Etude d'une nouvelle méthode d'accès (MAC) multi-canal dédiée aux réseaux locaux et réseaux personnels sans fil

Mahamat Habib Senoussi Hissein

Université Fédérale
Toulouse Midi-Pyrénées

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse - Jean Jaurès

Présentée et soutenue par :
MAHAMAT HABIB SENOUSSI HISSEIN

le 12 juillet 2017

Titre :

Étude et prototypage d'une nouvelle méthode d'accès aléatoire multi-canal multi-saut pour les réseaux locaux sans fil

---

École doctorale et discipline ou spécialité :
ED MITT : Domaine STIC : Réseaux, Télécoms, Systèmes et Architecture

Unité de recherche :
Institut de Recherches en Informatique de Toulouse (IRIT) équipe IRT

Directeur/trice(s) de Thèse :

Pr. Thierry VAL
Dr. Adrien VAN DEN BOSSCHE

Jury :
Pr. Abdennaceur KACHOURI, ENIS Sfax (Rapporteur)
Pr. Mohamed Salim BOUHLEL, ISBS Sfax (Rapporteur)
Karen GODARY, LIRMM Université Montpellier (Examinatrice)
Pr. Thierry VAL (Encadreur)
Dr. Adrien VANDEN BOSSCHE (Co-encadreur)

Remerciements

Je tiens tout d'abord à remercier mon directeur de thèse M. Thierry VAL d'avoir accepté de diriger et suivre mes travaux pendant ces quatre années d'expériences et surtout sa disponibilité et ses encouragements, je remercie le co-directeur Adrien VAN DEN BOSSCHE pour les remarques qu'il a apporté à ces travaux de thèse. Je tiens également à remercier M. Abdennaceur KACHOURI, professeur à l'ENIS Sfax et M. Mohamed Salim BOUHLEL, professeur à l'ISBS Sfax d'avoir montré leur intérêt pour mes travaux en acceptant la charge de rapporteur. Je remercie également Mme Karen Godary-Dejean, maître de conférences à LIRMM de l'université de Montpellier, d'avoir bien voulu juger ces travaux en tant qu'examinatrice.

J'adresse mes remerciements à tous mes collègues enseignants de l'INSTA particulièrement du département GI.

Merci à tous mes collègues du Labo de l'équipe IRT Blagnac pour les échanges des idées tant dans le domaine scientifique et d'autres.

Merci également à tous les amis Tchadiens de Toulouse avec lesquels nous avons toujours partagé les nostalgies du pays et surtout des moments agréables.

J'exprime ici toute ma gratitude et ma profonde reconnaissance à mon père SENOUSSI HISSEIN, pour tous les soutiens que vous m'avez apportés, nous sommes toujours heureux et très fiers de vous.

Mercie à tous mes frères et sœurs, vous êtes toujours montrées si patients de mon absence, merci encore pour votre soutien.

À toute ma famille d'Adjob, N'djamena, Oum-hadjer, Abéché.

À tous les parents et amis d'Adjob, Oum-hadjer, Abéché, N'djamena.

Table des matières

Liste des figures

Liste des tableaux

Liste des acronymes

ACK : Acknowledgement

ASN : Absolute Slot Number

AMNP : Ad hoc Multi-channel Negotiation Protocol

BER : Bit Error Rate

CDMA : Code Division Multiple Access

CoopMC : Cooperative Multi-Channel MAC

CRC : Cyclic redundancy Code

CSMA : Carrier Sense Multiple Access

CSMA/CA : Carrier Sense Multiple Access /Collision Avoidance

CW : Contention Window

DCA : Dynamic Channel Assignment

DCF : Distributed Coordination Function

EB : Enhanced Beacons

EDCA : Enhanced Distributed Channel Access

EDCF : Enhanced DCF

FDMA : Frequency Division Multiple Access

FER : Frame Error Rate

ISM : Industrial, Scientific and Medical

MAC : Medium Access Control

McMAC : Parallel Rendezvous Multi-Channel MAC Protocol

NACK : NO ACK

NAV : Network Allocation Vector

PAN : Personal Area Network

PCF : Point Coordination Function

QoS : Quality of Service

RBS : Reference Broadcast Synchronization

RDV : Rendez-Vous

RTS/CTS : Request To Send/ Clear to Send

SiSP : Simple Synchronization Protocol

SSCH : Slotted Seeded Channel Hopping

TDMA : Time Division Multiple Access

TMMAC : TDMA Multi-channel MAC

TSCH : Time Slotted Channel Hopping

WiNo : Wireless Node

Introduction générale

Depuis plus d'une décennie, la communauté scientifique spécialiste des réseaux et des protocoles traite de problématiques de recherches désignées par le terme générique « réseaux des capteurs sans fil » (RCSF) liées à plusieurs types d'applications. Initialement, ces applications se focalisaient souvent aux contrôles et à la surveillance de zones et d'environnements non accessibles, où l'accès par l'humain est trop dangereux. On peut par exemple citer : les zones sismiques, les surveillances des ponts, des zones nucléaires, les catastrophes naturelles.

Progressivement, les applications des réseaux de capteurs sans fil se sont élargies au-delà des simples domaines de la surveillance, pour toucher plusieurs domaines d'applications de la vie quotidienne. Ainsi, on les retrouve aujourd'hui dans les bâtiments et maisons intelligentes, les hôpitaux, les aéroports, les industries, et pour globalement des objets communicants pour accomplir une tâche bien définie. Cette application plus large des RCSF leur attribue une autre définition, c'est pourquoi on parle aujourd'hui de l'Internet des objets (IoT). Le but à atteindre par les objets communicants est généralement de répondre à un besoin de mobilité dévolue généralement au domaine désigné par le terme de réseaux mobiles ad hoc.

Par définition, les réseaux mobiles ad hoc sont constitués dynamiquement par un système autonome de nœuds mobiles qui sont connectés via des liaisons sans fil sans l'aide d'une infrastructure réseau existante ou d'une administration centralisée quelconque. Les nœuds sont libres de se déplacer de façon aléatoire et s'organiser de façon arbitraire. Ainsi, la topologie du réseau sans fil peut changer rapidement et de manière imprévisible. Un tel réseau peut fonctionner dans un mode autonome. Les réseaux mobiles ad hoc sont donc des réseaux sans infrastructure car ils ne nécessitent pas une infrastructure fixe, