocked.

D_{crdsa}: The average number of decoded packets by CRDSA when it is used alone.

D_{CT}: The average number of decoded packets by the complementary treatment R-SPOTiT.

D_{gain}: The percentage of the decoding gain resulting from the complementary treatment compared to CRDSA.

F_{ID}: The frame ID.

g: The interpolation of the 1/3 turbo coded PER with QPSK modulation in an AWGN channel environment, and a packet length of 150 symbols.

γ: The number of times that no preambles are detected on the whole frame (case C). Thus $\rho = 1$.

G_{data}: The extra guard data symbols from the region around the preamble location.

i: The preamble index.

it: The SIC iteration index in CRDSA.

$i_{N_{crdsa}}(\delta, n-1)$: The time slot index of the last replica suppressed by CRDSA at the previous iteration $n-1$, with a frame analysis index δ.

I_u: The summation of the number of interfering packets over both replicas of a user u.

$I_{u,r}$: The number of interfering packets on the analyzed slot with replica r and user u.

Ł: The number od collided preambles on a given slot.

λ: The index for randomly choosing a reference time slot by the complementary treatment.

n: Any iteration index of CRDSA / also the AWGN noise term.

N_{bt}: The total number of the coarse and fine tracking operations.

$N_{\mathrm{crdsa}}(\delta, n-1)$: The time slots where successful decoding took place at the previous CRDSA iteration $n-1$ at any frame analysis index δ (it could more than one packet).

N_{it}: The number of SIC iterations in CRDSA.

N_{P}: Number of used preambles.

$N^{\mathrm{u}}_{\mathrm{pm}}(\delta, \lambda)$: The number of potentially collided packets using the same detected preamble over the reference time slot, derived from the updated information table.

$N^{\mathrm{u}}_{\mathrm{po}}(\delta, \lambda)$: The number of potential packets in collision on the chosen reference time slot with index λ after the last update of the information table for a given frame analysis index δ.

$N_{\mathrm{pp}}(s)$: The number of potential preambles that could be transmitted on a given slot s among the N_{P} possible preambles.

$N^{\mathrm{u}}_{\mathrm{pp}}(s)$: The updated $N_{\mathrm{pp}}(s)$ after decoding occurred on slot s. Decoded packets are suppressed from the potential packet candidates.

$N_{\mathrm{scen}}(\delta, n)$: The number of slots where to perform correlations.

N_{PT}: The total number of pseudo-orthogonal Gold preambles with a given length.

P_{D}: The decoding probability.

P_{FA}: The false alarm probability.

$\phi_{u,r}$: The phase error of replica r of a user u on its given time slot.

p_i: The i^{th} Gold code signal.

Ψ: The total number of times that R-SPOTiT is used per frame.

P_{T}: The preamble region signal.

R: The ratio between the data and the preamble lengths.

ρ: It is equal to 0 when at least one preamble is detected on the analyzed slots, and it is equal to 1 when there are no detected preambles at any reference time slot.

R_{P}^{l}: The inter-correlation between the preamble region signal and the complex conjugate of the l^{th} preamble.

s: The time slot index.

σ^2: The power of the AWGN noise term plus the power of the interference over a given time slot.

τ_u: The time shift of a user u on the analyzed frame.

T_{h}: The preamble detection threshold.

T_{S} The symbol duration.

T_{search}: The search region over the preamble duration.

Chapter 5

$B_{\mathrm{inf}}(i, s)$: the lower bound of the slot set whose index is s, at a given level i of the S-SPOTiT distribution.

$B_{\mathrm{sup}}(i, s)$: The upper bound of the slot set whose index is s, at a given level i of the S-SPOTiT distribution.

$E_{i,s}$: A given slot set with index s at a given level i of the distribution of S-SPOTiT.

i: The index of a level in the S-SPOTiT distribution.

j: The index of the preamble, the preamble group, and of the value of the cyclic shift.

$M_{i,s}$: The central value of the slot set whose index is s, at a given level i of the S-SPOTiT distribution.

n: The index of a given element of $P\mathrm{g}_1$ and $P\mathrm{g}_2$.

$N_{\mathrm{E},i}$: The number of the slot sets at a given level i of the S-SPOTiT distribution.

N_{L}: The number of levels in the regular S-SPOTiT distribution.

N_{ss}^i: The number of slots in any of the slot sets at a given level i of S-SPOTiT distribution.

$P\text{g}_1$: The set of the first time slot positions of the first replicas, for each transmitter in a given group of users.

$P\text{g}_2$: The set of the second time slot positions of the second replicas, for each transmitter in a given group of users.

s: The index of a given slot, at a given level, in S-SPOTiT distribution.

Chapter 6

i: The index of a level in the irregular S-SPOTiT distribution.

k: an integer in $\mathbb{Z}^+$.

N_{G}: The number of user groups at each level i of the irregular Smart-SPOTiT distribution.

N_{L}: The number of levels in the irregular S-SPOTiT distribution.

N_{U}: The total number of users in the lower power of two regular Smart SPOTiT.

$\overline{N_{\text{U}}}(i)$: The number of users at each level i of irregular S-SPOTiT distribution, based on the regular distribution.

$\overline{N_{\text{U}}}$: The total number of users in the irregular S-SPOTiT, based on the regular one.

$\overline{N_{\widetilde{\text{U}}}}$: The total number of loop-free position couples in irregular Smart-SPOTiT or with the extension of S-SPOTiT with a given number of slots per frame.

s: The slot index of the extension of Smart-SPOTiT.

Υ: The number of unconsidered position couples in irregular S-SPOTiT when the regular method is used alone.

ζ: The difference between $\overline{N_{\text{U}}}$ and N_{U}, in terms of the user positions.

Chapter 7

a: The index of the ascending order of the replicas' positions of a same packet according to the beginning of the VF.

$C_{ARS}(u)$: The summation of the mean number of interfering packets of both replicas of a given packet u. We call it: the interference rate.

$C_r(u)$: The normalized interference rate $C_{ARS}(u)$ (over one replica).

$C_{sym}^{u,r}(s)$: The number of interfering packets at the symbol s of replica r belonging to the user u.

$C_{u,r}$: The mean number of interfering packets between the symbols of a replica r belonging to a user u.

$d_{u,j}$: The j^{th} distance between two replicas of a packet belonging to user u.

g: The interpolation of the 1/3 turbo coded PER with QPSK modulation in an AWGN channel environment, and a packet length of 150 symbols.

j: The index of the distances between replicas of a same packet.

N_d: The number of distances for N_R replicas belonging the same packet (same VF).

ν_p: The total number of preamble correlations for one packet localization.

N_s^p: The number of virtual time slots at distances derived from the AR-SPOTiT information table where to perform preamble search of the detected code.

N_{sym}: The number of symbols per packet (payload).

PER: Packet Error Rate with 1/3 turbo coding with QPSK modulation in an AWGN channel environment, and a packet length of 150 symbols.

R_a: The a^{th} replica after the ascending order of all replicas positions of a given packet on the VF.

$SNIR_{MIS}(u)$: The equivalent SNIR value experienced over both replicas' positions of a given packet u.

$SNIR_{MIS}(u, r)$: The SNIR value of a replica r, belong to a user u, computed according to $C_{u,r}$.

Chapter 1

Introduction

Contents

1.1 Context & Motivation

Satellite communications have largely been subject to study since the launch of the first artificial satellite (Sputnik), back in 1957. There have been many software and hardware developments and improvements regarding each of the space, ground, and control segments. Especially because a considerable potential is perceived by the communication area community. Indeed, the global coverage and the large capacities that satellites can offer motivated researchers to propose low cost, ubiquitous services that meet the constantly arising demands of a good quality worldwide connectivity. According to the United Nations Office for Outer Space Affairs (UNOOSA) [OS19], there are more than 5000 satellites in orbit around earth, around other planets (Mars, Venus), around natural satellites and around asteroids (Ryugu) as well. Indeed, they are used for various space missions in addition to the different earth applications. Specifically, in the latter, satellites can solve the white spot problem where there are no infrastructure for cellular coverage because of a wild environment or insufficient resources. In some developing countries and underserved areas, even a satellite solution appears to be expensive. In Senegal for example, a communication network within white spots, based on an ISM (Industrial, Science and Medical) long range radio communication have been developed for breeders [DIA17]. Nevertheless, constant efforts and novelties are put on the ground for better and cheaper satellite services. According to The Economist [SM16], flocks of cheap little satellites could transform the space business. Five Kilos-weight and 30 cm-long satellites are built up, by Planet [1], from smartphones and other devices' components within a week. This means that small numerous objects do not only apply for Internet of Things (IoT) terrestrial connected objects, but also to satellites. This can solve the connectivity problems in developing countries.

Satellite applications where numerous devices are involved can be time critical, considering the Round Trip Time (RTT), in addition to being resource consuming. Access strategies have then been set in different standards in order to efficiently organize communications. Communicating entities should then obey to the different protocols found in the various derivatives of the Digital Video Broadcasting standards [42197]; [V1.15a]; [V1.15b]; [V1.11]; [79003]; [A1511]. Demand Assignment Multiple Access (DAMA), unlike Random Access (RA) requires resource allocation requests, which adds signaling overheard in addition to the communication delay regarding the useful information transmission. Therefore, RA is more suitable for sporadic and short packet transmissions. However, numerous challenges are encountered and need to be dealt with.

We focus in this thesis on the multiple access techniques of a satellite communication's return link (between a user terminal and a gateway) where multiple terminals transmit packets over the same frequency bandwidth. Among the various RA and the dedicated access contributions, emerged some of the most efficient techniques, offering good system performance. ALOHA based RA methods are specifically targeted. We hence put our interest in recent solutions, mostly based on multi-replica transmission with successive interference cancellation (SIC) at reception in synchronous and asynchronous environments. On the one hand, Contention Resolution Diversity Slotted Aloha (CRDSA) considerably enhances the throughput due to the use of packet redundancy and SIC over well defined frames. It allows to efficiently manage packet collisions up to a certain channel load (in terms of the number of transmitters). Thus, the throughput collapses in high loads when no more packets can be retrieved (deadlock), or when only few are decoded. To cope with this problem, Multi-Replica Decoding using Correlation based Localisation (MARSALA) proposed to intervene as a complementary process to CRDSA when the latter is in a deadlock. First, it localizes replicas of packets in collision on a randomly chosen reference time slot using correlations. Then, replicas belonging to the same packet are combined. An association step meant to

[1] A private American company for a constant earth imaging, whose mission is to make global change visible, accessible, and actionable.

gather all packet's replicas along with channel parameters estimation are necessary to maximize the combination gain. More precisely, this allows to have a higher power of the packet of interest with a better probability of decoding. Nevertheless, the localization procedure of MARSALA that performs whole packet correlation operations adds a significant complexity to the receiver. As a matter of fact, global correlations are performed first between an arbitrary reference time slot and the rest of the slots on the frame in order to localize all replicas of collided packets. Then, additional correlations are needed to associate replicas of a same packet together. These are performed between a combined signal and the remaining correlation peak positions resulted from the previous global localization. Despite the complexity of MARSALA, significantly better performance are resulted, in terms of Packet Loss Ration (PLR) and throughput, compared to CRDSA.

1.2 Summary of contributions

Based on MARSALA's results, we sought to reduce the packet localization complexity regarding correlations when CRDSA can no longer retrieve packets. Until now, the receiver is not aware of the chosen positions of users. With the new method called Shared POsition Technique for Interfered random Transmissions (SPOTiT) we propose, a shared knowledge between the receiver and each of the terminals is introduced. The shared information is about the time slot locations on which each terminal transmits its replicas as well as the preamble to use (among a set of pseudo-orthogonal codes). The first version of SPOTiT dubbed R-SPOTiT (for Random SPOTiT) aims mainly to reduce the complexity of replicas localization process of the legacy technique MARSALA. It presents a less complex system without degrading performance and with no additional signaling information. It uses the generated common information between a transmitter and the receiver regarding replicas' potential positions and preambles, to target a lower number of slots for the localization correlations. Moreover, a detailed analysis of the number of correlation operations needed to locate replicas of collided packets is provided in this work. Scenarios with a single and multiple preambles, taking into account the preamble detection at CRDSA, are considered for a whole system complexity assessment. An optimal scheme for R-SPOTiT is deduced according to simulation results of the different scenarios that have been assessed.

Another contribution of this thesis proposes Smart SPOTiT (noted S-SPOTiT) as a hybrid solution that mixes both DAMA and RA in order to decrease the PLR floor. In fact, a centralized management of replicas' positions and preambles to use is made in a way that no loops are created. The loop phenomenon occurs when two or more packets are transmitted at the exact same positions, which creates an error floor at the PLR easily observed with MARSALA and R-SPOTiT. This version of SPOTiT requires a signaling information that is sent only once to the transmitters according to an optimal distribution. The latter includes a disposition of packets' locations on a frame without loops, associating them with preambles, and allowing a simple localization at the same time. Indeed, this distribution makes sure that one of the packet's replicas is the only one that could be transmitted in its time slot position with a given preamble; i.e. whenever a preamble is detected on a given position where a unique user could have used it, its packet is localized without correlations. S-SPOTiT showed promising results, especially with the disappearance of the error floor of the PLR. It is worth noting that the optimal distribution of S-SPOTiT relies on system parameters, such as the number of slots and the number of preambles, that are based on power of two values. It seemed then to be important to derive an irregular framework with any parameters in order to have a complete scheme of S-SPOTiT. Also, a dynamic loop-free scheme of S-SPOTiT, which offer a flexibility regarding the number of users is proposed. We recall that the core problem that S-SPOTiT tackled is the loop phenomenon. However, the latter is less significant in asynchronous transmission. This is why R-SPOTiT is considered in the asynchronous case.

Asynchronous RA solutions are characterized with no signaling overhead regarding synchronization information. As CRDSA emerged as a leading technique in synchronous transmissions, the definition of an asynchronous version of it was crucial. Asynchronous Contention Resolution Diversity ALOHA (ACRDA) represents the closest asynchronous method to CRDSA. CRDSA and ACRDA incur a deadlock when no more packets can be retrieved due to high channel loads. In synchronous transmissions, we recall that MARSALA allows to unlock some of the deadlock configurations which would relaunch CRDSA again. In asynchronous transmissions, Enhanced Contention Resolution Aloha (ECRA) uses different combining techniques for packets' replicas to offer high system performance in terms of PLR and throughput. These techniques MARSALA and ECRA can be costly in terms of localization complexity to the receiver. This is the reason R-SPOTiT was defined. Accordingly, we propose in this thesis AR-SPOTiT, an asynchronous design of R-SPOTiT, as a complementary method to ACRDA. It introduces a way to locate replicas on their virtual frames with less complexity and significantly higher system performance, in addition to the mitigation of the loop phenomenon.

1.3 Thesis Structure & Plan

The thesis will be structured in way that allows the reader to progressively understand different aspects of the application field we worked on. More specifically, the ALOHA based RA protocols which are used on the return link of a satellite communication system are addressed, along with the contributions we proposed to enhance some of them. Thus, the general organization of this dissertation is as follows:

- An introductory chapter will present a bibliographic background on traditional and recent access protocols used in wireless communications. As expected, a special focus is put on satellite communications' return link. This state of the art summary review turns out to be necessary to better understand the proposed solutions. Indeed, the latter rely on the recent advanced RA protocols, to perform as a complementary process.

- In the following chapter, we introduce the novel synchronous RA technique, R-SPOTiT, that aims mainly to reduce the localization complexity of MARSALA scheme without degrading performance. An initial comparative complexity analysis is presented along with system performance results.

- An extended version of R-SPOTiT is detailed in Chapter 4 for an overall complexity evaluation that considers preamble detection operations along with full packet localization. We present three system scenarios with different parameters and their related complexity in order to provide a fair comparison between R-SPOTiT and MARSALA.

- Chapter 5 presents the general principle of synchronous S-SPOTiT, an alternative solution to R-SPOTiT, which aims mainly to eliminate the loop phenomenon. It actually offers better system performance at the expense of an additive signaling information. In addition, an optimal power of two distribution scheme regarding system parameters is proposed.

- In Chapter 6, an irregular framework of S-SPOTiT using the power of two regular method aims to present a complete scheme with any set of system parameters. In addition, we propose an extension of S-SPOTiT towards a dynamic loop-free system that adds a flexibility in terms of the number of users.

- Unlike all of the previous synchronous technical chapters, Chapter 7 addresses the asynchronous environment by adapting R-SPOTiT. The operating mode of the proposed AR-SPOTiT is provided at transmission and reception along with a complexity case study. The latter concerns a small comparison between AR-SPOTiT and ECRA in terms of preamble detection.

- Finally, a general conclusion and discussions are provided alongside with open perspectives for future work in Chapter 8.

Chapter 2

Literature review

Contents

Including satellites in wireless communication systems provided the market with new applications and contributed to major technological advances. Quickly, the development of multiuser systems became necessary to respond to the exponentially arising demands of the civil, scientific and military communities. A general overview of wireless communication systems, including satellites and multiple access techniques, is presented in this chapter. It is meant to draw the path that led to use Random Access (RA) over satellites. The goal is to introduce recent RA techniques used in satellite communications which serve as a basis for better understanding the upcoming chapters. Thus, the main synchronous and asynchronous solutions on which we focused in this work, to propose enhancement schemes or for comparison matter, are described; in addition to some other leading techniques.

2.1 Wireless Communications

Recently, wireless communications have considerably evolved and expanded towards multiple sectors and industries. It is actually considered to be developing faster than any other engineering field since the first transatlantic radio transmission achieved by Guglielmo Marconi

[Rap96]. Indeed, wireless services and devices with relatively low cost are taking a big part in our everyday lives: from long range mobile communications and IoT like LoRa and SigFox to short range Blutooth, infrared, Zigbee and Wi-Fi [Mul02]. Many wireless Local Area Network (LAN) applications exploit the Industrial, Scientific and Medical (ISM) frequency band for multiple public and private services. These include voice and text, file transfer, Internet access, web browsing, video teleconference, supervision through sensors, entertainment... etc. Two modes of communication can be distinguished: a one way broadcasting performed by a single transmitter to numerous receivers, or a bidirectional point-to-point communication between a transmitter and a receiver. When multiple users of a given application need to send information to the same receiver or use the same radio channel, Media Access Control (MAC) techniques are needed to allow an efficient sharing of the channel. Multiple Access (MA) solutions are thus described in this section.

2.1.1 Multiple Access

In multiuser systems, sharing a radio spectrum is necessary to allow multiple users to have access to the same channel, commonly dubbed a multiuser channel. Thus, radio resources that define the use of the available carrier frequency bandwidth must be allocated to the different users in a way that keeps good system performance and capacity. There are two main categories of Multiple Access: Orthogonal MA and Non-Orthogonal MA that are described below.

Orthogonal MA

The orthogonal conventional MA (OMA) schemes have been largely used in cellular wireless mobile communication systems (1G, 2G, 3G, 4G) and satellites. OMA constructs separable resources according to time, frequency, code, space domain or a combination of these. The goal is to allocate each of the resources to a single user without creating interference with the others [TV05]. The purpose is to avoid any collision and mitigate retransmissions.

1. **Frequency Division Multiple Access (FDMA):** in this system, the frequency bandwidth is divided into disjoint subbands (channels), each of which is individually allocated to a single user, that are separated with guard bands to avoid overlapping resources.

A resource is defined here to be a subband. Continuous communications over time is possible as each subband is constantly occupied by the dedicated user. FDMA is more suitable for narrowband systems because the subband channels can be quite tight. In this absolute initial version of FDMA, the bandwidth is under-utilized when a user is in an idle state. Furthermore, including a new transmitter leads to modify the system equipments such as filters [RS90].

2. **Time Division Multiple Access (TDMA):** in this system the entire bandwidth is occupied by each user at different times (called time slots) allocated to each one of them. Such systems are characterized with a TDMA frame transmission which gathers a fixed number of slots. The time occupancy on a frame's time slot by a given user is repeated on a cyclic basis. This may add delay between successive transmissions, in addition to a potentially different channel environment, but with low battery consumption. The former requires a new state estimation of the channel [Gol05]. Similarly to FDMA, short yet sufficient guard time intervals are required to avoid interference between adjacent time slot signals.

3. **Code Division Multiple Access (CDMA):** it can be described as a hybrid combination of FDMA and TDMA since the entire carrier frequency bandwidth can be occupied simultaneously by all active users of the system. The separation between the different user signals is achieved through the selection of a unique spreading code by each one of them, through Direct Spreading (DS) or with Frequency Hopping (FH). The receiver, having a match filter for each used code, is able to retrieve the information transmitted by each user independently from the others. Despite the absence of synchronization between transmitters for resource utilization which is due to the code separation and the good interference mitigation, CDMA has a relatively low throughput performance [Kol02].

4. **Space Division Multiple Access (SDMA):** another type of radio resource separation is defined with spatial diversity. Indeed, having access to the channel at the same time and/or over the same carrier frequency is made possible thanks to directional multibeam-like antennas. Each of the users has a specific direction pointed to a given beam of the receiver [Hay13]. This requires good coordination and adaptation if a direction angle changes for a user.

5. **Orthogonal Frequency Multiple Access (OFDMA):** it is considered as a special

case of FDMA, where the subcarriers are orthogonal to each other. This makes the guard bands unnecessary, which increases the channel efficiency. The subcarriers assigned to the different users can be overlapping with each other without interfering with one another under orthogonality conditions. The latter is assured by having a consecutive subcarrier spacing equal to the inverse of one symbol duration. Indeed, the dot product between each of the subcarriers and another during one symbol transmission is zero.

Hybrid solutions that combine some of the previously described OMA schemes also exist. For instance, OFDMA can benefit from more diversity in terms of frequency if combined with tone hopping [Gol05]. These orthogonal multiple access techniques are mainly characterized with interference-free separable resources. In general, Multi User Detection (MUD) for OMA at reception is supposed to present poor complexity. However, orthogonality can be destroyed in many cases, especially because of channel impairments. Also, the emergence of new communication technologies that are likely to take a serious part in the upcoming fifth generation (5G) like Machine type communications and IoT shall meet numerous obstacles with the traditional OMA techniques for many reasons. Indeed, the massive number of devices that are expected in 5G for the different applications and services shall not be able to efficiently share the radio spectrum according to the legacy OMA schemes because of the insufficient resources. As a matter of fact, OMA techniques rely on a fixed number of resources according to which domain they have been created. Therefore, Non-Orthogonal Multiple Access (NOMA) is witnessing a great interest from the industrial and academic communities.

Non orthogonal MA

In Non-Orthogonal Multiple Access scheme [Sai+13], the radio resources, are no longer assigned individually to a single user each. Indeed, the overloading concept is adopted here to refer to having multiple users occupying the same resource. Thus, unlike all the deployed cellular communication systems, from the first-generation (1G) to the fourth-generation (4G) where OMA is used, the fifth-generation (5G) intends to implement NOMA techniques in order to support the massive communicating devices with a high throughput. These are generally low energy consumption, low latency devices with sporadic transmissions. However, it can be discussed that CDMA is considered as OMA only if there is no overloading, which means that there are more codes than users. Two major categories of NOMA can be distinguished: Power-domain NOMA and code-domain NOMA. Other domains also exist but are out of scope in this section.

- Power-domain NOMA: The main operations that are performed here are user multiplexing through superimposition coding at transmission with potential power allocation and Successive Interference Cancellation (SIC) at reception [Dai+18][Ben+13][Sai+13][NKH14]. SIC operations allow to remove the interference contribution to certain signals after retrieving a given decodable signal.
- code-domain NOMA: The full radio resource is shared between several users, having each a specific code [Dai+18][OKH12][Al-+11][Lu+15], which means that multiplexing is performed in the code domain. These are generally spreading codes with good properties such as a weak inter-correlation, sparsity and low density [vP09]. Thus unlike the underloaded CDMA, the spreading sequences are not orthogonal. In fact, the Low Density Spreading (LDS) techniques are considered as code-domain NOMA schemes, such as LDS-CDMA [HWT08], LDS-OFDM [Raz+12], Sparse Code Multiple Access (SCMA) [Zha+17], Pattern Division Multiple Access[Che+17], and Multi-User Shared Access (MUSA) [Yua+16].

These multiple access NOMA techniques that do not require any orthogonality between users can benefit from a grant-free application which reduces the transmission latency and signaling overhead. Indeed, no resource allocation requests are necessary. However, NOMA in general requires powerful MUD techniques at reception to cope with the overloading superimposition. This additive complexity at reception offers in return a higher spectral efficiency and throughput in addition to supporting a larger community of connected devices [Dai+18].

ALOHA based Random Access

Random Access (RA) can be considered as a grant-free NOMA technique in its concept as many users can transmit in the same resource, being aware of the risk of collision. The pioneer RA protocol is ALOHA [Abr70], which has been developed in the university of Hawaii in order to interconnect its islands. It is based on two main steps:

1. A packet is transmitted by a terminal whenever it is needed, without any signaling or resource allocation requests.

2. If the packet is correctly received by the base station, an acknowledgment is sent to the transmitter for confirmation. However, if the packet incurs a collision with time-overlapping transmitted packets from other users, then the transmitter should send again its packet after an arbitrary time out.

The main drawback of pure ALOHA protocol is that any total or partial interference destroys the packet of interest, which would require several retransmissions before it is correctly received. Therefore Slotted ALOHA (SA) [Abr77] proposed to organize transmissions in well defined time slots within a frame. This means that there are no partial interference, only simultaneously transmitted packets (same time slot) are lost. A synchronization at the network level is required between users in this case but it offers a lower Packet Loss Ratio (PLR) and a higher throughput. It has been shown in [Gol05] that SA systems can be further enhanced using error correction to avoid packet retransmissions to a certain level. More sophisticated ALOHA based RA techniques meant for satellite communications are described later in this chapter.

2.1.2 Satellite communications

The worldwide coverage that satellites offer in addition to the exponentially growing technology behind, have motivated many actors to develop efficient communications for Internet access, television and telephony. The integration of satellites in the different nowadays applications and innovations becomes then crucial. It is hence undoubtedly an important part of the wireless communication society today.

The conventional setup of any satellite communication system includes three main parts. These are the space segment, the control segment and the ground segment (see Figure 2.1) [MB02]. The latter consists of all types of terrestrial stations whose dimension can go from few centimeters to some meters, depending on the type of service they provide. There can be interface stations between the space segment and the terrestrial network (gateways) that generally provide services to user terminals. There are also Handsets and very small aperture terminals VSATs which communicate directly with the space segment. They interact with service stations such as hubs that relay services and information from the service provider. Herein, a single satellite or a constellation of satellites constitute the space segment, managed by the control segment. Indeed the control segment insures the stability of the satellite in orbit and is connected to the payload of the satellite through a TTC (Tracking, Telemetry and Command) link. The payload of a satellite makes him a bent pipe if it is transparent or regenerative if it includes complex operations. A bent pipe takes only in charge the amplification, the frequency transposition and the bandpass filtering of any received signals. On the other hand, in a regenerative satellite, operations like predistortion, demodulation/modulation, decoding/encoding are also performed.

Another important aspect to highlight is the communication link type. Such links are characterized with a source and a destination in addition to the intermediate nodes between them. This means that all operations that are involved in this link should be defined. Typically, two main links are outlined. On the one hand, the return link goes from a user terminal to a gateway or a service station regardless of the number of hopes and intermediate nodes (satellites for example). This means that this full connection includes an uplink, potential intersatellite links and a downlink. On the other hand, the forward link is the reverse configuration. We focus in this manuscript on the return link.

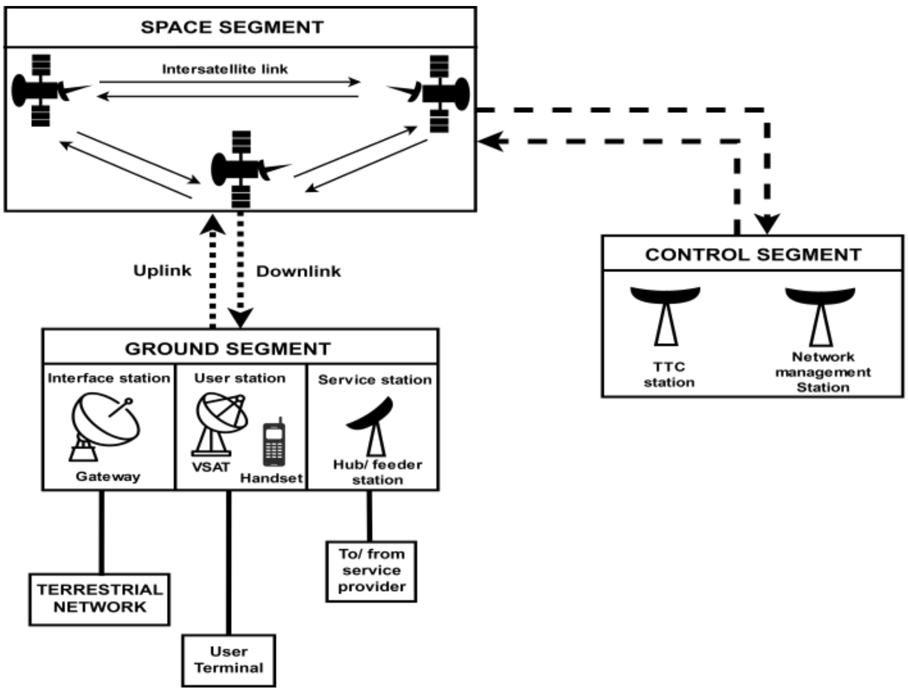


Figure 2.1: Overview of a satellite communication system interfacing with terrestrial entities.

The main satellite communications' standards, defined for the forward and return links, are presented below.

Initially, a consortium called Digital Video Broadcasting (DVB) was created to offer a worldwide digital television service through an international open standard. Satellite television was standardized in the DVB-S (DVB-Satellite) [42197], which has been later improved in the DVB-S2 (Second Generation) [V1.15a] through coding (low-density parity-check LDPC codes) and enhanced modulation schemes, VCM (Variable Coding and Modulation) and ACM (Adaptive Coding and Modulation). DVB-S2X (Extension) [V1.15b] extended the DVB-S2 towards more flexibility in the modulation schemes, better filtering ...etc. Also DVB-SH (Satellite services to Handhelds) [V1.11] introduced advanced service delivery based on Internet Protocol (IP) for small size terminals like mobile phones. Most of these standards operate on the forward link for broadcasting services that reach the user terminals. However, advanced applications that require bi-directional systems made it necessary to define the return link. DVB-RCS [79003] that stands for Return Channel Link via Satellite, and later its corresponding Second Generation (DVB-RCS2) [A1511] are hence the leading standards for a variety of interactive applications. IP based services are covered by the DVB-RCS2. Information are carried in MF-TDMA frames organized in time slots. Many waveform characteristics of reference are defined. They list the modulation, its order and the coding rate to use, the length of the useful information in bytes and symbols, and the total burst length in symbols in addition to its type as well. For example, the third waveform, in the DVB-RCS2, whose IDentifier (ID) is 3 uses QPSK (Quadrature Phase Shift Keying) modulation with a code rate of 1/3. The useful information length is 456 symbols and the total burst length is 664 symbols whose type is TRF1. The latter means that the time slot in the MF-TDMA frame occupies two BTU (Bandwidth-Time Unit).

Three main communicating parts can be distinguished in a return link of a satellite system, among which two can be combined: a gateway that relays information to a satellite, which in turn should deliver it to Satellite Terminals STs (user stations). If the satellite is regenerative, the gateway functionality is performed by the satellite's payload (on board) and thus only the satellite itself and the terminals are considered. It is worth noting that we focus, along this dissertation, on the return link of a satellite communication system, specifically on the way to access the channel at transmission, and retrieve the useful information at reception.

2.2 Access schemes in satellite systems

First, the conventional Demand Assignment Multiple Access (DAMA) techniques, heavily used in the return link satellite standards, are described. Then, an overall classification of the RA techniques is given. The latter constitutes the main subject of study in this thesis.

2.2.1 Demand Assignment Multiple Access

Demand Assignment Multiple Access (DAMA) is the major scheme to share the channel between multiple transmitters. It offers a dynamic allocation based on demand (whenever a packet is to be transmitted) rather than on users (a permanently allocated resource). In this deterministic approach, a resource is assigned to a single user, it is thus exploited only by him, but not permanently. Generally, in the DAMA methods found in the DVB-RCS2 standard, the Satellite Terminal (ST) needs to make an allocation request to the gateway, based on its needs for rate or for volume.

- Rate Based Dynamic Capacity (RBDC): it allocates dynamically a resource according to the requested rate by a user for a certain period of time. It is mainly used for unpredictable traffic profiles.
- Volume Based Dynamic Capacity (VBDC): it allocates dynamically a resource according to the requested volume by a user. It is mainly used for sparse traffic in a cumulative way (in terms of requests). It could be used for HTTP (HyperText Transfer Protocol) traffic for example. When the requests are not cumulative, an Absolute VBDC (AVBDC) is defined.

Nevertheless, it can also be found in the DVB-RCS2 standard assignment techniques which are not based on requests.

- Constant Rate Assignment (CRA): it is based on a constant rate that is fully provided without request according to the available capacity. It could be performed through Call Admission Control algorithms, which can be used in low latency and guaranteed bandwidth applications such as the Voice over IP (VoIP).

- Free Capacity Assignment (FCA): the unused capacity is allocated to STs without any prior request. As the unused capacity is not constant in time, this type of allocation is only efficient in low traffic environments.

These techniques, that are based or not on allocation requests, constitute the principal features in many protocols. For instance, the FCA combined with allocation requests and Piggy-Backing define the CFDAMA-PB protocol [LM93]. The Piggy-Backing part of the capacity reservation is inspired from the Contention-based Priority Oriented Demand Assignment (CPODA) [Jac+77]. However, as the efficiency of the DAMA techniques heavily relies of the traffic characteristics in addition to the number of active users, the performance of the previously mentioned methods can be easily degraded [DG+16]. Therefore, PRedective DAMA (PRDAMA) [ZYL02] that uses linear prediction for traffic has been proposed. As a matter of fact, a more efficient reservation scheme that allocates resources to terminals which are expected to need bandwidth, reduces the delay.

However, all the capacity requests and overhead that require the ST to estimate its needed rate adds significant propagation delay, especially if the satellite is geostationary (500 ms of Round Trip). For this reason, another scheme that does not depend on the traffic characteristic is deeply investigated in the literature, which is Random Access. It is first proposed to be used during the logon phase accompanied by data transmission in DAMA [HM83][CM88]. The performance in terms of delay are though not satisfying in high channel loads. Researchers found it then interesting to evaluate RA performance for the whole packet transmission.

2.2.2 Recent RA techniques

Interference has always been an issue to avoid in traditional communication systems. However, the expansion of the number of communicating devices and time-critical applications encouraged researchers to look for efficient solutions such as RA to be alternative or complementary to DAMA. The Random Access methods on which we will focus along this dissertation concern the ones based on the Physical Layer Network Coding (PLNC) [ZLL06][PY06] inspired from the conventional Network Coding (NC) [Ahl+00][LYN03]. The main feature of NC is to reduce the transmission delay with a higher spectral efficiency. Initially, a collision with its own transmitted packet is taken advantage of through self-interference removal. Generally, in a typical network scenario a communication between nodes go through intermediate relays. These should make linear combinations of the received packets before forwarding them further. A given end receiver (node) uses all received combinations to retrieve the original information with interference removal. The PLNC means that, unlike NC, the interference removal is performed at signal level, which will be referred to as Interference Cancellation (IC).

Moreover, Paired-Carrier Multiple Access (PCMA) [M.D98], first applied in satellite communications, is perceived as being the ancestor of the PLNC. Indeed, a self interference is allowed through the superimposition of the uplink and downlink carriers. The goal is then to suppress the known self interference (original uplink signal to be transmitted), in order to decode the other signal (original downlink signal to be received). As a result, the rental cost of the space segment is reduced, as more resource is available.

In what is following, are briefly presented some of the recent RA techniques based on ALOHA protocol for satellite communications. They can be classified into two main categories; synchronous and asynchronous solutions, also commonly called slotted and unslotted. On the one hand, the synchronous RA methods are characterized by a common time reference to all users that allow them to organize their transmissions according to time intervals, called time slots. A frame structure with a given number of slots is generally defined. Nevertheless, different time shift synchronization errors, up to a given number of symbols, can occur within a time slot for each packet, which requires channel estimation. On the other hand, in the unslotted methods, there is no coordination or synchronization between users. As packets are asynchronously transmitted, the incurred interference, which was total in the slotted environment, is mostly partial.

Synchronous solutions

The slotted RA protocols operate on well defined TDMA frames that are divided into a finite number of time slots. Thus the whole frame is transmitted over a single frequency. Nonetheless, the Multi-Frequency domain will be briefly addressed. The packets to transmit are randomly placed over the different time slots, which means collisions are tolerated. Synchronization between users is yet necessary here to maintain a common timing reference. Before starting to enumerate the main synchronous RA methods, performance evaluation metrics are presented.

Performance evaluation metrics

The performance of recent RA protocols in satellite communications is mainly evaluated, as for previous traditional methods, according to the peak throughput and the Packet Loss Ratio (PLR). These actually highly rely on the number of transmitters that describes the channel load, also named MAC (Media Access Control) layer load. This is due to the fact that the probability of collision on a time slot becomes significant with the increase of the number of transmitters (leading to a bigger packet loss). A normalized value of the channel load G, derived from the average number of packets per time slot, is defined as being the mean number of bits per symbol. It then depends on the used MODCOD with a given modulation order M and a coding rate R such as: $G = \lambda R \log_2(M)$ in bits per symbol. When N_U active users transmit N_R replicas each, on a frame of N_S time slots, the value of λ is as follows: $\lambda = N_\mathrm{U}/N_S$ in packets per slot. The value of λ and thus of G is normalized with respect to the number of replicas N_R. Obviously, the final value of G is also normalized according the MODCOD parameters. This normalization procedure is made in order to fairly compare any systems regardless of their chosen parameters (N_R, R and M).

Now that we have defined the MAC-layer load G that depends principally on the number of users, the number of slots and the MODCOD, the throughput and PLR can be expressed. Indeed, both of them are computed with respect to G. The PLR value at a given MAC-layer load G represents the ratio between the number of undecoded packets N_d and the total number of transmitted packets that corresponds to N_U. We recall that for given values of the MODCOD parameters and N_S, a different value of N_U matches a unique value of G.

$$PLR(G) = \frac{N_\mathrm{d}}{N_\mathrm{U}} \tag{2.1}$$

From (2.1), the probability of correct decoding of a packet at a given channel load $P(G) = 1 - PLR(G)$ is derived. The corresponding throughput T represents then simply the mean number of packets correctly decoded on a time slot. It can be normalized in bits per symbol using G instead of λ.

$$T(G) = G \cdot P(G) \quad \text{(bits/symbol)} \tag{2.2}$$

In what follows are briefly presented the main recent RA techniques with a longer description for the methods of interest we used in this thesis.

Contention Resolution Diversity Slotted ALOHA

Contention Resolution Diversity Slotted ALOHA (CRDSA) [CDD07] emerged as a leading technique for recent advances in slotted satellite communications RA techniques. It uses ALOHA protocol combined with multi-replica transmission and interference resolution using Successive Interference Cancellation (SIC) at reception.

Thus, two or more time slot positions are randomly selected on a given frame by each user to transmit a packet several times. Transmitting multiple replicas of a same packet on the same frame was initially introduced by Diversity Slotted Aloha (DSA) [CR83]. A packet's replica includes a payload joined with signaling information which points to the other replicas' positions, preceded by a preamble and succeeded by a postamble. Guard intervals are added in each time slot to counterbalance the potential timing errors (see Figure. 2.2). It is worth noting that the packet should be decoded in order to retrieve the signaling field pointers.

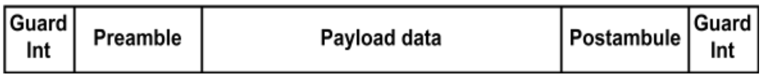


Figure 2.2: Packet structure in a typical TDMA frame.

The preamble, which is mainly used for channel estimation and determining the beginning of a packet, can be chosen from a set of pseudo-orthogonal codes as in the first version of CRDSA or it can be unique and common to all users. It can be BPSK (Binary Phase Shift Keying) or QPSK (Quadrature Phase Shift Keying) modulated. In case of multiple pseudo-orthogonal codes, replicas of a same packet use the same code. When a unique word is used as a preamble, channel estimation (frequency shift, timing offset and carrier phase) can be performed by exploiting the payload symbols of a clean packet (non collided replica) [Cas+12].

At reception, the frame is analyzed to look for non interfered replicas. Once a clean replica is found, decoding is attempted. The pointers towards the other replicas of the same packet are then exploited to reconstruct the other replicas' signals. All of the replicas can forthwith be removed from their respective positions. The removal of interference requires channel estimation and compensation for all replicas on their respective positions to avoid residual errors. Each of the timing offset, frequency shift and amplitude can be estimated from the decoded replica. However, carrier phase related to each replica has to be estimated on each position independently from the others (but the clean replica can be used as mentioned before). It is important to note that applying IC on the frame results in a smaller interference regarding the remaining signal, specifically in the time slots where decoding occurred. The next step consists in analyzing the frame again, in a successive way, and repeat the same process to unlock some of the configurations until all packets are decoded or until there are no more clean packets. The latter case describes a deadlock situation for CRDSA.

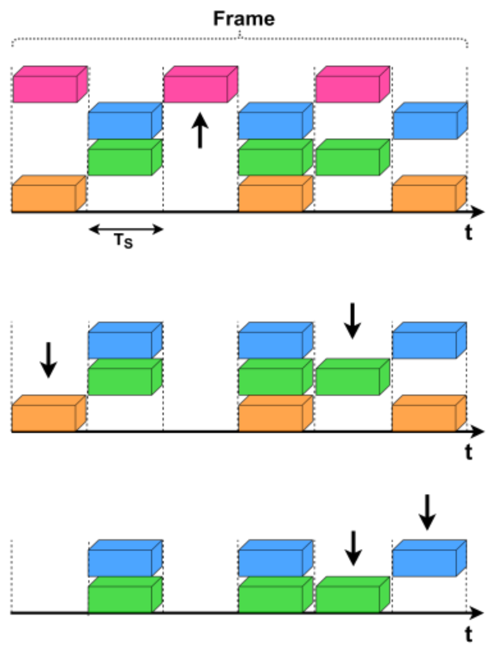


Figure 2.3: CRDSA SIC operations at reception with three replicas per packet.

In the example of Figure 2.3, the first analysis of the frame discovers the second replica of the pink packet which does not incur any collision. After the payload is retrieved, the signaling field regarding the other replicas' positions can be revealed. Thus the next step consists in reconstructing the other replicas (coding and modulation) and their respective signaling information which is the only difference between the three replicas. Then all of them are removed from the frame after channel parameters estimation and compensation. This operation contributes in solving the blue and orange packets as their first and last replicas, respectively, become clean. SIC operations are then performed in the same way as for the pink packet, which will allow to retrieve the last two packets. In this case, the whole system is solved. However, a deadlock situation could have been reached if no clean packets are revealed.

CRDSA proves its efficiency compared to the legacy Slotted-ALOHA scheme in terms of throughput and PLR. Indeed, while slotted ALOHA attains 0.36 bis/symbol, CRDSA reached 0.52 bits/symbol with two replicas per packet and 0.67 with three replicas (for 100 slots per frame and 100 information bits per packet).

Variants of CRDSA

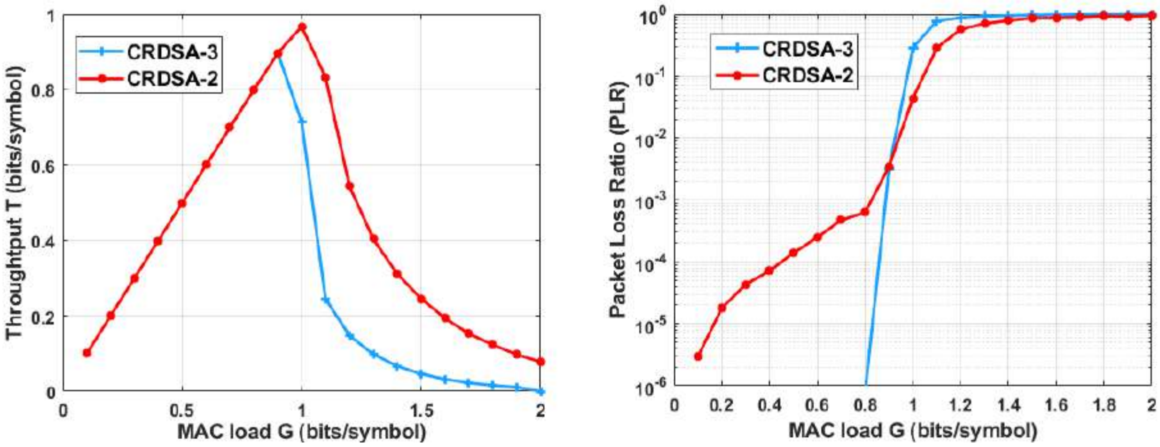
The variety of parameters in CRDSA pushed researchers to investigate the impact of choosing different values for them on the overall system performance. Each time a parameter is subject to study, a given performance or issue in CRDSA is targeted.

A. Packet power and coding rate

Compared to the initial version of CRDSA, CRDSA++ presented in [RHDG09], added and changed some of the parameters. For instance, it was the first time more than two replicas per packet were considered. This resulted in a less significant PLR floor that is due to the loop phenomenon. Also, power unbalance with a lognormal distribution between the received packets was considered, which offered better overall performance in terms of throughput and PLR. It is obviously due to the IC process that removes the contribution of the strongest signals first to be able to iteratively solve all the remaining packets. Moreover, another important aspect of CRDSA has been investigated in [dD14]. It mainly concerns the Forward Error Correction (FEC) coding rate. Using a stronger code allows better collision resolution as decoding is possible even in presence of interference. This clearly gets more effective with the increase of the value of E_S/N_0 (the energy per symbol to noise power spectral density ratio) and offers significantly higher performance in terms of throughput and PLR. In Figure 2.4 for instance, we can notice that by introducing 3GPP Turbo coding with rate 1/3 coupled with QPSK modulation, the maximum peak throughput reaches 1 bits/symbol with two replicas per packet (CRDSA-2), which is almost doubles compared to the 0.52 mentioned before. Also, the error floor caused by the loop phenomenon is drastically mitigated with CRDSA-3 (with 3 replicas per packet).

B. An irregular number of replicas per packet

G. Liva proposed in [Liv11] to use a different number of replicas per packet which can vary from one transmitter to another according to a bipartite graph. The latter representation is used in LDPC codes [Gal62] for FEC construction. In the proposed method IRSA that stands for Irregular Repetition Slotted ALOHA, the bipartite graph is constructed in accordance with



(a) Throughput (b) PLR

Figure 2.4: Performance evaluation in terms of PLR and throughput of CRDSA with 2 and 3 replicas per packet, 100 slots per frame, useful information of 100 bits, QPSK modulation, 3GPP coding of rate 1/3, equipowered packets, $E_S/N_0 = 10$ dB over and AWGN channel.

the frame structure and SIC operations. The number of replicas is then managed according to the analyzed convergence of the SIC process, which leads to a an irregular bipartite graph. The positive results regarding the maximum achieved throughput comes at the expense of an added complexity regarding the irregular number of replicas choice. In addition, the PLR performance is approximately similar to the one of CRDSA around 10^{-3} when the number of replicas is more than two. Nevertheless, in-depth analysis of IRSA under real channel conditions have been investigated in [MDA17a]; which showed that its performance is close to the one of CRDSA. Other studies and improvement schemes were particularly interested in IRSA such as [GS13][Pao15][Int+15][Sun+16][PLG17][Sha+19] where tracks like multiuser detection, a finite number of users and frame lengths, SIC at the slot level, priority settings for decoding, structure for low error floor and even NOMA-based IRSA have been explored with interesting enhancement results.

C. Multi-Frequency domain and Spread Spectrum

Another aspect to study is the multi-frequency domain. Indeed, a TDMA-like frame necessitates a transmission power peak that is multiplied by the number of slots compared to an FDMA system for example. It is because of the shorter time of transmission, which can be restrictive for low power terminals. Multi-Frequency (MF) CRDSA [Liv11]**f2** could reduce the peak transmission power but at the expense of poorer performance than the original CRDSA. In addition to that [MDA17b] also presented CRDSA with spread spectrum (SS-CRDSA).

The goal was to reduce the loop phenomenon in CRDSA with two replicas per packet. The loop phenomenon is a result of having two users or more transmitting at the exact same positions. It manifests itself as an observable error floor at the PLR curve. It was concluded that using the spread spectrum in the multi-frequency domain (SS-MF-CRDSA) contributes not only in reducing the transmission power peak but also in enhancing the performance. Though, the Multi-frequency aspect can yield to an additional complexity.

D. Time slot reservation scheme

One of the variants of CRDSA, called R-CRDSA [Lee+12], relies on a reservation scheme of time slots. These are slots where a successful decoding of a replica occurred on a previous frame for a given transmitter. In the next frame, the slot is reserved to that user until he finishes sending all his successive packets. In this case, a single replica of each packet is transmitted. Then the slot is no longer reserved to him. Knowing the frame information with time slots' status is required for each transmitter in R-CRDSA. This method offered a considerably higher throughput in addition to a further critical point. The latter is the starting point of the throughput collapse. However, PLR performance have not been studied. Another version with a Multiple Slot Reservation CRDSA is investigated in [Yun+15] according to how many packets a user have to transmit. It has been shown that fewer frames are required to transmit all the packets than in R-CRDSA with an expected smaller transmission delay.

E. Frameless scheme

CRDSA with a non frame structure method, based on a Sliding Window process, called SW-CRDSA has been proposed in [Mel+12]. This means that, similarly to DSA, a packet does not have to wait for a frame to be transmitted; it is directly sent over the next time slot which would also reduce the probability of undecodable configurations as in a time-limited frame schemes. As a result, a gain of 13% is observed for the throughput, compared to the standard CRDSA, with a higher peak. A smaller transmission delay is also achieved. Nevertheless, besides the PLR performance that has not been studied, memory analysis and detection appears to yield an extra complexity.

Multi-replicA decoding using CorRelation baSed locALisAtion

The combination of multi-replica transmission with SIC process by CRDSA proved to achieve remarkable performance enhancement regarding the peak throughput and PLR. Still, the randomness of the system can make some frames less solvable than others because of the deadlock situation that can occur at an early stage of the SIC operations. A deadlock situation is described as the absence of clean packets, which means that there are no more decodable packets. The latter situation can refer to the case where there are no longer non interfered replicas as in the initial version of CRDSA, or when the number of interference is higher than the tolerated amount of solvable collision thanks to coding.

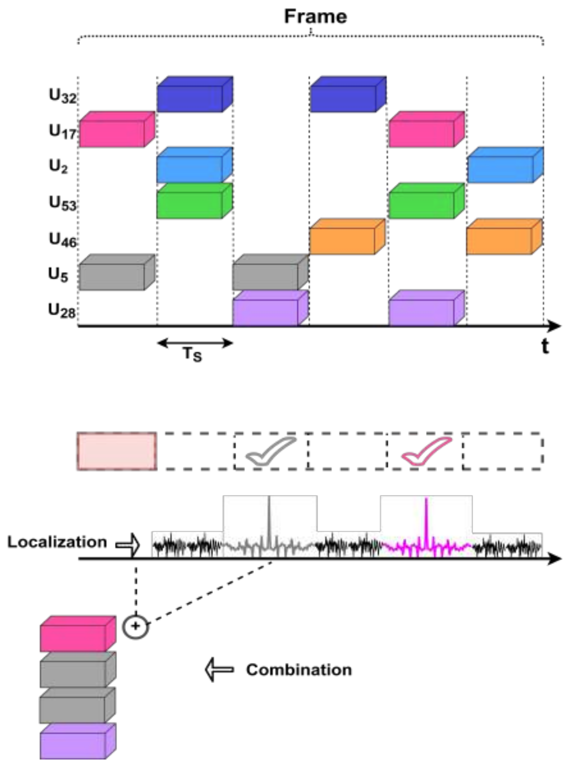
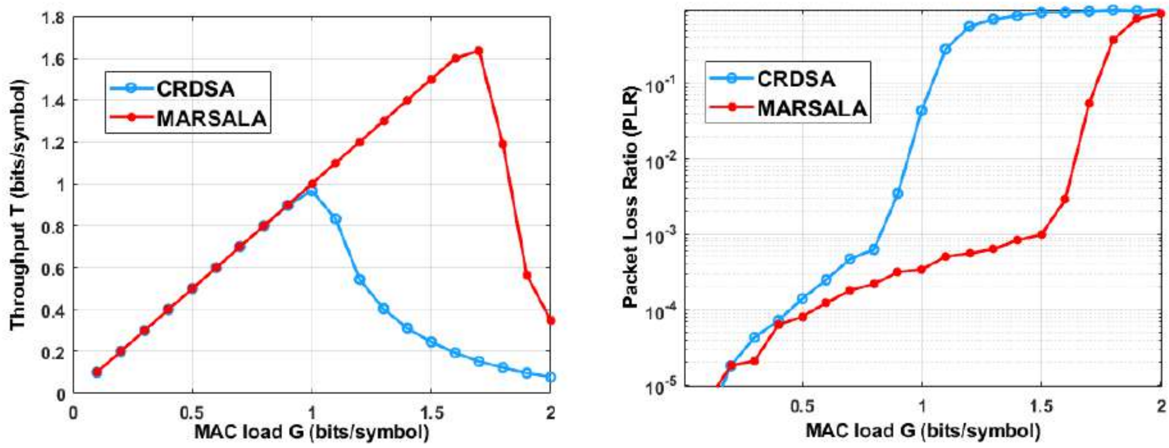


Figure 2.5: MARSALA process to unlock CRDSA's deadlock.

When CRDSA is blocked, the Signal to Noise plus Interference Ratio (SNIR) is low enough to prevent decoding and therefore, the locations of replicas on the frame can no longer be known. We recall that this information is only accessible when the payload is recovered. At this point, Multi-replicA decoding using corRelation baSed LocALisAtion (MARSALA) [Bui+15] can intervene to unlock the system and retrigger CRDSA again. First, a Reference Time slot (RTS) with collided packets is randomly chosen on the frame, then, correlations are made between the RTS and the remaining time slots on the frame. This step is meant to localize, on the frame, the other replicas of the collided packets on the RTS. Secondly, signal combination between localized replicas of the same packet is introduced by MARSALA before decoding. This will result in a higher power of the packet of interest and thus a potentially higher SNIR. If the number of replicas is more than two, an association step is added to the previous localization step in order to associate each N_R localized replicas to a given packet. In order to do that, it was first proposed to combine the RTS with the highest correlation peak signal, which corresponds to a second replica of one of the collided packets and then perform extra correlations between the combined signal and the remaining previous peak positions that regroup all replicas of collided packets on the RTS. The association step is obviously spared if the number of replicas is equal to two as any two combinations will associate the two replicas of a given packet. Moreover, a particularly important task for MARSALA to fulfill is the estimation and compensation of the timing offset and phase shift differences between replicas for a maximized coherent combination gain. Proposed solutions and signal processing details are provided in [Zid+15] and [Zid+16].



(a) Throughput (b) PLR

Figure 2.6: Performance evaluation in terms of PLR and throughput of CRDSA and MARSALA with 2 replicas per packet, 100 slots per frame, useful information of 100 bits, QPSK modulation, 3GPP coding of rate 1/3, equipowered packets, $E_S/N_0 = 10$ dB over and AWGN channel.

The frame state in Figure 2.5 displays a deadlock situation for CRDSA as there are no clean packets. In order to unlock the system, MARSALA proceeds with its two steps. First, an arbitrary time slot is selected as an RTS, in this case the first time slot. Then, the localization procedure that consists in data correlations between the RTS and the remaining slots on the frame is applied. Ideally, correlation peaks that point to the positions of the other replicas of collided packets on the RTS should be detected. In this case of two replicas per packet, the second replicas' positions of the collided packets (U_{17} and U_5) on the RTS show correlation peaks on their respective positions. One of the peaks is selected for the combination step; it could be the highest one. After estimation and compensation of the channel parameters, a coherent combination is performed and should offer a higher SNIR for U_5 packet as its power is four times higher. After interference cancellation, CRDSA is retriggered again. As a matter of fact, U_{17} and U_{28} can be removed because they have their first replicas, each, clean. At the next iteration U_{53} will have its second replica free from collision, which means that the packets can be retrieved and suppressed from the frame. However, another deadlock is reached afterwards because of the configuration of users U_{32}, U_2 and U_{46}. They form a cycle over three positions where their replicas are placed. This means that MARSALA is needed again to unlock the system and break the cycle. The alternation between the processes CRDSA and MARSALA should offer remarkable performance, compared to using CRDSA alone. We can observe on Figure 2.6 that the throughput is significantly enhanced when using MARSALA. It represents a gain of 67% and reaches its peak at a higher load (1.67 bits/symbols at a load of 1.7), unlike CRDSA that attains its 1 bit/symbol at a load of 1 bit/symbol. The PLR improvement is also noticeable as it allows the system to remain below the conventional PLR target of 10^{-3} (for satellite communications) at higher loads than when CRDSA is used alone.

Nevertheless, the PLR error floor that corresponds to the high probability of the loop phenomenon persists in MARSALA when two replicas are used. Increasing the number of replicas per packet leads to having a higher complexity, especially because of the added operations introduced by MARSALA. This is actually the main issue we focused on during this dissertation.

Coded Slotted ALOHA

In the proposed Coded Slotted ALOHA (CSA) [PLC11], diversity is exploited in a different way than in the previously explained methods. Unlike CRDSA and its variants, no whole replicas of the same packet are transmitted. Instead, the packet is time-divided into several segments, which are then coded together with an erasure coding. Before transmission, each of the coded segments are coded independently with an error correcting code to cope with potential transmission errors. The receiver tries to recover the clean segments on their respective time slots in order to retrieve the information about the other segments' positions and coding, belonging to the same fragmented packet. After decoding the maximum amount of segments, SIC process is applied to remove their contribution to the interference level. Despite the throughput improvement that CSA testifies, compared to IRSA and CRDSA, its implementation appears to be more complex because of the required signaling that grows fast with respect to the number of segments.

Multi-Slots Coded ALOHA

Another advanced synchronous RA protocol that focuses on the coding aspect is Multi-Slots Coded ALOHA (MuSCA) [BLB12][Bui12]. The difference with CSA is that MuSCA proceeds to the packet fragmentation after it is coded with a low rate FEC code, and has only one level of coding. Thus, each of the fragments should include a robustly coded signaling information about the other fragments' locations on the frame that is added to it. The signaling field should be distinguished from the useful information, it is therefore differently coded with a Reed-Muller code. This way, the receiver can first attempt to decode the signaling information in the same way as it will do for the useful one; decode one of the fields belonging to the same packet, reconstruct the other fields according to the decoded information then remove them through SIC operations. MuSCA offers a considerable gain in performance compared to CRDSA. Nevertheless, overhead is added due to the robust coding.

Asynchronous solutions

Two types of unslotted ALOHA RA methods can be distinguished according to whether Spread Spectrum (SS) is used or not. First, the main SS technique is herein presented, then, the non SS methods on which this thesis is partially build are described.

Performance of unslotted solutions are evaluated, as for slotted techniques, through their peak throughput and PLR. As these depend on the MAC-layer load G in bits per symbol, the latter does no longer rely on the mean number of packets per slot but rather on average number of packet arrivals over one packet duration. Interference can be total, if colliding packets arrive at the exact same time as for the packet of interest, or partial (in samples or symbols) if interfering packets arrive during or before the arrival of the packet of interest with one packet duration ($]-1;+1[$). Nonetheless, the computation of the PLR and the throughput remains the same as in (2.1) and (2.2).

Enhanced Spread Spectrum ALOHA

Enhanced Spread Spectrum ALOHA (E-SSA) proposed in [DD12] is an improved version of Spread Spectrum ALOHA (SSA) [Abr96]. Thus, E-SSA presents an asynchronous method based on SS, as in SSA, but with significantly enhanced performance. It exploits a Sliding Window (SW) scheme with iterative IC. This means that each time a packet is retrieved and removed (typically the packet with a highest SNIR) in the sliding window, the interference contribution becomes lower and thus SIC can be performed iteratively until there are no more decodable packets. At this point, the SW is moved forward by a window step. Typically, the SW size should be as long as three times the length of a packet and the window step 1/2 or 1/3 the SW length. These choices aim to keep the smallest window size possible for a good packet detection and keep the window step the largest possible for a reduced complexity. The performance evaluation presented in [DD12] imposes E-SSA as a very promising modern RA, especially if a lognormal distribution for the packet power is used. It can actually reach a throughput that is 110 times higher than the conventional SSA method [DG+16].

Enhanced Contention Resolution ALOHA

Enhanced Contention Resolution ALOHA (ECRA) [CK13] proposes to improve the scheme of Contention Resolution ALOHA [Kis11]. The latter is the first attempt to make CRDSA asynchronous by removing the slot notation. This means that packets can be randomly placed on the frame without any constraints on timing, which considerably reduces the loop phenomenon occurrence. In the same unslotted environment, ECRA proposes to combine, in its initial version, the non-interfered portions or the least interfered ones of replicas belonging to the same packet. This is called Selective combining (SC). This way, a new packet is formed with the lowest interference rate and hence a higher probability of decoding. Later, in the more recent version of ECRA, other combining techniques are used such as Maximum Ratio Combining (MRC) [Bre59] and Equal Gain Combining (EGC) [Cla+17][CKM18]. EGC consists in adding together all received replicas of the same packet as in MARSALA, while MRC, widely used in Multiple-Input Multiple-Output (MIMO) and Multiple-Input Single-Output (MISO), aims to add replicas together according to the SNIR level.

It is important to note that ECRA proceeds to packet decoding in two phases; the combining process (which is the second phase) is performed only when the first phase based on SIC has failed in retrieving clean packets (browsing the memory to look for decodable packets). The combination phase, obviously, requires perfect knowledge of the replicas locations. The latter is addressed in [Cla+17] where the fact that timing offset of a packet at reception is the same for all its replicas, is exploited. Therefore the delay between burst replicas is considered as a multiple of a virtual time slot duration. In addition, ECRA uses SIC at reception in its latest version with a decoding process that is based on a sliding window inspired from the asynchronous method ACRDA [De +14] presented below. ECRA presents a significant gain in performance compared to CRA in terms of throughput and PLR. It also proves to perform better than CRDSA when MRC is used.

Asynchronous Contention Resolution Diversity ALOHA

Asynchronous Contention Resolution Diversity Slotted ALOHA (ACRDA) [De +14] is considered as the closest version to an asynchronous CRDSA. A specific Virtual Frame (VF) is assigned to each user in which he can transmit his replicas within virtual time slots. Two transmission modes are defined for ACRDA. They are mainly distinguished by the way to put replicas on the VF; in the baseline mode replica are randomly placed on the VF within the delimited virtual time slots, while in the variant mode, the location of the first replica is forced to be placed into the first virtual slot of the VF. The latter allows to have a reduced transmission delay for non critical loads but appears to be less significant at Transport Control Protocol (TCP) layer level as reported in [De +14]. The results show that both modes are though equivalent in terms of PLR and throughput.

This synchronization free system between users maintains replicas of a same packet synchronized together within a VF. VFs have all the same length and start at different times with no correlations between them, which makes the loop probability very low even with a small number of replicas per packet. At reception, SIC operations are performed with a decoding process based on a Sliding Window (SW) that browses the whole memory according to a window step. This means that the SW is first settled at the beginning of the memory for analysis and decoding. The SIC process goes over the SW as long as there are clean packets. If there are no more solvable packets, the SW is shifted with a window step in order to include more packets gradually, until it reaches the end of the memory. Like E-SSA, a window size

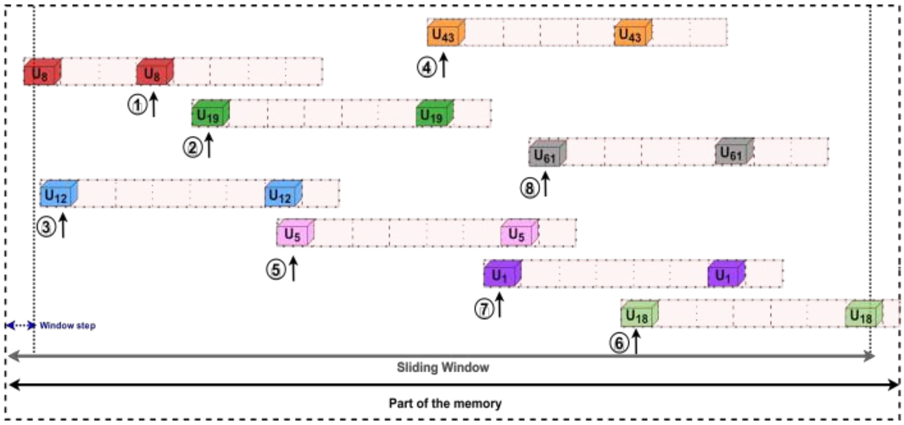
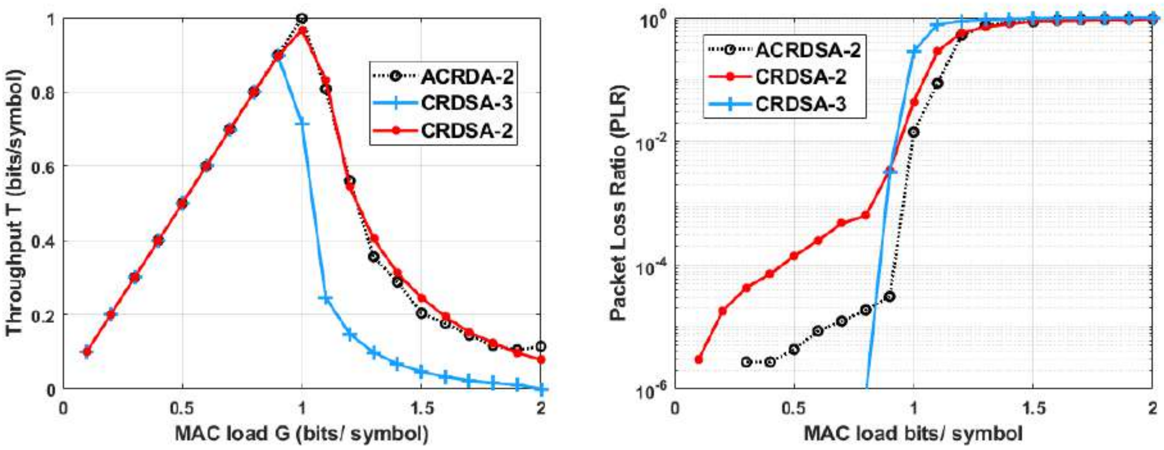


Figure 2.7: ACRDA decoding process with a sliding window and two replicas per packet using the variant mode.

of three times the length of a VF is recommended.



(a) Throughput (b) PLR

Figure 2.8: Performance evaluation in terms of PLR and throughput of ACRDA, compared to CRDSA with 2 replicas per packet, 100 slots per frame, useful information of 100 bits, QPSK modulation, 3GPP coding of rate 1/3, equipowered packets, $E_S/N_0 = 10$ dB over and AWGN channel.

Figure 2.7 shows a portion of the memory at reception for ACRDA decoding process. The SW is first placed at the beginning of the memory. We can observe that replicas of a same packet are positioned into a VF, independently from the others. The second replica of U_8 and the first replica of U_{19} are free from collision, which means that their respective useful information can be retrieved after demodulation and decoding. SIC is then performed to suppress these first replicas and their corresponding second replicas after signal reconstruction. U_{12} and U_{43} will have their first replicas clean as a result of the interference removal of packets U_8 and

U_{19}. Same SIC operations are held to solve packets U_{12} and U_{43}. The first replicas of U_5 and U_{18} will then be free from collision due to the removal of the second replicas of U_{12} and U_{43} respectively. The interference cancellation of U_5 frees the first replicas of U_{61} and U_1 from collision. All of them can be suppressed. However U_{18} has its second replica partially present of the SW. This means that in order for it to be removed, the sliding window has to move forward with a window step after having solved all possible decodable packets on the current SW.

The asynchronous nature of the ACRDA system makes its performance better than in CRDSA, especially because the error floor caused by the loop phenomenon is less significant when two replicas per packet are used (see Figure 2.8). Furthermore, the absence of synchronization between transmitters and the use of a smaller number of replicas per packet can keep the system complexity moderate with better or similar performance compared to CRDSA. However, packet search is more complex in ACRDA because there is no longer a time reference in slots on a common frame as in CRDSA.

Similarly to the latter, ACRDA can be transposed to the Multi-frequency domain, which is expected to reduce the transmission power of terminals.

2.3 Conclusion

We presented in this chapter the main different multiple access solutions used in satellite communications. Orthogonal multiple access offers a guarantee of no collision, and has long been the conventional way to share spectrum resources between multiple transmitters. The problem appears when massive connectivity of devices becomes necessary in recent applications due to the growing technology requirements. Such applications involve sporadic transmissions, low data rates, and are sensitive to delay. This is why NOMA techniques have been proposed to allow the occupancy of a resource, in time, frequency, code, space, or a combination of these by multiple transmitters possible. Furthermore, grant-free NOMA which do not rely on any resource occupancy request participates in significantly reducing the transmission delay, diminishes the signaling overhead, previously meant for allocation, and also keeps a longer lifetime of batteries.

Random access based on ALOHA protocol, which can be considered as a grant-free NOMA solution, plays a major role in potentially replacing DAMA methods for packet transmission in satellite communications. Many enhancement schemes have been proposed in order to improve the packet loss ratio and throughput, in addition to the spectral efficiency. Most of them rely on multi-replica transmission or spread spectrum at the transmitter side and successive interference cancellation at reception.

We focus in this dissertation on the non spread spectrum RA techniques based on ALOHA protocol. Such methods can induce a significant complexity at the receiver if the number of replicas is big and an error floor at the PLR performance if the number of replicas is small. At the same time, the collision probability with loss of information increases when the channel load grows, which makes it difficult to localize replicas and retrieve the useful information. These challenges are the main issues we tried to overcome in this work. All of them are addressed through the different upcoming chapters.

Chapter 3

Random Shared Position Technique for Interfered Random Transmissions

Contents

In slotted ALOHA RA legacy techniques, the receiver has no knowledge about packets' positions on the frame, which can make it difficult or complex to benefit from the information provided by the multiple replicas in low SNIR. We propose a Shared Position Technique for Interfered Random Transmissions (SPOTiT) as a softer way, than in MARSALA, to localize replicas of a same packet on a frame. This is due to the prior information provided by SPOTiT to the receiver regarding replicas' positions. A single preamble can be used or there can be a set of pseudo-orthogonal codes. In the latter case, the preamble associated to each user will be shared with the receiver as well. We present here R-SPOTiT (Random SPOTiT), which is the first version of SPOTiT that does not require any signaling overhead to provide a common information between the receiver and each of the transmitters. A second version of SPOTiT is addressed in Chapter 5, and its third version is explained in Chapter 7. The work presented in this chapter is mostly available in [Zam+18d] and [Zam+18c].

Later in this chapter, the reader will go through a whole description of R-SPOTiT, including transmission and reception characteristics. Before that, the problem statement in addition to the system model and assumptions are defined to properly set the environment we worked in. We also provide an initial modeling of the system complexity compared to MARSALA, which will be largely developed in Chapter 4. Finally, we present the simulation results and performance analysis of R-SPOTiT compared to CRDSA and MARSALA at the end of the chapter.

3.1 Problem statement

As indicated in Chapter 2, many RA techniques have been proposed according to the different needs and applications following the fast nowadays technological growth. Among the main encountered challenges, providing a high throughput with relatively non complex receivers becomes urgent. We recall that, among the ALOHA RA methods, MARSALA is meant to resolve CRDSA's deadlock when no more packets can be retrieved. First, it locates undecoded packets through time-domain correlations between an arbitrary reference time slot RTS (where the undecoded packets are positioned) and the remaining signal on the rest of the frame. Then, it coherently combines the localized replicas of the same packet before demodulation and decoding. It consequently offers significantly better PLR and throughput, but in return, it adds a processing complexity related to the correlation computation. Indeed, in order to localize replicas of a given packet, MARSALA proceeds to data correlations (the signal of the whole slot) between the RTS and all the other time slots of the frame, as in shown in Figure 2.5 of the previous chapter. Hence, not only the length of the signal to correlate is long, as it takes the whole slot, but also the number of correlations is maximum. In addition, the number of times that MARSALA intervenes to solve CRDSA's deadlock considerably increases when the number of transmitters gets bigger. This means that MARSALA's complexity grows, in a significant way, in high loads.

Taking into account the performance enhancement of MARSALA and the related complexity, the proposed solution R-SPOTiT aims to reduce the data localization correlations without degrading performance. These are the whole packet correlations between time slots.

3.2 System model and assumptions

In the considered system, there are N_{U} terminals, attached to a gateway or directly to the satellite if it is regenerative (see Figure. 3.1). Each one transmits in a synchronous way, over the same frequency, N_{R} replicas of the same packet. Each is positioned on one of the N_{S} time slots of a frame. We suppose each user waits for the next frame to send another packet. Thus, no more than one packet from a given user can be found on the same frame. The payload is a fixed-length set of symbols generated from N_{b} information bits which are

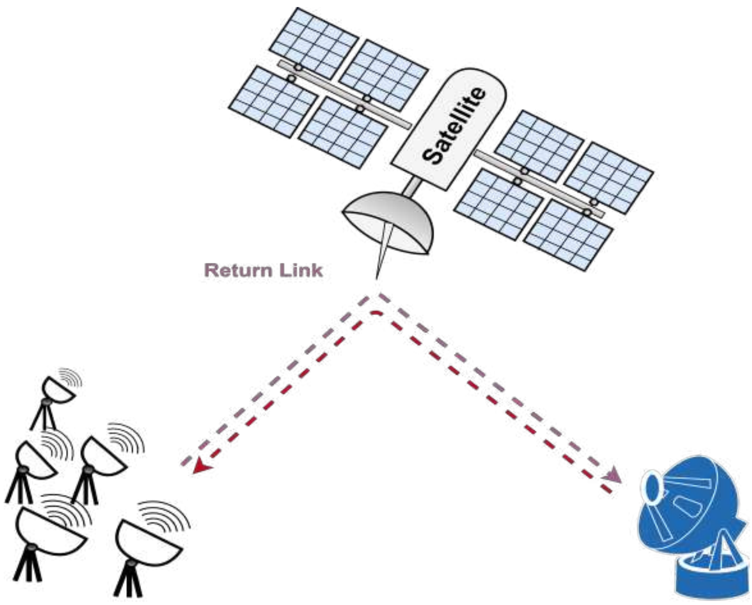


Figure 3.1: Satellite communication system example with a bent pipe satellite.

transformed into a MODCOD through coding and modulation. Packets are then formed by adding, at the beginning and at the end of the resulting payload symbols a preamble and a postamble respectively. We consider N_{P} pseudo-orthogonal preambles. These are the codes (constructed from cyclic codes and mostly used in multi-path channel transmissions) that are characterized with aperiodic inter-correlations. In addition, pilot fragments are randomly distributed in the packet for estimation matter. Guard intervals at the end of each slot are used to avoid interpacket interference due to potential synchronization errors.

At the receiver side, which can be the gateway or a satellite with a regenerative payload, CRDSA is applied first. It analyzes the frame and proceeds to collision-free packet detection and decoding on each time slot. We assume all packets are received with the same power (equipower). Replicas of the same demodulated and decoded packet are suppressed from their respective positions after one of them is decoded. Indeed, a non-collided replica points to the positions of the other replicas through a signaling field that is retrieved after demodulation and decoding. The frame is then analyzed again, thus applying SIC until there are no more decodable packets. A complementary treatment is triggered to resolve CRDSA's deadlock, which can be the legacy technique MARSALA or the new proposed method R-SPOTiT (see Figure 9.1) which is explained in the next section.

Both methods rely on replicas localization on the frame, and the combination of signals belonging to the same packet prior to decoding. The difference between MARSALA and R-SPOTiT is that the latter should require less complexity in the data localization process.

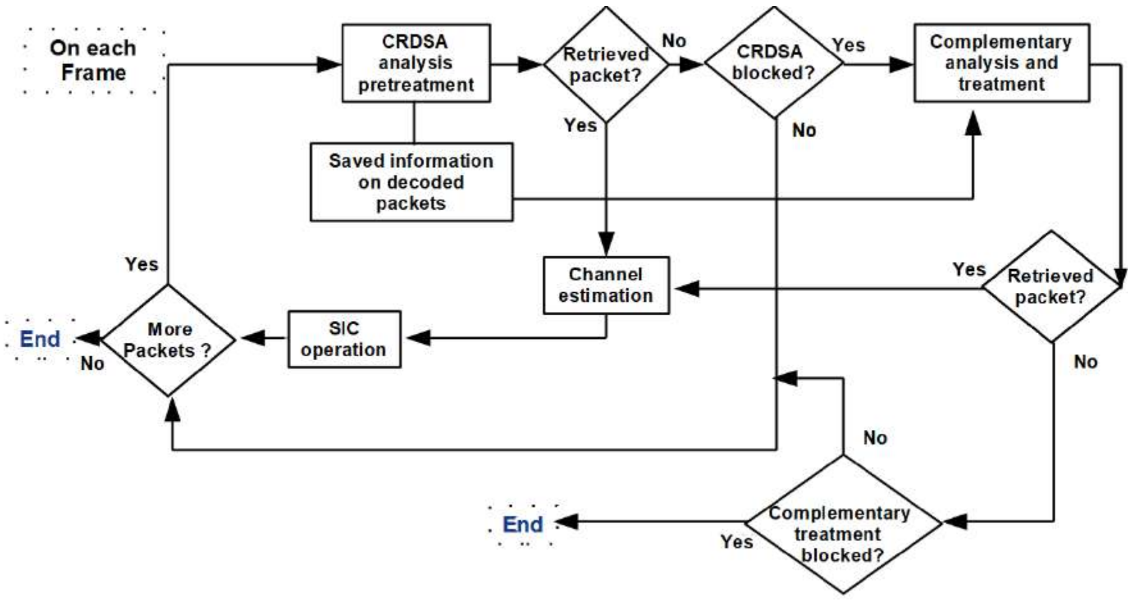


Figure 3.2: CRDSA with complementary treatment process

3.3 Random Shared Position Technique for Interfered random Transmissions

In this section the proposed multiple access solution R-SPOTiT is described when various pseudo-orthogonal preambles are used. Its general principle consists in the ability to beforehand communicate to the receiver the time slot positions of potentially transmitted packets and the associated preambles without extra signaling information. This is expected to reduce the localization complexity by limiting the time slot candidates where to perform data correlations. Thus, R-SPOTiT outlines a way of arranging packets on the frame and associating them with preambles. Transmission and reception aspects are detailed below.

3.3.1 Transmission

This part aims to explain how the transmitter selects its replicas' positions on the frame and the preamble to use in a way that makes the receiver be aware of them. Indeed, the main characteristic of R-SPOTiT is to provide a shared knowledge between the receiver and each terminal without any additional signaling information.

One solution is to use a PseudoRandom Number Generator (PRNG). It has been employed in [CKM18] and inspired from [CDD07] as signaling information that points to the position of a packet's replicas. However, it still needs in this case to be retrieved after demodulation and decoding of one of the replicas. Although, in R-SPOTiT, the PRNG uses the Identification Information (ID) known by both the transmitter and the receiver as a seed that generates the positions of replicas and a preamble. It is processed according to one of the two modes:

Fixed seed for each user

The unique Hardware IDentifier (HID) that is proper to each terminal is known by the receiver due to the logon phase. Indeed, each subscriber uses its identifier to login to the system. In other words, users send their identification information to the gateway to which they are attached to notify their presence and being active. Thus, when the HID is used as an entry seed to the PRNG, it makes sure that the receiver and each of the users are able to determine the same time slots on the frame and the preamble to be used at each transmission.

Dynamic seed for each user

In some applications where several users generate the same positions, and they transmit successively on the same frames, they create an unsolvable loop. A loop occurs when two or more packets are transmitted at the exact same positions on a frame, which makes the power of the packet of interest equal to the power of the interference after combination. As a result of the previously described scenario, continuous failure of retrieve will occur. To remedy this, a dynamic choice of positions and preambles is introduced. A dynamic combination can be used in order to have new time slots and preamble choices at each frame for each terminal. This shall involve an incremental identifier as an entry seed for the PRNG. For example, it can be obtained by adding U_{ID} the terminal HID to F_{ID} the frame ID, i.e. $F_{\mathrm{ID}} + U_{\mathrm{ID}}$ that is received or calculated using the synchronization information. Consequently, this dynamic combination between the HID and the frame ID avoids a continuous loop, for two users or more, in case of successive and simultaneous transmissions.

3.3.2 Reception

The receiver computes all replicas' positions and preamble choices of each subscriber using the predetermined seeds in the fixed or in the dynamic case and creates an information table. In Table 3.1 with two replicas per packet, $\mathrm{Slot}(u, r)$ refers to the time slot position of the replica r belonging to the user u, and P_u is the selected preamble for the same user u. This means the receiver knows all the potential users and their preambles that can transmit packets on each time slot of the frame. Thereafter, the pseudo-orthogonal characteristic of preambles is used to reduce the potential number of users on each time slot. A good preamble detection depends on the auto and cross-correlation properties of the code sequences in addition to their length. As a matter of fact, a detected preamble on a time slot will point to a certain number of users having that same preamble, from the receiver's information table. These users are the ones that could transmit data on that analyzed time slot. During the preamble detection phase, when a detected preamble points to a unique potential user (according to the information table), and its other replicas' positions exhibit as well a correlation peak of the same preamble, this indicates the presence of its packet. However if a correlation peak of a certain preamble on a specific time slot indicates, according to the information table, that it is associated to more than one user, the following strategies should then be applied.

1. Only preamble detection based method:

   The result of preamble detection made during CRDSA is stored and utilized by R-SPOTiT. As a matter of fact, the latter will first compare it with its information table. Then, it will check all replicas' positions of packets whose preamble is detected on the analyzed time slot. Positions that do not indicate the presence of the preamble of interest are eliminated from the potential transmitters. On the contrary, when one of

Table 3.1: Receiver's information table

Users	Position 1	Position 2	Preambles
$\mathbf{U_1}$	Slot(U_1,1)	Slot(U_1,2)	P_{U_1}
$\mathbf{U_2}$	Slot(U_2,1)	Slot(U_2,2)	P_{U_2}
$\mathbf{U_3}$	Slot(U_3,1)	Slot(U_3,2)	P_{U_3}
$\mathbf{U_4}$	Slot(U_4,1)	Slot(U_4,2)	P_{U_4}
$\mathbf{U_5}$	Slot(U_5,1)	Slot(U_5,2)	P_{U_5}
......	......	......	......

the positions or more show a correlation peak of the same preamble, and one of them points to a one potential user, this one is confirmed to have a packet on the current frame. However, when all replicas' positions of a user whose preamble is detected point to multiple possible packets, localization must resort to data correlations between slots.

2. Data localization correlations:
   The only preamble detection based method becomes difficult with the increase of the number of transmitters. Therefore data localization correlations over the whole slot are to be used, in addition to the preamble detection. Yet, in contrary to MARSALA which has $N_S - 1$ data localization correlations, only a small number is performed in R-SPOTiT. It is equal to the number of potential users having the same detected preamble on the slot when $N_R = 2$. Otherwise, data correlations are performed over the time slots containing the other replicas of potentially collided packets.

Once localization is successful, signal combination is performed between time slots containing replicas of the same packet before demodulation and decoding. Figure 9.2 summarizes the main differences between R-SPOTiT and MARSALA, seen as complementary treatments to CRDSA.

Example: In Figure 3.4, we assume all preambles are correctly detected and $N_R = 2$.
Let us take the first slot 'Slot 0' in the frame composition, with each color representing a distinct preamble. Each user u belongs to the set of N_U users whose time slot positions and preambles are selected through the PRNG. According to the information table lookup (Table 3.2) that concerns 'Slot 0', there are four potential users that can transmit one of their replicas in 'Slot 0': U_1 and U_{11} with the blue preamble we call P_1, U_{19} using the red preamble we call P_2 and U_{22} with the purple one we call P_3. The preamble detection of pseudo-orthogonal sequences on 'Slot 0' gives correlation peaks for P_1 and for P_2. This means the user U_{22} has

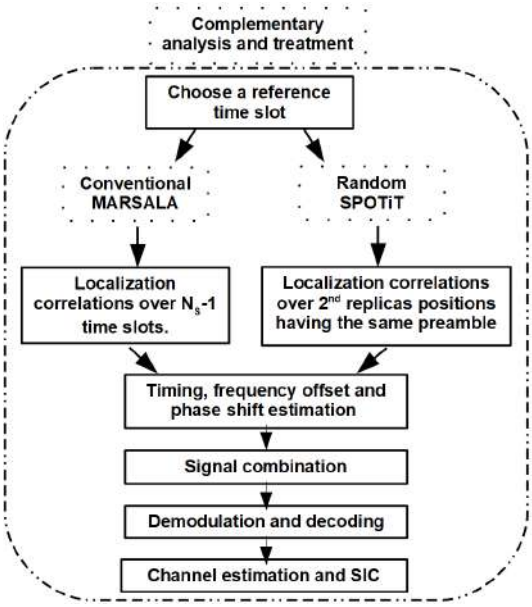


Figure 3.3: Explained complementary treatment process to CRDSA

not transmitted on this frame, thus only three candidates are to be investigated. Since U_{19} is the only potential user with the preamble P_2 that can send a packet on 'Slot 0', the red peak indicates its presence, especially, because its second replica's position shall exhibit a red correlation peak. As the receiver knows the location of its replica, from the information table, no data localization correlations are necessary. However, the blue correlation peak can even indicate the presence of one of the packets U_1 and U_{11} or both of them. In order to determine

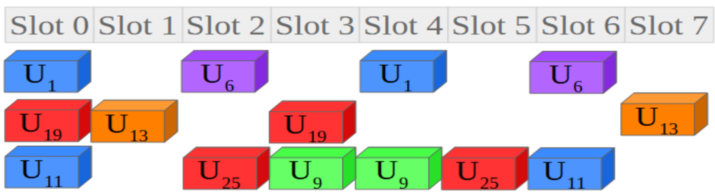


Figure 3.4: Example of a frame composition

which one has transmitted a packet, the result of preamble detection on both slots (slot 4 and slot 6) where the second replicas of U_1 and U_{11} is used. Slot 4 and slot 6 having both two correlation peaks a blue/green and blue/purple respectively will confirm the presence of U_1 and U_{11}. This is true because there is a unique potential transmitter with a blue preamble. In this example, only preamble detection was necessary. However, data localization correlations can be required in the case where more than one potential transmitter over all replicas' positions occurs, or when the false alarm of preamble detection probability is quite high.

Table 3.2: Time slot look up example

Slot 0	Slot 4	Slot 6
$\mathbf{U_1}$	$\mathbf{U_1}$	$\mathbf{U_6}$
$\mathbf{U_{11}}$	$\mathbf{U_4}$	$\mathbf{U_{11}}$
$\mathbf{U_{19}}$	$\mathbf{U_{23}}$	$\mathbf{U_9}$
$\mathbf{U_{22}}$		

3.4 Initial localization complexity analysis

This section will investigate the number of data localization correlations, assuming there are no loops, in case of MARSALA and R-SPOTiT. It concerns the correlations that are necessary to decode all the packets on an analyzed time slot and those needed to decode only one packet. We consider the worst case for R-SPOTiT when the only preamble detection based method has failed to locate more packets. We start the complexity analysis for any number of replicas before putting forward the case of two replicas which is less complex. As a matter of fact, when CRDSA is in a deadlock situation, MARSALA will randomly choose a reference time slot in order to perform necessary data correlations to locate the colliding packets' replicas. The number of localization correlations depends on the number of slots, the number of replicas and the number of the collided packets on this slot. MARSALA makes a two steps processing. At first, it locates all colliding packets' replicas on the frame using data correlations between the reference slot and the other remaining slots (first term of (3.1)). Then, it performs more correlations in order to associate the localized replicas to a given packet on the reference time slot (second term of (3.1)) before decoding and SIC. Therefore, one way to describe the whole process complexity is to compute the total number of correlations $N_{\text{MARSALA}}^{\text{Corr}}$ taking into account the localization and the association steps on a reference time slot:

$$N_{\text{MARSALA}}^{\text{Corr}} = (N_{\text{S}} - 1) + \sum_{c=1}^{N_{\text{Coll}}^{\text{Ref}}} \sum_{i=1}^{N_{\text{R}}-2} (N_{\text{R}} - 1)N_{\text{Coll}}^{\text{Ref}}(c) - i \tag{3.1}$$

where $N_{\text{Coll}}^{\text{Ref}}$ is the total number of collided packets over the reference time slot before MARSALA's decoding, $(N_{\text{R}} - 1)N_{\text{Coll}}^{\text{Ref}}$ is the total number of correlation peaks of replicas associated to $N_{\text{Coll}}^{\text{Ref}}$. Thus $(N_{\text{R}} - 1)N_{\text{Coll}}^{\text{Ref}}(c)$ is the number of the remaining correlation peaks after $c - 1$ SIC operations.

The association process is done by combining the signal of the reference time slot with the slot whose correlation peak is the highest. Afterwards, new data correlations with the rest of the peak slots are performed until the N_{R} replicas are associated.

On the other hand, R-SPOTiT depends on the number of replicas, the number of collided packets, the number of detected preambles and the number of potential transmitters. We consider the worst scenario when all collided packets having used the same preamble have the same timing offset. This means that when that preamble is detected, the receiver has no knowledge on which among all packet candidates have transmitted. As the receiver has knowledge about potential transmitters on a time slot and detected preambles, it will no longer be necessary to perform $N_{\mathrm{S}} - 1$ localization correlations as in MARSALA. Fewer data correlations $N_{\mathrm{SPOTiT}}^{\mathrm{Corr}}$ are needed to determine which users having the same detected preamble have transmitted on the analyzed time slot.

$$N_{\mathrm{SPOTiT}}^{\mathrm{Corr}} = (N_{\mathrm{R}} - 1) N_{\mathrm{PColl}}^{\mathrm{Ref}} \tag{3.2}$$

with $N_{\mathrm{PColl}}^{\mathrm{Ref}} = \sum_{p=1}^{N_{\mathrm{Det}}^{\mathrm{P}}} N_{\mathrm{pot}}^{\mathrm{Ref}}(p)$

- $N_{\mathrm{PColl}}^{\mathrm{Ref}}$ is the number of potential packets in collision on the reference time slot.
- $N_{\mathrm{Det}}^{\mathrm{P}}$ is the number of detected preambles.
- $N_{\mathrm{pot}}^{\mathrm{Ref}}(p)$ is the number of potential users with the detected preamble p that can transmit on the reference time slot.

Thus, for each detected preamble, R-SPOTiT performs data localization correlations only over the time slots containing the other replicas of potentially collided packets. These are the potentially collided over the analyzed reference time slot. No association is necessary because it is enough to confirm replicas presence by correlations on the well known time slots.

We have put our focus, in the complexity analysis, on the number of data correlations which are necessary to decode only one of the collided packets c over a time slot. This number $N_{\mathrm{MARSALA}}^{\mathrm{Corr(1),c}}$ for MARSALA is given in (3.3).

$$N_{\mathrm{MARSALA}}^{\mathrm{Corr(1),c}} = ((N_{\mathrm{S}})^{k} - 1) + \sum_{i=1}^{N_{\mathrm{R}}-2} (N_{\mathrm{R}} - 1) N_{\mathrm{Coll}}^{\mathrm{Ref}}(c) - i \tag{3.3}$$

$$k \in \begin{cases} 0 & \text{if } c = 1 \\ 1 & \text{for any } c > 1 \end{cases} \tag{3.4}$$

The first and the second term are respectively associated to the global localization process for all collided packets and the association process to localize the replicas of the packet of interest. k is equal to 1 when $c = 1$ and $k = 0$ for any other value of $c > 1$. This implies that the global localization process is made only once. In other words, $c > 1$ means that the global localization has been performed previously when tempting to decode one of the other collided packets on the same time slot. Actually, c is equal to 1 only once at the first localization of one of the collided packets on the analyzed time slot.

As for R-SPOTiT, the number of data localization correlations for one packet decoding $N_{\text{SPOTiT}}^{\text{Corr}(1),p,c}$ is related to a specific detected preamble p:

$$N_{\text{SPOTiT}}^{\text{Corr}(1),p,c} = ((N_{\text{R}})^k - 1)N_{\text{pot}}^{\text{Ref}}(p) \tag{3.5}$$

In this case, data localization correlations are only performed over the slots that contain all replicas of packets that can be potentially collided on same time slot. These shall have the same preamble p as for the packet of interest. These operations are also performed only once according to c, the same way as for the global localization of MARSALA explained above.

The next step analyzes the number of data localization correlations, for one packet decoding, in the case of two replicas. This case of having the minimum number of replicas is simpler in terms of complexity. As a matter of fact, this is a first step towards having a good solution for a less complex system.

- In MARSALA with two replicas per packet (MARSALA-2), the number of data correlations required to locate a packet before SIC does not depend on the association process:

$$N_{\text{MARSALA2}}^{\text{Corr}(1)} = ((N_{\text{S}})^k - 1) \tag{3.6}$$

- In R-SPOTiT-2 (with two replicas per packet), the number of correlations required to locate a packet with a preamble p becomes:

$$N_{\text{SPOTiT,2}}^{\text{Corr}(1),p,c} = \begin{cases} N_{\text{pot}}^{\text{Ref}}(p) & \text{k=1} \\ 0 & \text{for any } k \end{cases} \tag{3.7}$$

Remark

Let us take for instance the complexity related to the localization of one packet in terms of data correlations. MARSALA-2 and R-SPOTiT-2 have respectively $N_{\text{S}}-1$ and $N_{\text{pot}}^{\text{Ref}}(p)$ data correlations when $k=1$. This means that as long as the number of potential users having the same detected preamble is smaller than $N_{\text{MARSALA2}}^{\text{Corr(1)}}$, R-SPOTiT-2 is less complex. As mentioned before, $N_{\text{pot}}^{\text{Ref}}(p)$ depends on the total number of users, over the same frequency, attached to a gateway. Therefore, there is a maximum number of users beyond which the complexity between MARSALA-2 and R-SPOTiT-2 remains the same. A way to further minimize the complexity of the localization correlations in R-SPOTiT-2 is to start localization with the time slot that has the minimum $N_{\text{pot}}^{\text{Ref}}(p)$ for a preamble p. $N_{\text{pot}}^{\text{Ref}}(p)$ is retrieved from the information table at the receiver side. This can be applied from the beginning when the only preamble detection based method is to be proceeded. Nevertheless, the worst case can be described as when, with a certain number of users, the minimum $N_{\text{pot}}^{\text{Ref}}(p)$ for a preamble p is equal to $N_{\text{S}}-1$, i.e. $\min N_{\text{SPOTiT},2}^{\text{Corr}(1),p} = N_{\text{MARSALA2}}^{\text{Corr(1)}}$, and all these potential collided packets have their replicas on different time slots. This means that R-SPOTiT-2 should correlate the reference time slot with the $N_{\text{S}}-1$ different slots. In other words, R-SPOTiT-2 will have exactly the same behavior as MARSALA. However this case is extreme and depends also on the number of preambles which can increase or decrease the value of $N_{\text{pot}}^{\text{Ref}}(p)$.

3.5 Performance evaluation

To assess the system performance, an evaluation of the packet loss ration and throughput is realized along with a comparison of the corresponding data localization complexity between R-SPOTiT-2 and MARSALA-2. The PLR and throughput are obtained through a physical layer abstraction using the PER curve of the used MODCOD with an equivalent SNIR computation. We assume that the interference is approximated to AWGN (investigation and justification is provided in [dD14] Appendix B). Thus, the SNIR value is approximated to the SNR. We have considered that the payloads of packets are built from 100 information bits that are modulated with QPSK modulation and Turbo coded (3GPP coding) with rate 1/3, and are transmitted over a frame of 100 slots (in agreement with the CRDSA and MARSALA literature). Nevertheless, the size the frame, for a given application, should take into account the transmission delay and complexity if it is too long, and the high loop phenomenon if it

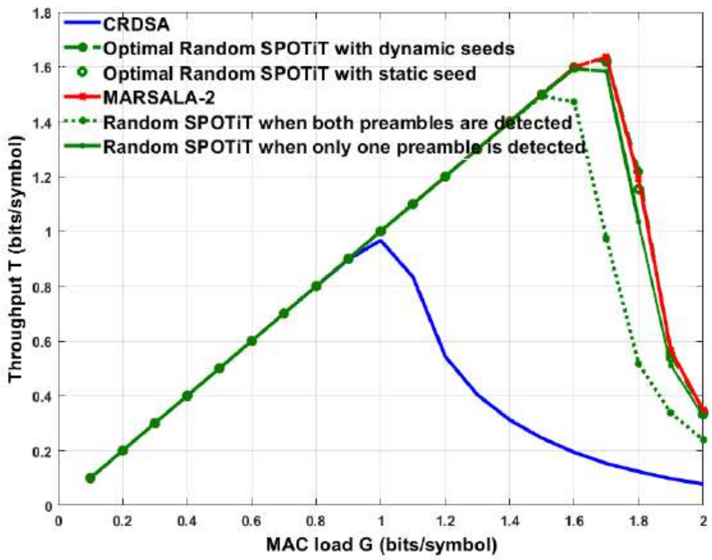


Figure 3.5: Throughput comparison between R-SPOTiT, MARSALA-2 and CRDSA, $E_S/N_0 = 10$ dB, 100 slots per frame, QPSK modulation, Turbo coding of rate 1/3 and equipowered packets of 100 bits and $N_R = 2$ replicas per packet.

is too short. We assume the channel model is an AWGN with an E_S/N_0 of 10 dB. The value of E_S/N_0 depends on the used system in terms of the used parameters, the transmit power of the transmitter, the gain of the receiver, the frequency bandwidth, channel attenuation and impairments,...etc. As we do not consider a specific system, we have chosen the value that is typically used for the equipowered case in the literature of E_S/N_0 [DG+16]. Gold pseudo-orthogonal sequences of length 31 are used as preambles, in agreement with CRDSA literature. The size of the preambles should also set a trade-off between the potential overhead if it is too long and the packet decodability in low SNIR if it is too short. Examples can be found in the DVB-RCS2 guidelines. 2000 users attached to the gateway are considered to be potentially transmitting over the same frequency.

We recall that R-SPOTiT, as for MARSALA, is a complementary method to CRDSA that is triggered whenever the latter finds itself in a deadlock. Assuming we have perfect channel estimation, Figure 9.3 and Figure 9.4 display the performance of R-SPOTiT-2 in terms of throughput and PLR in comparison to CRDSA with two replicas per packet and MARSALA-2. When the only preamble detection based method is used to decode a packet, both preambles shall be detected. In this case no data localization correlations are necessary for R-SPOTiT. Indeed, as replicas of a same packet have the same timing offset within a frame, the distance

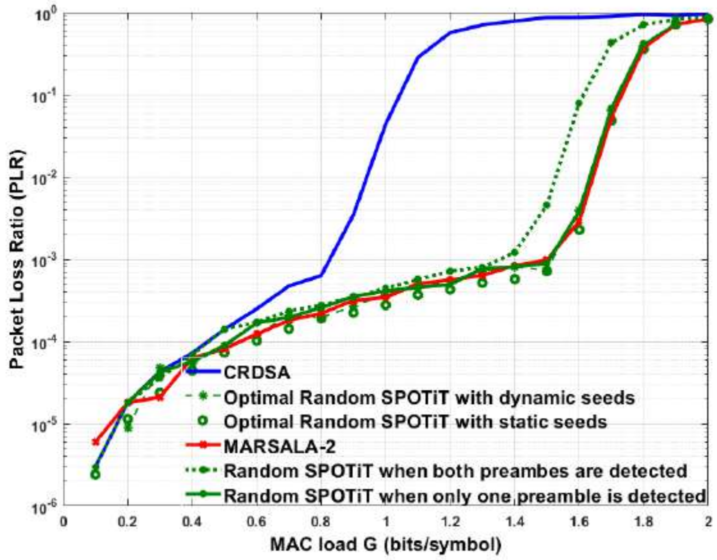


Figure 3.6: Packet Loss Ratio comparison between R-SPOTiT, MARSALA-2 and CRDSA, $E_S/N_0 = 10$ dB, 100 slots per frame, QPSK modulation, Turbo coding of rate 1/3 and equipowered packets of 100 bits and $N_R = 2$ replicas per packet.

between them is an integer number of slots, which confirms the packet's presence without extra correlations. As a result, a throughput of 1.5 bits/symbol is reached while MARSALA attains 1.64 bits/symbol. Nevertheless, performance can be enhanced when the shared information characteristic and the detected preambles are considered to perform data localization correlations. Indeed, only one detection of the two preambles of the same packet is required to perform data correlations over the second replicas' positions of potentially collided packets having the same detected preamble. Considering the decoding result from previous CRDSA and R-SPOTiT iterations, potential collided packets that have been decoded will be removed from the correlations to perform.

Figure 3.7 describes the average number of localization correlations needed to decode a packet in MARSALA-2 and R-SPOTiT-2. Data localization correlations for a packet decoding are performed only once at the first analysis by R-SPOTiT or MARSALA; assuming all positions are visible from the first analysis. This can be justified by the fact that a correlation over a whole slot is long enough, hence false alarms can be dismissed. Thus, when at least one of the replicas' preamble is detected in its respective position, R-SPOTiT reaches a throughput of about 1.6 bits/symbol with a negligible data localization correlation that goes up to about 0.3 with a MAC load of 2 bits/symbol. As mentioned before, the number of localization

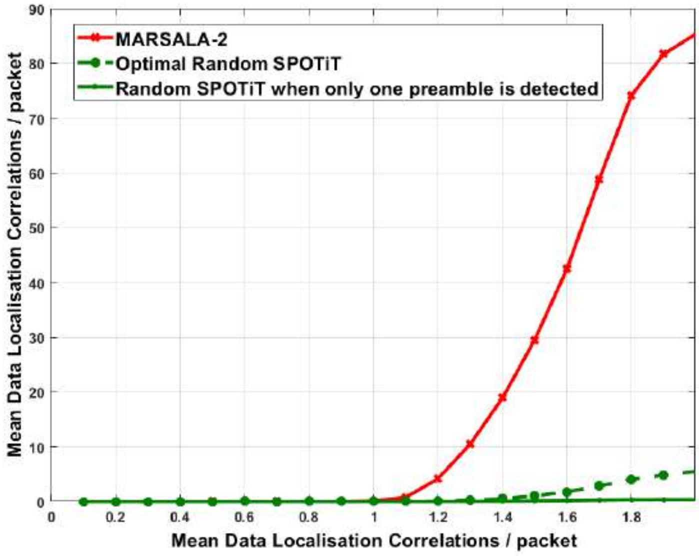


Figure 3.7: Complexity of MARSALA-2 vs R-SPOTiT in terms of localization correlations, $E_S/N_0 = 10$ dB, 100 slots per frame, QPSK modulation, Turbo coding of rate 1/3 and equipowered packets of 100 bits.

correlations depends on the MAC load and thus on the potential collided packets with the same preamble on the same slot. However, CRDSA process, allowing to decode a certain number of packets, reduces this number. Especially since once a packet is decoded in SPOTiT, CRDSA is unlocked and can therefore attempt to decode other packets. The probability of having at least more than one packet with the same preamble on the same slot, before and after the first CRDSA decoding is illustrated in Figure 3.8 with frames of 100 slots. In low MAC loads, and after one operation, CRDSA decoding considerably reduces $N_{pot}^{Ref}(p)$. It joins progressively the probability of occurrence of $N_{pot}^{Ref}(p)$ in high loads until the throughput collapses around 1.7 bits/symbol. As a result the number of data correlations when at least one preamble is detected is insignificant compared to MARSALA that has a mean of 85 data correlations at a load of 2 bits/symbol. In other words, localization complexity is reduced by a factor of 283.

R-SPOTiT can reach the same performance as MARSALA with extra data localization correlations in the case where none of the replicas' preambles are detected (optimal R-SPOTiT). As a matter of fact, on each slot, potential undetected preambles and all possible packets using these preambles are exploited along with the previous decoding result. Data localization correlations are then performed over the second replicas' positions of all potentially collided

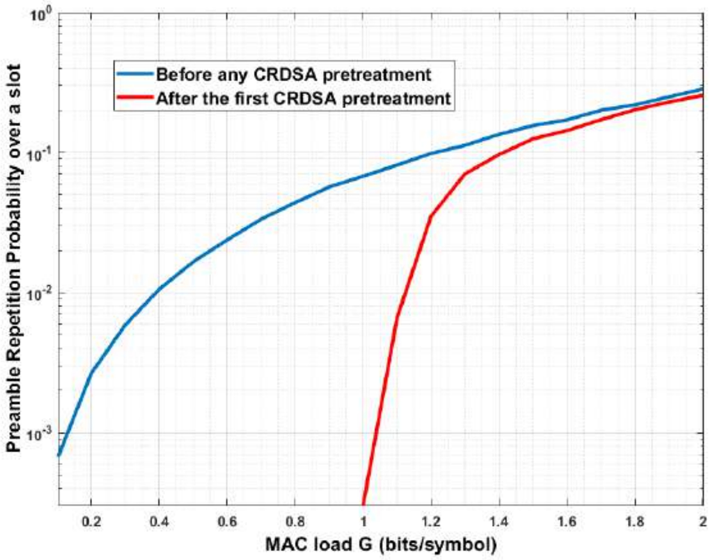


Figure 3.8: Probability of having at least more than one packet with the same preamble on the same time slot before and after CRDSA preprocessing, 100 slots per frame.

packets having the same preamble on the reference time slot. In this case, performance of R-SPOTiT remains the same in terms of PLR and throughput as in MARSALA-2. This is true regardless of the number of users and how replicas are placed on the frame. Indeed, a static arrangement based on fixed seeds or a dynamic ones is the same (see Figure 9.4 and Figure 9.3). This means that the probability of having repetitive loops on successive frames remains very low. Data correlations in this case attain a value of 5.5 while MARSALA reaches a value of 85. This means that the localization complexity is reduced by a factor of 15.5 approximately.

3.6 Summary & Conclusion

We presented in this chapter a novel RA technique called R-SPOTiT, an alternative solution to MARSALA, which is less complex in terms of data correlations required to localize replicas. We have seen that with a complete random processing using PRNG static or dynamic seeds to choose time slot positions and a preamble, it is possible to the receiver to have a prior knowledge on the potential frame composition. This includes replicas' positions and the preamble used by each user. However, the receiver is not aware of whom among all the potential users have their data transmitted on the analyzed frame. Therefore, the pseudo-orthogonal property of preambles is used to reduce the number of potential users. R-SPOTiT can either rely on the only preamble detection based method to localize packets' replicas or apply data correlations. The latter would be applied over the time slots that have potentially one of the packets' replicas with the same detected preambles. We resort to data correlations only when the first alternative fails. R-SPOTiT offers the same performance in terms of PLR and throughput as MARSALA with less complexity and without any additional signaling information. In the next chapter, the special case of R-SPOTiT with a single preamble will be assessed with a more detailed complexity evaluation along with the multi-preamble case and MARSALA.

Chapter 4

Detailed complexity Analysis for Random SPOTiT and MARSALA

Contents

In the previous chapter, R-SPOTiT mechanism has been introduced. It is described as an alternative solution to MARSALA that requires less complexity to localize packets after CRDSA is blocked. Indeed, R-SPOTiT proved to be efficient as it reaches MARSALA's performance with a lower number of localization correlations. The latter has been assessed previously regarding the decoding of one packet. In the current chapter, the overall complexity on a frame is evaluated. In addition, a single preamble case for R-SPOTiT is considered. Thus, a brief recall of the system parameters is presented in Section 4.1. Then, the extended version of R-SPOTiT is described in Section 4.2, complexity parameters in a single preamble or multi-preamble environment are described in Section 4.3, simulation scenarios with their related complexity are presented in Sections 4.4 and 4.5 respectively, and finally, simulation results are analyzed in Section 4.6. The work presented in this chapter is submitted in [Zam+18a].

4.1 System parameters overview

In this section, an overall recall of the system parameters presented in Chapter 3 is given. These parameters that characterize the structure of the signal to be transmitted, the frame, and the channel are the same that are used for the simulations presented in Section 4.6. Moreover, a flexibility regarding the number of preambles is added to R-SPOTiT in this chapter.

We put our focus in this dissertation and work on the return link of a satellite communication system. Thus, a plurality of users transmit their information over the same frequency within a frame of N_{S} time slots, on an AWGN channel. An E_{S}/N_0 equal to 10 dB is considered. The satellite can be a bent pipe that relays information to a gateway or it can be the main receiver if it is regenerative.

At each frame, two replicas ($N_{\mathrm{R}} = 2$) of transmitted packets are sent over different time slots. A packet is composed of a preamble which can be unique or from a selection of N_{P} Gold pseudo-orthogonal codes [Gol67], a postamble, randomly distributed pilot fragments for potential synchronization errors, and a payload. The latter is a set of 150 symbols generated after Turbo coding of rate 1/3 and QPSK modulation of a 100 bits fixed-length binary information. At reception, multiple iterations are made with CRDSA decoding process that applies SIC. It analyzes the frame, slot by slot, to look for replicas free from collision, decode them, reconstruct the other replica and then suppress both of them. This process is performed iteratively until no more packets are on the frame or until it is blocked. If CRDSA can no more retrieve packets, R-SPOTiT or MARSALA will take over the decoding process. In other words, the general idea of the considered system is to establish a two steps procedure for packets decoding at reception that come from a multi-access transmission channel. The first step being CRDSA until it reaches a deadlock, the second one is to unlock the system with one of the methods R-SPOTiT or MARSALA. We recall that MARSALA proceeds to packet decoding in two steps; the first one is the localization process using correlations, the second one is the combination of burst replicas. In other words, localization correlations are performed between an arbitrary time slot of reference (RTS) and all the other slots on the frame. This should allow to point to the positions of packets' replicas that are collided on the RTS. Once the localization is complete, replicas of the same packet are coherently added together in order to have a higher probability of decoding thanks to an SNIR value that is potentially higher than before combination.

4.2 Extension of R-SPOTiT

R-SPOTiT has been described in Section 3.3 of the previous chapter; yet a complete definition will be extended here to the case of a single preamble usage.

Indeed, a PRNG used at both the transmitter and the receiver, that has identification information as a seed and time slot positions with their associated preamble (if applied) as an output, will allow them to generate the same information. The input seed of the PNRG can be static, using for instance only the HID of each user, determined by the receiver thanks to the logon phase; or it can be dynamic if it exploits the frame ID (referring to F_{ID}) in addition to the HID. In the latter case, the time slot positions of each user varies from one frame to another, which avoids a potential continuous loop. In fact, unsolvable loops can be created in some applications where several users generate the same positions, and they transmit successively on the same frames.

On the one hand, each terminal will individually select, using the seed of the PRNG, the replicas' positions of its packet and the preamble to use if multiple codes are adopted. On the other hand, the receiver, knowing all users' HIDs and the F_{ID}, will be able to construct an information table that includes all possible positions in a single preamble case thanks to the same seed and PRNG. In a multi-preamble environment, the preamble choice for each user attached to the gateway will be associated to the time slot positions of the packet's replicas on the information table (see Figure 4.1).

During the localization process, R-SPOTiT looks for replicas of collided packets on a randomly chosen time slot of reference using data correlation operations. The latter is performed, in case of a single preamble, over the second replicas' positions of all potential packets. For example, with 100 time slots per frame and 2000 users attached to the gateway, there will be an average

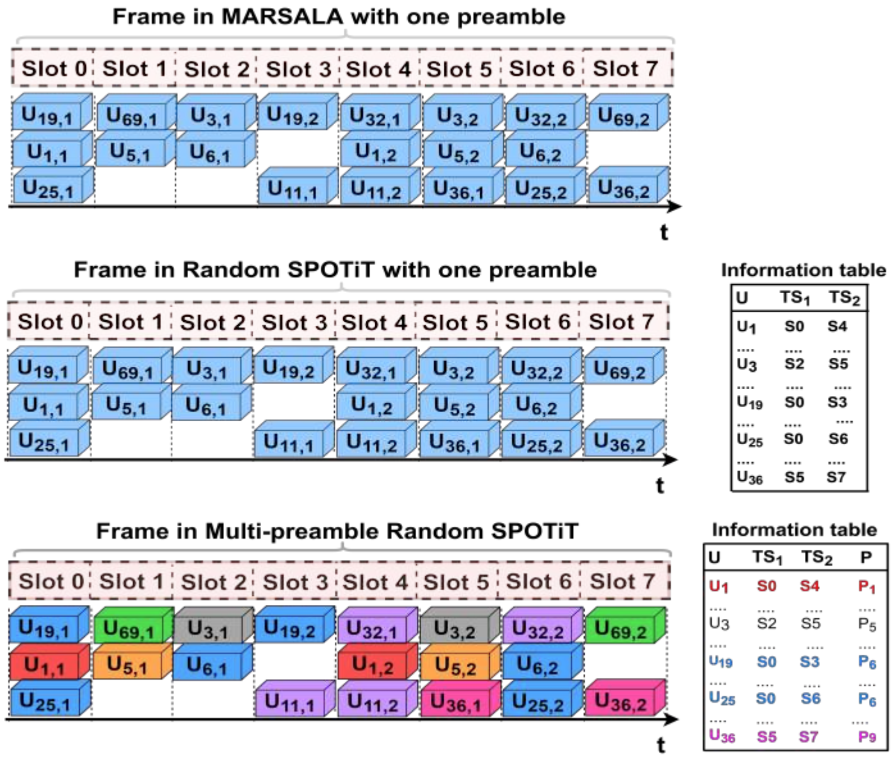


Figure 4.1: Frame structure example at reception.

of 40 correlations needed to decode a packet, while MARSALA requires 99 correlations. Furthermore, if multiple preambles are used, correlation are made over the second replicas' positions of all potential packets using the same detected preamble on the analyzed time slot of reference. Thus, for the same example mentioned before, when one of the preambles is detected on the analyzed slot, R-SPOTiT will make less than 40 correlations, because among the 40 potential transmitters, only the ones using the same detected preamble will be considered. Moreover, the receiver can also compare its information table and the received frame to see whether signal combination can be performed without localization correlations. For instance, if one of the replicas of a packet represents a unique potential transmitter on its corresponding time slot and its preamble is detected, then the packet's presence can be confirmed. Especially if the other replicas' positions exhibit a correlation peak of the same preamble. The whole R-SPOTiT process accompanied by CRDSA reaches MARSALA's performance in terms of throughput and PLR with an expected smaller overall complexity.

Figure 4.1 displays a frame structure example with the same packet distribution for R-SPOTiT, in a single and multi-preamble case, and for MARSALA. The difference between the two single preamble cases is the supplementary information that R-SPOTiT benefits from due to the information table. Indeed, the receiver is aware of the time slot positions TS_1 and TS_2. For instance, if the RTS is the slot 0, R-SPOTiT with a single preamble makes correlations only on slots 3, 4, and 6. In the multi-preamble case, information about the preamble P that each user U chooses to transmit a packet is also provided. This means that, if the blue preamble is detected on slot 0, which is the RTS, only two correlations are to be performed: on slot 3 and on slot 6, or no correlations at all if the red preamble is detected. At the same time, MARSALA would perform correlations over the $N_\mathrm{S} - 1$ time slots: from slot 1 to slot 7.

4.3 Complexity parameters in R-SPOTiT environment

In our overall localization complexity analysis, we focus on the preamble detection part that is necessary to CRDSA decoding process and on the replicas localization required in R-SPOTiT or MARSALA before signal combination. We recall that CRDSA is applied first until no more packets can be retrieved, then a complementary treatment R-SPOTiT or MARSALA is solicited. Once CRDSA is unlocked, it will be triggered again. This means that the total number of preamble detection correlations, for a given frame, is calculated over all iterations of CRDSA until it is blocked and when it is triggered again after R-SPOTiT or MARSALA process and until this complementary treatment is blocked as well. The whole process is ended under one of the three following conditions; when CRDSA alone has decoded all packets on the frame, when CRDSA plus the complementary treatment have decoded all packets or when they decoded the maximum number of packets before both of them are blocked due to the high level of collisions at high loads.

4.3.1 Preamble detection operations

Preamble detection is performed only at CRDSA. Indeed, the packet decoding process of the latter is attempted when one or more preambles are correctly detected over the analyzed frame. Consequently, CRDSA is blocked when no more preambles are detected or when packets cannot be retrieved even with detected preambles, due to the high level of collision.

We are interested here in the overall number of correlations over a frame until the system reaches a blocking situation. Let us consider Gold code preambles. Parallel correlations are made over the preamble search region to look for correlation peaks. A preamble detection method is proposed below, which is also valid for the single preamble case. Later, a preamble correlation will be expressed in terms of a data correlation.

Preamble correlations

A transmitted replica r of a user u on a given time slot can incur a phase error $\phi_{u,r} \in [0; 2\pi]$ and can be shifted in time with $\tau_u \in [-2T_\mathrm{S}; 2T_\mathrm{S}]$ where T_S is the symbol duration. This means that preamble correlation peak search is performed over four symbols duration ($4T_\mathrm{S}$). Each replica has a Gold code preamble of length 31. It corresponds to the pair of the maximum length pseudo noise sequences (m-sequences) for the shift registers which generate the Gold codes. Thus, we describe the preamble region signal P_T including the guard interval at time instant t as follows:

$$P_\mathrm{T}(t) = \sum_{i=1}^{L} p_i(t+\tau_i)\, e^{j\phi_{i,r}} + G_\mathrm{data}(t)\, e^{j\phi_d} + n(t) \tag{4.1}$$

where p_i is the i^{th} Gold code among the L collided preambles on the analyzed time slot, G_data is the extra guard data symbols from the region around the preamble location due to potential synchronization errors, and n is the AWGN noise term.

The receiver proceeds to preamble detection by correlating the received preamble region signal with the complex conjugates of the 31 gold codes.

$$R_\mathrm{P}^{l}(\tau) = \int_{0}^{T_\mathrm{search}} P_\mathrm{T}(t) p_l^*(t-\tau)\, dt \tag{4.2}$$

where T_search is the search region over the preamble duration.

R_P will have a peak for each transmitted preamble l at time instant τ_l referring to the autocorrelation function of each collided preamble. The packet decodability and the decision of preamble detection is affected by the number of collided packets. In order to provide with a first approximation, we have made a preliminary study. On the one hand, preamble detection probability is assessed with respect to the number of collided packets. On the other hand, the decoding probability is analyzed with respect to the number of interfering packets. Considering preambles as Gaussian random variables, the square modulus of correlation $|R_\mathrm{P}^{l}|^2$

can be represented as a Chi-square random variable with two degrees of freedom. A preamble l is decided to be detected if $|R_{\mathrm{P}}^{l}|^2$ is above a predetermined threshold. The detection threshold T_{h} can then be derived using the false alarm probability P_{FA}.

$$\begin{aligned} P_{\mathrm{FA}}(T_{\mathrm{h}}) &= P(|R_{\mathrm{P}}^{l}|^2 > T_{\mathrm{h}} \mid H_0) \\ &= 1 - P(R_\sigma < \frac{T_{\mathrm{h}}}{\sigma^2} \mid R_\sigma \sim \chi_2^2) = \exp\left(-\frac{T_{\mathrm{h}}}{2\sigma^2}\right) \end{aligned} \tag{4.3}$$

where H_0 is the hypothesis that the preamble l is absent, $R_\sigma = \frac{|R_{\mathrm{P}}^{l}|^2}{\sigma^2}$, and σ^2 is the noise power plus the interference power that is related to the collided Gold code sequences, over the analyzed time slot. It follows that:

$$T_{\mathrm{h}} = -2\sigma^2 \ln P_{\mathrm{FA}} \tag{4.4}$$

Note that this threshold is proper to the analyzed time slot as the number of interfering packets varies from one slot to another (which changes σ^2). Thus, a preamble is correctly detected on a given slot when the correlation of P_{T} with the right complex conjugate of the transmitted Gold code reaches a maximum that is above the detection threshold T_{h}. We start with the highest correlation peak. Once detected, and after demodulation and SIC, the gold code is suppressed, along with the whole packet, from the time slot. The preamble region signal can be analyzed again to look for the next highest correlation peak. The detection threshold at the next iteration depends then on the new level of σ^2 (as an interference is suppressed).

In this case of multi-preamble R-SPOTiT, the decision regarding a packet's presence (prior to the combination process, the demodulation and the decoding), is explained in Section 4.4.3, and summarized in Figure 4.2. The packet candidates refer to all the potentially collided packets with one of the replicas of a given packet. An additional condition regarding the preamble used by these candidates is set when one of the replicas' preamble is detected.

In Figure 4.3, with a false alarm probability set to 10^{-3} and $N_{\mathrm{R}} = 2$, the detection probability describes two cases. The first one is over one time slot; it concerns only one of the replicas of the packet of interest. The second one takes into consideration both replicas of a user when at least one of them is detected (preamble detection for a user) on its position or when both of them should imperatively be detected (optimal preamble detection for a user). The latter

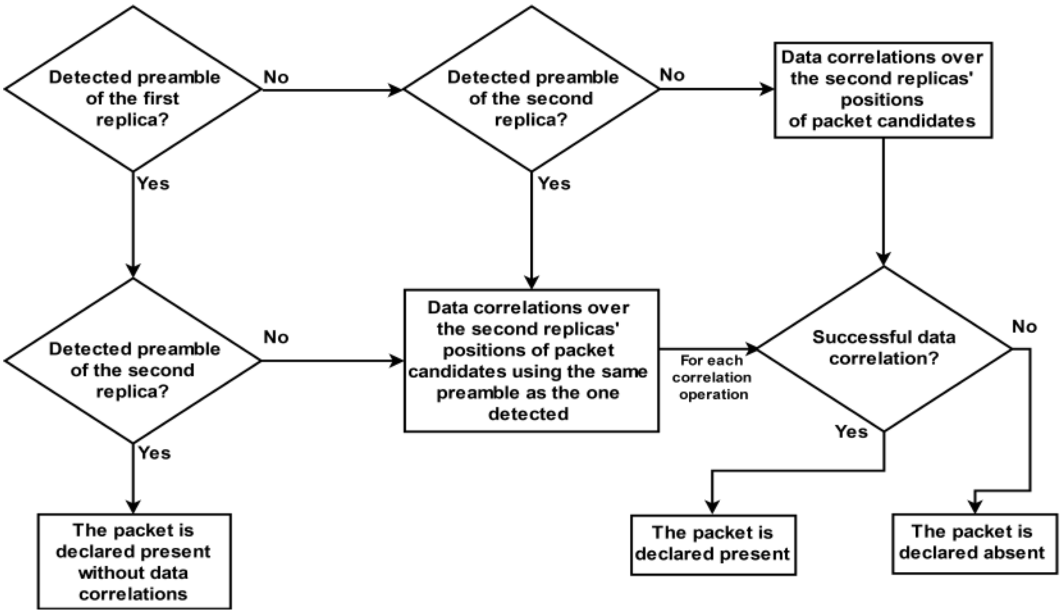


Figure 4.2: The way to detect a packet's presence according to the preamble detection and the data correlations.

case is computed when both replicas are interfered by the same number of packets. Any other scenario regarding the number of interfering packets each replica incurs can be derived from the first case. As a matter of fact this would be equal to $P_1^{int} + P_2^{int} - P_1^{int} P_2^{int}$, where P_1^{int} is the probability that the first replica is detected on its time slot position, P_2^{int} is the probability that the second replica is detected on its time slot position, and int represents the number of interfering packets which can be different from one time slot to another.

From R-SPOTiT and MARSALA perspective, signal combination enhances the SNIR value due to the higher power of the signal of interest. With the assumptions that interference is approximated to AWGN (investigation and justification is provided in [dD14] Appendix B), we can consider that the Packet Error Rate (PER) curve is associated to different values of SNIR defined in (4.5). The interpolation is proper to the chosen MODCOD. In our case we use QPSK modulation with Turbo coding of rate 1/3; this means the MODCODs are of 150 symbols. Thus, the decoding probability can be calculated for different numbers of interfering packets.

$$\mathrm{SNIR}(u, r) = \frac{E_\mathrm{S}/N_0}{E_\mathrm{S}/N_0 I_{u,r} + 1} \tag{4.5}$$

where $I_{u,r}$ is the number of interfering packets on the analyzed slot with replica r and user u.

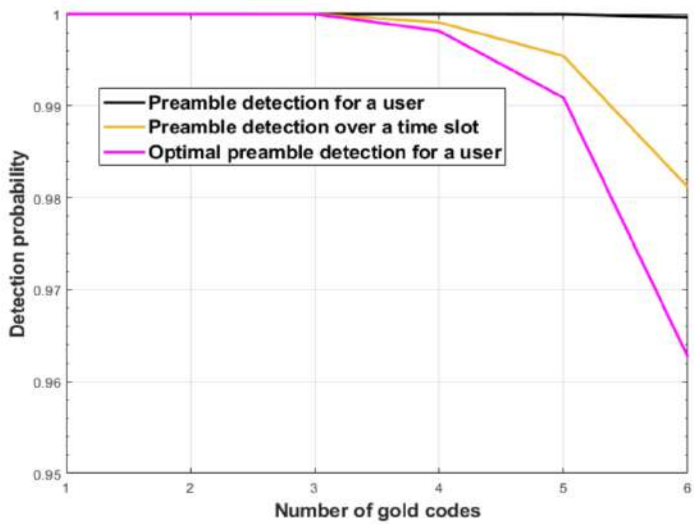


Figure 4.3: Users detection probability of gold code preambles of length 31, over an AWGN channel of an $E_S/N_0 = 10\ dB$.

If R-SPOTiT or MARSALA is applied, considering two replicas per packet, signal combination, based on a summation between the first and second replica, will quadruple the packet's power.

$$\text{SNIR}(u) = \frac{4E_S/N_0}{E_S/N_0 I_u + 2} \tag{4.6}$$

where $I_u = I_{u,r_1} + I_{u,r_2}$; with I_{u,r_1} the number of interfering packets with the first replica r_1 and user u, and I_{u,r_2} is the number of interference over the second replica's position r_2 of the same user u.

The generic expression that is applicable to all N_R is then:

$$\text{SNIR}(u) = \frac{{N_R}^2 E_S/N_0}{E_S/N_0 I_u + N_R} \tag{4.7}$$

I_u in this case is the interference rate that is calculated over all slots where replicas of the same packet are present. Considering fixed values of N_R and E_S/N_0, Equation (4.7) may be expressed otherwise according to 4.8.

$$\text{SNIR}(u) = f(I_u) \tag{4.8}$$

Indeed, the only variable is the random number of interference that depends on the channel load. We can now associate PER values of SNIR to I_u (see equation (4.9)) and derive the decoding probability $P_{\mathrm{D}}(u) = 1 - \mathrm{PER}$.

$$PER(u) = g(\mathrm{SNIR}(u)) = g \circ f(I_u) \tag{4.9}$$

Obviously, g is the interpolation of the PER curve (Figure 4.5) resulted from QPSK modulation and the 1/3 Turbo coding over an $AWGN$ channel. Figure 4.4 displays, for R-SPOTiT

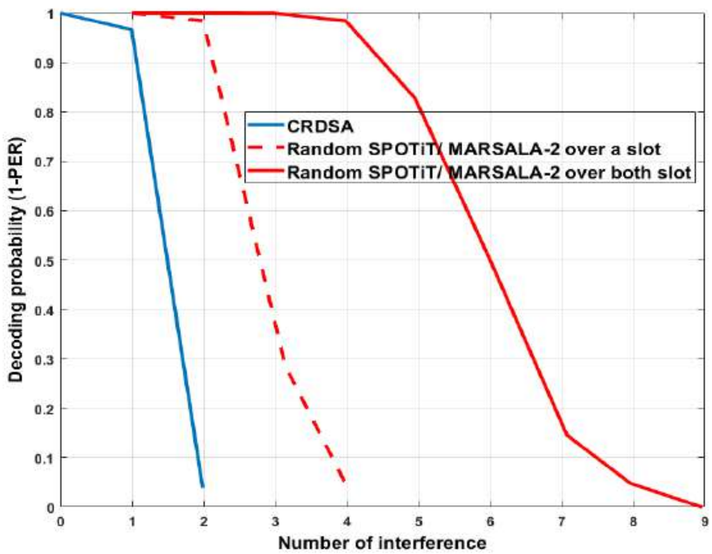


Figure 4.4: Decoding probability w.r.t. the number of interfering packets of 100 bits, QPSK modulation, Turbo coding of rate 1/3, $E_{\mathrm{S}}/N_0 = 10$ dB.

and MARSALA with two replicas, the decoding probability P_{D} with respect to the number of interfering packets. An interference rate over both positions represents I_u while a normalized value corresponds to $\frac{I_u}{2}$ which is the mean number of interference over one position among the two replicas' time slots. We consider no loops are created. If we take 0.98 as an acceptable value of decoding probability, we can see that with CRDSA alone, an equivalent interference length of one packet has a chance to be decoded. Meanwhile, Random SPOTiT and MARSALA with two replicas have an equivalent interference length of $I_{(u)} \approx 4$ packets over the two replicas' slots; or a mean of 2 packets per time slot. This means there are six total replicas over both slots including the replicas of the packet of interest. These six replicas can only refer to one scenario: two interfering packets with the packet of interest on each time slot. The possibility of having one interference on a time slot and three on the other one is dismissed because CRDSA can resolve a single interference scenario. We can see, according to Figure 4.3, that when both replicas are collided with two other packets (three Gold codes on each slot), the user preamble detection probability is equal to 1. This is valid in both scenarios when only one of the replicas' preamble peaks is necessary for detection or when both of them should be detected. It means that the preamble detection probability in this case matches the packet decoding probability we fixed at 0.98.

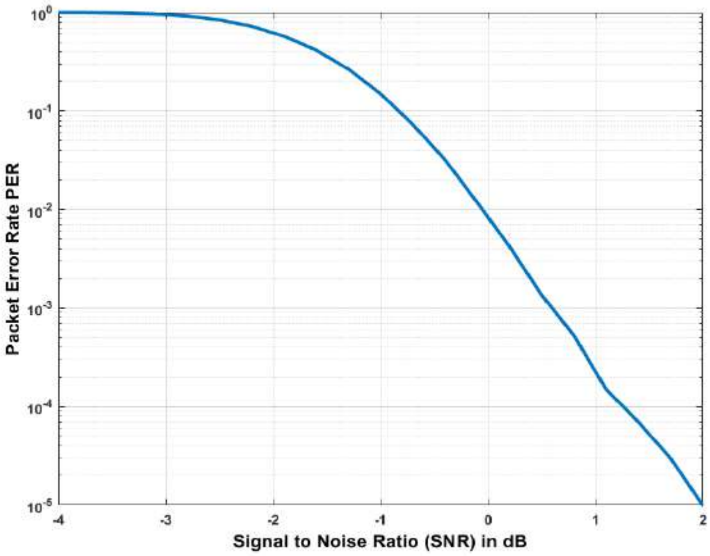


Figure 4.5: Packet Error Rate (PER) with 100 bits of useful information, QPSK modulation, Turbo coding of rate 1/3.

Method for preamble detection

A Two steps process is defined here for preamble detection; first, a coarse tracking is made considering a small number of samples per symbol, then a fine tracking is performed over the strongest symbol with more samples. In other words, the coarse step presents N_{C} correlations for one preamble detection and the fine one holds out N_{F} correlations for the same preamble. Hence, a total of $N_{\mathrm{bt}} = N_{\mathrm{C}} + N_{\mathrm{F}}$ correlations are considered necessary for every preamble to spot the closest location to its position, if transmitted, or the position itself. For example, if the coarse step presents two samples per symbol and the fine step eight samples, N_{bt} will be equal to 16 basic preamble correlations (two correlations over each of the four symbols during the coarse step plus eight correlations over the strongest one during the fine tracking). Figure 4.6 gives a general overview of how preamble detection is made during CRDSA. The dashed parts show the information that the receiver exploits, such as the information table of R-SPOTiT and the result of packet decoding.

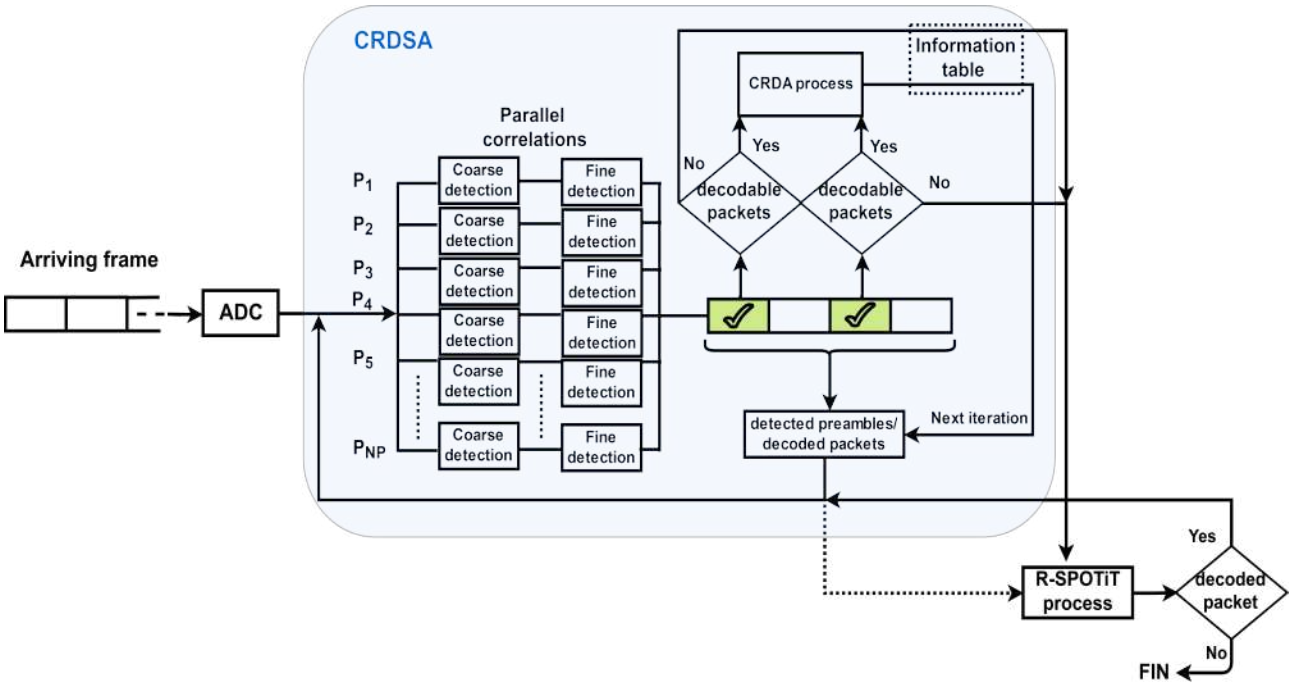


Figure 4.6: General overview of the receiver.

4.3.2 Data localization correlations

The data localization correlations are defined according to which complementary treatment comes to unlock CRDSA. As a matter of fact, MARSALA, not taking into account any preamble detection result from CRDSA, will proceed with the same coarse and fine tracking operations as for the preamble detection. This means that N_{bt} data correlations are included in one localization operation regarding one packet. Indeed, as the same preamble is used for all users, the beginning of a packet is almost impossible to determine in high loads, which is also the case for R-SPOTiT with a single preamble. However, R-SPOTiT with multiple preambles and its information table that considers the preamble detection result from the previous CRDSA iteration, requires only one data correlation for a localization operation. This is obviously due to the fact that the beginning of a packet is already determined thanks to its detected preamble at CRDSA.

Also, the process of randomly choosing a reference time slot in the complementary treatment is performed repetitively until a packet is decoded, which will unlock CRDSA.

When we take a closer look at the two correlations that are considered in our complexity analysis, we can figure out that the only difference between them is that the preamble correlation is shorter than the data one. Indeed, a correlation over a preamble of 31 symbols can be perceived as a fifth correlation over the 150 data symbols.

4.3.3 Total correlations per frame

To sum up, the total number of correlations C_T per frame includes the preamble detection operations C_{P} and the data localization operations C_{D}. C_{P} is considered for both coarse and fine tracking over all CRDSA iterations, before and after the complementary treatment, and until the whole system is blocked. C_{D} is performed by R-SPOTiT or MARSALA to retrigger CRDSA each time it is blocked until the whole system reaches a deadlock. The total number of correlations over a frame C_T is described below:

$$C_T = \sum_{\delta=1}^{\Delta} \left(\sum_{it=1}^{N_{\mathrm{it}}} C_{\mathrm{P}}(\delta, it) + \sum_{\lambda=1}^{\Lambda(\delta)} C_{\mathrm{D}}(\delta, \lambda) \right) \tag{4.10}$$

With δ the frame analysis index. Δ is the maximum value of δ that is reached when the whole system is blocked. Its value can vary from one frame to another. N_{it} is the number of CRDSA iterations. λ is the index for randomly choosing a reference time slot by the complementary treatment that can be more than once. $\Lambda(\delta)$ is the maximum number of reference time slot that is reached during a given frame analysis index δ.

When $\delta = 1$ the frame is analyzed by CRDSA with N_{it} iterations in addition to a complementary treatment (R-SPOTiT or MARSALA) that unlocks the latter if not all packets are decoded. In this case, decoding one packet is enough to unlock the system. To do that, $\Lambda(1)$ attempts are performed. In MARSALA, $\lambda \in [0; N_{\mathrm{S}}]$ which is the number reference time slots. In R-SPOTiT, λ can include several processes on one reference time slot, before selecting another one, according to the number of detected preambles. When CRDSA is retriggered, the frame is analyzed again and $\delta = 2$. When $\delta = \Delta$ given that $\Delta \geq 1$ (see Table 4.1 of Section 4.6), the whole system is solved if all packets are decoded or blocked when neither CRDSA nor R-SPOTiT or MARSALA can decode new packets. Basically, a one frame analysis complexity consists of preamble detection operations C_{P} at CRDSA and data localization operations C_{D} at R-SPOTiT or MARSALA. Also, in the rest of the chapter, we consider a basic correlation C_{b} that corresponds to the data correlation as a unit for complexity computation. Thus, a

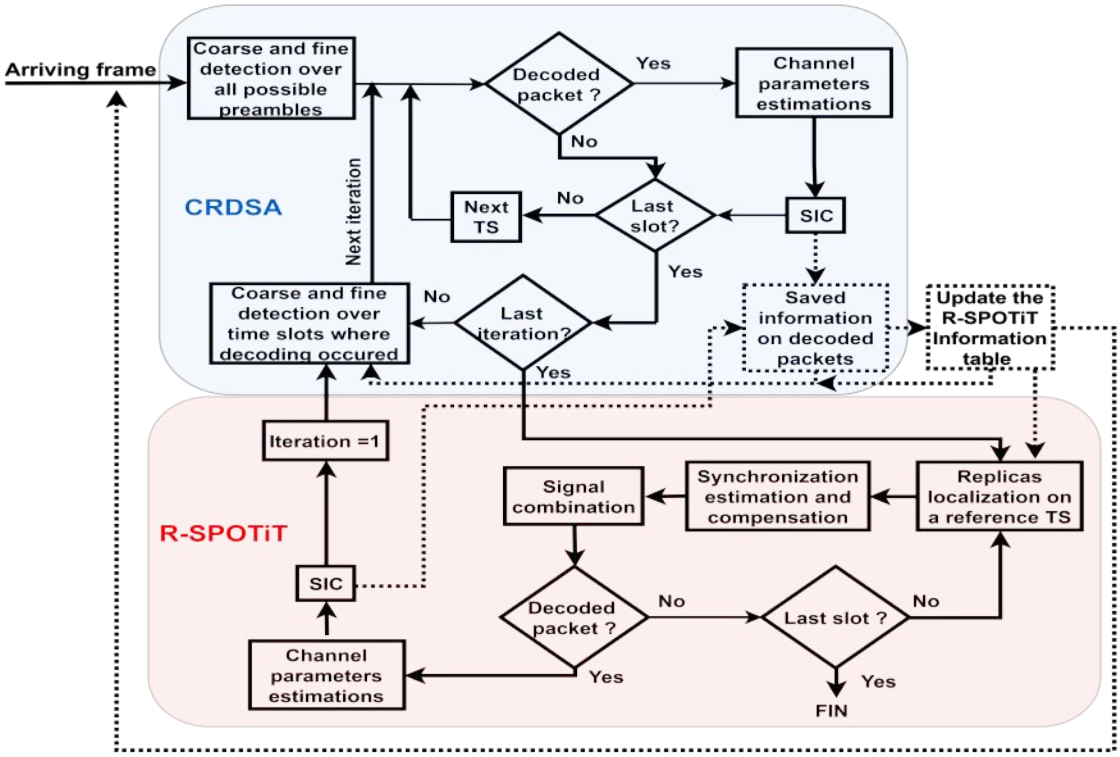


Figure 4.7: Detailed overview of the receiver.

basic preamble correlation C_{bp} will be expressed in regard to C_{b} such as $C_{\text{bp}} = \frac{C_{\text{b}}}{R}$. R is the ratio between the data and the preamble lengths ($R = 5$ if 150 symbols are considered for the data with a preamble of 31 symbols). A detailed overview of the receiver applying R-SPOTiT with CRDSA is provided in Figure 4.7. We recall that the dashed parts show the information that the receiver exploits (information table of R-SPOTiT and packet decoding result).

4.4 Simulation scenarios and general algorithms

In this section will be described each of the scenarios regarding CRDSA complemented by R-SPOTiT or by MARSALA. A single preamble is always considered for MARSALA. The complexity of R-SPOTiT will be assessed for a single preamble case as well as with multiple preambles. The scenarios below describe when the preamble detection correlations and data localization operations take place.

4.4.1 MARSALA with one preamble

For the sake of comparison with R-SPOTiT, we consider here a single preamble with the same length as the multiple pseudo-orthogonal preambles in R-SPOTiT. At the first frame analysis index ($\delta = 1$) and first iteration of CRDSA ($it = 1$), a preamble detection operation is performed over each slot of the frame, thus the number of preamble correlations is equal to $N_{\mathrm{S}} N_{\mathrm{bt}} C_{\mathrm{bp}}$. At an n^{th} iteration, preamble detection is only applied to $N_{\mathrm{crdsa}}(\delta, n-1)$ time slots where successful decoding took place at the previous iteration ($n-1$ with $\delta = 1$). Hence the number of preamble correlations is equal to $N_{\mathrm{crdsa}}(1, n-1) N_{\mathrm{bt}} C_{\mathrm{bp}}$.

When the frame analysis index is higher than 1 ($\delta > 1$), it means that MARSALA have been applied. At the first iteration of CRDSA, preamble detection is performed over the time slots where successful decoding of MARSALA happened at the previous frame analysis index. As only one packet decoding is required for MARSALA to trigger CRDSA again, the positions of the N_{R} replicas of the decoded packet are considered. At an n^{th} iteration, the preamble detection still takes into account CRDSA decoding of the previous iteration.

During MARSALA process, data localization operations are accomplished between a randomly chosen reference position and the rest of the time slots on the frame. If a packet is decoded, CRDSA is retriggered again, otherwise, another reference time slot is chosen and same operations are held one more time. This process is repeated $\Lambda(\delta)$ times until a packet is decoded, thus the number of basic data correlations is equal to $\Lambda(\delta)(N_{\mathrm{S}} - 1) N_{\mathrm{bt}} C_{\mathrm{b}}$. The maximum value of $\Lambda(\delta)$ can be reached without being able to decode any packet; in this case, the whole system is blocked.

4.4.2 Random SPOTiT with one preamble: scenario 1

In this scheme, a single preamble is used for R-SPOTiT in the same way as in MARSALA. This means that similarly to the latter, with $\delta = 1$ and $it = 1$, there are $N_{\mathrm{S}} N_{\mathrm{bt}} C_{\mathrm{bp}}$ correlations for preamble detection. Correspondingly, at an n^{th} iteration, preamble detection is only applied to $N_{\mathrm{crdsa}}(1, n-1)$ time slots where successful decoding took place at the previous iteration, hence the number of preamble correlations is equal to $N_{\mathrm{crdsa}}(1, n-1) N_{\mathrm{bt}} C_{\mathrm{bp}}$.

After R-SPOTiT intervenes when CRDSA is blocked ($\delta > 1$), the preamble detection of the first iteration at any index δ is performed over the time slots where successful decoding of one packet by R-SPOTiT happened at $\delta - 1$. An n^{th} iteration will still consider the positions of the packets decoded by CRDSA at the previous iteration.

During R-SPOTiT processing, data localization operations are held between a reference time slot and the second replicas' positions of all potentially collided packets on the RTS. The latter is derived from R-SPOTiT information table that is updated after each decoding. This means that when replicas of a packet are decoded on their positions, they will be suppressed from the potential packets in collision on their respective time slots. Similarly to MARSALA, if a packet is decoded, CRDSA will be launched again, if not, another reference time slot is chosen ($\Lambda(\delta) > 1$) and same operations are held.

4.4.3 Random SPOTiT with multiple preambles: scenario 2

One of the characteristics of R-SPOTiT is the use of pseudo-orthogonal preambles to reduce data correlations compared to MARSALA. Nevertheless, if MARSALA is operational with one preamble, the CRDSA part of the algorithm will be more complex with R-SPOTiT than with MARSALA. This is due to the parallel preamble detection operations. It becomes then important to assess the overall complexity of R-SPOTiT including preamble detection along with data localization and compare it with a single preamble case of MARSALA.

In this scenario, $N_{\mathrm{P}} \in [2; N_{PT}]$ pseudo-orthogonal Gold codes are considered where N_{PT} is the total number of pseudo-orthogonal Gold preambles with a given length. At the frame analysis index $\delta = 1$ and first iteration of CRDSA $it = 1$, $N_{\mathrm{pp}}(s)$ parallel preamble detection operations are performed on each time slot s, where $N_{\mathrm{pp}}(s) \in [1; N_{\mathrm{P}}]$. The value of $N_{\mathrm{pp}}(s)$ is derived from R-SPOTiT information table; it corresponds to the number of potential preambles that could be transmitted on a given slot s among the N_{P} possible preambles. Therefore, the number of preamble correlations is equal to $\sum_{s=1}^{N_{\mathrm{S}}} N_{\mathrm{pp}}(s) N_{\mathrm{bt}} C_{\mathrm{bp}}$. After each decoding, the information table of R-SPOTiT is updated. Thus, at an n^{th} iteration, the preamble detection process will take into consideration the previous decoding result ($N_{\mathrm{crdsa}}(\delta, n-1)$, as noted before) along with the updated information table $\left(\sum_{s=i_1}^{i_{N_{\mathrm{crdsa}}(\delta,n-1)}} N_{\mathrm{pp}}^{\mathrm{u}}(s) N_{\mathrm{bt}} C_{\mathrm{bp}}\right)$ where $N_{\mathrm{pp}}^{\mathrm{u}}(s)$ is simply the updated $N_{\mathrm{pp}}(s)$ after decoding occurred on slot s ($N_{\mathrm{pp}}^{\mathrm{u}}(s) < N_{\mathrm{pp}}(s)$). $i_{N_{crdsa}}(\delta, n-1)$ is the time slot index of the last replica suppressed by CRDSA at the previous iteration. When R-SPOTiT has been applied, with a frame analysis index exceeding one ($\delta > 1$), the preamble detection process at the first iteration of CRDSA should consider the time slot positions of one packet (N_{R}) where decoding happened with R-SPOTiT $\left(\sum_{s=i_1}^{i_{N_{\mathrm{R}}}(\delta-1)} N_{\mathrm{pp}}^{\mathrm{u}}(s) N_{\mathrm{bt}} C_{\mathrm{bp}}\right)$. $i_{N_R}(\delta-1)$ is the time slot index of the last replica belonging to the packet decoded by R-SPOTiT; which unlocked CRDSA. At the next iterations of CRDSA (any n), the previous $(n-1)$ one is always taken into account along with the updated information table.

Considering only packet localization complexity, three cases can be described during R-SPOTiT process. First, if both preambles of a given packet are correctly detected at the last CRDSA iteration, the packet will be considered localized and thus no data localization operations are needed. Secondly, when one of the two preambles is detected on a reference time slot, data localization operations are performed over the second replicas' positions of potentially collided packets having the same detected preamble. Considering that the replicas of a given user are synchronized at the frame level, a localization correlation is made only once due to the fact that we are supposed to know the packet time shift after its preamble is detected.

Thirdly, when there are no more detected preambles, the single preamble case can be copied. Indeed, preambles can be ignored, thus the information table will be exploited only regarding the packets' positions. A localization correlation in this case is performed N_{bt} times as no information about the packet's time shift is available. This way, R-SPOTiT with any number of preambles will have exactly the same performance as MARSALA.

4.5 Scenario based complexity computation

What has been expressed in (9.1) is the overall frame complexity during the whole CRDSA/R-SPOTiT or CRDSA/MARSALA process. This notation is adopted in this section to differentiate the two main terms of correlations; i.e. preamble correlations at CRDSA and data correlations during the complementary treatment. Each term will be detailed here according to the scenarios explained in Section 4.4. Two main cases are constructed depending on whether $\delta = 1$ or when $\delta > 1$. Moreover, each case will have two sub cases for CRDSA; when $it = 1$ and when $it > 1$ for the preamble detection.

On the one hand, C_{P} varies, in CRDSA/MARSALA, from one iteration n to another, with respect to δ, and according to $N_{\mathrm{scen}}(\delta, n)$: the total number of operations. It is equal to N_{S}, $N_{\mathrm{crdsa}}(\delta, n-1)$ or N_{R} which are respectively the number of slots, the number of slots where packets have been decoded at the previous iteration with CRDSA, and the number of slots where MARSALA decoded a packet at the previous frame analysis index. On the other hand, C_{P} also varies, in CRDSA/R-SPOTiT, with respect to $N_{\mathrm{scen}}(\delta, n)$. It is equal in this case to N_{S}, $N_{\mathrm{crdsa}}(\delta, n-1)$ or to N_{R} for R-SPOTiT with one preamble or to $\sum_{\mathrm{s}} N^{\mathrm{u}}_{\mathrm{pp}}(s)$ for R-SPOTiT with N_{P} preambles. N_{R} stands here for the time slots where packets have been decoded by R-SPOTiT with one preamble at the previous frame analysis index; $\sum_{\mathrm{s}} N^{\mathrm{u}}_{\mathrm{pp}}(s)$ is the number of possible preambles, considering the updated information table, on the time slots where packets have been decoded at the previous iteration of CRDSA or at the last frame analysis index. We recall that when a packet is decoded at a previous iteration or a past frame analysis index, it will be suppressed from the potential packet candidates in the information table. The latter is also being exploited at each analyzed time slot to avoid having parallel preamble detection operations over the whole number of used preambles N_P. In (4.11), (4.12), (7.2) and (4.14), C_{P} is conventionally expressed, in terms of basic data correlations C_b, according to each case δ and it described at the previous section and for each of the scenarios: MARSALA, R-SPOTiT with one preamble (R-SPOTiT(1)) and R-SPOTiT with N_P preambles (R-SPOTiT(N_P)).

- C_{P} when $\delta = 1$ and $it = 1$

$$C_{\mathrm{P}}(1,1) = N_{\mathrm{scen}}(1,1) N_{\mathrm{bt}} C_{\mathrm{bp}} = \begin{cases} N_{\mathrm{S}} N_{\mathrm{bt}} \frac{C_{\mathrm{b}}}{R} & \text{if MARSALA} \\ N_{\mathrm{S}} N_{\mathrm{bt}} \frac{C_{\mathrm{b}}}{R} & \text{if R-SPOTiT(1)} \\ \sum_{s=1}^{N_{\mathrm{S}}} N_{\mathrm{pp}}(s) N_{\mathrm{bt}} \frac{C_{\mathrm{b}}}{R} & \text{if R-SPOTiT}(N_{\mathrm{P}}) \end{cases} \tag{4.11}$$

- C_{P} when $\delta = 1$ and $it = n$

$$C_{\mathrm{P}}(1,n) = N_{\mathrm{scen}}(1,n) N_{\mathrm{bt}} C_{\mathrm{bp}} = \begin{cases} N_{\mathrm{crdsa}}(1,n-1) N_{\mathrm{bt}} \frac{C_{\mathrm{b}}}{R} & \text{if MARSALA} \\ N_{\mathrm{crdsa}}(1,n-1) N_{\mathrm{bt}} \frac{C_{\mathrm{b}}}{R} & \text{if R-SPOTiT(1)} \\ \sum_{s=i_1}^{i_{N_{\mathrm{crdsa}}(1,n-1)}} N^{\mathrm{u}}_{\mathrm{pp}}(s) N_{\mathrm{bt}} \frac{C_{\mathrm{b}}}{R} & \text{if R-SPOTiT}(N_{\mathrm{P}}) \end{cases} \tag{4.12}$$

- C_{P} when $\delta > 1$ and $it = 1$

$$C_{\mathrm{P}}(\delta,1) = N_{\mathrm{scen}}(\delta-1,1) N_{\mathrm{bt}} C_{\mathrm{bp}} = \begin{cases} N_{\mathrm{R}} N_{\mathrm{bt}} \frac{C_{\mathrm{b}}}{R} & \text{if MARSALA} \\ N_{\mathrm{R}} N_{\mathrm{bt}} \frac{C_{\mathrm{b}}}{R} & \text{if R-SPOTiT(1)} \\ \sum_{s=i_1}^{i_{N_{\mathrm{R}}}(\delta-1)} N^{\mathrm{u}}_{\mathrm{pp}}(s) N_{\mathrm{bt}} \frac{C_{\mathrm{b}}}{R} & \text{if R-SPOTiT}(N_{\mathrm{P}}) \end{cases} \tag{4.13}$$

- C_P when $\delta > 1$ and $it = n$

$$C_\text{P}(\delta, n) = N_\text{scen}(\delta, n) N_\text{bt} C_\text{bp} = \begin{cases} N_\text{crdsa}(\delta, n-1) N_\text{bt} \frac{C_\text{b}}{R} & \text{if MARSALA} \\ N_\text{crdsa}(\delta, n-1) N_\text{bt} \frac{C_\text{b}}{R} & \text{if R-SPOTiT(1)} \\ \sum_{s=i_1}^{i_{N_\text{crdsa}}(\delta,n-1)} N_\text{pp}^\text{u}(s) N_\text{bt} \frac{C_\text{b}}{R} & \text{if R-SPOTiT}(N_\text{P}) \end{cases} \quad (4.14)$$

The number of data correlations C_D is expressed in (4.15):

- C_D for any value of δ and λ

$$C_\text{D}(k, \lambda) = \begin{cases} (N_\text{S} - 1) N_\text{bt} C_\text{b} & \text{if MARSALA} \\ N_\text{po}^\text{u}(k, \lambda) N_\text{bt} C_\text{b} & \text{if R-SPOTiT(1)} \\ \left(N_\text{pm}^\text{u}(\delta, \lambda) C_\text{b}\right)^{(1-\rho)} \left(N_\text{po}^\text{u}(k, \lambda) N_\text{bt} C_\text{b}\right)^{\rho} & \text{if R-SPOTiT}(N_\text{P}) \end{cases} \quad (4.15)$$

At each frame index δ, if necessary, C_D is performed $\Lambda(\delta)$ times until a packet is successfully decoded. The value of C_D is fixed for MARSALA. However, in R-SPOTiT, it can be softened in the same way as for C_P using the information table. In case of a single preamble, $N_\text{po}^\text{u}(\delta, \lambda)$ represents the number of potential packets in collision on the chosen reference time slot with index λ after the last update of the information table for a given frame analysis index δ. When multiple preambles are used in SPOTiT, C_D is characterized by $N_\text{pm}^\text{u}(\delta, \lambda)$ that is the number of potentially collided packets using the same detected preamble over the reference time slot, derived from the updated information table. This means that at least one preamble is detected on the analyzed slots, therefore, we note the boolean variable $\rho = 0$. When in contrary, there are no detected preambles at any reference time slot ($\rho = 1$), $N_\text{po}^\text{u}(\delta, \lambda)$ is equal, similarly to the single preamble case, to the overall number of potentially collided packets on the chosen reference time slot whatever are their preambles, taking into consideration the updated information table.

4.6 Simulation results

The simulations conducted in this work apply to the scenarios described above. For each one of them, the overall frame complexity including preamble detection and packet localization is assessed with respect to the number of users that is converted to a channel load in bits/symbol.

In scenario 2, the length N_{PT} of the preambles is 31, which are generated using the preferred polynomial pair $\{x^5 + x^2 + 1\ ,\ x^5 + x^4 + x^3 + x^2 + 1\}$, and N_{P} varies in $\{2, 3, 5, 15, 31\}$.

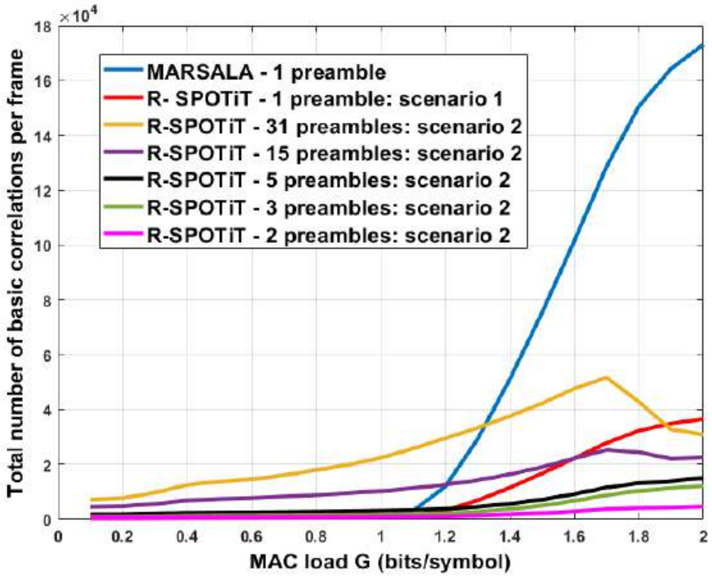


Figure 4.8: Overall frame complexity, with $N_{\mathrm{bt}} = 16$ and $R = 5$ in CRDSA/MARSALA and CRDSA/R-SPOTiT environments

We observe on Figure 9.5 that in case of a single preamble, MARSALA's complexity exceeds the one of R-SPOTiT starting approximatively from a channel load of 1.1 bits/symbol. The difference between the two becomes larger with the increase of the number of transmitters. MARSALA's complexity is, on average, four times higher than the one of R-SPOTiT. At low loads, complexity is negligible because no complementary treatment is necessary. Thus, no heavy data localization operations are performed, only the single preamble detection operation. However, when multiple preambles are used in R-SPOTiT, low loads experience a number of correlations higher than in a single preamble case whereas high loads exhibit up to a certain point, depending on the number of preambles, more complexity than the single preamble case. Then it becomes lower. Actually, with 31 and 15 preambles, the number of basic correlations is higher than MARSALA until 1.2 bits/symbol and 1.3 bits/symbol respectively, then it evolves gradually, but in a less significant way compared to MARSALA, until 1.7 bits/symbol when the throughput collapses. At this point, the whole system ends earlier when no more packets can be retrieved. Therefore, the number of preamble detection operations decreases. Each of the complexity curves of R-SPOTiT with a 15 and 31 preambles crosses the single preamble case at 1.6 bits/symbol and 1.8 bits/symbol, respectively,

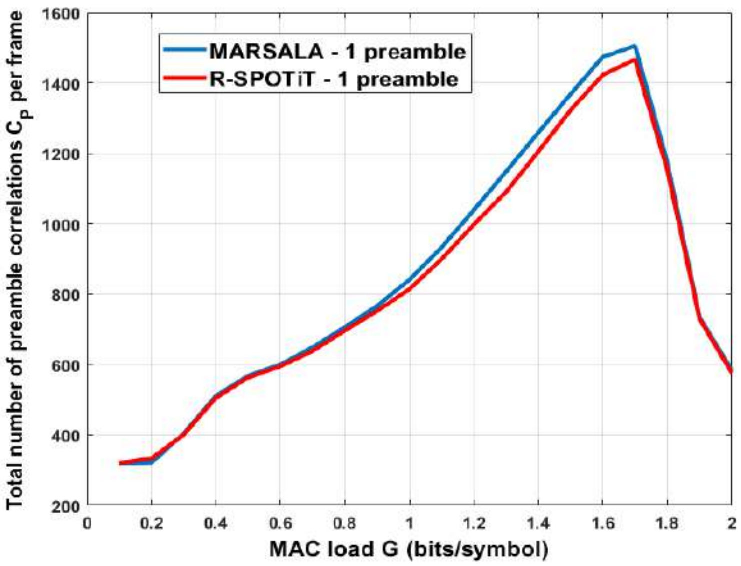


Figure 4.9: Preamble localization complexity in case of MARSALA and single preamble R-SPOTiT

to become less complex. However, this region is around the throughput collapse point (1.7 bits/symbol).

When N_P is lower, 2, 3 and 5, the complexity at low loads is smaller and closer to the single preamble case than to R-SPOTiT with 31 and 15 preambles. They cross MARSALA's curve at around 1 bits/symbol and evolve differently, in a less significant way. They also cross the single preamble R-SPOTiT curve at around 1.2 bits/symbol and present a smaller number of basic correlations compared to the single preamble case and for any number of preambles.

To understand better the contribution of C_P and C_D, in each case, to the overall localization complexity at different channel loads, separate metrics are presented. Figure 4.9 and Figure 4.10 show, one at a time, C_P and C_D for MARSALA and R-SPOTiT with a single preamble. The number of correlations in the preamble detection is negligible next to the number of correlations for the data localization. Therefore, the overall complexity curve of MARSALA follows the shape of C_D curve. Same comment can be dedicated to R-SPOTiT with one preamble, except that it exhibits considerably lower C_D than MARSALA. However, in R-SPOTiT with 31 and 15 preambles (Figure 4.11 and Figure 4.12), C_P is high enough to make the overall complexity curve follow its shape. When the number of preambles is equal to 5, to 3 or to 2 (Figure 4.11 and Figure 4.12), the overall complexity shape converges towards C_D because C_P is low.

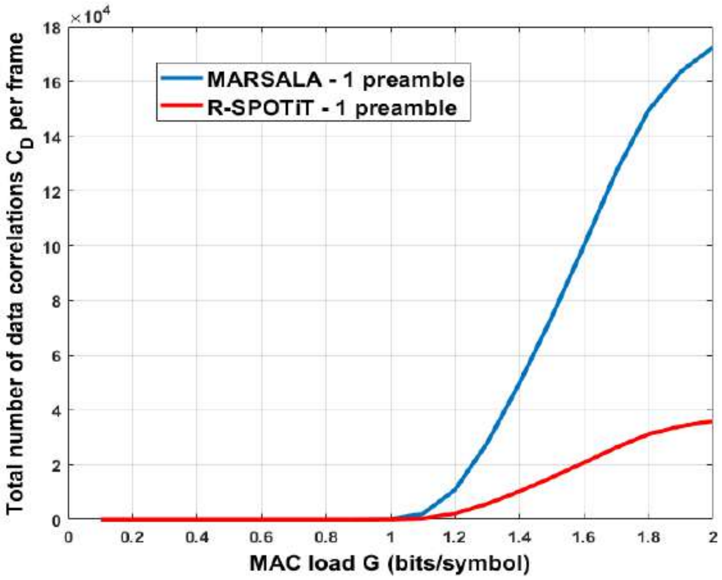


Figure 4.10: Data detection complexity in case of MARSALA and single preamble R-SPOTiT

Consequently, according to the average operational load of a given application, one can select the most adequate complementary treatment with a suitable number of preambles, the smallest in this case, that would take over the decoding process after a deadlock. R-SPOTiT with two preambles is hence the best candidate.

Indeed, the Gold code preamble detection appears to be better when the number of preambles is small. We have computed the total number of times Ψ that R-SPOTiT is used per frame, and how it is distributed ($\Psi = \alpha + \beta + \gamma$) according to three cases. Case A is characterized with α that is the number of times that a preamble of a given packet is detected on the reference time slot, and that its second replica's position exhibits a correlation peak for the same preamble. Case B is characterized by β that is the number of times that only one of the two preambles is detected. Finally, case C defines γ that is the number of times that no preambles are detected on the whole frame. We recall that during case A, the packet is considered localized and thus no data correlations are performed. However, during case B when $\rho = 0$ (when at least one preamble is detected on an analyzed slot), correlations are made over all second replicas' positions of potentially collided packets on the reference time slot that use the same detected preamble. Furthermore, during case C when $\rho = 1$ (when there are no detected preambles in any RTS), correlations are made over all second replicas' positions of all potentially collided packets on the reference time slot regardless of their preambles. Figure 4.13 shows the mean usage percentage of R-SPOTiT per frame during

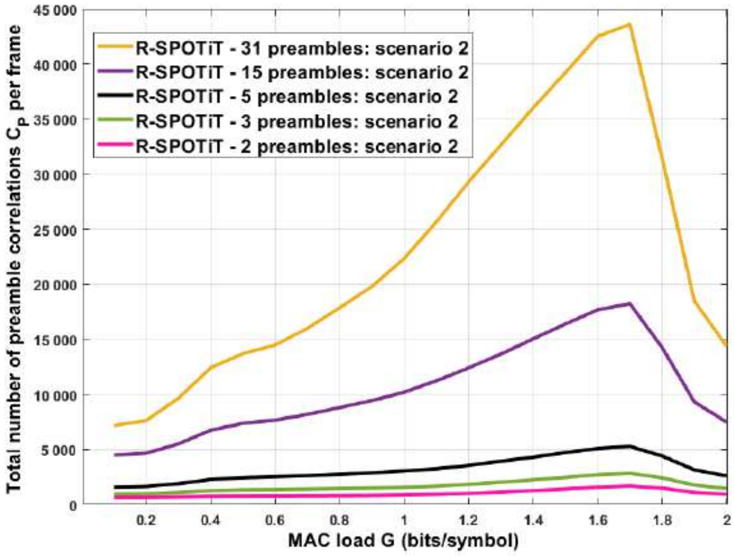


Figure 4.11: Preamble localization complexity in of case multi-preamble R-SPOTiT

each of the cases A, B and C. We can deduce that the probability of having both preambles detected on their respective time slots is higher when the number of preambles is low (see the 2-preambles curve). The probability that none of the preambles are detected is hence the lowest when the number of preambles is the smallest. We recall that in this case ($\rho = 1$) a localization correlation is applied N_{bt} times because the beginning of a packet is unknown. For this reason the total number of data correlations C_D per frame is higher when the number of preambles is high. This is mainly due to the pseudo-orthogonality of preambles. Since preambles are not perfectly orthogonal, a linear combination of some of them can lead to indistinguishable correlation peaks.

Furthermore, the complexity introduced in low loads is not very significant in multi-preamble R-SPOTiT compared to the complexity of MARSALA in high loads. Therefore, it is preferable to use multi-preamble R-SPOTiT. Figure 4.14 sums up the total number of basic correlations, with the various methods and different numbers of preambles for an average load of 1.6 bits/symbol. We can see that the less complex system to use, in this case of pseudo-orthogonal Gold codes, is the multi-preamble R-SPOTiT with 2 preambles.

Moreover, in order to have an idea on how many times the complementary treatment is solicited by CRDSA before the whole system finds itself in a deadlock, we derived experimentally, average values of Δ, which is the maximum frame analysis index δ, in terms of the

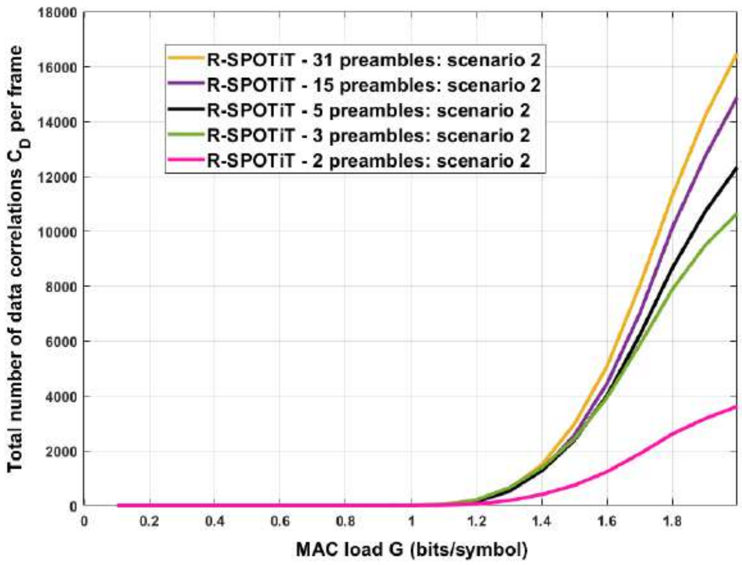


Figure 4.12: Data detection complexity in case of multi-preamble R-SPOTiT

channel load. These values of Δ define, as well, the average number of packets decoded by R-SPOTiT or MARSALA at each load as only one packet is decoded at each intervention, thus at each δ. Table 4.1 exhibits the different values of Δ with respect to the channel load and the corresponding number of transmitters N_{U}. In addition, each value of N_{U} is associated to the average number of decoded packets by CRDSA when it is used alone, noted D_{crdsa}, and by the complementary treatment D_{CT}, each derived from its corresponding PLR. The complementary treatment here refers to R-SPOTiT or MARSALA that have approximately the same PLR performance. The decoding gain resulting from the complementary treatment compared to CRDSA (Δ) is also derived in percentage (D_{gain}).

Table 4.1: Average simulation values of the maximum frame analysis index

G(b/s)	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2**
N_{U}	15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255	270	285	300
D_{crdsa}	15	30	45	60	75	90	105	120	134	143	118	77	57	44	30	31	26	17	24	14
D_{CT}	15	30	45	60	75	90	105	120	135	150	165	180	195	210	224	239	239	173	81	50
$\boldsymbol{\Delta}$	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**2**	**6**	**13**	**23**	**35**	**47**	**56**	**43**	**23**	**14**
D_{gain}	-	-	-	-	-	-	-	-	0.7%	4.7%	28.5%	61%	70.8%	79%	86%	86.6%	83.5%	57.7%	20%	12%

Indeed, we can observe that the number of times (Δ) the complementary treatment intervenes to solve CRDSA's deadlock becomes significant in high channel loads. $\Delta = 1$ means that, on

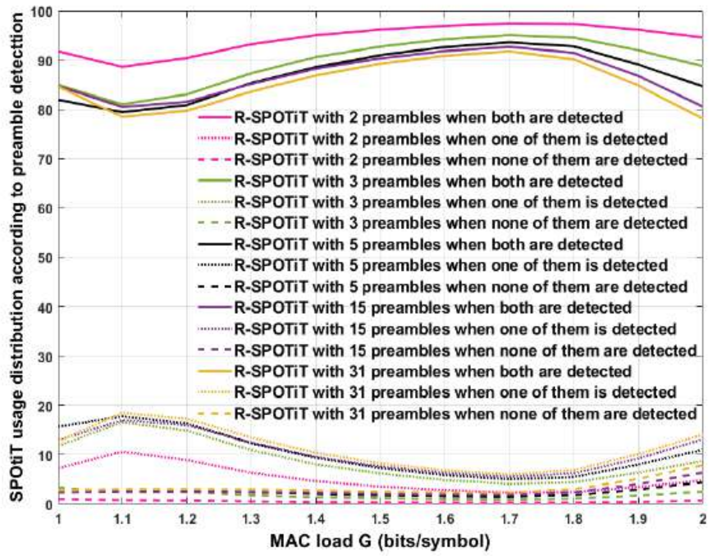


Figure 4.13: R-SPOTiT usage distribution, with $N_{\text{bt}} = 16$ and $R = 5$ per frame in percentage.

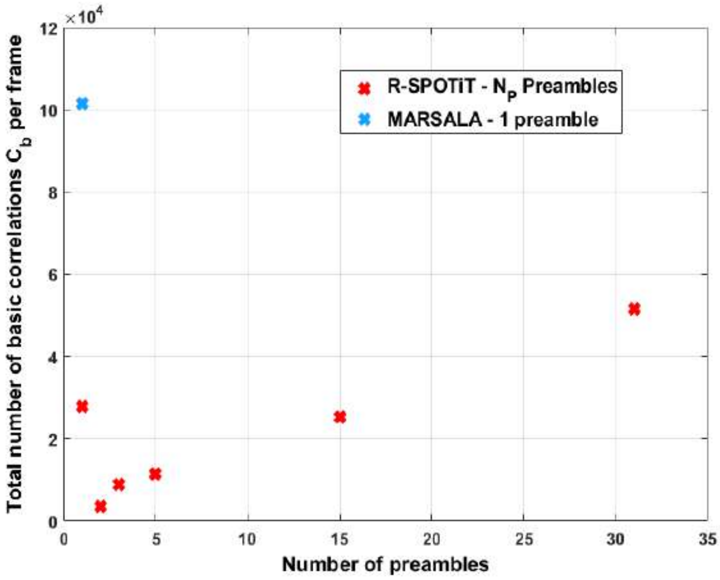


Figure 4.14: Overall frame complexity, with $N_{\text{bt}} = 16$ and $R = 5$ in CRDSA/MARSALA and CRDSA/R-SPOTiT with different numbers of preambles, for a channel load of 1.6 bits/symbol.

average, R-SPOTiT or MARSALA had not taken part of the decoding process, or that their contribution was minimal. Then, until 1.7 bits/symbol, Δ values increase up to 56, then they decrease gradually to reach 14 at a load of 2 bits/symbol along with the throughput collapse. Also, it is noticeable that above the PLR target set to 10^{-3}, the number of decoded packets when CRDSA is used alone (D_{crdsa}) decreases gradually with the increase of the number of transmitters, compared to the total number of users. The same comment can be made regarding D_{CT}, except that with the latter the decrease of the number of decoded packets compared to the ones transmitted starts at a channel load of 1.7 bits/symbol, unlike CRDSA that experiences it at a load of 0.9 bits/symbol. Nevertheless, having the complementary treatment that could be R-SPOTiT or MARSALA triggered each time CRDSA incurs a deadlock, offers a decoding gain that can reach 86.6% with 47 interventions at a channel load of 1.6 bits/symbol.

4.7 Summary & conclusion

The main issue addressed in this chapter is the packet localization complexity, over a whole frame, induced by the detection correlations in R-SPOTiT and MARSALA, each accompanied by CRDSA. This includes the impact of preamble detection performed during CRDSA in addition to the data localization operations accomplished by R-SPOTiT and MARSALA complementary treatments. For an efficient detection and less complexity, we introduced a coarse and a fine tracking with different numbers of samples per symbol. In a single preamble scenario, we showed that CRDSA/MARSALA and CRDSA/R-SPOTiT are equivalent in low channel load environments because only preamble detection is performed. This means that only CRDSA is needed for decoding. However in high loads, MARSALA's complexity considerably surpasses R-SPOTiT with the single preamble, and is on average four times higher. This means that, compared to MARSALA, R-SPOTiT with one preamble is preferable to use. Nevertheless, when the number of preambles in R-SPOTiT is higher, the system is more complex in low channel loads because of the parallel preamble correlations. In high loads, opposite phenomenon is observed. The multi-preamble R-SPOTiT becomes less complex than the single preamble case. This is due to the simplified data localization operations of R-SPOTiT that exploits the pseudo-orthogonal preambles. Also, Gold code preambles turned out to be more effective regarding their detection when their number is small. Indeed, R-SPOTiT with two preambles presents the smallest complexity in this case. Furthermore, the complexity difference in low loads between the single preamble R-SPOTiT and the multi-preamble case is less significant than the difference between them in high loads, especially if the number of preambles is small. To summarize, the single preamble case of R-SPOTiT is preferable in low loads, whereas the multi-preamble scenario is more suitable in high channel loads and reaches the least complex system with two preambles. Also, the total number of basic correlations per frame of the single preamble case is smaller than the one of the multi-preamble case. However, the difference between them is less considerable in high loads where the single preamble case complexity surpasses the one of the multi-preamble case. Therefore, we believe that using R-SPOTiT with two preambles is less costly to the receiver.

Finally, we have worked in this chapter on the complexity evaluation between two Random access methods, R-SPOTiT and MARSALA, that have the same system performance in terms of PLR and throughput. However, R-SPOTiT and MARSALA with two replicas par packet incur a PLR floor in low network loads because of the high probability of loops occurrence in comparison to a higher number of replicas system. In the next chapter, we propose a smart version of SPOTiT with a no-loop packet positioning on the frame.

Chapter 5

Smart Shared Position Technique for Interfered Random Transmissions

Contents

As pointed out in the previous chapters, due to MARSALA's efficiency and the complexity regarding its implementation, Random SPOTiT came out with a new solution aiming to reduce the localization correlations. It uses a shared information between the receiver and each of the transmitters, in addition to potential pseudo-orthogonal preambles, to target a lower number of slots for the localization correlations. As a matter of fact, it exploits the commonly known identification information by each user and the receiver as a seed of a PRNG. This latter is used by the transmitter to select replicas time slot positions and the preamble to be used (if applicable), and at the same time allows the receiver to be aware of them. The localization correlations are hence made only on potential replicas' time slot positions of packets with the same detected preamble that can be collided on the analyzed slot. Despite the good performance it offers which matches MARSALA's throughput and PLR, the latter suffers from an error floor. Basically, The number of loops remains almost the same with MARSALA, which means that the later does not solve the loop phenomenon problem found in CRDSA.

In this chapter, we propose a new technique, called Smart SPOTiT (S-SPOTiT), aiming to improve the PLR performance. Indeed, the PLR curve of R-SPOTiT and MARSALA have a floor corresponding to the probability that some users choose the same replicas' positions, commonly called loops. The expected improvement shall be mostly due to a centralized management of time slot positions and the preamble to use by each transmitter. The main idea is thus to allow the receiver to manage time slot positions and the preamble to be used for each transmitter without having more than one user transmitting its replicas on the same time slot positions. It can be viewed as a mix between DAMA and RA. The RA aspect lies in the fact that a resource is permanently allocated to a community of users and not to only one user. Each of them transmit data whenever it is needed. This allocation depends on an optimal distribution that eliminates data loops between users. Another additional challenge of S-SPOTiT is to keep a relatively simple localization process.

First, the system model is presented in Section 5.2. It has many parameters in common with R-SPOTiT system that will be recalled with supplementary information that characterizes S-SPOTiT, which in turn is described in Section 5.3. An optimal distribution scheme for S-SPOTiT of replicas' positions is included. Simulation results are presented later in Section 5.4. The work presented in this chapter is published in [Zam+18b] and [Zam+18c].

5.1 Problem Statement

As explained before, CRDSA decoding process reaches a deadlock when the maximum number of decodable packets is attained. At this point, MARSALA intervenes to solve this blocking situation. It is based on a computation of correlations between an arbitrary reference time slot and the remaining signal on the rest of the frame. This makes it possible to locate, then to combine replicas of the same packet, in order to have a higher probability of decoding. As a result, MARSALA considerably enhances the throughput and offers a lower PLR. In return, it adds a noticeable complexity to the receiver related to the packet localization correlations. Eventually, R-SPOTiT came out with a new solution aiming to reduce the localization correlations. It uses a shared information between the receiver and each of the transmitters, in addition to the pseudo-orthogonal characteristic of preambles, in its initial version, to target a lower number of slots for the localization correlations. Thus, R-SPOTiT drastically reduces the receiver's complexity without degrading performance and with no additional signaling information. Indeed, same throughput and PLR as in MARSALA are observed. However, the packet loss ratio in each of CRDSA, MARSALA et R-SPOTiT suffers from an error floor which is due to the loop phenomenon, especially in low loads and with a small number of replicas per packets. The latter case with the smallest number of replicas is targeted in this work because it presents the lowest localization complexity, which makes it necessary to deal with its PLR floor.

5.2 System overview

Most of the system assumptions are common to the ones of the previous chapters. Namely, N_{U} users associated to a gateway transmit synchronously, over the same frequency, N_{R} replicas, each on a time slot within a frame of N_{S} slots. Assuming that each user waits for the next frame to send another packet, the worst scenario in this case happens when all of the N_{U} users transmit their replicas on the same frame. The transmitters are synchronized with each other and receive gradually synchronization tables. Packets have equal power, and are composed of a payload after coding and modulation of N_{b} information bits, a preamble, and a postamble. We consider N_{P} pseudo-orthogonal preamble codes (e.g. Gold sequences). Guard intervals are used at the end of each slot to avoid interpacket interference due to potential synchronization errors. When a frame is received, CRDSA will first attempt to decode a maximum number of packets through SIC by browsing slots one by one until it can no longer retrieve information. A complementary treatment is triggered afterwards to resolve CRDSA's deadlock. It can be the legacy MARSALA, or the proposed S-SPOTiT. The difference between those methods is that the latter avoids loops and requires an extra signaling information (sent only once), which is a novelty compared to R-SPOTiT. Actually, S-SPOTiT is considered as a version of SPOTiT since it introduces additional information that is shared between each transmitter and the receiver thanks to the newly added signaling information. This includes the frame structure at the worst case scenario that is known by the receiver. A detailed description of S-SPOTiT is provided in the next section.

5.3 Smart SPOTiT

This section describes a method of assigning to each user time slot positions on the frame in a way that no loops are created. Another goal is to wisely distribute the associated preamble to each packet's position, among a set of pseudo orthogonal codes, in order to simplify replicas localization. Therefore, the set goal is to make sure that each potentially transmitted packet has one of its replicas having a unique preamble on its time slot position. This will allow to determine which users have sent data on the analyzed frame without proceeding to data localization correlations as in MARSALA or in R-SPOTiT. This is applicable only if preambles are correctly detected. Finally, after replicas are localized, combination is performed before demodulation and decoding.

5.3.1 General Principle

In this S-SPOTiT operating mode two main characteristics are to be pointed out:

1. The time slot positions: the receiver manages the time slot positions of replicas on the frame and the associated preamble for each user. It makes sure to differentiate from the others the potentially collided preamble on the same slot of one of the packet's replicas and eliminates data loops. These optimal time slot positions and preamble choice must be communicated to the transmitters as signaling information. It is sent only once and can be added to the logon phase. On the one hand, the PLR is expected to be improved due to the disappearance of the error floor, easily observable in low loads for CRDSA, MARSALA and R-SPOTiT, which is created by data loops. On the other hand, the intelligent layout regarding the choice of time slot positions and preambles reduces the level of complexity in terms of correlations. As a matter of fact, each preamble used by a packet will be unique on one of its replicas' positions; this means that no data localization correlations are necessary when the preamble is detected.

2. Preambles: the pseudo-orthogonality of preambles is taken advantage of to restrain the localization correlations. We recall that their detection probability relies mainly on having good auto and cross-correlation properties, in addition to their length. When a preamble is detected on a slot, the receiver can guess whether this preamble is unique or not. It will consequently confirm the presence of packets that have a unique preamble on

one of their replicas' positions, specifically when their other replicas exhibit a correlation peak. In other words, only preamble detection correlations can be utilized to correctly localize replicas. As stated in [CDD07], these preambles can also be used for an initial carrier phase estimation.

5.3.2 An optimal distribution scheme

In this part, we present an optimal distribution scheme for S-SPOTiT. This pattern shall present a loop-free time slot positions and preambles to be used for a community of transmitters with packets of two replicas each. It is based on a construction that relies on levels. Each level includes user groups, as shown in the example for Figure 5.1. The number of user groups, which is equal to the number of preambles (justified later in this section), is the same at each level. The only difference is the number of users in each group that is reduced by half from one level to another. P_1 is the blue preamble and P_2 is the red preamble. The number

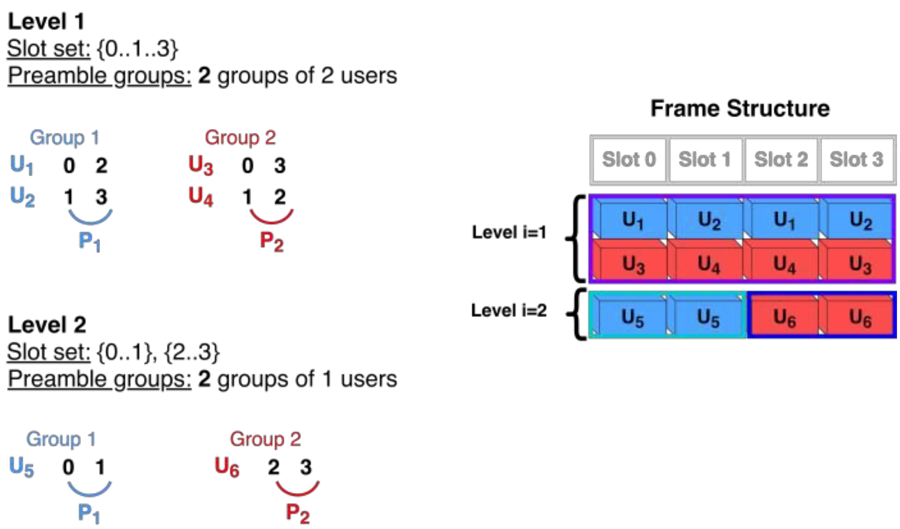


Figure 5.1: Optimal frame structure, $N_S = 4$, $N_R = 2$, $N_P = 2$.

of levels and how to construct each user group is explained below. It is worth noting that this distribution includes the maximum number of loop-free position couples in the frame (of 4 slots in the example). First, the group structure of level one is explained, then the whole system is proposed.

Group structure of level one

In order to have an optimal disposition of time slot positions without loops, several methods can be applied. One way is to create groups of users that contain each, loop-free time slot couples for a certain number of users. In addition, we put the condition that no slot can be assigned to two different users; it is used only once within a group of transmitters. From this initial idea, we propose to have the first replicas' positions of the first group of users on the first half of slots, and their corresponding second replicas' positions on the remaining slots (the other half). Thus there can be $\frac{N_\mathrm{S}}{2}$ users in this group, with a unique replica per slot. In the example of Figure 5.1, users U_1 and U_2 of the first group of level 1 have their first replicas on slots 0 and 1, and their second replicas on slots 3 and 4 respectively. In order to construct the second group of users, a scheme based on a cyclic shift applied to the position of the second replica of each user of the first group is proposed (see Group 2 of Figure 5.1). It is worth noting that it is not possible to make another cyclic shift in the example of Figure 5.1 because it will lead to the Group 1 which already exists. We can conclude that, with an even number of slots, there can be $\frac{N_\mathrm{S}}{2}$ position couples in $\frac{N_\mathrm{S}}{2}$ user groups (2 users in each of the 2 groups in the example of Figure 5.1), which is also the maximum value of the cyclic shift.

The next step is to optimally distribute the preambles to ensure a simple packet localization procedure. Thus, transmitters having each a couple of time slot positions, that belong to a group, use the same pseudo-orthogonal sequence as a preamble (Group 1 of level 1 in Figure 5.1 uses P_1, and Group 2 uses P_2). They are characterized by an index j where $j \in [0; N_\mathrm{P}-1]$ is the preamble index, which is also the value of the cyclic shift applied to the second replicas' positions. Also, the number of groups is equal to the number of preambles. Hence, user groups are referred to as preamble groups. Since a time slot is only assigned once within a preamble group, data localization correlations are avoided. As a matter of fact, a correctly detected preamble will indicate the presence of a single user, if the level 1 is used alone.

To sum up, each user u sends his two replicas on two time slots that we note $P\mathrm{g}_1(u)$ and $P\mathrm{g}_2(u)$. The structure of a preamble group j of level one, including all first and second slots (couples) for any N_S, is defined as follows:

$$G_j = \begin{cases} Pg_1 \in \left\{0,\ 1,\ ..\ \frac{N_\mathrm{S}}{2} - 1\right\} \\ \\ Pg_2 \in \{\ \frac{N_\mathrm{S}}{2} + [\frac{N_\mathrm{S}}{2} + j](\mathrm{mod}\frac{N_\mathrm{S}}{2}), \\ \qquad \frac{N_\mathrm{S}}{2} + [\frac{N_\mathrm{S}}{2} + j + 1]\ (\mathrm{mod}\frac{N_\mathrm{S}}{2})\ .., \\ \qquad \frac{N_\mathrm{S}}{2} + [\frac{N_\mathrm{S}}{2} + j + \frac{N_\mathrm{S}}{2} - 1]\ (\mathrm{mod}\frac{N_\mathrm{S}}{2})\ \} \end{cases} \tag{5.1}$$

Pg_1 and Pg_2 are the set of the first time slot positions of the first replicas and the set of the second time slot positions of their corresponding second replicas, respectively, for each transmitter in the group with the cyclic shift j.

It is important to note that an n^{th} element of Pg_1 corresponds imperatively to the n^{th} element of Pg_2. For example, when $j = 1$ the slot 0 as the position of a given first replica can only be matched with a second replica whose position is $\frac{N_\mathrm{S}}{2} + [\frac{N_\mathrm{S}}{2} + 1](\mathrm{mod}\frac{N_\mathrm{S}}{2})$. Another example of a couple of time slots for a given user is $\left\{1, \frac{N_\mathrm{S}}{2} + [\frac{N_\mathrm{S}}{2} + 2](\mathrm{mod}\frac{N_\mathrm{S}}{2})\right\}$.

It is desired not to assign, more than once, one of the N_P preambles to a group of time slot couples in order to avoid a potential collision between users with the same preamble on a time slot. Consequently, there are N_P possible cyclic shifts, thus N_P groups, having each $\frac{N_\mathrm{S}}{2}$ users and a distinct preamble. This gives at the end a total number of $\frac{N_\mathrm{S}}{2} \times N_\mathrm{P}$ transmitters, over the same frequency, which are attached to the receiver. However, some of the possible loop-free time slot couples are not used because of the preamble uniqueness necessity on each slot for complexity matter.

In order to exploit all loop-free possible combinations, let us first make the assumptions that the number of time slots is a power of two and that the number of preambles is equal to the number of slots divided by two: $N_\mathrm{P} = \frac{N_\mathrm{S}}{2}$. We can now create subsets with a dimension reduced by two from the basic level as explained and illustrated in the upcoming example. This will result in a certain number of levels $N_\mathrm{L} = log_2(N_\mathrm{S})$. At each level, there is the same number of preamble groups. The difference lies in the fact that a preamble group of a level $i + 1$ constitutes a subset of the preamble group at level i that uses the same preamble, and with a reduced number of users. In fact, a preamble group, at a level i occupies a certain number of time slot positions and has a certain number of users. Its corresponding subset (preamble group) with the same preamble at level $i + 1$ occupies half of its time slot positions with a half the number of users. Independently, each preamble group at any level keeps the same properties regarding preamble uniqueness over the occupied slots and the loop-free condition.

In the end, the total number of users from all preamble groups of the different levels correspond to the binomial coefficient $\binom{N_\mathrm{S}}{N_\mathrm{R}} = \frac{N_\mathrm{S} \times (N_\mathrm{S}-1)}{2}$, which represents the maximum number of loop-free time slot couples.

Preamble group structure for $N_\mathrm{S} = 2^{N_\mathrm{L}}$ and $N_\mathrm{P} = \frac{N_\mathrm{S}}{2}$

The composition of each preamble group at each level is formally described, below, in terms of time slot positions and preamble choice for each user. We recall that the group design along with the cyclic shifting serves creating a loop-free system of time slot positions, while associating each of the groups to a single preamble simplifies the packets localization procedure.

Each level $i \in [1; N_\mathrm{L}]$ includes $N_{\mathrm{E},i} = 2^{i-1}$ sets of slots $E_{i,s}$ with $s \in [1; N_{\mathrm{E},i}]$, each of which contains $\frac{N_\mathrm{P}}{2^{i-1}}$ preamble groups of $\frac{N_\mathrm{S}}{2^i}$ users. Each set $E_{i,s}$ is associated to $N_\mathrm{ss}^i = \frac{N_\mathrm{S}}{2^{i-1}}$ slots defined as $E_{i,s} = \{(s-1) \times N_\mathrm{ss}^i, ..., s \times N_\mathrm{ss}^i - 1\}$. The latter is otherwise expressed as $E_{i,s} = \{B_\mathrm{inf}(i,s), ..., B_\mathrm{sup}(i,s)\}$ with $B_\mathrm{inf}(i,s)$ the lower bound which is equal to $(s-1) \times N_\mathrm{ss}^i$, and $B_\mathrm{sup}(i,s)$ the upper bound which is equal to $s \times N_\mathrm{ss}^i - 1$.

There are N_P groups for level 1 (as seen earlier), each having a separate preamble, of $\frac{N_\mathrm{S}}{2}$ users. The next level can simply be formed by redefining the bounds of the new sets of slots resulting from the division by two of the previous level's set. Therefore the number of users in each preamble group belonging to these new slot sets will be reduced to half compared to the previous level.

Each set of slots of the form $\{B_\mathrm{inf}(i,s), ..., M_{i,s}, ..., B_\mathrm{sup}(i,s)\}$, with $M_{i,s}$ the central value of the set; $M_{i,s} = B_\mathrm{inf}(i,s) + \frac{B_\mathrm{sup}(i,s) - B_\mathrm{inf}(i,s) + 1}{2} - 1$, having $N_\mathrm{P}(i) = \frac{N_\mathrm{P}}{2^{i-1}}$ preambles at a given level i, will be divided into two slot sets at level $i+1$ of the form $\{B_\mathrm{inf}(i,s), ..., m_{i+1}^s, ..., M_{i,s}\}$, $\{M_{i,s}+1, ..., m_{i+1}^s, ..., B_\mathrm{sup}(i,s)\}$ each having $\frac{N_\mathrm{P}(i)}{2}$ preambles. At this level $i+1$, m_{i+1}^s is calculated in the same way as $M_{i,s}$ at the previous level i. It should be noted that each set of slots has different preamble groups from those of the other sets. As stated before, in order to create the preamble groups, cyclic shifts are performed only on the positions of the second replicas Pg_2, within the same set of slots $E_{i,s} = \{B_{\text{inf}}(i,s), ..., M_{i,s}, ..., B_{\text{sup}}(i,s)\}$. Preamble groups of a given slot set occupy, all, exactly the same slots, it means that for each group, the first replicas are on the slots $[B_{\text{inf}}(i,s); M_{i,s}]$ and the second replicas are on the slots $[M_{i,s}+1; B_{\text{sup}}(i,s)]$.

Numerical example

Let us consider the case with the following values: the number of slots $N_{\text{S}} = 8$, the number of preambles $N_{\text{P}} = 4$, the number of replicas per packet $N_{\text{R}} = 2$, the number of levels $N_{\text{L}} = 3$, and the total number of users $N_{\text{U}} = 28$.

The worst case scenario can be illustrated when every user u, with $u \in [U_1; U_{N_{\text{U}}}]$, has sent a packet on the same frame. Thus, we must construct preamble groups of all N_L levels. Each preamble is represented by a color.

The N_P user groups of level 1 occupy the whole frame with the largest slot set $\{0..3..7\}$. Each group has four users with the same preamble (see Level 1 in Figure 9.6). The first column of each preamble group gathers the time slots of the first replicas of all four users, while the second column regroups their second replicas' positions. The next step is to create the following level by dividing the slot set of Level 1 into two equal sub slot sets (see Level 2 in Figure 9.6). Every slot set at Level 2 contains half of the number of preamble groups of the previous level with half the number of users. Indeed, the first slot set $\{0..1..3\}$ for example puts together the first and second preamble groups with preamble P_1 and preamble P_2 respectively, each with two users. Finally the third level will have a total of four sub slot sets derived from Level 2 slot sets. Each with a single preamble group of one user (see Level 3 of Figure 9.6).

Figure 9.7 shows the disposition of each potential packet on a frame of the previously described distribution. In other words, This represents the worst case scenario (when all users transmit on the same frame). Each of the preambles includes a group of users in every level. Let us see the blue preamble for example. Its groups are: first, the preamble group in Level 1, to which belong the users U_1, U_2, U_3, U_4, secondly, the preamble group of Level 2, to which belong the users U_{17}, U_{18}, and finally the preamble group in Level 3, to which belongs the user U_{25}. Two properties can be noticeable. First, a set of slots of a level $i+1$ is associated to a number of slots which are half of the slots of the previous level i. Thus, $E_{2,1}$, for instance, has the

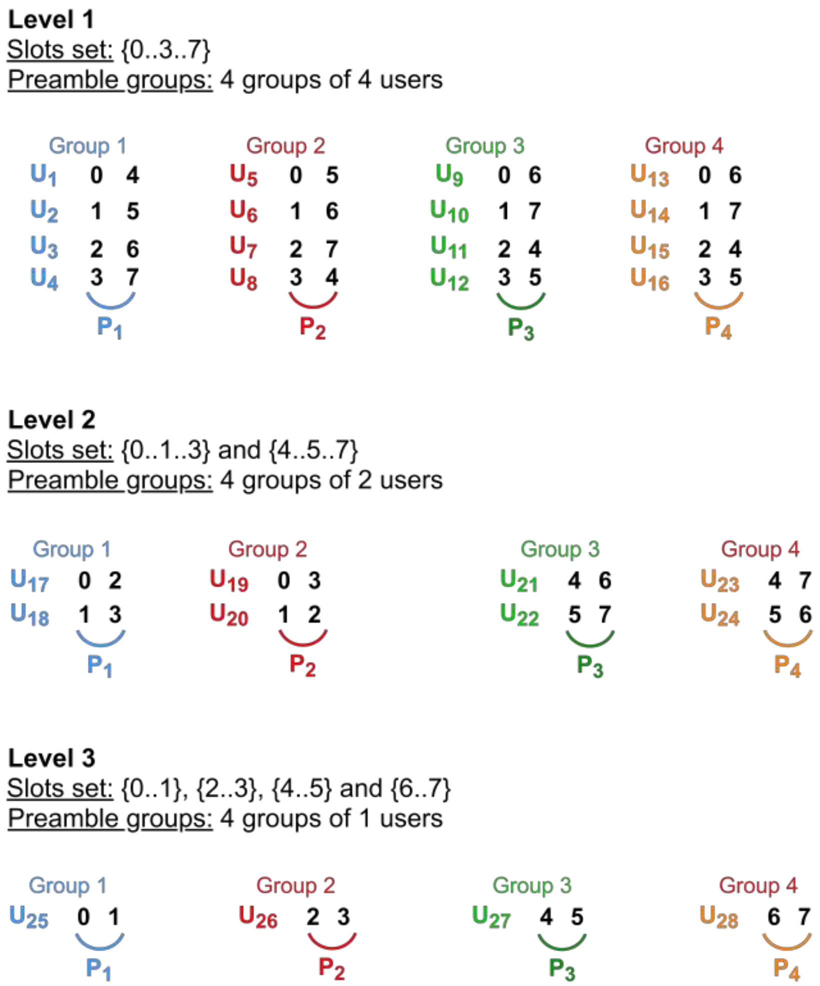


Figure 5.2: Level construction of an optimal frame distribution, $N_{\mathrm{S}} = 8$, $N_{\mathrm{R}} = 2$, $N_{\mathrm{P}} = 4$.

packets of its preamble groups (blue and red) on slots $[0;3]$, which are half of the slots where the packets of the preamble groups of $E_{1,1}$ can be transmitted $[0;7]$. Secondly, a set of slots of a level $i+1$ regroups half of the preambles of the previous level i. As a matter of fact, $E_{2,1}$ is associated to two preambles (blue and red) which are half of the preambles of the previous level $E_{1,1}$ (blue, red, green, and brown). Considering these two properties, at each level, every preamble group has one of the replicas of its packet assigned to a unique preamble on their time slot positions, compared to the following levels. Indeed, at level 1 for example the blue preamble group has the second replicas of its packets U_1, U_2, U_3, U_4 having a unique preamble on their respective slots $[4;7]$, compared to the following levels. This is valid for all preamble groups at any level. the latter is investigated and proved next through a lemma and a theorem. Therefore, we assume there is no restriction on the number of detectable preambles, and decodable packets over a time slot. The detection operation is attempted using preamble correlations over each slot, and is performed during CRDSA.

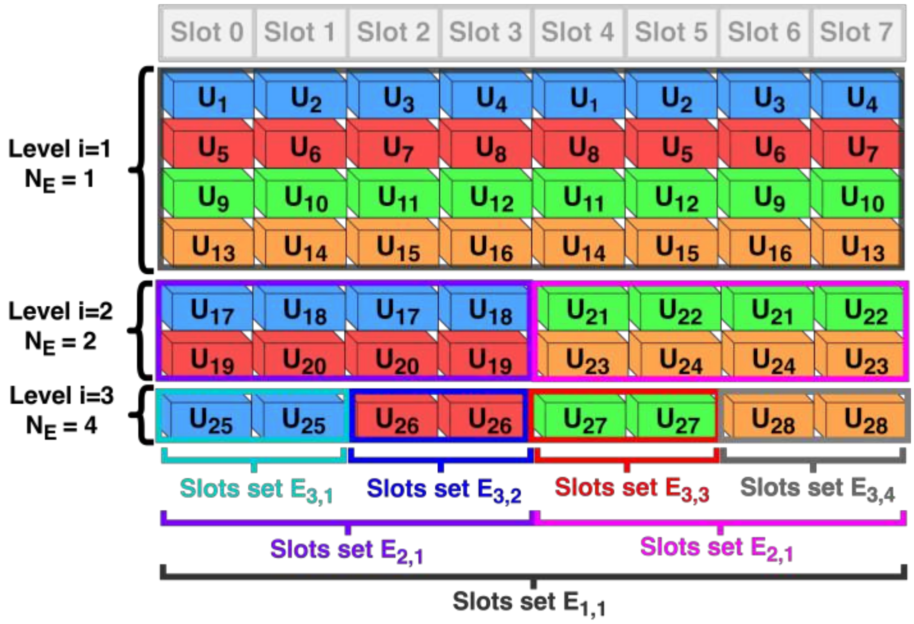


Figure 5.3: Eight slots frame disposition at the worst scenario, $N_R = 2$, $N_P = 4$.

5.3.3 Theoretical analysis of Smart SPOTiT optimal distribution

In this part, the proposed scheme for S-SPOTiT is proved to be optimal, regarding the maximum number of loop-free users with a simple localization process, through a lemma and a theorem. As a matter of fact, the main properties of the optimal power of two distribution that includes user positions and their associated preamble are highlighted. For the sake of simplification, we recall that S-SPOTiT disposition is characterized by two main components; Pg_1 which is the set of all first replicas' positions of users belonging to a preamble group, while Pg_2 represents the set of their corresponding second replicas' positions. We also assume that preamble detection is successful.

Lemma. *Each preamble group at any level has one of its two components Pg_1 or Pg_2 not interfered with any packet of the associated group that uses the same preamble at the higher level.*

Proof. Let us take any preamble group j of level i having packets on the slot set $\{B_{\text{inf}}(i, s), .., M_{i,s}, ..., B_{\text{sup}}(i, s)\}$ and the corresponding preamble group of level $i+1$. A preamble group of any level that has any set of slots has $\frac{N_S}{2^i}$ users. We consider a minimum distance between $B_{\inf}(i+1,s)$ and $B_{\sup}(i+1,s)$ bigger than 1, such as:

$$G_{i,j} = \begin{cases} Pg_1 \in \{B_{\inf}(i,s),\ B_{\inf}(i,s)+1,\ ...\ M_{i,s}\} \\ \\ Pg_2 \in \{(M_{i,s}+1) + [j \left(\text{mod} \frac{N_P}{2^{i-1}}\right)](\text{mod} \frac{N_P}{2^{i-1}}), \\ (M_{i,s}+1) + [j \left(\text{mod} \frac{N_P}{2^{i-1}}\right) + 1]\ (\text{mod} \frac{N_P}{2^{i-1}}), ..., \\ (M_{i,s}+1) + [j \left(\text{mod} \frac{N_P}{2^{i-1}}\right) + \frac{N_S}{2^i} - 1]\ (\text{mod} \frac{N_P}{2^{i-1}})\} \end{cases} \tag{5.2}$$

With $j \in [(s-1) \times \frac{N_S}{2^i}; s \times \frac{N_S}{2^i} - 1]$. At the higher level $i+1$, the group using the same preamble j as the one in level i will belong either to the set of slots $\{B_{\inf}(i,s), .., m^s_{i+1}, .., M_{i,s}\}$ or to $\{M_{i,s}+1, .., m^s_{i+1}, .., B_{\sup}(i,s)\}$. These are the sets of slots to which belongs one of the two components Pg_1 and Pg_2 of level i. We recall that an n^{th} element of Pg_1 corresponds imperatively to the n^{th} element of Pg_2. Let us take the first slot set for instance:

$$G_{i+1,j} = \begin{cases} Pg_1 \in \{B_{\inf}(i,s),\ B_{\inf}(i,s)+1,\ ...m^1_{i+1,s}\} \\ \\ Pg_2 \in \{(m^1_{i+1}+1) + [j \left(\text{mod} \frac{N_P}{2^i}\right)](\text{mod} \frac{N_P}{2^i}), \\ (m^1_{i+1}+1) + [j \left(\text{mod} \frac{N_P}{2^i}\right) + 1](\text{mod} \frac{N_P}{2^i}), ..., \\ (m^1_{i+1}+1) + [j \left(\text{mod} \frac{N_P}{2^i}\right) + \frac{N_S}{2^i} - 1](\text{mod} \frac{N_P}{2^i})\} \end{cases} \tag{5.3}$$

It can be noticed that the slot sets $\{B_{\inf}(i,s), .., m^s_{i+1}, .., M_{i,s}\}$ or $\{M_{i,s}+1, .., m^s_{i+1}, .., B_{\sup}(i,s)\}$ at level $i+1$ correspond exactly to Pg_1 or Pg_2 respectively, of the main slot set $\{B_{\inf}(i,s), .., M_{i,s}, .., B_{\sup}(i,s)\}$ at level i. Therefore, a preamble group j of the level i has one of its the two components Pg_1 and Pg_2 not interfered with any packet of the preamble group that has the same preamble j at level $i+1$. □

Theorem. *A single-frequency data loop-free system, requiring only preamble detection to localize packets' replicas, is built with a maximum number of users that is equal to the binomial coefficient* $\binom{N_S}{N_R}$ *:*

$$N_U = \frac{N_S \times (N_S - 1)}{2} \tag{5.4}$$

Proof. We organize the proof in two parts. The first one concerns the number of users and the second one concerns the localization and decoding ability.

Part 1 : Number of users

At each level i, there are N_{P} preamble groups of $\frac{N_{\mathrm{S}}}{2^i}$ users each, i.e. a total of $N_{\mathrm{U}}(i) = N_{\mathrm{P}} \times \frac{N_{\mathrm{S}}}{2^i} = \frac{N_{\mathrm{P}}^2}{2^{i-1}}$ transmitters. Hence the total number of transmitters throughout the N_U system is:

$$\begin{aligned} N_{\mathrm{U}} &= \sum_{i=1}^{N_{\mathrm{L}}} N_{\mathrm{U}}(i) \\ &= N_{\mathrm{P}}^2 \times \sum_{i=1}^{N_{\mathrm{L}}} \frac{1}{2^{i-1}} = 2N_{\mathrm{P}}^2 \times \left(\sum_{i=0}^{N_{\mathrm{L}}} \frac{1}{2^i} - 1 \right) \\ &= 2N_{\mathrm{P}}^2 \times \left(\frac{2^{N_{\mathrm{L}}+1} - 1}{2^{N_{\mathrm{L}}}} - 1 \right) = 2N_{\mathrm{P}}^2 \times \left(\frac{2N_{\mathrm{S}} - 1 - N_{\mathrm{S}}}{N_{\mathrm{S}}} \right) \\ &= \frac{N_{\mathrm{S}} \times (N_{\mathrm{S}} - 1)}{2} = \binom{N_{\mathrm{S}}}{N_{\mathrm{R}}} \end{aligned} \tag{5.5}$$

We can observe that this number of users corresponds to the maximum number of position couples without loops on a frame of N_{S} slots.

Part 2: Localization and decoding ability

Let us take a given preamble and the worst scenario where all groups of this preamble belonging to the different levels have transmitted on the same frame and see how the SIC can help us to decode all the packets.

According to the Lemma above, each preamble group at a given level i has one of its two components $P\mathrm{g}_1$ and $P\mathrm{g}_2$ not interfered with any of the packets of the corresponding higher level group. On the one hand, localization can be performed for each preamble group packets on one of the replicas time slot position. Thus, no extra localization correlations are necessary. On the other hand, the entirety of the level $i+1$ packets indeed occupies half of all the slots of the level i. By reasoning in the same way for the rest of the levels, level i will always have one of its components not interfered by any packet of any higher levels. Therefore, a preamble group of a given level is decodable using SIC if the packets of all lower levels have been decoded. In other words, an algorithm that starts the decoding operation from level 1 packets will unlock the higher levels one by one until no more packets are on the frame. □

Example: Let us take the previous numerical example of section 5.3.2 and choose the blue preamble. There are three blue preamble groups, one at each level. According to the theorem above, the packet localization can be realized without data correlations if the decoding process starts with the first level; assuming preamble detection is successful. In Figure 5.4, we can observe the worst case scenario of the blue preamble where all packets using the latter have transmitted on the same frame. Moreover, it is noticeable that all second replicas of packets in the preamble group of Level 1 cannot be collided to other packets of the same preamble. This means that all packets of higher levels can only transmit over the first replicas' positions of Level 1 packets. Consequently, the first decoding phase concerns packets of Level 1. It has as a target the second replicas localization using preamble detection. Once all these positions exhibit a blue correlation peak, the presence of information from U_1, U_2, U_3, U_4 is confirmed and hence the decoding process using SIC can be applied. The next step consists in executing the same operations for the packets of the second level. Indeed, after applying SIC on the packets of the first level, the second replicas of packets in Level 2 will be unique regarding their preamble utilization. This process is carried out successively until reaching the last level (in this example the next one). In fact, This procedure remains accurate for all worst case scenarios of any preamble.

Figure 5.4: Worst case scenario for the blue preamble, $N_S = 8$, $N_R = 2$, $N_P = 4$.

5.3.4 Preamble group structure when the number of preambles is not a power of two

As concluded previously, in S-SPOTiT distribution each group uses the same preamble and each preamble is used by a unique group at each level. Therefore, if N_{P} is not equal to $\frac{N_{\mathrm{S}}}{2}$, fewer cyclic shifts will be applied, thus fewer preamble groups are created. It is then considered that $N_{\mathrm{S}} = 2^{N_{\mathrm{L}}}$ and $N_{\mathrm{P}} \neq \frac{N_{\mathrm{S}}}{2}$.

Let us recall, when $N_{\mathrm{P}} = \frac{N_{\mathrm{S}}}{2} = 2^{N_{\mathrm{L}}-1}$, the total number of users which is maximum according to (5.5) is $N_{\mathrm{U}} = \sum_{i=1}^{N_{\mathrm{L}}} \frac{N_{\mathrm{P}}^2}{2^{i-1}}$. When N_{P} is no longer half of N_{S}, N_{U} becomes: $N_{\mathrm{U}} = \sum_{i=1}^{N_{\mathrm{L}}} \frac{N_{\mathrm{P}} N_{\mathrm{S}}}{2^i}$. We can then state that N_{U} depends on the number of preambles which corresponds, in turn, to the number of preamble groups applying each a different cyclic shift.

Case 1 - N_{P} is a power of two lower than $\frac{N_{\mathrm{S}}}{2}$:

N_{P} can be expressed as $N_{\mathrm{P}} = 2^{N_{\mathrm{L}}-1-d}$ with $d \in [1; N_{\mathrm{L}}-1]$. This means that the total number of users is reduced by 2^d:

$$N_{\mathrm{U}} = \frac{N_{\mathrm{S}}(N_{\mathrm{S}}-1)}{2^{d+1}} = \frac{\binom{N_{\mathrm{S}}}{N_{\mathrm{R}}}}{2^d} \tag{5.6}$$

Case 2 - N_{P} is lower than $\frac{N_{\mathrm{S}}}{2}$ and is not a power of two:

In this case, N_{P} can be expressed as $N_{\mathrm{P}} = 2^{N_{\mathrm{L}}-1} - k$ with $k \in [1; 2^{N_{\mathrm{L}}-1} - 1]$. This means that the total number of users becomes:

$$N_{\mathrm{U}} = (N_{\mathrm{S}}-1)\left(\frac{N_{\mathrm{S}}}{2} - k\right) = \binom{N_{\mathrm{S}}}{N_{\mathrm{R}}} - k(N_{\mathrm{S}}-1) \tag{5.7}$$

Figure 5.5 displays the total number of users according to the number of preambles for the example of 128 slots. When it is a smaller power of two, this number of users is divided by two at each value. Thus, it follows a linear evolution. When N_{P} is still smaller than $\frac{N_{\mathrm{S}}}{2}$ but not a power of two, it also evolves according to a linear function superimposed to the smaller power of two one.

These results regarding the total number of users when $N_{\mathrm{P}} \neq \frac{N_{\mathrm{S}}}{2}$ makes sure to keep the same properties as in the optimal distribution. This means that no loops can be created between any of the transmitters. In addition, a simple packet localization is realized due to

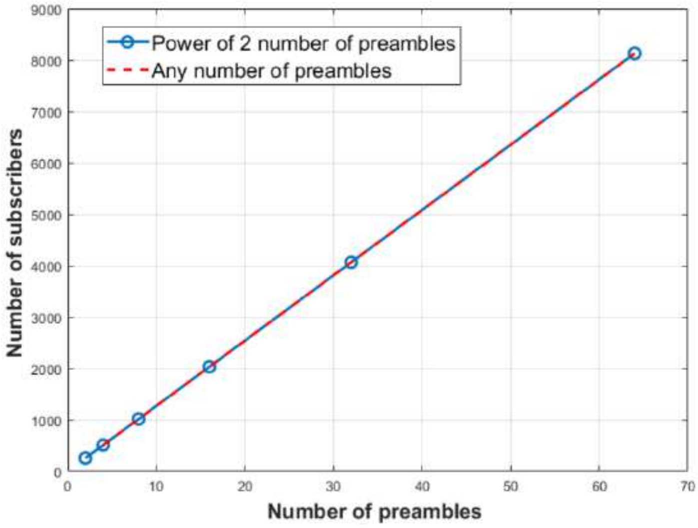


Figure 5.5: Impact of N_{P} when it is not equal to $\frac{N_{\mathrm{S}}}{2}$

the condition of having one of the replicas not interfered with any other packet of the same used preamble.

5.4 Simulation and results

We have chosen a system with two replicas per packet for complexity matter, which means that there is no need for association correlations (to define which replicas belong to the same user). However, what degrades the performance of a system with two replicas compared to a higher number of replicas system is that the probability of loops is more significant. As a result, a PLR floor formed in low network loads can be observed in CRDSA, MARSALA and R-SPOTiT. Actually, only a loop of two packets can be solvable by CRDSA with the parameters we took; QPSK modulation and 3GPP Turbo coding of rate 1/3 over an AWGN channel. Indeed, in this case, CRDSA is able to decode a packet in presence of a single interference. Thus when one of the two couple replicas in collision are alone on a slot; the loops will be solved by CRDSA's SIC, hence breaking the loop. This can be observed using the PER of the MODCOD. It is important to note, according to the simulation result in Figure 9.8, that the probability of having two or more loops represents exactly the CRDSA error pattern in low network loads until the number of collisions becomes large enough (0.6 bits/symbol), due to the number of transmitters, to prevent decoding. The loop occurrence probability and its impact on CRDSA's performance is investigated in Appendix D of [dD14].

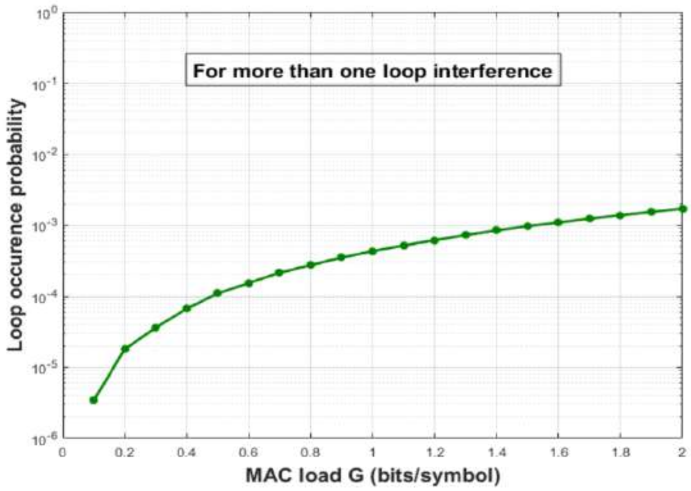


Figure 5.6: Probability of more than one interference loop in a CRDSA-like frame of 100 time slots.

For R-SPOTiT and MARSALA, it is perfectly matched up to a load of 1.5 bits/symbol (see Figure 9.9). Beyond this load, the number of collisions becomes larger and the level of SNIR will no longer allow correct demodulation and decoding.

S-SPOTiT which is based on an optimal management, regarding replicas' positions on the frame and the choice of preamble, prevents loops and makes sure to have a unique preamble on one of the packet's replicas' positions. The goal is to further simplify the packet localization, and improve the PLR performance by removing the error floor created by loops. On the one hand, we have seen that this distribution can prevent data localization correlations and rely only on preamble detection for packet localization. Then, in order to be able to compare the PLR with MARSALA and R-SPOTiT with two preambles, we have chosen to use a loop-free system with 100 slots and the first level $i = 1$ of of S-SPOTiT with 50 preambles. As a result, Figure 9.9 shows that the PLR floor is no longer present. The throughput enhancement (Figure 9.10) is insignificant because its collapse occurs at a load of 1.7 bits/symbol. At this level, the PLR is degraded in the same way for R-SPOTiT, MARSALA and S-SPOTiT.

5.5 Conclusion

In this chapter was introduced the second version of the novel proposed technique SPOTiT. Indeed, S-SPOTiT is a synchronous random access technique over a multiuser channel, the same way as R-SPOTiT and MARSALA, which can be complementary to the legacy CRDSA.

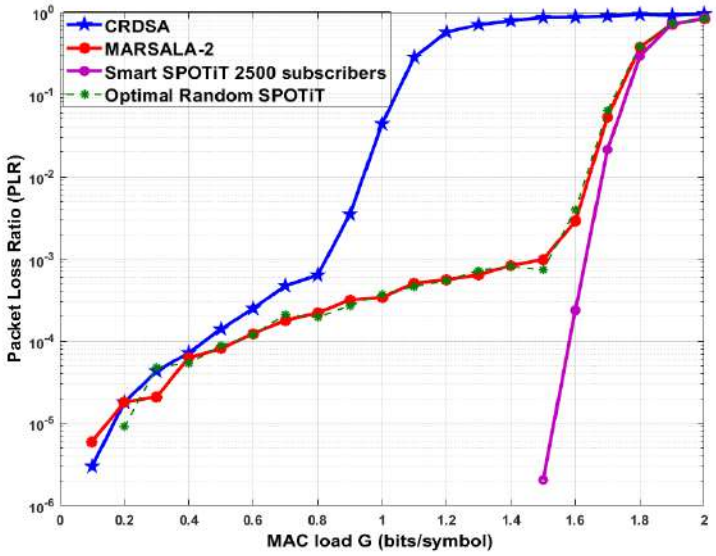


Figure 5.7: PLR of S-SPOTiT, MARSALA-2 and CRDSA with QPSK modulation and turbo coding of rate 1/3 , 100 slots/ frame, 100 information bits, AWGN channel and $E_s/N_0 = 10$ dB

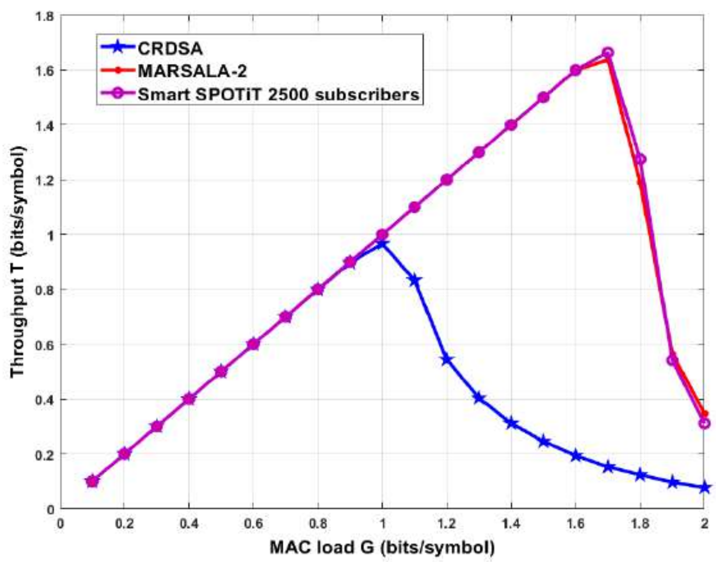


Figure 5.8: Throughput of S-SPOTiT, MARSALA-2 and CRDSA with QPSK modulation and Turbo coding of rate 1/3, 100 slots/ frame, 100 information bits, AWGN channel and $E_s/N_0 = 10$ dB

In particular when there are no more packets to be retrieved by the latter. S-SPOTiT is mainly based on a centralized management, insured by the receiver, of time slot positions and the preamble to use for each transmitter. An optimal frame content distribution is constructed to prevent loops between users and makes sure one of the replicas of the same packet has a unique preamble on its time slot position. Consequently, better PLR performance is resulted and only preamble detection is used to localize packets' replicas.

The optimal distribution of S-SPOTiT proposed in this chapter relies on a power of two system. Indeed, this concerns the number of slots and preambles which is half of the latter. However, when the number of slots and preambles does not respect the power of two condition, the proposed disposition becomes partially invalid. Therefore, the next chapter addresses S-SPOTiT with an arbitrary number of slots per frame and any number of preambles that shall not depend on the latter. Furthermore, this new disposition should be derived in order to fairly compare the performance of S-SPOTiT to R-SPOTiT and MARSALA.

Chapter 6

Irregular framework and extension of Smart SPOTiT

Contents

A regular loop-free scheme for S-SPOTiT was described in the previous chapter with a number of slots equal to a power of two value. An optimal distribution was defined according to a number of levels that depends on the number of slots. In this chapter, we target two main aspects. First, an investigation of S-SPOTiT is made to setup a scheme with an irregular number of slots which is not in the form of a power of two. Secondly, a configuration of a dynamic system, based on a progressive addition of new time slots when needed, is proposed. The way to allocate any number of preambles is also subject to study.

The first part of this chapter aims to introduce a way to construct a loop-free system with any number of slots, that is based on the regular scheme we have seen on the previous chapter. We also show how to adjust and adapt the system with a given number of slots according to new irregular parameters, which are the total number of users and preambles, that can create data loops. The second part proposes a way to permanently keep a loop-free system regardless of the number of users. It actually presents a trade-off between the PLR error floor and a potential additional complexity that is due to the frame length.

6.1 Irregular framework for Smart SPOTiT

6.1.1 Introduction and system parameters

The original principle of SPOTiT was to be able to communicate to the receiver information about each transmitter regarding its packets' locations. The main goal was to mitigate the complexity of the packet localization to solve CRDSA's deadlock as an alternative to MARSALA. The first version R-SPOTiT does not require any signaling information to provide the receiver with packets' locations. It relies on a PRNG which is provided with a seed easily known by the receiver such as identification information. Moreover, using two replicas per packet appears to be a good choice to keep a relatively low complexity. However, this configuration creates a PLR error floor because of the high probability of loops. For this reason, we proposed S-SPOTiT as a second version of SPOTiT which is based on the same principle. In other words, the receiver chooses the packet's positions of all the transmitters in order to avoid loops and sends the related positions to each one of them (transmitters). Nevertheless, the additional signaling information is transmitted only once during the logon phase. We recall that the optimal distribution of S-SPOTiT, proposed in the previous chapter, is based on a regular scheme which requires a power of two number of slots with half of it as a number of preambles. This helps conceiving a certain number of levels, each with preamble groups gathering an equal number of users that have the same preamble. We tackle now the irregular case when the number of slots is not equal to a power of two value with an independent number of preambles. These are the only parameters that are modified compared to the previous chapter.

We recall in this paragraph the chosen notations for the system parameters and assumptions. Our synchronous scheme is characterized by a frame of N_{S} time slots in which are carried packets sent by a community of transmitters among a set of N_{U} users. These packets transport information to the receiver (gateway or satellite) in the form of data packets which include a payload. The payload is preceded by a preamble, chosen among a set of N_{P} pseudo-orthogonal codes (Gold codes in our case). The way to put packets on the frame is not completely random with S-SPOTiT. It actually depends on the optimal distribution in the regular or irregular schemes. As a matter of fact, each of the users receives instructions from the receiver during the logon on where to put its packets and which preamble to use. One of the important features of the regular scheme is the number of levels N_{L} which is the construction basis of the optimal distribution. We recall that N_{L} depends on the number of slots whose value is a power of two. However, in this chapter the irregular scheme is the center of interest.

At the receiver side, same operations are held. CRDSA, as the first step of the decoding process, is followed by S-SPOTiT when reaching a deadlock. The whole mechanism ends when all packets are solved or when both algorithms (CRDSA and S-SPOTiT) are blocked after having retrieved the maximum number of transmitted information.

In order to define the most adequate irregular framework for S-SPOTiT, its main two characteristics should be studied. First, a loop-free system is constructed regardless of how preambles are distributed. Basically, it includes the whole process of designing the optimal distribution based on levels, their slot sets and user groups with the cyclic shift. This is referred to as the regular method because it is based on the power of two system design. A user group defined here is a preamble group, as it has been introduced in the previous chapter, when ignoring the assignment of the preamble. Secondly, the way to distribute preambles on users with a simplified localization complexity is treated.

6.1.2 Loop-free scheme and frame structure using the regular method

The general idea is to first design an irregular loop-free system based on the closest regular scheme. However, as the number of slots of the latter does not match the one of the irregular scheme, the total number of loop-free configurations cannot be reached. This is why it is necessary to identify and complete the irregular S-SPOTiT with a maximum number of loop-free users.

In order to define the system parameters' values when S-SPOTiT is irregular, the regular scheme is taken as a basis. First, each parameter is expressed with respect to the closest lower power of two value of the regular framework. Then, the whole system is built according to the new criteria.

As seen in the previous chapter, the regular scheme relies on a number of slots equal to a power of two value ($N_\mathrm{S} = 2^{N_\mathrm{L}}$) with N_L levels. At each level i, there are $N_\mathrm{P} = \frac{N_\mathrm{S}}{2}$ preamble groups. The total number of users at each level is $N_\mathrm{U}(i) = N_\mathrm{P}\frac{N_\mathrm{S}}{2^i}$, equally distributed over the preamble groups ($\frac{N_\mathrm{S}}{2^i}$ in every group). Thus, N_P also refers to the number of possible groups at each level. Let's recall that at each level there are $N_{E,i} = 2^{i-1}$ slot sets to which are associated equal number of groups. Note that the number of groups associated to each slot set is equal to the number of users belonging to these groups. This is due to the cyclic shifting characteristic over the second replicas' positions. N_P can hence be expressed as $N_\mathrm{P} = N_{E,i}\frac{N_\mathrm{S}}{2^i}$. Actually, the same number of preambles/preamble groups is used at each level. It is always equal to $\frac{N_\mathrm{S}}{2}$. This may not be the case for the irregular scheme.

Let us base the irregular framework on the same principle as for the power of two scheme. In other words, a distribution is constructed based on levels, each one gathering a fixed number of user groups. For simplicity matter, we ignore the preambles for now. Thus, the number of groups is noted N_G instead of N_P.

The number of slots per frame is then expressed as follows: $N_\mathrm{S} = 2^{N_\mathrm{L}} + k$ where 2^{N_L} is the closest lower power of two value and $k \in \mathbb{Z}^+$. The number of levels for the irregular framework is equal to N_L. Each level contains $N_\mathrm{G}(i)$ user groups equal to the number of slot sets multiplied by the number of groups at each slot set.

$$N_\mathrm{G}(i) = N_{E,i}\left\lfloor \frac{N_\mathrm{S}}{2^i} \right\rfloor = 2^{i-1}\left\lfloor \frac{N_\mathrm{S}}{2^i} \right\rfloor \tag{6.1}$$

This means that $\left\lfloor \frac{N_\mathrm{S}}{2^i} \right\rfloor$ is the number of user groups in each slot set of a given level i. This value represents also the number of users in each group of users at the same level, as explained earlier. Consequently, the number of users at each level is expressed in $\overline{N_\mathrm{U}}(i)$. The total number of users in the whole irregular system, using the regular method, is noted $\overline{N_\mathrm{U}}$.

$$\overline{N_\mathrm{U}}(i) = 2^{i-1}\left\lfloor \frac{N_\mathrm{S}}{2^i} \right\rfloor^2 \tag{6.2}$$

The slot sets, having each a lower bound and an upper one are, as explained in the previous chapter, resulting from the division by two of the number of slots at level one. Then the new resulting slot sets are in turn divided by two at the next level. The same rule is applied to the irregular scheme except that the integer part of the result is taken into account when dividing.

The total number of users is computed as a summation of the number of users of all levels. According the assumptions we took, $\overline{N_{\mathrm{U}}}$ is always bigger than $N_{\mathrm{U}} = 2^{N_{\mathrm{L}}-1}\left(2^{N_{\mathrm{L}}} - 1\right)$; the total number of loop-free position couples of the lower power of two regular scheme.

$$\overline{N_{\mathrm{U}}} = \sum_{i=1}^{N_{\mathrm{L}}} 2^{i-1} \left\lfloor \frac{N_{\mathrm{S}}}{2^i} \right\rfloor^2 > 2^{N_{\mathrm{L}}-1}\left(2^{N_{\mathrm{L}}} - 1\right) \tag{6.3}$$

Next, we express $\overline{N_{\mathrm{U}}}$ with respect to N_{U} to determine the number of extra couple positions.

$$\begin{aligned}
\overline{N_{\mathrm{U}}} &= \sum_{i=1}^{N_{\mathrm{L}}} 2^{i-1} \left\lfloor \frac{2^{N_{\mathrm{L}}} + k}{2^i} \right\rfloor^2 \\
&= \sum_{i=1}^{N_{\mathrm{L}}} 2^{i-1} \left[2^{2N_{\mathrm{L}}-2i} + 2^{N_{\mathrm{L}}-i+1} \left\lfloor 2^{-i}k \right\rfloor + \left\lfloor 2^{-i}k \right\rfloor^2 \right] \\
&= \sum_{i=1}^{N_{\mathrm{L}}} 2^{N_{\mathrm{L}}} \left(2^{N_{\mathrm{L}}-i-1} + \left\lfloor 2^{-i}k \right\rfloor \right) + 2^{i-1} \left\lfloor 2^{-i}k \right\rfloor^2 \\
&= 2^{2N_{\mathrm{L}}-1} \sum_{i=1}^{N_{\mathrm{L}}} \frac{1}{2^i} + \sum_{i=1}^{N_{\mathrm{L}}} \left\lfloor 2^{-i}k \right\rfloor \left(2^{N_{\mathrm{L}}} + 2^{i-1} \left\lfloor 2^{-i}k \right\rfloor^2 \right) \\
&= 2^{2N_{\mathrm{L}}-1} \left(\frac{2^{N_{\mathrm{L}}+1}}{2^{N_{\mathrm{L}}}} - 1 \right) + \sum_{i=1}^{\mathrm{L}} \left\lfloor 2^{-i}k \right\rfloor \left(2^{N_{\mathrm{L}}} + 2^{i-1} \left\lfloor 2^{-i}k \right\rfloor^2 \right) \\
&= 2^{N_{\mathrm{L}}-1} \left(2^{N_{\mathrm{L}}} - 1 \right) + \sum_{i=1}^{N_{\mathrm{L}}} \left\lfloor 2^{-i}k \right\rfloor \left(2^{N_{\mathrm{L}}} + 2^{i-1} \left\lfloor 2^{-i}k \right\rfloor^2 \right) \\
&= N_{\mathrm{U}} + \sum_{i=1}^{N_{\mathrm{L}}} \left\lfloor 2^{-i}k \right\rfloor \left(2^{N_{\mathrm{L}}} + 2^{i-1} \left\lfloor 2^{-i}k \right\rfloor^2 \right)
\end{aligned} \tag{6.4}$$

We note ζ the second term of $\overline{N_{\mathrm{U}}}$. This means that $\zeta = \sum_{i=1}^{N_{\mathrm{L}}} \left\lfloor 2^{-i}k \right\rfloor \left(2^{N_{\mathrm{L}}} + 2^{i-1} \left\lfloor 2^{-i}k \right\rfloor^2 \right)$ and $\overline{N_{\mathrm{U}}}$ becomes:

$$\overline{N_{\mathrm{U}}} = N_{\mathrm{U}} + \zeta \tag{6.5}$$

Although, the maximum number of loop-free position couples in the irregular S-SPOTiT framework corresponds, as for the regular scheme, to the binomial coefficient $\overline{N_{\widetilde{\mathrm{U}}}} = N_{\mathrm{S}}(\frac{N_{\mathrm{S}}-1}{2})$ with $N_{\mathrm{S}} = 2^{N_{\mathrm{L}}} + k$. Let us now compute the number of unconsidered position couples Υ with this irregular scheme.

$$\begin{aligned}\Upsilon = \overline{N_{\widetilde{\mathrm{U}}}} - \overline{N_{\mathrm{U}}} &= 2^{-1}\left(2^{N_{\mathrm{L}}} + k\right)\left(2^{N_{\mathrm{L}}} + k - 1\right) - \left[2^{N_{\mathrm{L}}-1}\left(2^{N_{\mathrm{L}}} - 1\right) + \zeta\right] \\ &= k\left(2^{N_{\mathrm{L}}} + \frac{k-1}{2}\right) - \zeta\end{aligned} \tag{6.6}$$

We suspect that these unused combinations in the irregular S-SPOTiT framework must be due to the indivisible number of slots in a slot set, at a given level, on two equal parts for position couples. Indeed, the floor integer part we took to compute the number of groups at each level leaves out one slot when a set is not divisible on two equal integer parts. For example, a number of slots of 15 cannot be divisible on two equal parts, unless one slot is ignored. If the last slot (slot 14) is left out, two equal slot sets having each 7 slots are defined. However in this case, the slot 14 remains permanently empty although it is available. Thus, the unused combinations, with the empty slots, need to be added to the scheme.

Frame Structure

The position couples that constitute a user group at a given level are determined, after having defined the slot sets, according to Equation (5.2) in the previous chapter. It is hence applicable to the irregular S-SPOTiT scheme. Let us take an example of $N_{\mathrm{S}} = 10$ slots. The lower power of two value is a regular frame of 8 slots, which means that $N_{\mathrm{L}} = 3$. By following the scheme described earlier, the number of user groups at each level, according to Equation (6.1) is 5 at the first level, 4 at the second level and also 4 at the last level. This gives a distribution with $\overline{N_{\mathrm{U}}} = 37$ users, which respectively corresponds to 25 couples of positions at the first level, 8 couples of positions at level 2 and 4 couples of positions at level 3. Although, the total number of loop-free couples of positions is equal to 45 according to $\overline{N_{\widetilde{\mathrm{U}}}}$, which means that there are $\Upsilon = 8$ couples of positions left.

The 37 positions which can be precisely defined according to equation 5.2 are represented in Figure 6.1. We can observe that blank spots occur when the number of slots is not divisible by two. Therefore, when dividing the number of slots (5) in the slot sets of level 2 ($\{0;4\}$ and $\{5;9\}$) on two equal parts that are necessary to define users' positions, the result which is equal to 2.5 is not an integer number of slots .

The eight packets' positions left with no loops can be easily deductible. First replicas of users $38, 39, 40$ and 41 constitute the diagonal of the slot set $\{0;3\}$ preceding the first blank spot

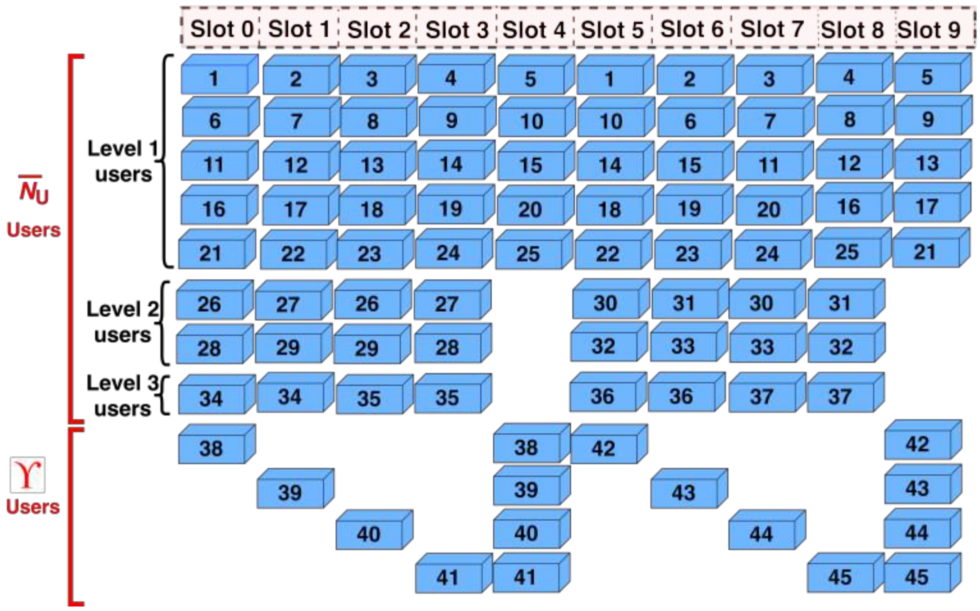


Figure 6.1: Loop-free frame disposition at the worst scenario, $N_S = 10$, $N_R = 2$.

which is slot 4; their second replicas are all placed on the blank spot itself. Same comment can be made regarding packets 42, 43, 44 and 45 positioned on the slot set $\{5; 9\}$ with slot 9 as a blank spot. These $\Upsilon = 8$ packets were actually easy to add to $\overline{N_U}$ because of the small number of slots per frame. However when the frame is longer, the number of blank spots will be dispersed on the numerous levels. This makes it difficult to construct the whole system, especially because it is necessary to first rely of the adapted loop-free scheme described earlier that generates $\overline{N_U}$ positions and then look for the blank spots based on which Υ new positions should be defined using the diagonal method to have a maximum of $\overline{N_{\widetilde{U}}}$ loop-free users.

6.1.3 Irregular number of total users and preambles

In previous work, we went through a loop-free optimal distribution of S-SPOTiT with an irregular number of slots. It is of interest now to characterize a system with an irregular number of preambles and total number of users.

The way to allocate preambles to the loop-free packets defined earlier with the regular method is important to ensure a simple localization. Indeed, in the power of two system, one of the two replicas of all packets in the loop-free distribution cannot collide with another packet having the same preamble. This makes sure to have only one packet candidate when this preamble is detected, which confirms its presence without localization correlations. Moreover, when the total number of users in the system is lower or higher than the total number of loop-free couples, the optimal distribution should be revised and adapted to the new parameters.

Following the regular method, it should be relatively simple to distribute preambles on the user groups for the irregular S-SPOTiT. It has to respect the condition of having, at each slot, a single user with a given preamble. We recall that this constraint is only applicable at the first level. Indeed, when the decoding process using SIC starts with packets of level 1, the next level users will witness the same characteristic, which means that one of the replicas cannot be collided with any other packet using the same preamble. This implies that packets of all levels benefit from this condition as long as all previous levels' packets are solved. On this basis, the regular scheme can be constructed. We can observe on Figure 6.2, that the optimal number of preambles that respects the previously mentioned condition for a simple packet localization is $N_{\mathrm{P}} = \frac{N_{\mathrm{S}}}{2} + 1$ (in this case).

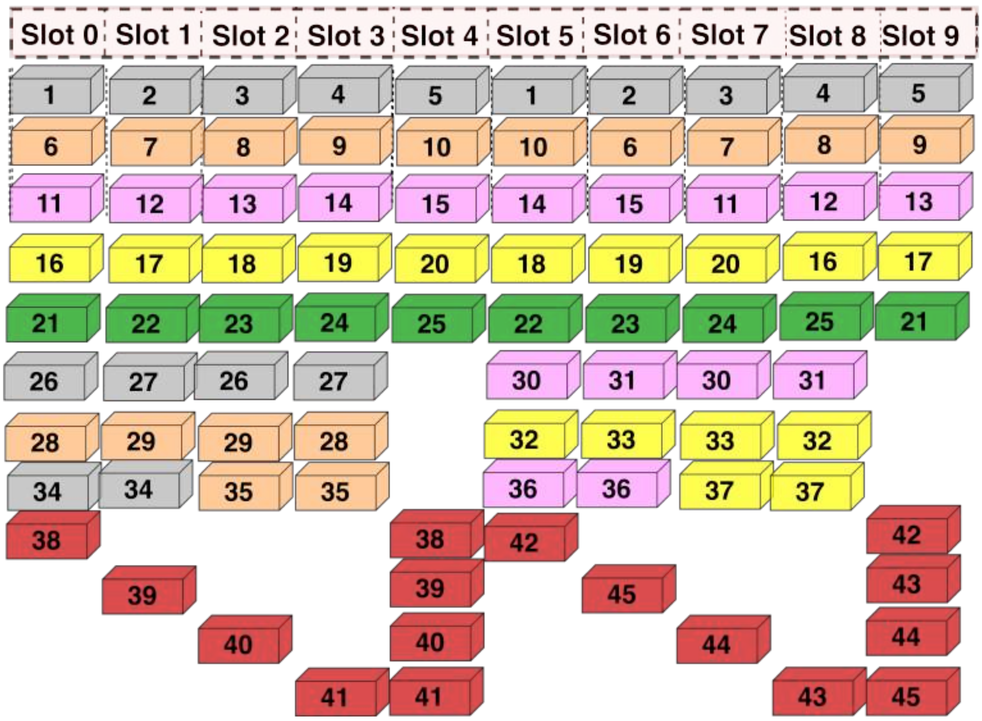


Figure 6.2: Preamble distribution in irregular S-SPOTiT using the regular method, $N_{\mathrm{S}} = 10$, $N_{\mathrm{R}} = 2$.

Nevertheless, according to the results presented on Chapter 4 regarding Gold preamble detection, using a small number of preambles is better. Also, the number of registered users can be different from $\overline{N_{\widetilde{\mathrm{U}}}}$, i.e. larger or smaller. Therefore, we need to find a way to adapt the optimal distribution according to the number of users and preambles. We will first start considering an optimal number of preambles and any total number of users. Two cases are to be distinguished: when the number of subscribers is lower than $\overline{N_{\tilde{\mathrm{U}}}}$ and when it is larger.

In the first case, there are two possible ways to build the system. The first one is to take successively a user group (named also preamble group is this dissertation) from level 1 and then all the associated user groups from the remaining levels. Users are added dynamically until reaching the given number of users which is less than $\overline{N_{\tilde{\mathrm{U}}}}$. The second method is to take successively user groups from level one until all are used and then from next levels if necessary. This latter technique ensures having a homogeneous number of potential transmitters per time slot and therefore it is chosen to build the system.

For the second case when the given number of users is bigger than $\overline{N_{\tilde{\mathrm{U}}}}$, the trivial way to build the system is to repeat positions from the user groups one by one and level by level dynamically until reaching the total desired number. This means that each time a user is added above the maximum $\overline{N_{\tilde{\mathrm{U}}}}$ loop-free positions, a loop is created.

The next step resides in distributing preambles on the new built system with a given number of users when N_{P} is not optimal for a simple localization. A small value of N_{P} undoubtedly induces data localization complexity because the characteristic which guarantees a simple localization is no longer valid. Keeping preambles uniformly distributed on the system users appears to be a good choice. This way of building an adaptable distribution is also applicable to the regular power of two system. Let us take an example of 10 slots per frame, 2 preambles and a total number of users that does not exceed 15 for a given application. According to the choices we have taken, one of the two preambles will be associated to 8 users and the other one to the remaining 7 users (see Figure 6.3).

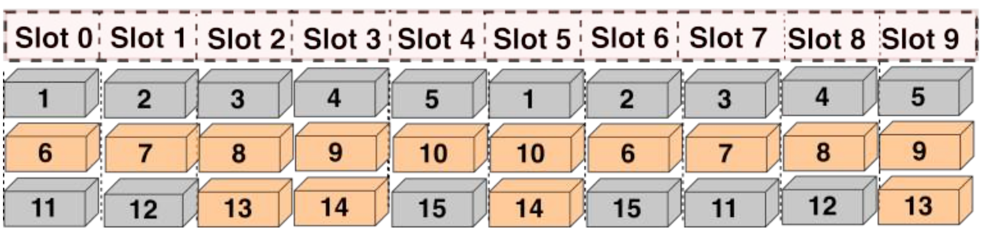


Figure 6.3: Adapted irregular S-SPOTiT distribution with $N_{\mathrm{S}} = 10$ and $N_{\mathrm{R}} = 2$ to a case with 15 users and two preambles.

6.2 Extension of S-SPOTiT: a dynamic system configuration

In this section, we propose a design of a loop-free dynamic system regarding the number of slots, in agreement with the number of users. Indeed, some critical applications need to operate under a certain PLR target, which makes the loop phenomenon extremely restrictive. Initially, the goal is thus to permanently be able to eliminate loops between packets regardless of the number of users. We recall that with the regular or irregular S-SPOTiT, whenever the number of users exceeds the maximum number of the loop-free configurations, which is equal to the binomial coefficient $N_S(N_S-1)/2$, loops are created. This is clearly due to the repetition of the already used configurations (position couples), which is valid as long as the number of slots is a fixed value. Therefore, the traditional S-SPOTiT happens to be unsuitable for the type of systems that, on one hand, are subject to a PLR target, and on the other hand, can be resized in terms of the number of users for a massive connectivity (5G applications).

In the perspective of creating a dynamic loop-free adaptable system, we propose an extension of S-SPOTiT based on a diagonal method to construct the whole distribution. Its main feature it that whenever a loop-free user needs to be added to the system, no modification of the previously defined positions is involved, thus there is no additional signaling information. The adjustment of the system concerns only the newly added users, which receive a unique signaling information as for the traditional S-SPOTiT.

6.2.1 Loop-free scheme and frame structure with the diagonal method

One of the ways to built a loop-free system with the maximum number of users, as an extension of S-SPOTiT, is to use a method we call "diagonal". It is inspired from the previously applied method to complete the irregular scheme based on levels, slot sets and user groups. This diagonal method appears to be a good choice to build the whole framework. To do so, a progressive construction of the potential loop-free frame content is created according to the time slots. Each time new users need to be added to a complete distribution (which already have a maximum number of loop-free position couples), a new time slot is introduced in order to keep a system without any loops. The construction of a diagonal distribution with two replicas per packet is as follows:

- As there are two replicas per packet, the smallest frame length is with $N_S = 2$ and a single packet, which corresponds to the maximum number of loop-free position couples. Each of the slots (slot 0 and slot 1) contains one replica of the unique packet. Indeed, no more packets can be placed on this frame without creating a loop (see Figure 6.4).

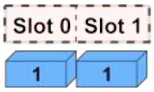


Figure 6.4: Loop-free packets in a frame of two slots, $N_R = 2$.

- The next step consists in adding slots progressively according to the total number of users. Thus, each time a new slot is added, new positions are included using the diagonal method. The positions of the first replicas are placed diagonally with respect to the previous slots, and their corresponding second replicas' positions are placed on the added slot itself. Consequently, for a slot index $s \in \{2; N_S - 1\}$, there are s possible positions that can be generated diagonally. The diagonal method takes then literally effect when the number of slots exceeds two. Figure 6.5 shows how the diagonal method is used each time a slot is added.

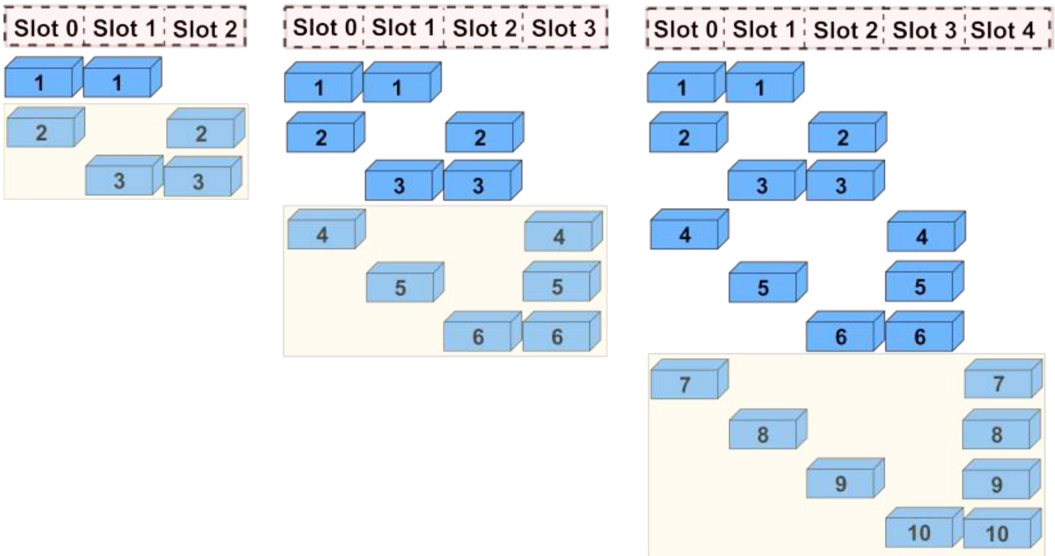


Figure 6.5: Progressive scheme for the S-SPOTiT digonal method $N_R = 2$.

Note that for schematic simplicity, replicas belonging to the same packet are uniquely represented on the same line. This process is carried out until reaching the total number of users. To sum up, with a given value of N_S, users are added progressively, starting with one packet over two slots (slot 0 and slot 1), then s packets at each slot index s. This mechanism stops when the value of s reaches $N_S - 1$. The value of N_S is hence set according to the desired number of users.

It can be concluded that the total number of loop-free positions $\overline{N_{\widetilde{U}}}$ achieved with a certain number of slots N_S with a last index $N_S - 1$ is equal to the summation of the diagonal distributions of the last slot ($s = N_S - 1$) and of all the previous slots, including the first position of the unique packet. Thus $\overline{N_{\widetilde{U}}}$ is computed as follows:

$$\overline{N_{\widetilde{U}}} = \sum_{s=1}^{N_S - 1} s = \frac{N_S(N_S - 1)}{2} \tag{6.7}$$

It is important to note that this distribution keeps a homogeneous number of users per slot at the worst case scenario when all of them transmit on the same frame. In addition, each time a loop-free user is added to the distribution, no change of positions occur for the previously defined position couples. This means that the system can be resized without resetting new information nor introducing extra signaling for the existing disposition.

This resulting diagonal system is characterized with a flexibility regarding the number of users, the absence of loops between packets, and with a small signaling information that involves only the newly added users. In order to be achieved, this extension of S-SPOTiT might necessitate longer frames than in the traditional systems. Consequently, a trade-off can be made regarding the complexity and potential transmission delay introduced by the longer frames and the loop phenomenon elimination, according to the applications' needs.

Nevertheless, the loop-free feature of the diagonal method needs to be proved. In this regard, a theorem is proposed below.

Theorem. *A slotted loop-free system with two replicas per packet can be constructed dynamically, regardless of the number of users, thanks to the addition of new time slots.*

Proof. The proof is organized according to the recursive behavior of the diagonal system. As the dynamic part of the distribution consists in adding a time slot each time new users join the system, the distribution is constructed progressively, starting from a set of two slots, in a way that no loops are created along the process.

- First, a single packet is defined on a frame of two slots. As a matter of fact, in order to avoid loops, only one couple of two positions can be defined over two slots.

- Each slot that is added with a given index s, introduces up to s new user positions. All their first replicas are placed on the previous s slots respectively, and their corresponding second replicas are all placed on the added slot itself (with index s). As all of the first replicas have distinct positions, no loops can be created between the newly added users. In addition, no loops can be created between them and the previous position couples. As a matter of fact, the added slot with a given index s, which introduces new user positions, did not exist in the previous frame structure. Therefore, whenever a new time slot is introduced, a new longer frame structure is defined.

- The previous statement is valid for all the slots to be added. In fact, slots are added progressively until reaching the desired number of users. It is achieved with a certain number of slots N_{S}. The last diagonally added packets at the last slot with index $N_{\mathrm{S}} - 1$ cannot include loops between them and neither with all the previously added users. Consequently, the whole system is loop-free.

□

It is worth noting that at a given added slot s, it is not necessary to include all of the s possible positions. It should depend on the necessary number of additional users.

6.2.2 Preamble distribution

Given the results of Chapter 4, a small number of preambles is preferable. In this case, the same choice as in the regular and irregular S-SPOTiT should be taken. The preambles should be homogeneously associated to the total number of users. A Round-robin algorithm can be used to equally distribute the preambles on the progressively added users.

Numerical Example

Let us take an example with 45 users and 2 preambles. The construction of the diagonal distribution is illustrated in Figure 6.6 along with a Round-robin distribution of the two preambles. The blue packets use the first preamble, and the red ones use the second preamble. We can see that during the worst case scenario, where packets from all users are transmitted in the same frame, The Round-robin distribution is mostly fair. Among the nine users in each

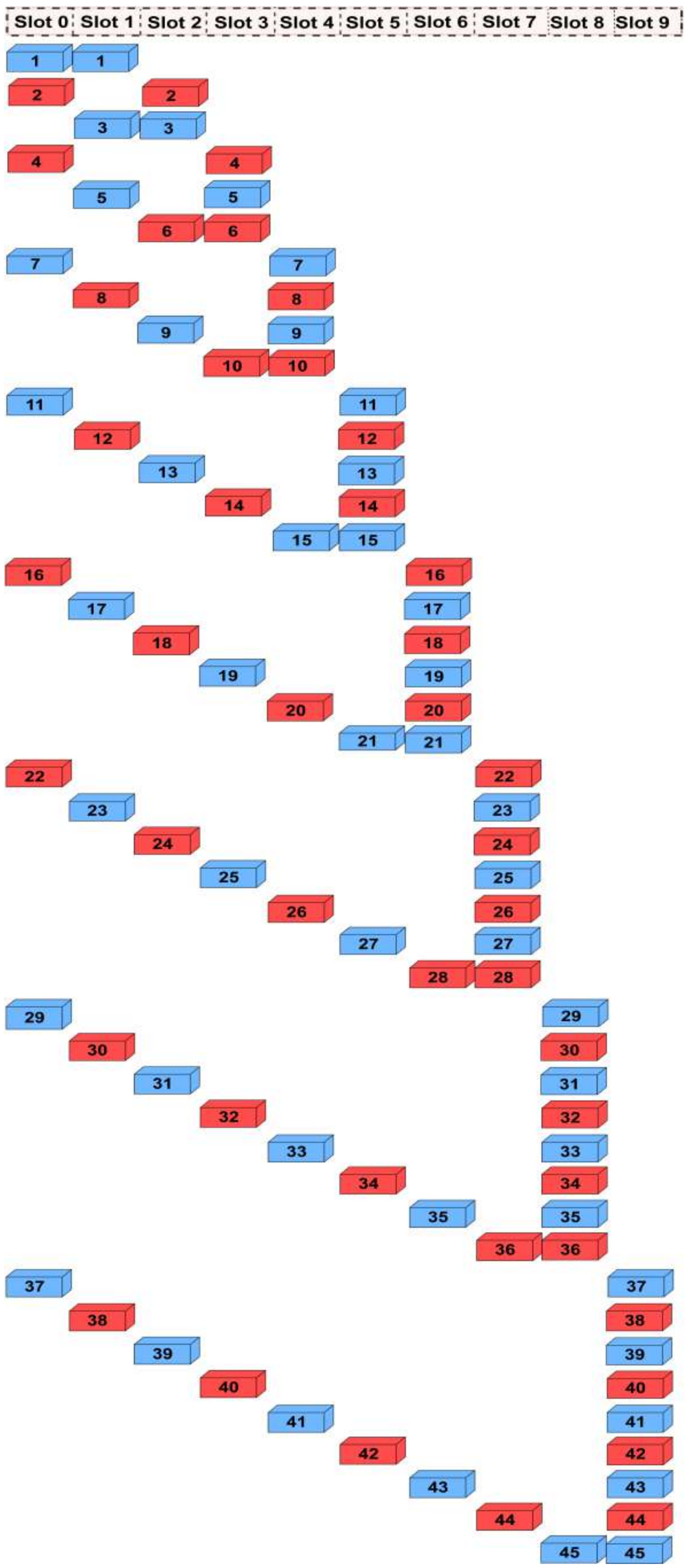


Figure 6.6: Extension S-SPOTiT with the diagonal distribution for $N_{\mathrm{U}} = 45$ and $N_{\mathrm{R}} = 2$.

slot, four of them use the first preamble and five use the other preamble, except for two slots where the distribution is over six and three users respectively for the blue and red preambles.

6.3 Discussion

When the number of preambles is small, the characteristic of the regular and irregular S-SPOTiT which ensures simple packet localization is interrupted. That is, a packet can no longer have one of its replicas using a unique preamble on its time slot position. In this situation the localization complexity of S-SPOTiT is perceived as equivalent to the one of R-SPOTiT. Indeed, the more the channel load is bigger the more often a given preamble is used. Moreover, since the complementary methods R-SPOTiT and S-SPOTiT intervene to solve CRDSA's deadlock at high loads, the probability of having a single preamble usage in one of the replicas positions of transmitted packets remains very low. This means that data correlations are needed to properly localize packets rather than relying on preamble detection alone. The only difference between S-SPOTiT and R-SPOTiT is that the former maintains a homogeneous distribution of preambles and users over the frame. Nevertheless, this uniform disposition exist only at the worst case where every single user has transmitted a packet on the same frame or with a certain number of transmitters with specific users. Otherwise the frame disposition depends on the number of transmitters which is random in addition to the randomness of which among all users are the transmitters.

6.4 Summary and conclusion

The complementary method S-SPOTiT was studied in this chapter with irregular parameters such as the number of slots per frame, the total number of users, and the number of preambles. In addition, it has been extended towards a dynamic loop-free distribution regardless of the number of users. The irregular method is based on the scheme of the regular S-SPOTiT with a power of two number of slots and preambles. It relies on a construction built with levels and slot sets resulting from a division by two of the original slot set of the whole frame. However, by choosing an irregular number of slots per frame, blank spots are created each time a slot set is not divisible by two. These blank spots represent empty time slot positions at a given slot set of a given level. In order to exploit the whole number of loop-free positions, we proposed to fill the blank spots with a diagonal method. The latter inspired the author to build a permanently loop-free dynamic scheme for S-SPOTiT using a diagonal distribution. New time slots are added to the frame each time new loop-free user positions are required. As the frame can be longer than in the traditional systems, because of the required additional positions and the loop-free condition, a compromise can be set according to the applications' needs in terms of performance, complexity and transmission delay.

We recall that S-SPOTiT offers a way to eliminate the loop phenomenon by introducing signaling information that is sent only once to each transmitter. Nevertheless, asynchronous transmissions considerably mitigate the loop phenomenon thanks to the partial interference that is encountered. For this reason, we investigate SPOTiT in an asynchronous environment in the next chapter.

Chapter 7

Asynchronous Packet Localization with Random SPOTiT

Contents

7.1 Introduction

In synchronous transmissions, R-SPOTiT comes to rescue CRDSA by unlocking it, in the same way as MARSALA but with less complexity. As a matter of fact, CRDSA process finds itself in a deadlock when no more packets can be retrieved due to the high MAC loads or to the loop phenomenon. R-SPOTiT locates packets' replicas on the frame, in a less complex way than MARSALA, and then combines them for a higher signal to noise plus interference ratio but still suffers from the loop phenomenon in case of two replicas per packet. In synchronous transmissions, Smart SPOTiT that constructs an optimal distribution of packets' positions on the frame has been proposed as a solution to suppress loops. Nevertheless, ACRDA considerably reduces the latter thanks to its asynchronous property and offers a lower PLR. On that basis, we believe ACRDA coupled with an asynchronous version of R-SPOTiT (AR-SPOTiT) should offer better performance.

AR-SPOTiT can also be seen as a complementary alternative, to ECRA, for replicas localization in an asynchronous environment. As a matter of fact, AR-SPOTiT permits to localize replicas with a shared information that does not require signaling overhead between the receiver and each of the transmitters.

The remainder of this chapter is as follows. Section 7.2 gives an overview of the considered asynchronous satellite system; then a brief summary about Synchronous R-SPOTiT is presented in Section 7.3. Asynchronous decoding characteristics and AR-SPOTiT are detailed afterwards in Section 7.4 and Section 7.5 respectively. Finally, Simulation results are presented in Section 7.6 before concluding with Section 7.7. The work of this chapter is accepted in [Zam+19].

7.2 System Overview

Same environment, as in the previous chapters, is considered. Thus, we focus on packets reception of multi-access transmissions coming from a random return link channel via satellite. N_U users transmit, asynchronously according to ACRDA, N_R replicas on different positions. Each burst is located within a specific virtual frame of $N_S = 100$ virtual time slots, over the same frequency. This means that no signaling information is necessary, thus no synchronization tables are received by terminals. Yet, each packet has a signaling field about its replicas locations on the Virtual Frame VF independently from the others.

In a general way, at the physical layer level, the set of $N_b = 100$ information bits of a user are turned into a MODCOD using QPSK modulation $M = 4$ and 3GPP turbo coding of rate $K = 1/3$. The supposed equipowered packets are then formed by adding, at the beginning and at the end of the resulted payloads of $N_{sym} = N_b/Klog_2(M)$ symbols a preamble and a postamble respectively, then at a known location in the payload a signaling field regarding other replicas positions on the VF. Each packet and its replicas are placed on the VF time slots, associated to a given user, according to two ACRDA modes: all positions are randomly selected on the VF; or having the first replica imposed to the first virtual time slot, and the other replicas randomly placed on the VF. The latter allows to have a reduced transmission delay for non critical loads but appears to be less significant at Transport Control Protocol (TCP) layer level as reported in [De +14]. The results show that both modes are though equivalent in terms of PLR and throughput. We have chosen the first mode for AR-SPOTiT.

We assume the channel model is AWGN with a $E_S/N_0 = 10\ dB$.

At the receiver side, ACRDA analyzes the memory for preamble detection through a sliding window SW (see scheme in Figure 9.7) and then performs SIC operations to all decodable packets until it is blocked. The preamble detection operation will be adapted to a set of pseudo-orthogonal codes. We fix the maximum SIC iterations at 15, window size at 3 times the VF size which is the same for all users and a window step of $0,15$ the VF size. Typically, these choices aim to keep the smallest window size possible for a good packet detection and keep the window step the largest possible for a reduced complexity. The PLR has been tested with different sizes of SW for ACRDA in [De +14] where two to three times the VF size is recommended. Before the SW is shifted by a window step, AR-SPOTiT can intervene. As a matter of fact, this complementary treatment is triggered to resolve ACRDA's deadlock when no more packets can be retrieved on the current SW. Perfect estimation of the channel parameters such as the timing offset, the phase shift and the frequency offset is assumed. Once AR-SPOTiT has localized replicas' positions, they are combined and attempted to be decoded again. When AR-SPOTiT is locked, the SW steps forward and ACRDA starts decoding again.

7.3 Brief recall of Synchronous Random SPOTiT

R-SPOTiT, presented in Chapter 3, has been proposed as an alternative solution to MARSALA that aims to solve CRDSA's deadlock. Indeed, when CRDSA can no longer decode packets on a given frame, MARSALA takes over the decoding process. First, it localizes replicas of collided packets on a randomly chosen reference time slot through correlations that are made over the whole frame. Then it combines replicas that belong to the same packet (according to correlation peaks) prior to decoding and SIC. If more than two replicas per packet are used, an association step takes place with extra correlations that affiliates replicas of interest to the same packet.

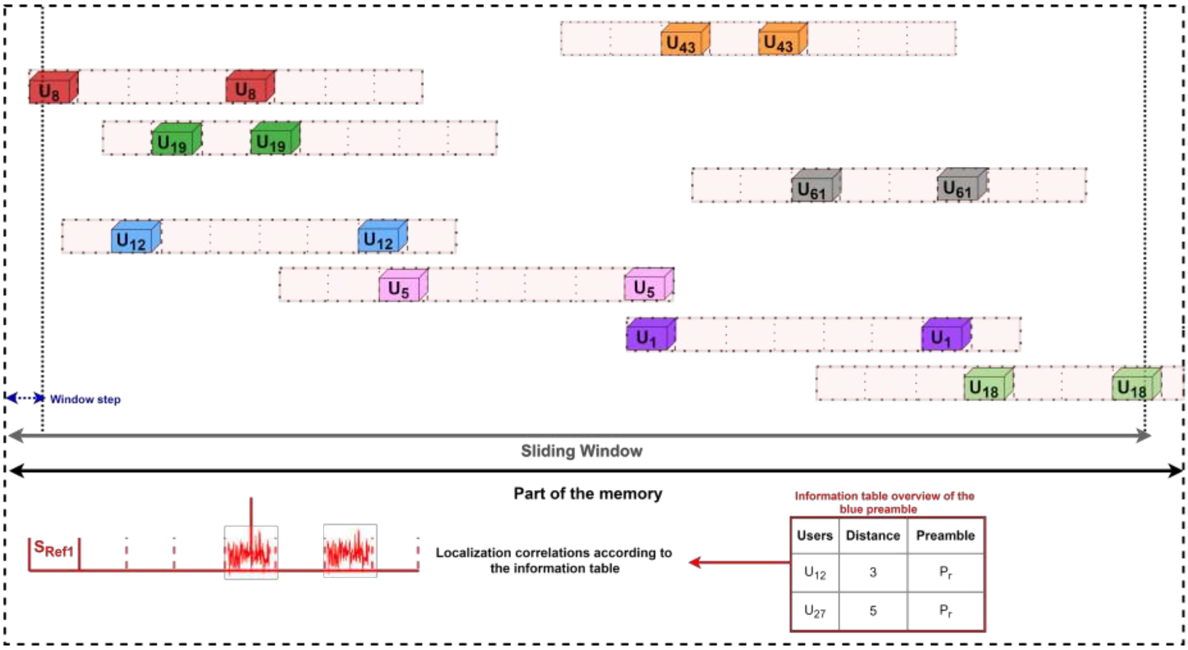


Figure 7.1: Asynchronous MARSALA contribution to ACRDA at reception.

In the perspective of reducing packets data localization complexity, R-SPOTiT introduced a possibility of having the information about all potential packets' positions available at the receiver side, without extra signaling information. The goal was to reduce the number of data correlation operations that are performed to localize packets' replicas. Therefore, a PRNG with shared seeds is exploited at both the transmitter and the receiver in order to generate the time slot positions of users. The seed can be static using the HID or dynamic through a combination between the latter and the frame identifier. If multiple preambles are used, the PRNG can also be used to select the preamble code for each user. In order to generate this information, seeds are taken individually, at each terminal, as an entry to the PRNG that gives as output the time slot positions for each one of them and the used preamble if applied. The receiver, knowing all identifiers of terminals attached to it (thanks to the logon phase) and the received frame identifier, constructs an information table that includes all possible positions on the frame and the potential preambles that could be detected there. This way, when a reference time slot is selected, fewer correlations are made, rather than the $N_S - 1$ correlations of MARSALA. Indeed, when a preamble is detected, these would refer to all replicas' positions of potentially collided packets on the reference time slot that use the same detected preamble. In the case of a single preamble, correlations are made over all replicas' positions of potentially collided packets on the reference time slot.

7.4 Asynchronous decoding characteristics with replicas combination

It has been shown in [GC05] that the approximation of the interference term to an AWGN channel is accurate when the number of colliding packets is high enough in an unfaded environment but remains imprecise when this number is weak. Here, we assume the SNR is approximated to the SNIR as it is the case for ACRDA. Thus, in the same way as in Chapter 3 the PER with respect to the SNIR of the used MODCOD is exploited to determine the packet decoding probability in terms of the number of total and partial interference.

In a rough way, a mean interference SNIR ($SNIR_{MIS}(u,r)$) can be determined to approximate the simulation results. The $SNIR_{MIS}(u,r)$ of a replica r belonging to a user u computes the mean number of interfering packets between symbols $C_{u,r}$. ECRA uses the same type of SNIR but with a combined less interfered packet from all replicas. The average mutual information (MI) over a replica has also been taken into account in ECRA for interference modeling. Actually, MI has mainly been used in terrestrial wireless networks and Multiple Input Multiple Output (MIMO) applications [SZS07] [Jen+08]. In [Cha+12] performance prediction methods have been investigated, in the context of Digital Video Broadcasting - Satellite Handled (DVD-SH) despite the significantly varying Land Mobile Satellite (LMS) channel. It proved that the mutual information based method offers a 0.1 dB better precision than the SNR based techniques in addition to a faster simulation time. However in this chapter, we consider only the equivalent SNIR mean value of a packet as a preliminary study. The mean number of interfering packets $C_{u,r}$ over a replica r of user u is expressed as follows

$$C_{u,r} = \frac{1}{N_{sym}} \sum_{s=1}^{N_{sym}} C_{sym}^{u,r}(s) \tag{7.1}$$

With: $C_{sym}^{u,r}(s)$ the number of interfering packets at symbol s level (of replica r of user u), and N_{sym} the number of symbols per packet.

As stated before, synchronous R-SPOTiT is able to retrieve more packets when CRDSA fails but with less complexity than MARSALA. We suggest that AR-SPOTiT can use the same mechanism as in synchronous systems given that signal combination takes place within the virtual frame between replicas of the same user independently from the others. Consequently, when AR-SPOTiT intervenes, signal combination takes place. Hence, the power of the packet of interest is expected to become significant. For two replicas r_1 and r_2, $C_{u,r}$ becomes $C_{ARS}(u) = C_{u,r_1} + C_{u,r_2}$ that we call the interference rate. The equivalent SNIR value experienced over both replicas' positions $SNIR_{MIS}(u)$ can be expressed according to C_{ARS} as follows:

$$\begin{aligned} SNIR_{MIS}(u) =& f\left(C_{ARS}(u)\right) \\ =& \frac{{N_R}^2 \times E_S/N_0}{E_S/N_0 \times C_{ARS}(u) + N_R} \end{aligned} \tag{7.2}$$

Furthermore, each value of $SNIR_{MIS}(u)$ approximated to $SNR(u)$ is associated to a certain PER value such as:

$$PER(u) = g\left(SNIR_{MIS}(u)\right) = g \circ f(C_{ARS}(u)) \tag{7.3}$$

The function g depends on the used MODCOD. In our case, it corresponds to the interpolation of the 1/3 turbo coded PER with QPSK modulation in an AWGN channel environment, and a packet length of 150 symbols.

It is also important to recall that a whole packet interference with a loop is unlikely to happen in this kind of scheme because of the asynchronous nature of transmissions. We have then considered that the interference power is always proportional to E_S/N_0. For a fixed E_S/N_0, each interference rate matches an SNIR value, which in turn corresponds to a PER value of the MODCOD. Knowing that the decoding probability is equal to $1 - PER$, Figure 7.2 shows the latter with respect to the interference rate for ACRDA and AR-SPOTiT with two replicas (AR-SPOTiT-2) and three replicas (AR-SPOTiT-3). A given value of the interference rate over all positions corresponds to $C_{ARS}(u)$ while a normalized one refers to $C_r(u) = C_{ARS}(u)/N_R$. The interference rate belongs to $\mathbf{R}^+$; for example a value of 2.5 means all interfered portions of a packet constitute a collision as long as two packets and a half. We can clearly see on the figure the considerable intake of replicas combination in AR-SPOTiT compared to ACRDA; the average number of tolerated interference per slot (normalized) is doubled in AR-SPOTiT with two replicas, and tripled with three replicas.

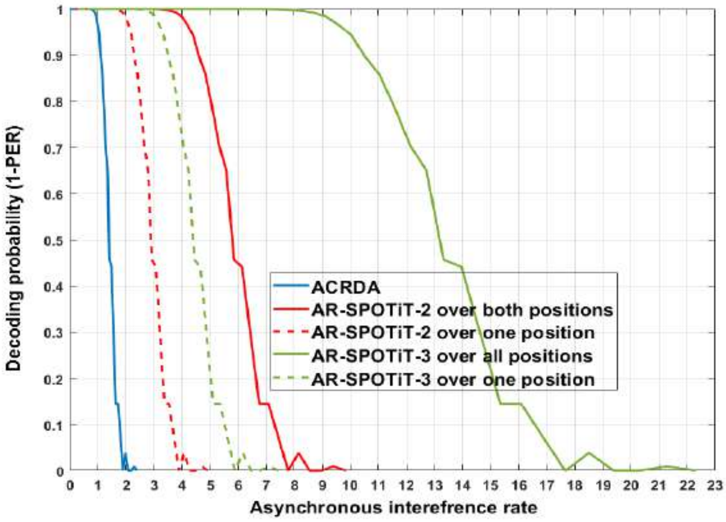


Figure 7.2: Decoding probability with respect to the number of interference of 150 symbols, over an AWGN channel of $E_S/N_0 = 10dB$.

7.5 Asynchronous Random SPOTiT

The main contribution of AR-SPOTiT is to present a way to localize replicas at reception after ACRDA is blocked. Indeed, by having a shared information between the transmitter and the receiver about all replicas' positions in their virtual frames, their localization should not require extra signaling information. We recall that a deadlock situation of ACRDA means that the latter can no longer retrieve packets. The difference between R-SPOTiT and AR-SPOTiT is explicated in Subsection 7.5.2. Here is a description of transmission and reception operations in AR-SPOTiT.

7.5.1 Transmission

At transmission, the way to put each packet in its corresponding virtual frame is governed by AR-SPOTiT. Indeed, replicas positions are selected using a PRNG that has the HID as an entry seed. The same seed is used to select one preamble among a set of pseudo-random codes. As a matter of fact, multiple preambles are considered in AR-SPOTiT. This reduces the complexity of the system as explained later in this section.

All seeds here are static (replicas' positions of each packet will always have the same positions at each new VF) because there is no virtual frame identification that could serve a dynamic seed, but this should not be an issue as the probability of having a repetitive loop remains very low due to the asynchronous nature of the system. Especially because the timing offset between the transmitted virtual frames is random as there is no synchronization between users.

7.5.2 Reception

An information table is constructed by the AR-SPOTiT receiver in which the distance between replicas of a same users u are included, in terms of the number of virtual slots between them. These distances are computed by the receiver after the replicas' positions are derived using the PRNG with the HID seeds that are available at reception.

There are $N_d = N_R(N_R - 1)/2$ distances for N_R replicas belonging the same packet (same VF). To compute these distances, first the virtual time slot positions $R_a \in [0; N_S - 1]$ of the N_R replicas are sorted in an ascending order $a \in [1 : N_R]$, thus $R_1 < R_2 ... < R_{N_R}$. Then, subtraction operations are performed between all sorted positions as follows:

$$\{d_{u,j}\} = \begin{cases} d_{u,1} = R_2 - R_1 - 1, d_{u,2} = R_3 - R_2 - 1, ..., \\ d_{u,3} = R_3 - R_1 - 1, ..., \\ d_{u,N_d - N_R + 2} = R_{N_R} - R_{N_R - 1} - 1, \\ d_{u,N_d - N_R + 3} = R_{N_R} - R_{N_R - 2} - 1, ..., \\ d_{u,N_d} = R_{N_R} - R_1 - 1 \end{cases} \tag{7.4}$$

With $d_{u,j} \in [0; N_S - 1]$ the j^{th} (with $j \in [1; N_d]$) distance between two replicas of a packet belonging to user u. An AR-SPOTiT information table with three replicas per packet ($N_d = 3$) and multiple preambles is presented in Table 7.1, with P_u the preamble associated to user u.

The distance, in slots, between the second and first replica of user u is $d_{u,1}$. The distance between the third and second replica of the same user is $d_{u,2}$, and finally $d_{u,3}$ is the distance between the third and first replica of user u. It is worth noting that the replicas are sorted in an ascending order; thus a first replica, after sorting, of any user is the one whose position is the closest to the beginning of the VF, while the position of the third replica (after sorting) is the furthest.

Table 7.1: AR-SPOTiT information table

Users	Distances between positions in slots			Preambles
$\mathbf{U_1}$	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$	P_{U_1}
$\mathbf{U_2}$	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$	P_{U_2}
$\mathbf{U_3}$	$d_{3,1}$	$d_{3,2}$	$d_{3,3}$	P_{U_3}
......	......	......	......	......
$\mathbf{U_{N_U}}$	$d_{N_U,1}$	$d_{N_U,2}$	$d_{N_U,3}$	$P_{U_{N_U}}$

AR-SPOTiT decoding mechanism can take place according to two options: it can first exploit the resulting ACRDA information when a preamble is detected but the packet is not decodable due to the high interference level; or rely on power detection to reveal a whole packet's presence when preamble detection fails in determining a clear presence of a preamble. We consider in this chapter of the dissertation that, even after ACRDA is blocked, preambles are correctly detected, and thus the beginning of the virtual time slot is known.

The localization process can now take place, and the correlations meant to localize replicas' positions are hence performed. They are made at distances, in slots obtained from the information table, to replicas' positions of packets with the detected preamble. In other words, the reference time slot used in synchronous MARSALA and R-SPOTiT is always, in AR-SPOTiT, set as the virtual time slot where a preamble is correctly detected. One can deduce that due to the fact that the virtual frames are independent one towards another. Having a single preamble will lead to make localization correlations, at the worst case, over the next $N_S - 1$ slots from where the single preamble has been detected. Furthermore, we set the correlations to be made in both directions, at distances before and after the virtual reference time slot because before decoding we do not have any knowledge on which replica's preamble is detected. Indeed, an a^{th} replica can be detected instead of first detecting an a^{th-1} replica. In this case, a correlation is necessary in the virtual time slot positions preceding and succeeding the RTS. This is valid as long as the $d_{u,j}$ is smaller than the distance between the virtual RTS and the beginning of SW, and smaller than the distance between the virtual RTS and the end of SW. Otherwise the correlations regarding a certain distance are made on a single direction. Therefore, using a single preamble that encounters $(2 \times N_S) - 2$ correlations at its worst case should be too complex to be considered in AR-SPOTiT.

Figure 9.7. shows an example of a portion of a receiver's memory with eight VFs of $N_S = 8$ slots, belonging to eight different users, and a sliding window. Even if the 8^{th} VF is not

completely on the SW, the latter will move forward step by step, after finishing interference cancellation on the current step, in order to browse the whole memory. Indeed, at the preamble detection phase, as and when a preamble is discovered, AR-SPOTiT will perform localization correlations on specific virtual time slots derived from the distances between replicas on the information table. SIC operation will afterwards take place to eliminate interference. When no more packets can be retrieved, the SW will move with one step at a time, and the mechanism of ACRDA with AR-SPOTiT is triggered again, until it reaches the end of the memory.

The difference between AR-SPOTiT and R-SPOTiT, in a multi-preamble environment, is that when a preamble is detected in AR-SPOTiT, the potential user candidates are all the packets using that preamble , which means that data correlations are made all over their other replicas' positions (derived from the different distances of the information table). However, in R-SPOTiT, an additional condition helps mitigating more the correlation complexity. Indeed, as transmissions are made over a commonly shared time-slotted frame between users, when a preamble is detected on a given slot, correlations are made over the other replicas' positions of potentially collided packets having the same preamble. These are the ones that have one of their replicas positioned on the analyzed slot. Thus, in order to choose one of the methods R-SPOTiT or AR-SPOTiT, a trade-off between the receiver's complexity, regarding packet search, which is lower in R-SPOTiT (the absence of synchronization in AR-SPOTiT makes it high), and the loop phenomenon mitigation of AR-SPOTiT should be considered.

7.5.3 Complexity case study

In this section, an overview of the localization complexity regarding detection correlations is considered with a proposed refinement of AR-SPOTiT and ECRA. The main idea is to evaluate the impact, on replicas localization, when having a shared information between each terminal and the receiver along with the usage of multiple preambles. Two replicas per packet are used. Each detected preamble is assumed to belong to the first replica which comes first on the VF (after sorting positions). Thus, the second replica search is only performed on one direction towards the right side for both algorithms AR-SPOTiT and ECRA.

AR-SPOTiT mechanism with a single preamble can be approached to the two phases ECRA but with additional information. Indeed, the first SIC phase of ECRA is performed by

ACRDA (first phase in AR-SPOTiT), and its second step that is mainly based on SC (Selective Combining), EGC (Equal Gain Combining) or MRC (Maximum Ratio Combining) is handled by AR-SPOTiT localization and combining, but with extra knowledge from the receiver's information table.

Let us assume having a correct energy detection to search for a packet-like entity at AR-SPOTiT and ECRA combining phase after ACRDA and ECRA's SIC phase have failed in retrieving more packets. At a detected entity starting position, preamble search can be performed with a unique code word used in an ECRA-like algorithm and N_{P} preambles in AR-SPOTiT. In the latter case, the first preamble to be detected above a given threshold will be taken into account for packet decoding. In other words, one correlation is made in ECRA and $\frac{N_{\mathrm{P}}}{2}$ as a mean number of preamble correlation value is taken for AR-SPOTiT. At this point, replicas localization of the packet of interest is necessary. On the one hand, ECRA will proceed with correlations at the next virtual time slots starting from the detected preamble, considering that replicas of the same user have the same timing offset on a VF. An assumption can be made here to stop at the first detected preamble spaced by an integer number of virtual time slots before going through the whole frame duration, thus a mean value of $\frac{N_{\mathrm{S}}}{2}$ correlations is considered. This assumption is particularly relevant in asynchronous transmissions because not only frames start at different random times, but each packet encounters a random timing offset. Therefore, the probability of having two packets with exactly the same starting position is unlikely to happen. On the other hand, AR-SPOTiT is able to define a number of specific virtual time slots N_s^p at distances derived from its information table where to perform preamble search of the detected code. This actually depends on the overall number of users attached to the gateway N_{U} and the number of preambles N_{P}. Also, AR-SPOTiT can also benefit from the assumption of stopping the preamble search at the first position where the same preamble is found, especially because an integer number of time slots as a distance between both replicas is implicitly taken into account in the algorithm of AR-SPOTiT. Consequently, a mean value of the number of preamble detection correlations is such as: $N_s^p = \frac{N_{\mathrm{U}}}{2N_{\mathrm{P}}}$ after having initially made $N_{\mathrm{P}}/2$ correlations. The total is thus $N_s^p + N_{\mathrm{P}}/2 = (N_{\mathrm{P}}^2 + N_{\mathrm{U}})/2N_{\mathrm{P}}$.

Besides, since the number of users registered at a given gateway N_{U} is known, the latter can reduce the number of preambles to be used by transmitters to a single one. As a matter of fact, when the estimated $(N_P^2 + N_U)/2N_{\mathrm{P}}$ for a given N_{P} and a given N_{U} exceeds $N_{\mathrm{S}}/2$, N_{P}

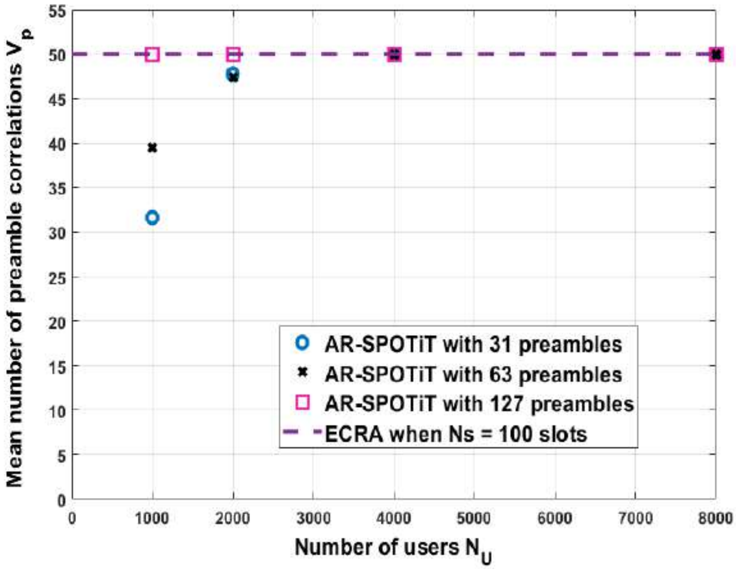


Figure 7.3: Mean number of preamble detection correlations meant to localize a packet's second replica with 100 time slots per VF.

can be set to 1. When a single preamble is used in AR-SPOTiT, all slots should be tested, but the first one to be compatible with the packet of interest will be taken into account. Thus, similarly, to ECRA, a mean value of $N_{\mathrm{S}}/2$ preamble correlations are performed.

To summarize, the total number of preamble correlations for one packet localization ν_p is expressed below in case of AR-SPOTiT and ECRA with two replicas per packet.

$$\nu_p = \begin{cases} \frac{N_S}{2} & \text{if ECRA} \\ \min\left(\frac{N_P^2+N_U}{2N_P}, \frac{N_S}{2}\right) & \text{if AR-SPOTiT} \end{cases} \tag{7.5}$$

Hereafter, we take a number of virtual slots per VF equal to 100 and 200 in order to compute the mean number of preamble detection correlations for ECRA as it only depends on N_{S}. In addition, different numbers of users $N_{\mathrm{U}} = \{1000, 2000, 4000, 8000\}$ and preambles $N_{\mathrm{P}} = \{31, 63, 127\}$ are taken to compute ν_p for AR-SPOTiT according to (7.5). The results are summarized in Figure 7.3 when $N_{\mathrm{S}} = 100$ and in Figure 7.4 when $N_{\mathrm{S}} = 200$.

When the number of slots is equal to 100, AR-SPOTiT requires less preamble correlations to localize a packet's replica than ECRA. This is true when $N_{\mathrm{U}} = 1000$ and $N_{\mathrm{U}} = 2000$ and with

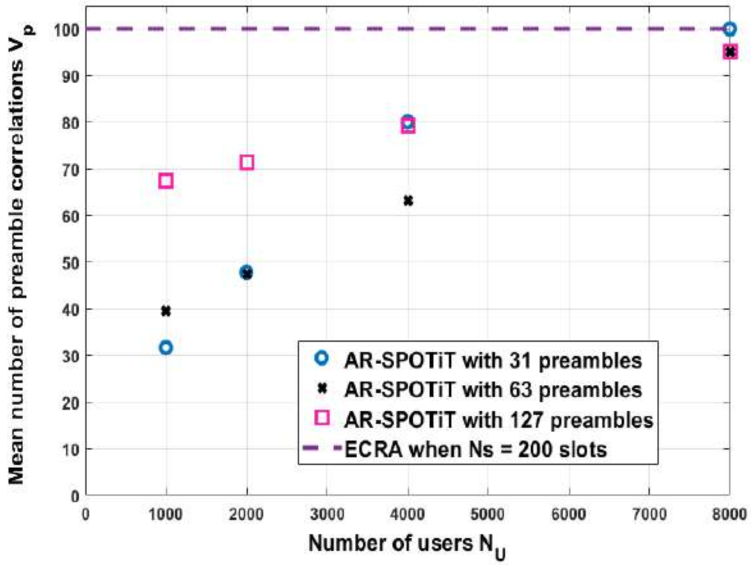


Figure 7.4: Mean number of preamble detection correlations meant to localize a packet's second replica with 200 time slots per VF.

a number of preambles equal to 31 and 63. Otherwise, the number of preambles will be reduced to a single one and then, ν_p for AR-SPOTiT will meet ECRA result. With 127 preambles, ECRA is less complex with any number of users, thus AR-SPOTiT meets its performance by using only one preamble. However, when the number of slots is equal to 200, AR-SPOTiT presents less preamble detection correlations meant to localize a packet's replica, compared to ECRA, for any number of users and preambles, except for the combination of 31 preambles with 8000 users. Therefore, a small number of slots is preferable to use when applying ECRA, and a number of preambles of 63 appears to be a good choice for AR-SPOTiT.

7.6 Experimental results and analysis

The considered scenarios in this section use the parameters presented in Section 7.2. We compare R-SPOTiT with AR-SPOTiT, in terms of PLR and throughput.

In R-SPOTiT, preamble search operations are performed at each slot during CRDSA. The symbols over which these operations are made depend on the timing offset and the guard interval. As a matter of fact, in this synchronous slotted case, the beginning of each slot and the maximum timing offset are known. However in AR-SPOTiT, preamble research, using correlators, is made along the sliding window, over the whole memory at each sample until a preamble is found during ACRDA. This can make the receiver's complexity significant if the number of preambles is big. For this reason, it is preferable to keep a small number of pseudo-orthogonal codes. We used here 63 Gold preambles according to the previous subsection. Also, we have chosen in our scenario $N_R = 2$ replicas per packet because it is less complex in terms of association correlations after the first replica is localized. Moreover, as asynchronous transmissions mitigate data loops between packets, the PLR error floor that CRDSA experiences in low loads is lower in ACRDA when the number of replicas is equal to 2 (see Figure 9.11). Indeed, with a channel load that is between 0.1 bits/symbol and 0.8 bits/symbol, CRDSA experiences an error floor that goes, approximately, from 3×10^{-6} to 6×10^{-4} respectively. For the same load, ACRDA presents an error floor that goes from 3×10^{-6} to 2×10^{-5}.

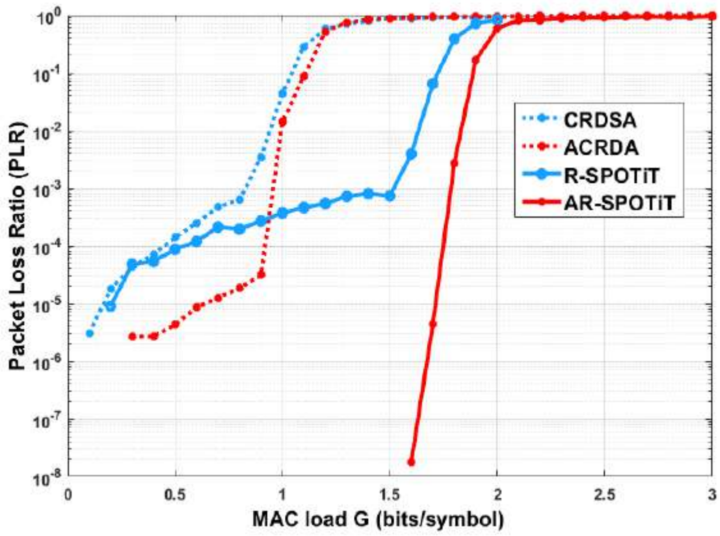


Figure 7.5: ACRDA with AR-SPOTiT, CRDSA with R-SPOTiT, ACRDA and CRDSA Packet Loss Ratio, 100 information bits, QPSK modulation, Turbo code of rate 1/3, AWGN channel and $E_S/N_0 = 10$.

On the one hand, we can notice (Figure 9.12) that starting from a channel load of 1.6 bits/symbol, the throughput is higher with AR-SPOTiT compared to R-SPOTiT that reaches its maximum of 1.64 bits/symbol at a channel load of 1.7 bits/symbol. Therefore, AR-SPOTiT is preferable in high loads. Furthermore, we can observe on the same figure that AR-SPOTiT significantly enhances the throughput when coupled to ACRDA; 1.8 bits/symbol reached at a channel load of 1.8 bits/symbol approximately versus 1 bit/symbol at a load of 1 bits/symbol

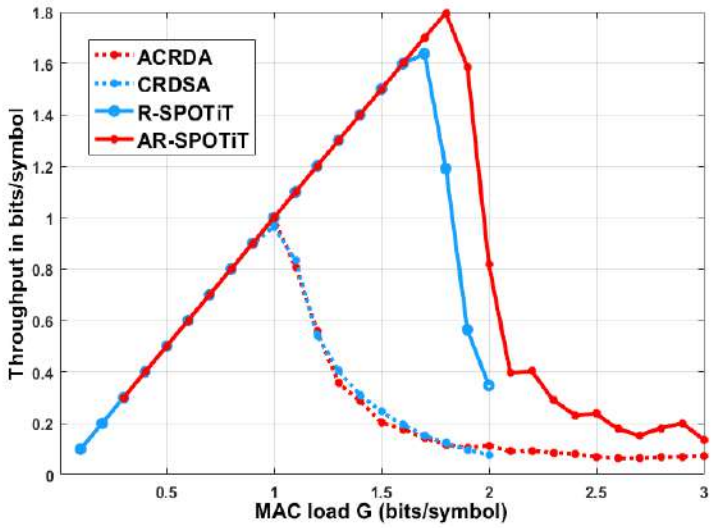


Figure 7.6: ACRDA withAR-SPOTiT, CRDSA with R-SPOTiT, ACRDA and CRDSA throughput, 100 information bits, QPSK modulation, Turbo code of rate 1/3, AWGN channel and $E_S/N_0 = 10$.

when ACRDA is used alone. On the other hand, AR-SPOTiT significantly reduces the PLR compared to R-SPOTiT (Figure 9.11) in addition to the disappearance of the error floor. It attains approximately 4.4×10^{-6} at a channel load of 1.7 bits/symbol unlike R-SPOTiT that reaches 6×10^{-2} at the same load. An asynchronous scheme offering better results than a synchronous one is mainly due to the type of interference that is partial. Nevertheless, a compromise that takes into account the overall complexity and system performance has to be set according to applications' needs.

7.7 Summary and conclusion

In this Chapter, we presented an asynchronous version of R-SPOTiT that comes as a complementary process to ACRDA with better performance. Indeed, while ACRDA reaches a throughput of 1 bits/symbol, 1.77 bits/symbol is attained with AR-SPOTiT, which is higher than the throughput reached by R-SPOTiT in addition to the elimination of the error floor of the PLR.

The accomplishment of an asynchronous version of R-SPOTiT, which was originally designed to reduce MARSALA's complexity, allowed us to introduce a new way to localize replicas as a second option along with ECRA. In addition, according to Section 7.5.3 results and parameters, AR-SPOTiT significantly reduces the preamble detection correlations.

Chapter 8

Conclusion and perspectives

8.1 Conclusion

This thesis provided an overview of Recent ALOHA based RA protocols for satellite communications. More particularly, our focus is put on the leading technique CRDSA, which was subject to many enhancement schemes. Despite all of the proposed variants of CRDSA that decided to add or modify some parameters, MARSALA scheme proposed to keep CRDSA as a main decoding process and intervenes only when the latter incurs the deadlock (no solvable packets); generally in high channel loads. Such an action intends to solve some of the configurations (collided packets) in order to make CRDSA operational again. This was a major contribution that offered considerable gain in system performance in terms of packet loss ratio and throughput. It adds however a processing complexity at the receiver due to its operations that unlock CRDSA. This involves the localization of replicas through correlations that allow to benefit from the packets' replications using signal combination. Indeed, since the positions of replicas belonging to a given packet are only known in CRDSA after one of them is decoded, if a deadlock situation occurs (no clean replicas) then the whole decoding process stops. This is where MARSALA tries to get the replicas' positions of at least one packet using correlations between a reference time slot (containing one replica in collision with other packets) and the remaining slots on the frame. The goal is to further combine all replicas of the same packet in order to have a potentially higher SNIR for a better chance of decoding. After SIC, the frame can consequently reveal clean packets and thus CRDSA is retriggered again.

Throughout this thesis, we particularly tried to overcome two main challenges that a complementary treatment to CRDSA can encounter. The first issue is the computational complexity regarding packet localization at reception after CRDSA is locked. The second problem is the loop phenomenon when the number of replicas is small, which causes an error floor at the PLR performance. These issues are anyhow connected; to keep the localization complexity low, a small number of replicas is required, but the latter induces a higher loop phenomenon occurrence. To solve this combined issue, we addressed the packet localization part, where until now, there were no prior information at the receiver, on any potential packet's position. The major contribution of this thesis is the introduction of a novel technique SPOTiT based on a shared localization information between the receiver and each of the transmitters, whose study is summarized below.

The proposed Shared Position Technique for Interference random Transmissions (SPOTiT) relies mainly on providing the receiver information about all potential packets' positions on a frame/virtual frame, if they transmit. Its main purpose is to refine the localization complexity by reducing the number of candidate time slots/virtual time slots where to perform correlations meant to localize packets' replicas. In addition, pseudo-orthogonal preambles can also be used for better results. SPOTiT is operational at the same level as MARSALA; i.e. when CRDSA can no longer decode new packets. Three solutions are then proposed. The first one, R-SPOTiT, aims to mitigate the localization complexity by making potential packets' positions available at reception, but without any additional signaling information. The second solution, S-SPOTiT on the other hand, involves a signaling information that is sent to each user about the positions he should transmit in after having defined an optimal loop-free disposition. In addition to a potentially simple localization, S-SPOTiT targets the loop phenomenon as well. As the latter problem is naturally mitigated with asynchronous transmissions, because of the nature of interference which is mostly partial, an adaptation of SPOTiT in such an environment seemed compelling. AR-SPOTiT as a complementary process to ACRDA (An asynchronous version of CRDSA) has then been defined. It does not require the signaling information that served S-SPOTiT to eliminate the loop phenomenon, but it is more complex at reception because of the asynchronous nature of transmission.

R-SPOTiT

To make the receiver be aware of the packets' positions, R-SPOTiT relies on generating this information through a PRNG whose seeds are known ID information. This way, the receiver does not have to perform correlations over the whole frame to localize a packet, but only on certain time slots where a potential packet candidate could have transmitted. This includes potentially collided packets on the time slot of reference, using the same preamble. Results show that R-SPOTiT can achieve the same system performance as MARSALA in terms of PLR and throughput but with significantly lower data correlation complexity. Furthermore, an in-dept analysis we performed regarding the overall complexity, including preamble detection as well as data correlations, showed that a system with R-SPOTiT is still a better choice than MARSALA. An interesting result recommends using R-SPOTiT with a small number of preambles if these are pseudo-orthogonal (such as Gold codes).

S-SPOTiT

The purpose of S-SPOTiT is to eliminate the loop phenomenon and keeping the lowest localization complexity possible. As such, we defined a level-based distribution of time slot positions and the preambles to use for a community of users in a way that no loops are created, in addition to a potentially simple localization. The latter is insured by having one of the replicas of a given packet using a unique preamble on its time slot position; which means that no localization correlations are necessary if this preamble is detected. This feature is valid as long as all packets having the same preamble at previous levels have been decoded. As a result, a significantly low PLR with no error floor is observed. A complete Scheme of S-SPOTiT, with irregular parameters, has been provided to adapt its optimal distribution to any real system scenario. In addition, S-SPOTiT has been extended towards a dynamic loop-free system regarding the number of users and slots, which adds an interesting flexibility of the network scalability.

AR-SPOTiT

AR-SPOTiT can be perceived as an asynchronous version of R-SPOTiT, as well as an alternative to S-SPOTiT regarding the loop phenomenon mitigation. It also presents a way of localizing packets in an asynchronous environment with a potentially lower complexity than ECRA. It exploits the distance between replicas within a virtual frame of ACRDA and the PRNG concept of R-SPOTiT to achieve better system performance. As a matter of fact, it offers a higher throughput than the R-SPOTiT and S-SPOTiT, and with better PLR performance. We also concluded that AR-SPOTiT is less complex than ECRA in terms of preamble detection correlations.

8.2 Perspectives & remaining challenges

Despite the primary promising results, obtained with the three derivatives of SPOTiT in the synchronous and asynchronous environments, enhancement schemes can be expected.

The complexity assessment of R-SPOTiT that we have accomplished throughout this thesis presented results based on the pseudo-orthogonal Gold code preambles. However, other types of codes might offer better detection abilities, through which we might observe improvements regarding the localization complexity mitigation. For instance, Zadoff-Chu sequences used in 3GPP Long term Evolution (LTE), present interesting orthogonality properties, and they are subject to many proposed enhancement strategies in random access-like systems.

In addition to the loop-phenomenon mitigation that S-SPOTiT offers, the packet localization complexity should be low due to its property of having one of the replicas using a unique preamble on its time slot position. However, when this feature is broken because of irregular parameters (the number of slots, the number of preambles and the number of users), as explained in Chapter 5, it would be interesting to assess the encountered localization complexity and compare it with R-SPOTiT using the same parameters. This should provide an overall idea of which system to use, depending on chosen compromises (loop phenomenon, complexity, signaling information, PLR and throughput). On the same path, AR-SPOTiT should also be part of the complexity assessment as it also participate in drastically reducing the loop phenomenon without signaling information and offering a higher throughput. In return, the process of looking for preambles in an asynchronous environment turns out to be computationally consuming because of the absence of slotted common frames, as this requires to browse the whole memory at sample/symbol level.

The condition of equally distributing preambles on the slots for the extension of S-SPOTiT is not always respected with a Round-robin technique. Therefore, an in-depth study is necessary to define the best solution.

Even though the impact of real channel conditions and estimation of synchronization parameters have been studied for MARSALA in [Zid16], it would be interesting to check their validity with SPOTiT and its variants. On the one hand, a strategy based on combining the use of the EM (Expectation Maximization) algorithm at the preamble and postamble along with an initialization of the channel parameters that relies on an auto-correlation operation, instead of a random initialization has shown good results in terms of PER compared to the traditional EM. On the other hand, an estimation using pilot symbol assisted modulation has also been proposed for MARSALA in the same research work. When these are combined with a joint estimation and decoding, which should enhance the SIC performance, to estimate the different channel parameters, very low loss of the system performance is observed.

Also, the MRC combining technique assessed by ECRA in an asynchronous environment and MARSALA for synchronous transmissions, showed a significant enhancement of the throughput. Furthermore, packet power unbalance using a half normal distribution in MARSALA presented the best results when coupled to MRC. Therefore, we think applying MRC to the different variants of SPOTiT with packet power unbalance is expected to be considerably beneficial in terms of gain in performance.

Apart from varying the different system parameters or introducing real channel conditions, such as Poisson traffic, which we consider to be an important aspect to check before implementation, there are still open doors to exploit in order to have the best theoretical version of SPOTiT. Let us take for instance the critical region of the throughput curve in any of the variants of SPOTiT. This should correspond to the part of the curve that comes after the critical point where the the throughput collapses (1.7 bits/symbol in R-SPOTiT). We envision that having a retransmission mechanism such as HARQ (Hybrid automatic repeat request) would smooth the throughput collapse, especially in non time-critical applications. The information table of SPOTiT can be exploited in order to define which among all users should retransmit an additional replica and in which time slot, according to the application's characteristics. Moreover, the maximum number of retransmissions per user should be set for a given scenario.

Finally, this thesis tackled some of the issues of ALOHA based RA protocols. However, the very promising spread spectrum techniques, like E-SSA and its variants, are also subject of interest to inspect and might take part of our future work.

8.3 List of Publications

Patent

- S. Zamoum, M. Gineste, J. Lacan, M-L. Boucheret et J-B. Dupe, "procédé et système de transmission de packets de données à travers un canal de transmission (RA) à accès aléatoire", N° 071277 FR RQDLV 14-05-18 YTA-LRE, May 2018.

International conferences

- S. Zamoum, J. Lacan, M-L. Boucheret, J-B. Dupe, M. Gineste, "Shared Position Technique for Interfered Random Transmissions in Satellite Communications", 9th Advanced Satellite Multimedia Systems Conference and the 15th Signal Processing for Space Communications Workshop (ASMS/SPSC), 2018.
- S. Zamoum, J. Lacan, M-L. Boucheret, M. Gineste, J-B. Dupe, Deterministic "Distribution of Replicas Positions for Multiuser Random Transmissions in Satcoms", IEEE Global Communications Conference (GLOBECOM), 2018.
- S. Zamoum, J. Lacan, M-L. Boucheret, J-B. Dupe, M. Gineste, "Asynchronous Packet Localization with Random SPOTiT in Satellite Communications", The 22nd International Symposium on Wireless Personal Multimedia Communications (WPMC), accepted, 2019.

Journals

- S. Zamoum, J. Lacan, M-L. Boucheret, J-B. Dupe, M. Gineste, "Complexity Analysis of Recent Aloha Random Access Techniques in Satellite Communications", International Journal of Satellite Communications and Networking, Submitted, 2019.

Chapter 9

Résumé étendu en Français

9.1 Introduction

9.1.1 Context & Motivation

Depuis le lancement du premier satellite artificiel (Spoutnik), en 1957, les télécommunications par satellites ont fait l'objet de nombreuses études. Il y a eu de nombreux développements et améliorations de logiciels et de matériel concernant chacun des segments de l'espace, du sol et de contrôle. D'autant plus qu'un potentiel considérable est perçu par les différents acteurs en télécommunications. En effet, la couverture mondiale ainsi que les larges capacités que les satellites peuvent offrir motivent les chercheurs à proposer des services omniprésents et peu coûteux qui répondent aux exigences accrues pour une connectivité mondiale de bonne qualité. Selon le Bureau des Nations Unies pour les affaires spatiales (UNOOSA) [OS19], il y a aujourd'hui plus de 5000 satellites en orbite autour de la terre, autour d'autres planètes (Mars, Vénus), autour des satellites naturels et autour d'astéroïdes (Ryugu) également. En effet, ils sont utilisés pour diverses missions spatiales en plus des différentes applications terrestres. Plus précisément, dans ce dernier cas, les satellites peuvent résoudre le problème des zones blanches où il n'y a pas d'infrastructure pour une couverture cellulaire en raison d'un environnement difficilement accessible ou de ressources insuffisantes. Dans certains pays en voie de développement et dans des régions mal desservies, une solution par satellite semble aussi être coûteuse. Au Sénégal par exemple, un réseau de communication dans des zones blanches, basé sur une communication radio longue portée, utilisant les bandes de fréquence ISM (Industrial, Science and Medical), a été développé pour les éleveurs [DIA17]. Néanmoins, des efforts constants sont déployés sur le terrain pour offrir des services par satellite de meilleure qualité et moins chers. Selon The Economist [SM16], des petits satellites bon marché pourraient transformer l'industrie spatiale. Cinq satellites d'un poids de 30 kilos et d'une longueur de 30 cm sont construits par Planet [1] à partir de smartphones et d'autres composants d'appareils en une semaine.

Les applications satellites entraînant une grande population d'utilisateurs peuvent être critique en termes de temps de communication, compte tenu du temps d'aller-retour (RTT), en plus d'être coûteuses en termes de ressources. Des stratégies d'accès ont alors été définies dans différentes normes, permettant d'organiser efficacement les communications. Les utilisateurs doivent obéir aux différents protocoles des multiples dérivées des normes de diffusion vidéo numérique (DVB: Digital Video Broadcasting) [42197]; [V1.15a]; [V1.15b]; [V1.11]; [79003]; [79003]; [A1511]. Contrairement à l'accès aléatoire, l'assignation de la demande d'accès multiple (DAMA) nécessite des demandes d'allocation de ressources, ce qui ajoute de la signalisation en plus du délai de communication concernant la transmission des informations utiles. Par conséquent, l'accès aléatoire est plus approprié pour les transmissions sporadiques avec de courts paquets. Cependant, de nombreux défis se posent et doivent être relevés.

Nous visons dans cette thèse les techniques d'accès multiples sur la liaison retour d'une communication par satellite (entre un terminal utilisateur et une passerelle) où plusieurs terminaux transmettent des paquets sur la même bande passante de fréquence. Parmi les différentes contributions dans l'accès aléatoire et de l'accès dédié, ont émergé certaines techniques efficaces, offrant de bonnes performances du système. Les méthodes d'accès aléatoire basées sur le protocole ALOHA sont spécifiquement ciblées. Nous nous sommes donc intéressés aux solutions récentes, basées principalement sur une transmission multi-réplique avec suppression d'interférences successives (SIC) à la réception dans des environnements synchrones et asynchrones. D'une part, CRDSA (Contention Resolution Diversity Slotted Aloha) améliore considérablement le débit grâce à l'utilisation de la redondance des paquets et du SIC sur des trames bien définies. Il permet de gérer efficacement les collisions de paquets jusqu'à une certaine charge de canal (en termes de nombre d'émetteurs). Ainsi, le débit s'effondre en cas de charges élevées lorsqu'il n'est plus possible de récupérer des paquets (deadlock), ou lorsque seuls quelques paquets sont décodés. Pour faire face à ce problème, MARSALA (Multi-Replica Decoding using Correlation based Localisation) a proposé d'intervenir en complément de CRDSA lorsque ce dernier est dans une situatoin de blocage. Tout d'abord, il localise les répliques de paquets en collision sur une tranche de temps de référence choisi au hasard à l'aide de corrélations. Ensuite, les répliques appartenant au même paquet sont

[1]Une société privée américaine pour une imagerie terrestre constante, dont la mission est de rendre le changement global visible, accessible et réalisable.

combinées. Une étape d'association destinée à rassembler toutes les répliques de paquets ainsi qu'une estimation des paramètres de canal sont nécessaires pour maximiser le gain de la combinaison. Plus précisément, cela permet d'avoir une plus grande puissance du paquet d'intérêt avec une meilleure probabilité de décodage. Néanmoins, la procédure de localisation de MARSALA, qui effectue des opérations de corrélation de paquets entiers, ajoute une complexité considérable au récepteur. En fait, des corrélations globales sont d'abord effectuées entre une tranche de temps de référence arbitraire et le reste des tranches sur la trame afin de localiser toutes les répliques des paquets en collision. Ensuite, des corrélations supplémentaires sont nécessaires pour associer les répliques d'un même paquet. Celles-ci sont effectuées entre un signal combiné et les positions des pics de corrélation restantes résultant de la localisation globale précédente. Malgré la complexité de MARSALA, les performances sont nettement meilleures, en termes de taux de perte en packet (PLR: Packet Loss Ratio) et de débit, par rapport à CRDSA.

9.1.2 Contributions

Tenant compte du gain apporté par MARSALA, nous avons cherché à réduire la complexité de localisation des paquets en ce qui concerne les corrélations lorsque CRDSA n'est plus en mesure de récupérer des paquets. Jusqu'à présent, le récepteur ne connaît pas les positions choisies par les utilisateurs sur une trame donnée. Avec la nouvelle méthode SPOTiT (Shared POsition Technique for Interfered Random Transmissions) que nous proposons, une connaissance partagée entre le récepteur et chacun des terminaux est introduite. Les informations partagées concernent les tranchess de temps sur lesquelles chaque terminal transmet ses répliques ainsi que le préambule à utiliser (parmi un ensemble de codes pseudo-orthogonaux). La première version de SPOTiT appelée R-SPOTiT (pour Random SPOTiT) vise principalement à réduire la complexité du processus de localisation des répliques de l'ancienne technique MARSALA. Il présente un système moins complexe sans dégrader les performances et sans aucune information de signalisation supplémentaire. Il utilise l'information commune générée entre un émetteur et le récepteur concernant les positions et les préambules potentiels des répliques, pour cibler un nombre inférieur de tranches de temps pour effectuer les corrélations de localisation. De plus, une analyse détaillée du nombre d'opérations de corrélation nécessaires pour localiser les répliques de paquets en collision est fournie dans ce travail. Les scénarios comportant un seul préambule et plusieurs préambules, considérant la détection de ceux-ci au niveau de CRDSA, sont pris en compte pour l'évaluation la complexité du système dans son ensemble. Un schéma optimal pour R-SPOTiT est déduit en fonction des résultats de simulation des différents scénarios qui ont été évalués.

Une autre contribution de cette thèse propose Smart SPOTiT (noté S-SPOTiT) comme solution hybride qui mélange à la fois le DAMA (Demand Assignment Multiple Access) et l'accès aléatoire afin de diminuer le plancher d'erreur du PLR. En fait, une gestion centralisée des positions des répliques et des préambules à utiliser est faite de telle sorte qu'aucune boucle ne soit créée. Le phénomène de boucle se produit lorsque deux ou plusieurs paquets sont transmis exactement aux mêmes positions, ce qui crée un plancher d'erreur au niveau du PLR facilement observable avec MARSALA et R-SPOTiT. Cette version de SPOTiT nécessite une information de signalisation qui n'est envoyée qu'une seule fois aux émetteurs selon une distribution optimale. Ce dernier inclut une disposition des emplacements des paquets sur une trame sans boucles, en les associant à des préambules, et en permettant une localisation simple en même temps. En effet, cette distribution s'assure qu'une des répliques du paquet est la seule qui puisse être transmise dans sa position avec un préambule donné ; cela veut dire qu'à chaque fois qu'un préambule est détecté sur une position donnée où un utilisateur unique aurait pu l'utiliser, son paquet est localisé sans aucune opération de corrélation. S-SPOTiT a donné des résultats prometteurs, notamment avec la disparition du plancher d'erreur du PLR. Il convient de noter que la distribution optimale de S-SPOTiT repose sur des paramètres du système, tels que le nombre de tranches temporelles et le nombre de préambules, qui sont sous la forme d'une puissance de deux. Il a donc semblé important de dériver un modèle irrégulier avec des paramètres quelconques afin d'avoir un schéma complet de S-SPOTiT. De plus, un schéma dynamique sans boucle de S-SPOTiT, qui offre une flexibilité quant au nombre d'utilisateurs, est proposé. Nous rappelons que le problème central que S-SPOTiT a abordé est le phénomène de boucle. Cependant, cette dernière est moins importante en transmission asynchrone. C'est pourquoi R-SPOTiT est considéré dans le cas asynchrone.

Les solutions RA asynchrones se caractérisent par l'absence de surcharge de signalisation en ce qui concerne les informations de synchronisation. Etant donnée que CRDSA s'impose comme étant une technique de pointe dans le domaine des transmissions synchrones, la définition d'une version asynchrone de cette méthode était cruciale. ACRDA (Asynchronous Contention Resolution ALOHA) représente la méthode asynchrone la plus proche de la CRDSA. CRDSA et ACRDA se trouvent dans une situation de blocage lorsqu'il n'est plus possible de récupérer des paquets en raison d'une charge élevée du réseau. Dans les transmissions synchrones, nous rappelons que MARSALA permet de débloquer certaines des configurations de blocage qui relanceraient CRDSA. Dans les transmissions asynchrones, ECRA (Enhanced Contention Resolution Aloha) utilise différentes techniques de combinaison des répliques de paquets pour offrir des performances système élevées en termes de PLR et de débit. Ces techniques MARSALA et ECRA peuvent être coûteuses en termes de complexité de localisation pour le récepteur. C'est la raison pour laquelle R-SPOTiT a été défini dans le cas synchrone. En conséquence, nous proposons dans cette thèse AR-SPOTiT, une conception asynchrone de R-SPOTiT, comme étant une méthode complémentaire à ACRDA. Elle introduit un moyen de localiser les répliques sur leurs trames virtuelles d'une façon moins complxe et avec des performances système nettement plus élevées, en plus de l'atténuation du phénomène de boucle.

9.2 Random Shared Position Technique for Interfered Random Transmissions

Dans cette section, la solution proposée pour l'accès multiple R-SPOTiT est décrite et évalué lorsque différents préambules pseudo-orthogonaux sont utilisés. Son principe général consiste à pouvoir communiquer, au préalable, au récepteur les positions des tranches de temps des paquets potentiellement transmis et les préambules associés sans informations de signalisation supplémentaires. Cela devrait atténuer la complexité de localisation car les itranches de temps candidats où il est nécessaire d'effectuer les corrélations de données est réduit. Ainsi, R-SPOTiT décrit un moyen d'organiser les paquets sur la trame et leur associer des préambules. Les aspects de transmission et de réception sont détaillés ci-dessous.

9.2.1 Transmission

La solution proposée consiste à utiliser un générateur de nombre pseudo-aléatoire (PRNG) selon deux modes:

Graine fixe: l'identifiant matériel (HID) qui est propre à chaque terminal est connu par le récepteur grâce à la phase de connexion. En effet, chaque abonné utilise son identifiant unique pour se connecter au système. Ainsi, lorsque le HID est utilisé comme graine d'entrée pour le PRNG, le récepteur et chacun des utilisateurs déterminent les mêmes tranches de temps dans la trame où envoyer les répliques et le préambule à utiliser à chaque transmission.

Graine dynamique: dans certaines applications où plusieurs utilisateurs génèrent les mêmes positions et transmettent successivement sur les mêmes trames, ils créent une boucle insoluble. Une boucle se produit lorsque deux paquets ou plus sont transmis exactement aux mêmes positions sur une trame, ce qui rend la puissance du paquet d'intérêt égale à la puissance de l'interférence après la combinaison. À la suite du scénario décrit précédemment, un échec continu de décodage des packet se produira. Pour y remédier, un choix dynamique de positions et de préambules est introduit. Une combinaison dynamique peut être utilisée afin de renouveler le choix des tranches de temps et du préambule à chaque trame pour chaque terminal. Ceci impliquera un identifiant incrémental en tant que graine d'entrée pour le PRNG. Par exemple, cela peut être obtenu en ajoutant U_{ID} le HID du terminal à F_{ID} l'identifiant de la trame. Par conséquent, cette combinaison dynamique entre l'identifiant HID et l'identifiant de trame évite une boucle continue, pour deux utilisateurs ou plus, en cas de transmissions successives et simultanées.

9.2.2 Réception

Le récepteur calcule toutes les positions des répliques et le choix du préambule de chaque abonné en utilisant les graines prédéterminés dans le cas fixe ou dynamique pour ensuite créer une table d'informations. Cela signifie que le récepteur connaît tous les utilisateurs potentiels et leurs préambules qui sont capables de transmettre des paquets à chaque tranche de temps dans la trame. De plus, la caractéristique pseudo-orthogonale des préambules est utilisée pour réduire le nombre potentiel d'utilisateurs sur chaque itranche de temps. En fait, un préambule détecté sur une position donnée indiquera un certain nombre d'utilisateurs ayant le même préambule, à partir de la table d'informations du récepteur. Ces utilisateurs sont ceux qui pourraient transmettre des données sur cet tranche de temps qui est analysé.

Une fois la localisation réussie, la combinaison de signaux est effectuée entre des tranches de temps contenant des répliques du même paquet avant la démodulation et le décodage. La figure 9.2 résume les principales différences entre R-SPOTiT et MARSALA, considérées comme des traitements complémentaires du CRDSA (9.1).

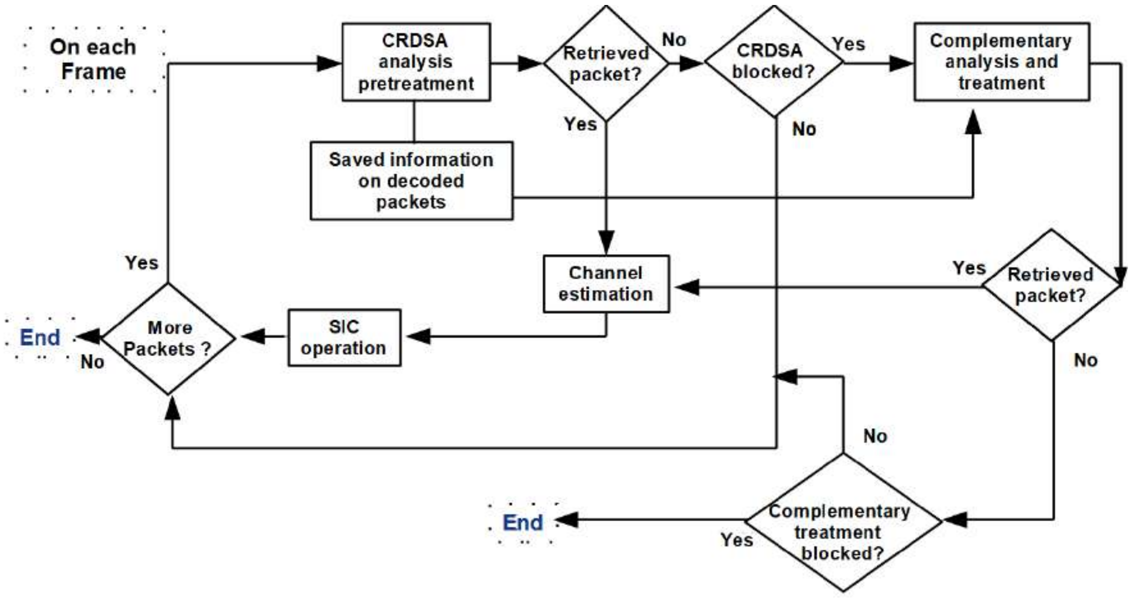


Figure 9.1: CRDSA with complementary treatment process

9.2.3 Débit et taux de perte en paquets

Afin d'évaluer les performances globales du système, nous nous intéressons au débit, au taux de perte en paquets (PLR). De plus, une comparaison quant à la complexité de localisation entre R-SPOTiT et MARSALA avec deux répliques par paquet est réalisée. Le PLR et le débit sont obtenus à travers une abstraction de couche physique en utilisant la courbe du taux d'erreurs de packet (PER) du MODCOD utilisé avec un calcul SNIR équivalent. Nous avons considéré que l'information utile des paquets sont construites à partir de 100 bits modulés avec une modulation QPSK et un Turbo code (codage 3GPP) avec un taux de 1/3, et sont transmis sur une trame de 100 tranches temporelles sur la même fréquence. Nous supposons que le modèle de canal est un AWGN avec un E_S/N_0 de 10 dB. Des séquences pseudo-orthogonales de longueur 31 sont utilisées comme des préambules. 2000 utilisateurs connectés à la passerelle sont considérés comme potentiels émetteurs. En supposant que nous ayons une estimation parfaite du canal, les Figures 9.3 et Figure 9.4 affichent les performances de R-SPOTiT-2 en termes de débit et de PLR par rapport à CRDSA avec deux réplicas par paquet et MARSALA-2. Lorsque la seule méthode basée sur la détection de préambule est utilisée pour décoder un paquet, les deux préambules doivent être détectés. Dans ce cas, aucune corrélation de localisation de données n'est nécessaire pour R-SPOTiT. En effet, comme les répliques d'un même paquet ont le même décalage temporel dans une trame, la

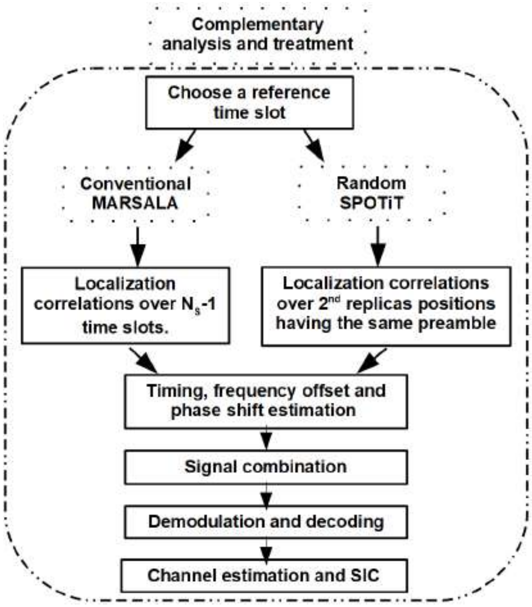


Figure 9.2: Explained complementary treatment process to CRDSA

distance qui les sépare est un nombre entier de tranches, ce qui confirme la présence du paquet sans corrélations supplémentaires. En conséquence, un débit de 1.5 bits / symbole est atteint tandis que MARSALA atteint 1.64 bits / symbole. Néanmoins, les performances peuvent être améliorées lorsque l'information sur les préambules détectés est considérée pour effectuer des corrélations de localisation des données. En effet, une seule détection des deux préambules du même paquet est nécessaire pour effectuer des corrélations de données sur les positions des deuxièmes répliques de paquets potentiellement en collision ayant le même préambule détecté. Compte tenu du résultat de décodage des itérations précédentes CRDSA et R-SPOTiT, les paquets en collision potentiels qui ont été décodés seront supprimés des corrélations à exécuter.

9.2.4 Evaluation de la complexité de localisation

Dans notre analyse globale de la complexité de localisation, nous nous concentrons sur la partie de détection des préambule qui est nécessaire au processus de décodage CRDSA et sur la localisation des répliques requise dans R-SPOTiT ou MARSALA (le traitement complémentaire) avant la combinaison du signal. L'ensemble du processus est terminé dans l'une des trois conditions suivantes; lorsque CRDSA seul a décodé tous les paquets de la trame, lorsque CRDSA plus le traitement complémentaire ont décodé tous les paquets ou quand ils

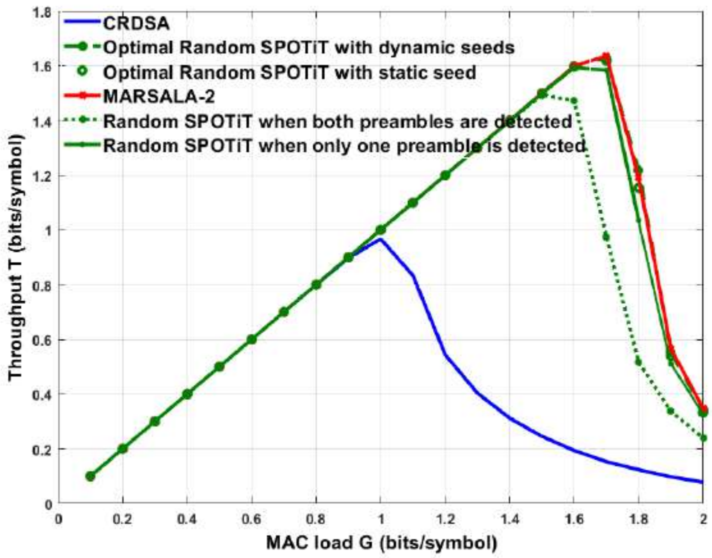


Figure 9.3: Throughput comparison between R-SPOTiT, MARSALA-2 and CRDSA, $E_S/N_0 = 10$ dB, 100 slots per frame, QPSK modulation, Turbo coding of rate 1/3 and equipowered packets of 100 bits and $N_R = 2$ replicas per packet.

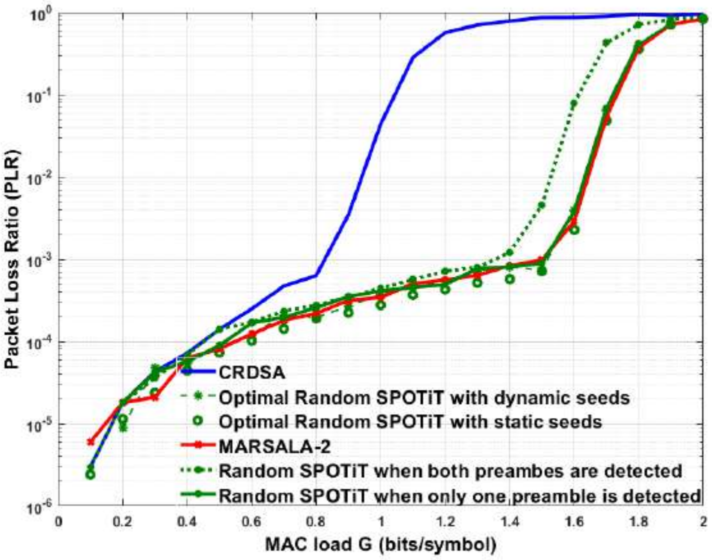


Figure 9.4: Packet Loss Ratio comparison between R-SPOTiT, MARSALA-2 and CRDSA, $E_S/N_0 = 10$ dB, 100 slots per frame, QPSK modulation, Turbo coding of rate 1/3 and equipowered packets of 100 bits and $N_R = 2$ replicas per packet.

ont décodé le nombre maximal de paquets avant que les deux ne soient bloqués en raison du nombre élevé de collisions dans les hautes charges.

En résumé, le nombre total de corrélations C_T effectuées par trame inclut les opérations de détection des préambules C_{P} et les opérations de localisation de données C_{D}. C_{P} est pris en compte durant toutes les itérations de CRDSA, avant et après le traitement complémentaire et jusqu'à ce que le système entier soit bloqué. C_{D} est exécuté par $R-SPOTiT$ ou MARSALA pour redéclencher CRDSA chaque fois qu'il est bloqué jusqu'à ce que tout le système atteigne une situation de blocage. Le nombre total de corrélations sur une trame C_T est décrit ci-dessous:

$$C_T = \sum_{\delta=1}^{\Delta} \left(\sum_{it=1}^{N_{\mathrm{it}}} C_{\mathrm{P}}(\delta, it) + \sum_{\lambda=1}^{\Lambda(\delta)} C_{\mathrm{D}}(\delta, \lambda) \right) \tag{9.1}$$

Avec δ, l'index pour l'analyse de la trame. Δ est la valeur maximale de δ qui est atteinte lorsque tout le système est bloqué. Sa valeur peut varier d'une trame à l'autre. N_{it} est le nombre d'itérations CRDSA. λ est l'index permettant de choisir au hasard une tranche temporelle de référence selon le traitement complémentaire. $\Lambda(\ delta)$ est le nombre maximum de tranches temporelles de référence atteints pendant un index d'analyse de trame donné δ.

Les simulations effectuées dans ce travail s'appliquent aux différents scénarios: MARSALA avec un préambule unique et commun à tous, R-SPORiT avec un préambule unique et commun à tous (scenario 1), et R-SPORiT avec de multiples préambules pseudo-orthogonaux (scenario 2) . Pour chacun d'eux, la complexité globale de la trame, y compris la détection des préambules et la localisation des paquets, est évaluée par rapport au nombre d'utilisateurs convertis en charge de canal en bits / symbole. Dans le scénario 2, la longueur des préambules varie dans $\{2, 3, 5, 15, 31\}$.

Sur la figure 9.5, nous observons que dans le cas d'un préambule unique, la complexité de MARSALA est supérieure à celle de R-SPOTiT à partir, approximativement, d'une charge de canal de $1,1$ bits/symbole. La différence entre les deux augmente avec l'augmentation du nombre d'émetteurs. La complexité de MARSALA est en moyenne quatre fois supérieure à celle de R-SPOTiT. Dans les faibles charges, la complexité est négligeable car aucun traitement complémentaire n'est nécessaire. Ainsi, aucune opération de localisation de données lourde n'est effectuée, seule l'opération de détection de préambule unique. Cependant, lorsque plusieurs préambules sont utilisés dans R-SPOTiT, les faibles charges subissent un nombre

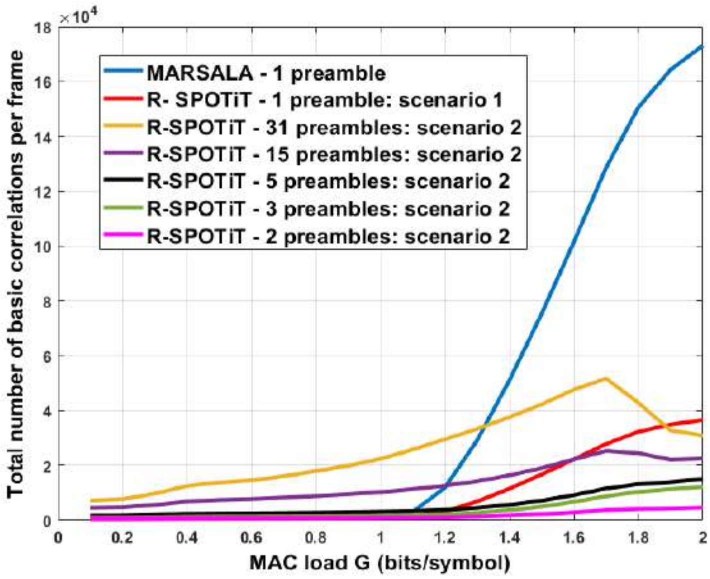


Figure 9.5: Overall frame complexity, with $N_{\rm bt} = 16$ and $R = 5$ in CRDSA/MARSALA and CRDSA/R-SPOTiT environments

de corrélations plus élevé que dans le cas de l'unique préambule, alors que les charges élevées peuvent atteindre jusqu'à une certaine charge, dépendant du nombre de préambules, plus de complexité que pour le cas du préambule unique. Ensuite, ça devient moins complexe. En réalité, avec 31 et 15 de préambules, le nombre de corrélations de base est supérieur à MARSALA jusqu'à $1,2$ bits/symbole et $1,3$ bits/symbole respectivement, puis il évolue progressivement, mais de manière moins significative que dans le cas de MARSALA, jusqu'à $1,7$ bits/symbole. Chacune des courbes de complexité de R-SPOTiT avec 15 et 31 préambules croise la courbe du cas du préambule unique au niveau des charges $1,6$ bits/symbole et $1,8$ bits/symbole, respectivement, pour ensuite devenir moins complexe. Cependant, cette région se situe autour du point d'effondrement du débit (1.7 bits/symbole).

Lorsque $N_{\rm P}$ est plus bas, 2, 3 et 5, la complexité dans les faibles charges est plus petite et plus proche du cas à préambule unique que de R-SPOTiT avec 31 et 15. Leurs courbes franchissent la courbe de MARSALA à environ 1 bits/symbole et évoluent différemment, de manière moins significative. Ils croisent également la courbe R-SPOTiT au préambule unique à environ $1,2$ bits/symbole et présentent un nombre moins important de corrélations.

9.3 Smart Shared Position technique for Interfered Random Transmissions

Smart SPOTiT (S-SPOTiT) décrit une méthode d'attribution à chaque utilisateur des positions de tranches de temps sur une trame de manière à ce qu'aucune boucle ne soit créée. Un autre objectif est de répartir judicieusement les préambules associés aux positions des répliques de chaque paquet, parmi un ensemble de codes pseudo-orthogonaux, afin de simplifier la localisation des répliques. Par conséquent, l'objectif est de s'assurer que chaque paquet potentiellement transmis ait une de ses répliques ayant un préambule unique sur sa position temporelle. Cela permettra de déterminer quels utilisateurs ont envoyé des données sur la trame analysée sans procéder à des corrélations de localisation de données comme dans MARSALA ou dans R-SPOTiT. Ceci ne s'applique que si les préambules sont correctement détectés. Enfin, une fois les répliques localisées, la combinaison des répliques appartenant à un même paquet est effectuée avant démodulation et décodage.

9.3.1 Principe général

Dans le mode de fonctionnement S-SPOTiT, deux caractéristiques principales doivent être soulignées:

1. Les positions des répliques: le récepteur gère les positions des répliques sur la trame et le préambule associé pour chaque utilisateur. Il veille à différencier le préambule d'une réplique d'un paquet donnée des autre paquets potentiellement en collision sur la même tranche temporelle et élimine les boucles. Les positions optimales et le choix de distribution des préambules doivent être communiqués aux émetteurs sous la forme d'informations de signalisation. Cela est envoyé une seule fois et peut être ajouté à la phase de connexion. D'une part, le PLR devrait être amélioré du fait de la disparition du plancher d'erreur, facilement observable à faibles charges pour CRDSA, MARSALA et R-SPOTiT, créé par les boucles de données. D'autre part, la disposition judicieuse quant au choix des positions des paquets et des préambules associés réduit le niveau de complexité en termes de corrélations. En fait, chaque préambule utilisé par un paquet sera unique sur l'une des positions de ses répliques; cela signifie qu'aucune corrélation de localisation de données n'est nécessaire lorsque ce préambule est détecté.

2. Préambules: nous utilisons la pseudo-orthogonalité des préambules afin de limiter les corrélations de localisation. Lorsqu'un préambule est détecté sur une position, le récepteur peut deviner si ce préambule est unique ou non. Il confirmera par conséquent la présence du paquet ayant un préambule unique sur l'une des positions de leurs répliques, en particulier lorsque leurs autres répliques présentent un pic de corrélation. En d'autres termes, seules les corrélations de détection de préambule peuvent être utilisées pour localiser correctement les répliques.

9.3.2 Exemple d'une disposition optimale

Considérons le scénario avec les valeurs suivantes: le nombre de tranches temporelles $N_{\mathrm{S}} = 8$, le nombre de préambules $N_{\mathrm{P}} = 4$, le nombre de répliques par paquet $N_{\mathrm{R}} = 2$, le nombre de niveaux dans la distribution optimale $N_{\mathrm{L}} = 3$ et le nombre maximal d'utilisateurs sans boucles $N_{\mathrm{U}} = 28$.

Le pire scénario peut être illustré lorsque chaque utilisateur u, avec $u\ in[U_1; U_{N_{\mathrm{U}}}]$, a envoyé un paquet sur la même trame. Ainsi, nous devons construire des groupes d'utilisateurs sur tous les niveaux de N_L et leur associer un préambule chacun. Chaque préambule est représenté par une couleur.

Les groupes d'utilisateurs N_P du niveau 1 occupent l'intégralité de la trame avec le plus grand nombre de tranches temporelles $\{0..3..7\}$. Chaque groupe a quatre utilisateurs avec le même préambule (voir Niveau 1 dans la figure 9.6). La première colonne de chaque groupe de préambule regroupe les créneaux horaires des premières répliques des quatre utilisateurs, tandis que la deuxième colonne regroupe les positions de leurs secondes répliques. La différences entre les groupes d'utilisateurs L'étape suivante consiste à créer le niveau suivant en divisant l'ensemble des tranches temporelles de niveau 1 en deux sous-ensembles égaux (voir Niveau 2 dans la figure refsm1). Chaque ensemble défini au niveau 2 contient la moitié du nombre de groupes de préambules du niveau précédent avec la moitié du nombre d'utilisateurs. En effet, le premier ensemble de tranches temporelles $\{0..1..3\}$ regroupe par exemple les premier et deuxième groupes d'utilisateurs avec les préambules P_1 et P_2 respectivement, chacun avec deux utilisateurs. Enfin, le troisième niveau comportera un total de quatre sous-ensembles de tranches temporelles dérivés des ensembles du niveau 2. Chacun avec un seul groupe de préambule d'un utilisateur (voir Niveau 3 de la figure 9.6). La figure ref img2 montre la dis-

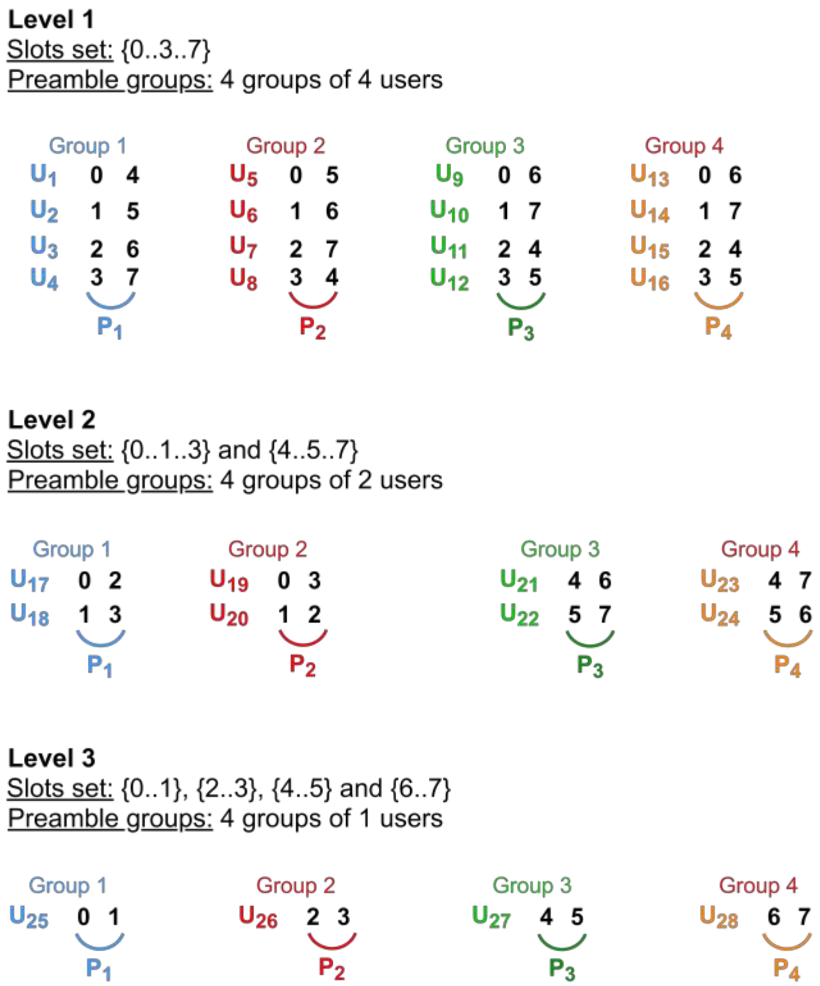


Figure 9.6: Level construction of an optimal frame distribution, $N_{\mathrm{S}} = 8$, $N_{\mathrm{R}} = 2$, $N_{\mathrm{P}} = 4$.

position de chaque paquet potentiel sur une trame selon la distribution décrite précédemment. En d'autres termes, cela représente le pire des cas, lorsque tous les utilisateurs transmettent sur la même trame. Chacun des préambules comprend un groupe d'utilisateurs à chaque niveau. Prenons le préambule bleu par exemple. Ses groupes sont: premièrement, le groupe de préambule de niveau 1, auquel appartiennent les utilisateurs U_1, U_2, U_3, U_4, deuxièmement, le groupe de préambule de niveau 2, auquel appartiennent les utilisateurs U_{17}, U_{18}, et enfin le groupe de préambule du niveau 3, auquel appartient l'utilisateur U_{25}. Deux propriétés peuvent être perceptibles. Tout d'abord, un ensemble des tranches temporelles de niveau $i+1$ est associé à un nombre de tranches temporelle correspondant à la moitié de celui du niveau précédent i. Ainsi, $E_{2,1}$, par exemple, a les paquets de ses groupes de préambule (bleu et rouge) sur les tranches $[0;3]$, qui correspondent à la moitié des tranches où se trouvent les paquets des groupes d'utilisateurs de $E_{1,1}$ ($[0;7]$). Deuxièmement, un ensemble de positions de niveau $i+1$ regroupe la moitié des préambules du niveau précédent i. En fait, $E_{2,1}$ est associé à deux préambules (bleu et rouge) qui représentent la moitié des préambules du niveau précédent $E_{1,1}$ (bleu, rouge, vert et marron). En considérant ces deux propriétés, à chaque niveau, chaque groupe d'utilisateurs a l'une des répliques de ses paquets affectées à un préambule unique sur sa position, par rapport aux niveaux suivants. En effet, au niveau 1, par exemple, le groupe du préambule bleu a les deuxièmes répliques de ses paquets U_1, U_2, U_3, U_4 ayant un préambule unique sur leurs tranches temporelles respectives [4; 7], par rapport aux niveaux suivants. Ceci est valable pour tous les groupes de préambules à tous les niveaux.

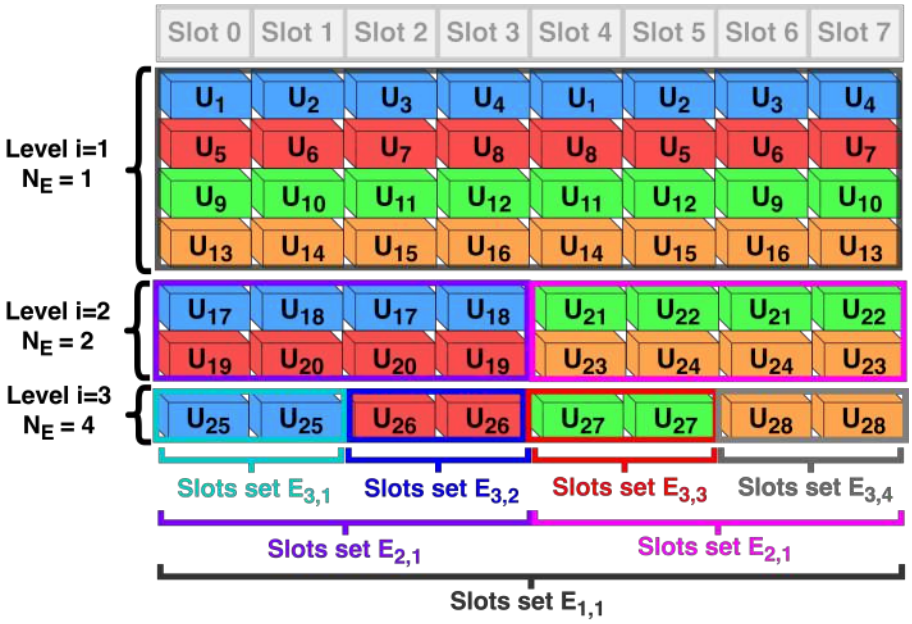


Figure 9.7: Eight slots frame disposition at the worst scenario, $N_R = 2$, $N_P = 4$.

9.3.3 Evaluation des performances

Nous avons choisi un système avec deux répliques par paquet pour des raisons de complexité. Cependant, ce qui dégrade les performances d'un système à deux réplicas par rapport à un nombre plus élevé de réplicas, c'est que la probabilité de boucles qui est plus importante. En conséquence, un plancher d'erreur au niveau du PLR peut être observé dans CRDSA, MARSALA et R-SPOTiT. En réalité, seule une boucle de deux paquets peut être résolue par CRDSA avec les paramètres que nous avons pris; Modulation QPSK et codage 3GPP Turbo du débit 1/3 sur un canal AWGN. En effet, dans ce cas, CRDSA est capable de décoder un paquet en présence d'une seule interférence. Il est important de noter, selon le résultat de

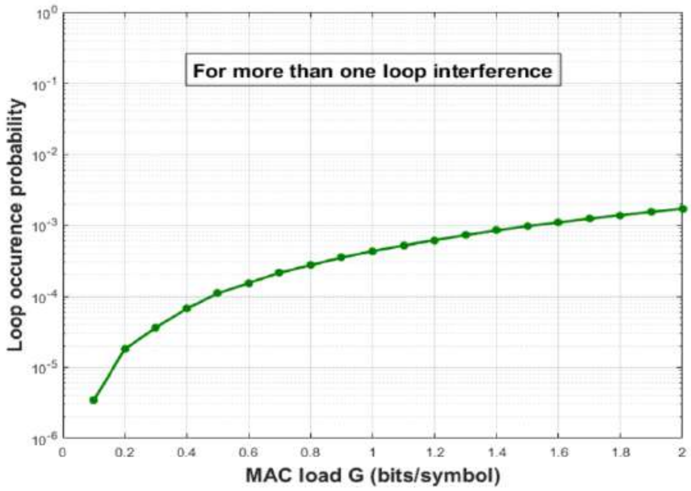


Figure 9.8: Probability of more than one interference loop in a CRDSA-like frame of 100 time slots.

la simulation dans la figure 9.8, que la probabilité d'avoir deux boucles ou plus représente exactement le plancher d'erreur dans CRDSA sur les charges faibles jusqu'à ce que le nombre de collisions devienne suffisamment important (0.6 bits/symbole), en raison du grand nombre d'émetteurs. Pour R-SPOTiT et MARSALA, nous remarquons une concordance entre la probabilité d'avoir deux boucles ou plus et la courbe du PLR jusqu'à une charge de $1,5$ bits/symbole (voir Figure 9.9). Au-delà de cette charge, le nombre de collisions devient plus important et le niveau de SNIR ne permet plus une démodulation et un décodage corrects.

S-SPOTiT qui repose sur une gestion optimale, concernant les positions des répliques sur le cadre et le choix du préambule, évite les boucles et s'assure d'avoir un préambule unique sur l'une des positions des répliques d'un paquet. L'objectif est de simplifier davantage la localisation des paquets et d'améliorer les performances du PLR en supprimant le plancher d'erreur créé par les boucles. D'une part, nous avons vu que cette distribution peut empêcher les corrélations de données des données et ne s'appuyer que sur la détection du préambule pour la localisation des paquets. Ensuite, afin de pouvoir comparer le PLR avec MARSALA et R-SPOTiT avec deux préambules, nous avons choisi d'utiliser un système sans boucle avec des trames de 100 tranches temporelles et 50 préambules avec le premier niveau $i = 1$ de la distribution optimale. Par conséquent, la figure 9.9 montre que le plancher d'erreur du PLR n'est plus présent. L'amélioration du débit (figure 9.10) est insignifiante car son effondrement se produit à une charge de 1.7 bits/symbole. A ce niveau, le PLR est dégradé de la même manière pour R-SPOTiT, MARSALA et S-SPOTiT.

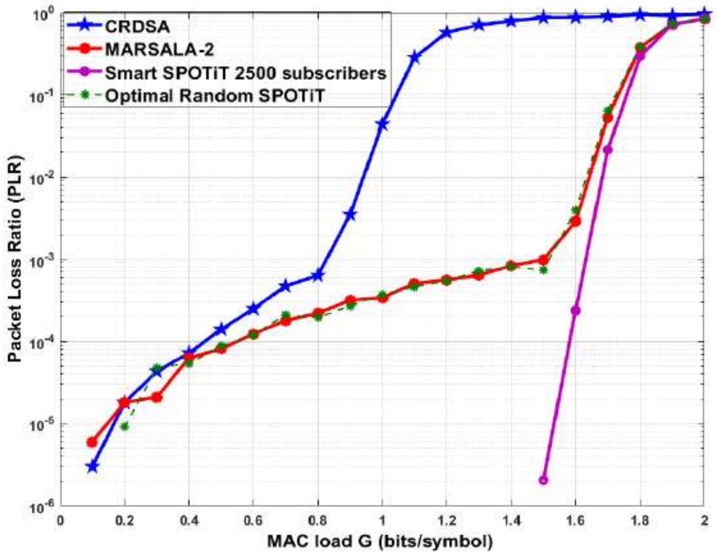


Figure 9.9: PLR of S-SPOTiT, MARSALA-2 and CRDSA with QPSK modulation and turbo coding of rate 1/3 , 100 slots/ frame, 100 information bits, AWGN channel and $E_s/N_0 = 10\ dB$

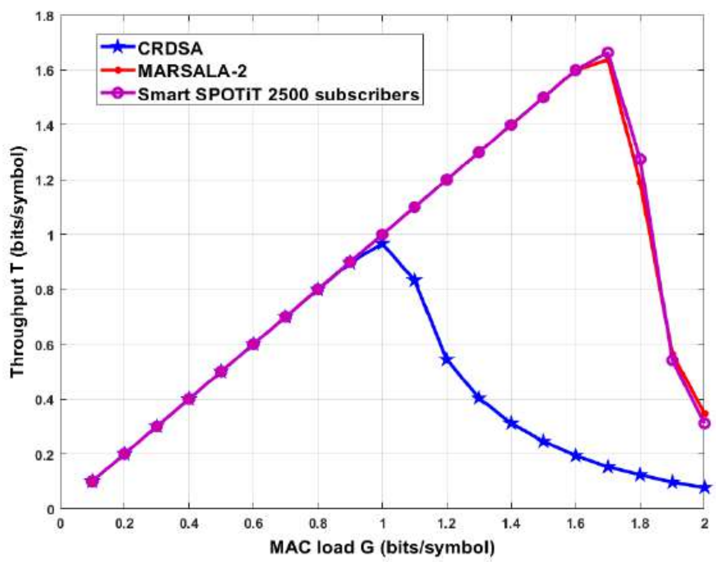


Figure 9.10: Throughput of S-SPOTiT, MARSALA-2 and CRDSA with QPSK modulation and Turbo coding of rate 1/3, 100 slots/ frame, 100 information bits, AWGN channel and $E_s/N_0 = 10\ dB$

9.3.4 Schéma irregulier et extension de S-SPOTiT

Un schéma régulier sans boucle pour S-SPOTiT a été décrit dans le chapitre précédent avec un nombre de tranches temporelles sous la forme d'une puissance de deux. Une distribution optimale a été définie en fonction d'un nombre de niveaux qui dépend du nombre de tranches. Cependant deux limitations peuvent être déduites quant au schéma régulier: premièrement, les paramètres de la distribution optimale en termes de nombre de tranches temporelles et préambules sous la forme d'une puissance de deux ne sont pas forcément disponibles et praticables dans des systèmes réels; deuxièmement, le nombre maximal d'utilisateurs sans boucles est limité en raison de la délimitation fixe de la trame temporelle. De ce fait, nous avons ciblé deux aspects principaux. Tout d'abord, une étude sur S-SPOTiT est effectuée pour mettre en place un schéma avec un nombre irrégulier de tranches temporelles qui n'est pas sous la forme d'une puissance de deux. Deuxièmement, une configuration d'un système dynamique, basée sur l'ajout progressif de nouvelles tranches en cas de besoin, est proposée. La manière d'attribuer un nombre quelconque de préambules a également été étudié.

L'idée générale est de concevoir d'abord un système sans boucle irrégulier basé sur le schéma régulier le plus proche. Cependant, comme le nombre d'emplacements de ce dernier ne correspond pas à celui du schéma irrégulier, le nombre total de configurations sans boucle ne peut pas être atteint. C'est pourquoi il sera nécessaire d'identifier et de compléter le S-SPOTiT irrégulier avec un nombre maximum d'utilisateurs sans boucle. Le construction optimale est décrite en détails dans le chapitre 6.

En ce qui concerne l'extension de S-SPOTiT, nous proposons de concevoir un système dynamique sans boucle en fonction du nombre de tranches temporelles, en accord avec le nombre d'utilisateurs. En effet, certaines applications critiques sont contrainte de fonctionner sous une certaine valeur de PLR (PLR target), ce qui rend le phénomène de boucle extrêmement restrictif. Au départ, l'objectif est donc de pouvoir éliminer en permanence les boucles entre les paquets, quel que soit le nombre d'utilisateurs. Il est important de noter qu'avec la version régulière et irrégumière de S-SPOTiT, chaque fois que le nombre d'utilisateurs dépasse le nombre maximal de configurations sans boucle, des boucles sont créées à l'ajout de nouveaux usagers. Cela est clairement dû à la répétition des configurations déjà utilisées, ce qui reste valide tant que le nombre de tranches temporelles est une valeur fixe. Par conséquent, le S-SPOTiT traditionnel n'est pas adapté au type de systèmes qui, d'une part, sont soumis à une cible PLR et, d'autre part, peuvent être redimensionnés en termes de nombre d'utilisateurs pour une connectivité massive (Applications 5G).

9.4 Localisation des paquets dans un environnement asynchrone utilisant R-SPOtiT

Dans les transmissions synchrones, R-SPOTiT débloque CRDSA, de la même manière que MARSALA mais avec moins de complexité. En effet, le processus CRDSA se trouve dans une situation de blocage lorsqu'il n'y a plus de paquets décodables en raison des charges MAC élevées ou du phénomène de boucle. R-SPOTiT localise les répliques de paquets sur une trame, de manière moins complexe que MARSALA, puis les combine pour obtenir un rapport signal sur bruit plus interférences plus élevé. Cepndant, il souffre du phénomène de boucle si deux répliques par paquet sont utilisées. Dans les transmissions synchrones, Smart SPOTiT, qui construit une répartition optimale des positions des paquets sur la trame, a été proposé comme solution pour supprimer les boucles. Néanmoins, ACRDA réduit considérablement ce dernier grâce à sa propriété asynchrone et offre un PLR plus faible. Sur cette base, nous pensons que l'ACRDA, associé à une version asynchrone de R-SPOTiT (AR-SPOTiT), devrait offrir de meilleures performances.

AR-SPOTiT peut également être considéré comme une alternative complémentaire à l'ECRA pour la localisation de réplicas dans un environnement asynchrone. En fait, AR-SPOTiT permet de localiser des répliques avec des informations partagées ne nécessitant aucune information de signalisation entre le récepteur et chacun des émetteurs.

9.4.1 mécanisme général

Lors de la transmission, la manière de placer chaque paquet dans la trame virtuelle est régie par AR-SPOTiT. En effet, les positions de réplicas sont sélectionnées à l'aide d'un PRNG dont le HID est une graine d'entrée. La même graine est utilisée pour sélectionner un préambule parmi un ensemble de codes pseudo-aléatoires. En fait, plusieurs préambules sont pris en compte dans AR-SPOTiT.

Toutes les graine sont statiques car il n'existe aucune identification de trame virtuelle pouvant servir une graine dynamique, mais cela ne devrait pas être un problème, vu la probabilité d'avoir des boucles répétitives reste très faible en raison de la nature asynchrone du système. Surtout parce que le décalage temporel entre les trames virtuelles transmises est aléatoire en raison d'absence de synchronisation entre les utilisateurs.

Une table d'informations est construite par le récepteur de AR-SPOTiT dans laquelle la distance entre les répliques d'un même utilisateur u est incluse, en termes de nombre de tranches virtuelles entre eux. Ces distances sont calculées par le récepteur après que les positions des répliques ont été générées à l'aide du PRNG en utilisant les graines HID disponibles à la réception.

Le mécanisme de décodage de AR-SPOTiT peut s'effectuer selon deux options: il peut d'abord exploiter les informations ACRDA obtenues dans le cas où le préambule est détecté, mais le paquet n'est pas décodable en raison du niveau élevé d'interférence; Ou bien une détection de puissance peut être effectuée afin de révéler la présence d'un paquet.

Le processus de localisation peut à présent avoir lieu. Les corrélations censées localiser les positions des répliques sont donc effectuées selon le préambule détecté. Elles sont faites à des distances multiple de tranches virtuelles obtenus à partir de la table d'informations, correspondant aux positions des répliques des paquets utilisant le préambule détecté. En d'autres termes, la tranche temporelle de référence utilisée dans MARSALA et R-SPOTiT synchrones est toujours, dans AR-SPOTiT, défini comme étant la tranche virtuelle dans laquelle un préambule est correctement détecté.

9.4.2 Performances globales du système

Dans R-SPOTiT, les opérations pour la détection des préambules sont effectuées à chaque tranche temporelle durant le processus du CRDSA. En effet, dans ce cas synchrone, le début de chaque tranche et le décalage de synchronisation maximal sont connus. Cependant, dans AR-SPOTiT, des recherches sur le préambule, utilisant des corrélateurs, sont effectuées le long de la fenêtre glissante, sur toute la mémoire, au niveau échantillon ou symbole, jusqu'à ce qu'un préambule soit trouvé lors de l'ACRDA. Cela peut rendre la complexité du destinataire importante si le nombre de préambules est grand. Pour cette raison, il est préférable de conserver un petit nombre de codes pseudo-orthogonaux. Nous avons utilisé ici des préambules de Gold de longueur 63 symboles. De même, nous avons choisi dans notre scénario $N_R =$

2 répliques par paquet car ce cas est moins complexe. De plus, comme les transmissions asynchrones atténuent les boucles de données entre paquets, le plancher d'erreur au niveau du PLR rencontré par CRDSA lors de charges faibles est moins significatif dans ACRDA lorsque la nombre de réplicas est égal à 2 (voir la figure ref 3b6). En effet, avec une charge de canal charge est comprise entre 0.1 bits/symbole et 0.8 bits/symbole, CRDSA rencontre un plancher d'erreur entre 3×10^{-6} et 6×10^{-4} respectivement. Pour la même charge, ACRDA présente un plancher d'erreur allant de 3×10^{-6} à 2×10^{-5}.

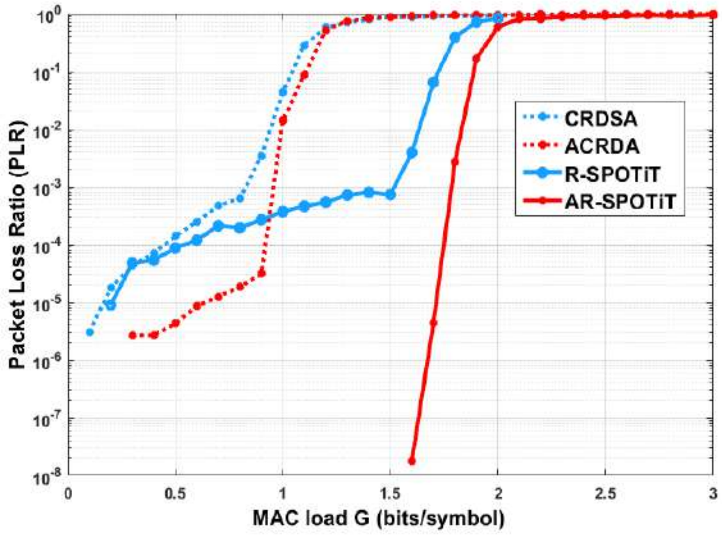


Figure 9.11: ACRDA with AR-SPOTiT, CRDSA with R-SPOTiT, ACRDA and CRDSA Packet Loss Ratio, 100 information bits, QPSK modulation, Turbo code of rate 1/3, AWGN channel and $E_S/N_0 = 10$.

D'une part, nous pouvons remarquer (Figure 9.12) que, à partir d'une charge de canal de 1.6 bits/symbole, le débit est plus élevé avec AR-SPOTiT par rapport à R-SPOTiT qui atteint son maximum de 1.64 bits/symbole à une charge de canal de 1.7 bits/symbol. Par conséquent, AR-SPOTiT est préférable pour les charges élevées. De plus, on peut observer que AR-SPOTiT améliore considérablement le débit lorsqu'il est couplé à l'ACRDA; 1.8 bits/symbole est atteint pour une charge de canal de 1.8 bits/symbole approximativement, par rapport à 1 bit/symbole pour une charge de 1 bit/symbole lorsque ACRDA est utilisé seul. En revanche, AR-SPOTiT réduit considérablement le PLR par rapport à R-SPOTiT (Figure 9.11) en plus de la disparition du plancher d'erreur. Il atteint environ 4.4×10^{-6} avec une charge de canal de 1.7 bits/symbole contrairement à R-SPOTiT qui atteint 6×10^{-2} à la même charge. Un schéma asynchrone offrant de meilleurs résultats qu'un synchrone est principalement dû au

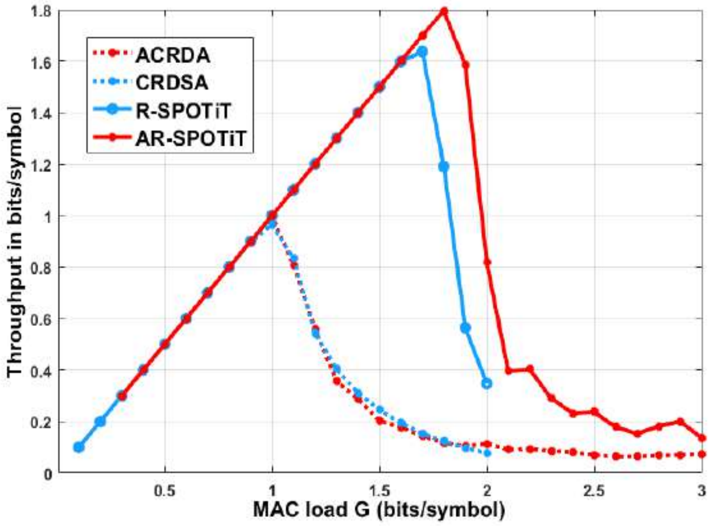


Figure 9.12: ACRDA withAR-SPOTiT, CRDSA with R-SPOTiT, ACRDA and CRDSA throughput, 100 information bits, QPSK modulation, Turbo code of rate 1/3, AWGN channel and $E_S/N_0 = 10$.

modèle d'interférences qui sont majoritairement partielles.

9.5 Conclusion générale et perspectives

Cette thèse a fourni une vue d'ensemble des protocoles RA basés sur ALOHA pour les communications par satellite. Nous nous sommes focalisés plus particulièrement sur la technique CRDSA, qui a fait l'objet de nombreuses améliorations. Malgré toutes les variantes proposées de CRDSA qui ont décidé d'ajouter ou de modifier certains paramètres, le système MARSALA a proposé de conserver le CRDSA en tant que processus de décodage principal et d'intervenir uniquement lorsque ce dernier se trouve dans une situation de blocage (pas de paquets pouvant être résolus); généralement dans des charges de canal élevées. Une telle action vise à résoudre certaines configurations (paquets en collision) afin de rendre le CRDSA opérationnel. Cette contribution majeure a permis d'améliorer considérablement les performances du système en termes de taux de perte de paquets et de débit. Toutefois, une complexité supplémentaire de traitement est perçus au niveau du récepteur. Cela implique la localisation de répliques par le biais de corrélations permettant de tirer profit de la réplication des paquets en utilisant une combinaison de signaux. En effet, puisque les positions des répliques appartenant à un paquet donné ne sont connues dans CRDSA qu'après le décodage de l'une d'entre elles, si une situation de blocage se produit, alors le processus de décodage complet s'arrête. C'est là que MARSALA essaie d'obtenir les positions d'au moins un paquet des répliques en utilisant des corrélations entre une tranche temporelle de référence (contenant une réplique en collision avec d'autres paquets) et les autres tranches de la trame. L'objectif est de combiner toutes les répliques du même paquet afin d'obtenir un SNIR potentiellement plus élevé pour une meilleure probabilité de décodage. Après le SIC, la trame peut par conséquent révéler des paquets sans collision et CRDSA est de nouveau déclenché.

Tout au long de cette thèse, nous avons particulièrement essayé de surmonter deux défis qu'un traitement complémentaire au CRDSA peut rencontrer. Le premier problème est la complexité des calculs concernant la localisation des paquets à la réception après le blocage de CRDSA. Le deuxième problème est le phénomène de boucle lorsque le nombre de répliques est faible, ce qui provoque un plancher d'erreur au niveau des performances du PLR. Ces deux problématiques sont liées; afin de maintenir une faible complexité de localisation, un petit nombre de répliques est requis, mais cette dernière induit un phénomène de boucle plus important. Pour résoudre ce problème combiné, nous avons abordé la partie de la localisation des paquets, où jusqu'à présent, le récepteur ne disposait d'aucune information préalable sur la position des éventuels paquets. L'apport majeur de cette thèse est l'introduction une nouvelle technique SPOTiT basée sur une information de localisation partagée entre le récepteur et chacun des émetteurs, dont l'étude est résumée ci-dessous.

La technique proposée SPOTiT repose principalement sur le partage d'informations avec le récepteur sur les positions de tous les paquets potentiels sur une trame /une trame virtuelle, si si un paquet est transmis. Son objectif principal est d'affiner la complexité de localisation en réduisant le nombre de tranches temporelles/ virtuelles où effectuer des corrélations destinées à localiser les répliques des paquets. De plus, des préambules pseudo-orthogonaux peuvent également être utilisés pour de meilleurs résultats. SPOTiT est opérationnel au même niveau que MARSALA; C'est-à-dire lorsque CRDSA ne peut plus décoder de nouveaux paquets. Trois solutions sont alors proposées. La première, R-SPOTiT, vise à atténuer la complexité de localisation en rendant les positions potentielles des paquets disponibles à la réception, mais sans aucune information de signalisation supplémentaire. La seconde solution, S-SPOTiT, en revanche, implique une information de signalisation qui est envoyée à chaque utilisateur sur les positions dans lesquelles il doit transmettre après avoir défini une disposition optimale sans boucle. Outre une localisation potentiellement simple, S-SPOTiT cible également le phénomène de boucle. Ce dernier problème étant naturellement atténué par les transmissions asynchrones, du fait de la nature des interférences, qui est en grande partie partielle, une adaptation de SPOTiT dans un tel environnement a semblé convaincante. AR-SPOTiT en tant que processus complémentaire à ACRDA (une version asynchrone de CRDSA) a ensuite été défini. Les informations de signalisation ayant servi à S-SPOTiT n'ont pas besoin d'éliminer le phénomène de boucle, mais elles sont plus complexes à la réception en raison de la nature asynchrone de la transmission.

R-SPOTiT

Pour que le destinataire connaisse la position des paquets, R-SPOTiT s'appuie sur la génération de ces informations via un PRNG dont les graines sont des informations d'identification connues. De cette manière, le récepteur n'a pas à effectuer des corrélation sur l'ensemble de la trame pour localiser un paquet, mais uniquement sur certaines tranches temporelles où un paquet potentiel pourrait avoir été transmis. Cela inclut les paquets potentiellement en collision sur la tranche de temps de référence, utilisant le même préambule. Les résultats montrent que R-SPOTiT peut atteindre les mêmes performances système que MARSALA en termes de PLR et de débit, mais avec une complexité de corrélation de données nettement inférieure. En outre, une analyse approfondie avait été effectué concernant la complexité globale, incluant la détection du préambule ainsi que les corrélations de données, a montré qu'un système avec R-SPOTiT reste un meilleur choix que MARSALA. Un résultat intéressant recommande d'utiliser R-SPOTiT avec un petit nombre de préambules s'ils sont pseudo-orthogonaux (tels que les codes Gold).

S-SPOTiT

Le but de S-SPOTiT est d'éliminer le phénomène de boucle et d'assurer une faible complexité de localisation. En tant que tel, nous avons défini une répartition par niveau des positions des paquets et des préambules à utiliser pour une communauté d'utilisateurs de manière à ce qu'aucune boucle ne soit créée, en plus d'une localisation potentiellement simple. Cette dernière fait en sorte que l'une des répliques d'un paquet donné utilise un préambule unique sur sa position; ce qui signifie qu'aucune corrélation de localisation n'est nécessaire si ce préambule est détecté. Cette fonctionnalité est valide tant que tous les paquets ayant le même préambule aux niveaux précédents ont été décodés. Par conséquant, un PLR sans plancher d'erreur est observé. Un schéma complet de S-SPOTiT, avec des paramètres irréguliers, a été fourni pour adapter sa distribution optimale à tout scénario de système réel. De plus, S-SPOTiT a été étendu à un système dynamique sans boucle en ce qui concerne le nombre d'utilisateurs et de tranches temporelles, ce qui ajoute une flexibilité intéressante en termes de nombre d'utilisateurs ainsi qu'une évolutivité du réseau.

AR-SPOTiT

AR-SPOTiT peut être perçu comme une version asynchrone de R-SPOTiT, ainsi qu'une alternative à S-SPOTiT en ce qui concerne l'atténuation du phénomène de boucle. Il présente également un moyen de localiser les paquets dans un environnement asynchrone avec une complexité potentiellement inférieure à celle d'ECRA. Il exploite la distance entre les répliques dans une trame virtuelle de ACRDA ainsi que le PRNG de R-SPOTiT pour améliorer les performances du système. En effet, il offre un débit plus élevé que ceux du R-SPOTiT et S-SPOTiT, et avec de meilleures performances PLR. Nous avons également conclu qu'AR-SPOTiT était moins complexe que l'ECRA en termes de corrélation de détection du préambule.

Malgré les principaux résultats prometteurs, obtenus avec les trois dérivés de SPOTiT dans les environnements synchrones et asynchrones, des schémas d'amélioration sont attendus.

L'évaluation de la complexité de R-SPOTiT que nous avons réalisée tout au long de cette thèse a présenté des résultats basés sur les préambules pseudo-orthogonaux de codes de Gold. Cependant, d'autres types de codes pourraient offrir une meilleure détection. Par exemple, les séquences de Zadoff-Chu utilisées dans l'évolution à long terme (LTE) de 3GPP, présentent des propriétés d'orthogonalité intéressantes et sont sujettes à de nombreuses stratégies d'amélioration proposées dans des systèmes à accès aléatoires.

En plus de l'atténuation du phénomène de boucle offerte par S-SPOTiT, la complexité de localisation des paquets devrait être faible en raison de sa propriété d'avoir l'une des répliques utilisant un préambule unique sur sa position. Cependant, lorsque cette fonctionnalité est interrompue à cause de paramètres irréguliers (nombre d'emplacements, nombre de préambules et nombre d'utilisateurs), il serait intéressant d'évaluer la complexité de localisation rencontrée et de la comparer avec R-SPOTiT en utilisant les mêmes paramètres. Cela devrait donner une idée globale du système à utiliser, en fonction des compromis choisis (phénomène de boucle, complexité, informations de signalisation, PLR et débit). Sur le même chemin, AR-SPOTiT devrait également prendre part de l'évaluation de complexité car il participe également à la réduction drastique du phénomène de boucle, sans information de signalisation et en offrant un débit plus élevé. En retour, le processus de recherche de préambules dans un environnement asynchrone s'avère fastidieux en termes de calcul en raison de l'absence de trames communes aux utilisateurs, car cela nécessite de parcourir toute la mémoire au niveau de l'échantillon ou symbole.

La condition de répartition égale des préambules sur les tranches temporelles dans l'extension de S-SPOTiT n'est pas toujours respectée lorsqu'une technique de Round-Robin est utilisée. Par conséquent, une étude approfondie est nécessaire pour définir la meilleure solution.

L'impact des imperfections du canal et l'estimation des paramètres de synchronisation ont été étudiés pour MARSALA dans [Zid16]. Toutefois, il serait intéressant de vérifier leur validité avec SPOTiT et ses variantes. D'une part, une stratégie consistant à combiner l'utilisation de l'algorithme EM (Expectation Maximization) au préambule et au postambule avec une initialisation des paramètres de canal reposant sur une opération d'auto-corrélation, au lieu d'une initialisation aléatoire, a donné de bons résultats. les résultats en termes de PER comparés au EM traditionnel. D'autre part, une estimation utilisant une modulation pilote assistée par symbole a également été proposée pour MARSALA dans le même travail de recherche. Lorsque ceux-ci sont combinés avec une estimation et un décodage conjoints, ce qui devrait améliorer les performances du SIC, pour estimer les différents paramètres de canal, une très faible perte des performances du système est observée.

En outre, la technique de combinaison MRC évaluée par ECRA dans un environnement asynchrone et MARSALA pour les transmissions synchrones a montré une amélioration significative du débit. De plus, la variabilité des puissances des paquets utilisant une distribution log-normale dans MARSALA présentait les meilleurs résultats lorsqu'elle utilisée conjointement avec la combinaison MRC. Par conséquent, nous pensons que l'application du MRC aux différentes variantes de SPOTiT avec une variabilité des puissances des paquets devrait être considérablement bénéfique en termes de gain de performance.

Outre que la variation des différents paramètres du système ou l'introduction des imperfections du canal, telles que le trafic de Poisson, que nous considérons être un aspect important à vérifier avant l'implémentation, il reste encore des portes à exploiter pour obtenir la meilleure version théorique de SPOTiT. Prenons par exemple la région critique de la courbe de débit dans l'une des variantes de SPOTiT. Cela devrait correspondre à la partie de la courbe qui suit le point critique où le débit s'effondre ($1,7$ bits/symbole dans R-SPOTiT). Nous prévoyons qu'un mécanisme de retransmission tel que HARQ (demande de répétition automatique hybride) préviendrait l'effondrement prématuré du débit, en particulier dans les applications non critiques en termes de temps de transmission. La table d'informations de SPOTiT peut être exploitée afin de définir quels utilisateurs parmi tous les utilisateurs doivent retransmettre une réplique supplémentaire et dans quelle tranche temporelle, en fonction des caractéristiques et besoins de l'application. De plus, le nombre maximal de retransmissions par utilisateur doit être défini pour un scénario donné.

Enfin, cette thèse a abordé certaines des questions liées aux protocoles d'accès aléatoire basés sur ALOHA. Toutefois, les techniques très prometteuses à étalement de spectre, telles que E-SSA et ses variantes, sont également intéressantes à étudier et pourraient faire partie de nos travaux futurs.

Bibliography

[42197] DVB Document ETSI EN 300 421. “Digital Video Broadcasting (DVB); Farming Structure, Channel Coding and Modulation for 11/12 GHz Satellite Services”. In: (1997) (cit. on pp. 2, 14, 138).

[79003] DVB Document ETSI EN 301 790. “Digital Video Broadcasting (DVB); Interaction Channel for Satellite Distribution Systems”. In: (2003) (cit. on pp. 2, 14, 138).

[A1511] DVB Document ETSI A155-1. “Digital Video Broadcasting (DVB); Second Generation DVB Interactive Satellite System (RCS2; Part2)”. In: (2011) (cit. on pp. 2, 14, 138).

[Abr70] N. Abramson. “The ALOHA System - Another Alternative for Computer Communications”. In: 37 (1970), pp. 281–285 (cit. on p. 11).

[Abr77] N. Abramson. “The Throughput of Packet Broadcasting Channels”. In: *IEEE Transactions on Communications* 25.1 (1977), pp. 117–128 (cit. on p. 12).

[Abr96] N. Abramson. “Spread ALOHA CDMA Data Communications”. In: *US Patent 5,537,397*. 1996 (cit. on p. 28).

[Ahl+00] R. Ahlswede et al. “Network Information Flow”. In: *IEEE Transactions on Information Theory* 46.4 (2000), pp. 1204–1216 (cit. on p. 16).

[Al-+11] M. Al-Imari et al. “Subcarrier and Power Allocation for LDS-OFDM System”. In: *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*. 2011, pp. 1–5 (cit. on p. 11).

[Ben+13] A. Benjebbour et al. “Concept and Practical Considerations of Non-orthogonal Multiple Access (NOMA) for Future Radio Access”. In: *2013 International Symposium on Intelligent Signal Processing and Communication Systems*. 2013, pp. 770–774 (cit. on p. 11).

[BLB12] H. Bui, J. Lacan, and M. Boucheret. “An Enhanced Multiple Random Acess Scheme for Satellite Communications”. In: *Wireless Telecommunications Symposium 2012*. 2012, pp. 1–6 (cit. on p. 27).

[Bre59] D. G. Brennan. “Linear Diversity Combining Techniques”. In: *Proceedings of the IRE* 47.6 (1959), pp. 1075–1102 (cit. on p. 28).

[Bui12] Huyen Chi Bui. “Thesis on: Access Methods based on Physical Layer Network Coding”. In: (2012) (cit. on p. 27).

[Bui+15] H-C. Bui et al. “A Multi-Replica Decoding Technique for Contention Resolution Diversity Slotted Aloha”. In: *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*. 2015 (cit. on p. 24).

[Cas+12] E. Cassini et al. “Packet Data Transmission over a Shared Transmission Channel”. In: *US Patent Application No. 8,094,672*. 2012 (cit. on p. 19).

[CDD07] E. Casini, R. De Gaudenzi, and O. Del Rio Herrero. “Contention Resolution Diversity Slotted ALOHA (CRDSA): An Enhanced Random Access Scheme for Satellite Access Packet Networks”. In: *IEEE Transactions on Wireless Communications* 6.4 (2007), pp. 1408–1419 (cit. on pp. 19, 37, 83).

[Cha+12] W. Chauvet et al. “Physical Layer DVB-SH Performance Prediction Based on Mutual Information”. In: *International Journal on Satellite Communications and Networking* 30.5 (2012), pp. 193–211 (cit. on p. 119).

[Che+17] S. Chen et al. “Pattern Division Multiple Access—A Novel Nonorthogonal Multiple Access for Fifth-Generation Radio Networks”. In: *IEEE Transactions on Vehicular Technology* 66.4 (2017), pp. 3185–3196 (cit. on p. 11).

[CK13] F. Clazzer and C. Kissling. “Enhanced Contention Resolution Aloha - ECRA”. In: *SCC 2013; 9th International ITG Conference on Systems, Communication and Coding*. 2013, pp. 1–6 (cit. on p. 28).

[CKM18] F. Clazzer, C. Kissling, and M. Marchese. “Enhancing Contention Resolution ALOHA Using Combining Techniques”. In: *IEEE Transactions on Communications* 66.6 (2018), pp. 2576–2587 (cit. on pp. 28, 37).

[Cla+17] F. Clazzer et al. “Detection and Combining Techniques for Asynchronous Random Access with Time Diversity”. In: *SCC 2017; 11th International ITG Conference on Systems, Communications and Coding*. 2017, pp. 1–6 (cit. on pp. 28, 29).

[CM88] D. M. Chitre and J. S. McCoskey. “VSAT networks: Architectures, Protocols, and Management”. In: *IEEE Communications Magazine* 26.7 (1988), pp. 28–38 (cit. on p. 16).

[CR83] G. Choudhury and S. Rappaport. “Diversity ALOHA - A Random Access Scheme for Satellite Communications”. In: *IEEE Transactions on Communications* 31.3 (1983), pp. 450–457 (cit. on p. 19).

[Dai+18] L. Dai et al. “A Survey of Non-Orthogonal Multiple Access for 5G”. In: *IEEE Communications Surveys Tutorials* 20.3 (2018), pp. 2294–2323 (cit. on p. 11).

[DD12] O. Del Rio Herrero and R. De Gaudenzi. “High Efficiency Satellite Multiple Access Scheme for Machine-to-Machine Communications”. In: *IEEE Transactions on Aerospace and Electronic Systems* 48.4 (2012), pp. 2961–2989 (cit. on p. 28).

[dD14] O. del Rio Herrero and R. De Gaudenzi. “Generalized Analytical Framework for the Performance Assessment of Slotted Random Access Protocols”. In: *IEEE Transactions on Wireless Communications* 13.2 (2014), pp. 809–821 (cit. on pp. 21, 44, 58, 94).

[De +14] R. De Gaudenzi et al. “Asynchronous Contention Resolution Diversity ALOHA: Making CRDSA Truly Asynchronous”. In: *IEEE Transactions on Wireless Communications* 13.11 (2014), pp. 6193–6206 (cit. on pp. 29, 117).

[DG+16] Riccardo De Gaudenzi et al. “Random Access Schemes for Satellite Networks: from VSAT to M2M -A Survey”. In: *International Journal of Satellite Communications and Networking* 36 (2016) (cit. on pp. 16, 28, 45).

[DIA17] Moussa DIALLO. “COWShED Project, Internet Society, Beyond the Net”. In: (2017) (cit. on pp. 1, 137).

[Gal62] R. Gallager. “Low-density parity-check codes”. In: *IRE Transactions on Information Theory* 8.1 (1962), pp. 21–28 (cit. on p. 21).

[GC05] A. Giorgetti and M. Chiani. “Influence of Fading on the Gaussian Approximation for BPSK and QPSK with Asynchronous Cochannel Interference”. In: *IEEE Transactions on Wireless Communications* 4.2 (2005), pp. 384–389 (cit. on p. 119).

[Gol05] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, 2005 (cit. on pp. 9, 10, 12).

[Gol67] R. Gold. “Optimal Binary Sequences for Spread Spectrum Multiplexing”. In: *IEEE Transactions on Information Theory* 13.4 (1967), pp. 619–621 (cit. on p. 52).

[GS13] M. Ghanbarinejad and C. Schlegel. “Irregular Repetition Slotted ALOHA with Multiuser Detection”. In: *2013 10th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*. 2013, pp. 201–205 (cit. on p. 22).

[Hay13] S. Haykin. *Digital Communication Systems*. Wiley, 2013 (cit. on p. 9).

[HM83] Hyong Lee and J. Mark. “Combined Random/Reservation Access for Packet Switched Transmission Over a Satellite with On-Board Processing: Part I - Global Beam Satellite”. In: *IEEE Transactions on Communications* 31.10 (1983), pp. 1161–1171 (cit. on p. 16).

[HWT08] R. Hoshyar, F. P. Wathan, and R. Tafazolli. “Novel Low-Density Signature for Synchronous CDMA Systems Over AWGN Channel”. In: *IEEE Transactions on Signal Processing* 56.4 (2008), pp. 1616–1626 (cit. on p. 11).

[Int+15] G. Interdonato et al. “Intra-slot Interference Cancellation for collision resolution in Irregular Repetition Slotted ALOHA”. In: *2015 IEEE International Conference on Communication Workshop (ICCW)*. 2015, pp. 2069–2074 (cit. on p. 22).

[Jac+77] Irwin Jacobs et al. “Cpoda - a Demand Assignment Protocol for Satnet”. In: *Proceedings of the Fifth Symposium on Data Communications*. SIGCOMM ’77. Snowbird, Utah, USA: ACM, 1977, pp. 2.5–2.9 (cit. on p. 16).

[Jen+08] T. L. Jensen et al. “Mutual Information Metrics for Fast Link Adaptation in IEEE 802.11n”. In: *2008 IEEE International Conference on Communications*. 2008 (cit. on p. 119).

[Kis11] C. Kissling. “Performance Enhancements for Asynchronous Random Access Protocols over Satellite”. In: *2011 IEEE International Conference on Communications (ICC)*. 2011, pp. 1–6 (cit. on p. 28).

[Kol02] Michael O. Kolawole. *Satellite Communication Engineering*. Marcel Dekker New York, 2002, xiii, 263 p. : (cit. on p. 9).

[Lee+12] M. Lee et al. “R-CRDSA: Reservation-Contention Resolution Diversity Slotted ALOHA for Satellite Networks”. In: *IEEE Communications Letters* 16.10 (2012), pp. 1576–1579 (cit. on p. 23).

[Liv11] G. Liva. “Graph-Based Analysis and Optimization of Contention Resolution Diversity Slotted ALOHA”. In: *IEEE Transactions on Communications* 59.2 (2011), pp. 477–487 (cit. on p. 21).

[Liv11] G. Liva. "Method for Contention Resolution in Time Hopping or Frequency Hopping". In: *US Patent Application No. US/2011/0096795 A1*. 2011 (cit. on p. 22).

[LM93] T. Le-Ngoc and J. I. Mohammed. "Combined Free/Demand Assignment Multiple Access (CFDAMA) Protocols for Packet Satellite Communications". In: *Proceedings of 2nd IEEE International Conference on Universal Personal Communications*. Vol. 2. 1993, 824–828 vol.2 (cit. on p. 16).

[Lu+15] L. Lu et al. "Prototype for 5G New Air Interface Technology SCMA and Performance Evaluation". In: *China Communications* 12.Supplement (2015), pp. 38–48 (cit. on p. 11).

[LYN03] S. . R. Li, R. W. Yeung, and Ning Cai. "Linear Network Coding". In: *IEEE Transactions on Information Theory* 49.2 (2003), pp. 371–381 (cit. on p. 16).

[MB02] G. Maral and M. Bousquet. *Satellite Communications Systems: Systems, Techniques and Technology*. Novartis Foundation Symposium. Wiley, 2002 (cit. on p. 12).

[M.D98] M.Dankberg. "Paired Carrier Multiple Access (PCMA) for Satellite Communications". In: (1998) (cit. on p. 17).

[MDA17a] A. Mengali, R. De Gaudenzi, and P. Arapoglou. "Enhancing the Physical Layer of Contention Resolution Diversity Slotted ALOHA". In: *IEEE Transactions on Communications* 65.10 (2017), pp. 4295–4308 (cit. on p. 22).

[MDA17b] A. Mengali, R. De Gaudenzi, and P. Arapoglou. "Enhancing the Physical Layer of Contention Resolution Diversity Slotted ALOHA". In: *IEEE Transactions on Communications* 65.10 (2017), pp. 4295–4308 (cit. on p. 22).

[Mel+12] A. Meloni et al. "Sliding Window-based Contention Resolution Diversity Slotted ALOHA". In: *2012 IEEE Global Communications Conference (GLOBECOM)*. 2012, pp. 3305–3310 (cit. on p. 23).

[Mul02] N.J. Muller. *Wireless A to Z*. Mainframe Series: Quick Reference. McGraw-Hill Education, 2002 (cit. on p. 8).

[NKH14] N. Nonaka, Y. Kishiyama, and K. Higuchi. "Non-Orthogonal Multiple Access Using Intra-Beam Superposition Coding and SIC in Base Station Cooperative MIMO Cellular Downlink". In: *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*. 2014, pp. 1–5 (cit. on p. 11).

[OKH12] N. Otao, Y. Kishiyama, and K. Higuchi. “Performance of Non-orthogonal Access with SIC in Cellular Downlink using Proportional Fair-based Resource Allocation”. In: *2012 International Symposium on Wireless Communication Systems (ISWCS)*. 2012, pp. 476–480 (cit. on p. 11).

[OS19] Index of Objects Launched into Outer Space. “United Nations Office for Outer Space Affairs (UNOOSA)”. In: (2019) (cit. on pp. 1, 137).

[Pao15] E. Paolini. “Finite Length Analysis of Irregular Repetition Slotted Aloha (IRSA) Access Protocols”. In: *2015 IEEE International Conference on Communication Workshop (ICCW)*. 2015, pp. 2115–2120 (cit. on p. 22).

[PLC11] E. Paolini, G. Liva, and M. Chiani. “High Throughput Random Access via Codes on Graphs: Coded Slotted ALOHA”. In: *2011 IEEE International Conference on Communications (ICC)*. 2011, pp. 1–6 (cit. on p. 26).

[PLG17] E. Paolini, G. Liva, and A. Graell i Amat. “A Structured Irregular Repetition Slotted ALOHA Scheme with Low Error Floors”. In: *2017 IEEE International Conference on Communications (ICC)*. 2017, pp. 1–6 (cit. on p. 22).

[PY06] P. Popovski and H. Yomo. “The Anti-Packets Can Increase the Achievable Throughput of a Wireless Multi-Hop Network”. In: *2006 IEEE International Conference on Communications*. Vol. 9. 2006, pp. 3885–3890 (cit. on p. 16).

[Rap96] Theodore S Rappaport. *Wireless communications: Principles and practice*. English (US). Prentice Hall, 1996 (cit. on p. 8).

[Raz+12] R. Razavi et al. “On Receiver Design for Uplink Low Density Signature OFDM (LDS-OFDM)”. In: *IEEE Transactions on Communications* 60.11 (2012), pp. 3499–3508 (cit. on p. 11).

[RHDG09] O del Rio Herrero and Riccardo De Gaudenzi. “A high-performance MAC Protocol for Consumer Broadband Satellite Systems”. In: 2009, pp. 512–512 (cit. on p. 21).

[RS90] Raphael Rom and Moshe Sidi. *Multiple Access Protocols: Performance and Analysis*. Berlin, Heidelberg: Springer-Verlag, 1990 (cit. on p. 9).

[Sai+13] Y. Saito et al. “Non-Orthogonal Multiple Access (NOMA) for Cellular Future Radio Access”. In: *2013 IEEE 77th Vehicular Technology Conference (VTC Spring)*. 2013, pp. 1–5 (cit. on pp. 10, 11).

[Sha+19] X. Shao et al. "NOMA-Based Irregular Repetition Slotted ALOHA for Satellite Networks". In: *IEEE Communications Letters* 23.4 (2019), pp. 624–627 (cit. on p. 22).

[SM16] Satellites: The Small and the Many. "The Economist". In: (2016) (cit. on pp. 1, 137).

[Sun+16] J. Sun et al. "Irregular Repetition Slotted ALOHA with Priority (P-IRSA)". In: *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. 2016, pp. 1–5 (cit. on p. 22).

[SZS07] K. Sayana, J. Zhuang, and K A. Stewart. "Link Performance Abstraction based on Mean Mutual Information per Bit (MMIB) of the LLR Channel". In: 2007 (cit. on p. 119).

[TV05] David Tse and Pramod Viswanath. *Fundamentals of Wireless Communication.* New York, NY, USA: Cambridge University Press, 2005 (cit. on p. 8).

[V1.11] DVB Document ETSI TS 302 583 V1.2.1. "Digital Video Broadcasting (DVB); Framing Structure, Channel Coding and Modulation for Satellite Services to Handheld Devises (SH) below 3 GHz". In: (2011) (cit. on pp. 2, 14, 138).

[V1.15a] DVB Document ETSI EN 300 421 V1.1.2. "Digital Video Broadcasting (DVB); Implementation Guidelines for the Second Generation System for Broadcasting, Interactive Services, News Gathering and other Broadband Satellite Applications; Part 1: DVB-S2". In: (2015) (cit. on pp. 2, 14, 138).

[V1.15b] DVB Document ETSI EN 300 421 V1.1.2. "Digital Video Broadcasting (DVB); Implementation Guidelines for the Second Generation System for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Applications; Part 2: S2 Extensions (DVB-S2X)". In: (2015) (cit. on pp. 2, 14, 138).

[vP09] J. van de Beek and B. M. Popovic. "Multiple Access with Low-Density Signatures". In: *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*. 2009, pp. 1–6 (cit. on p. 11).

[Yua+16] Z. Yuan et al. "Multi-User Shared Access for Internet of Things". In: *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. 2016, pp. 1–5 (cit. on p. 11).

[Yun+15] Yunseong Lee et al. “MR-CRDSA: Multiple Reservation-Contention Resolution Diversity Slotted ALOHA for Battle-field Communication”. In: *MILCOM 2015 - 2015 IEEE Military Communications Conference*. 2015, pp. 85–90 (cit. on p. 23).

[Zam+18a] S. Zamoum et al. “Complexity Analysis of Recent ALOHA Random Access Techniques in Satellite Communications”. In: *International Journal of Satellite Communications and Networking, Submitted*. 2018, pp. 1–8 (cit. on p. 52).

[Zam+18b] S. Zamoum et al. “Deterministic Distribution of Replicas Positions for Multiuser Random Transmissions in Satcoms”. In: *2018 IEEE Global Communications Conference (GLOBECOM)*. 2018, pp. 1–6 (cit. on p. 80).

[Zam+18c] S. Zamoum et al. “N: 071277 FR RQDLV 14-05-18 YTA-LRE”. In: *Patent: Procédé et Système de Transmission de Packets de Données à Travers un Canal de Transmission (RA) à Accès Aléatoire*. 2018 (cit. on pp. 34, 80).

[Zam+18d] S. Zamoum et al. “Shared Position Technique for Interfered Random Transmissions in Satellite Communications”. In: *2018 9th Advanced Satellite Multimedia Systems Conference and the 15th Signal Processing for Space Communications Workshop (ASMS/SPSC)*. 2018, pp. 1–8 (cit. on p. 34).

[Zam+19] S. Zamoum et al. “Asynchronous Packet Localization with Random SPOtiT in Satellite Communications”. In: *The 22nd International Symposium on Wireless Multimedia Communications, Submitted*. 2019 (cit. on p. 116).

[Zha+17] M. Zhao et al. “An Improved Uplink Sparse Coded Multiple Access”. In: *IEEE Communications Letters* 21.1 (2017), pp. 176–179 (cit. on p. 11).

[Zid+15] K. Zidane et al. “Effect of Residual Channel Estimation Errors in Random Access Methods for Satellite Communications”. In: *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*. 2015 (cit. on p. 25).

[Zid+16] K. Zidane et al. “Estimation of Timing Offsets and Phase Shifts between Packet Replicas in MARSALA Random Access”. In: *2016 IEEE Global Communications Conference (GLOBECOM)*. 2016 (cit. on p. 25).

[Zid16] Karine Zidane. “Thesis on: Improving Synchronous Random Access Schemes for Satellite Communications”. In: (2016) (cit. on pp. 134, 162).

[ZLL06] S. Zhang, S. C. Liewi, and P. P. Lam. “Hot Topic: Physical-Layer Network Topic”. In: *ACM MOBICOM*. 2006 (cit. on p. 16).

[ZYL02] Zhifeng Jiang, Yuheng Li, and V. C. M. Leung. “A predictive Demand Assignment Multiple Access Protocol for Broadband Satellite Networks Supporting Internet Applications”. In: *2002 IEEE International Conference on Communications. Conference Proceedings. ICC 2002 (Cat. No.02CH37333)*. Vol. 5. 2002, 2973–2977 vol.5 (cit. on p. 16).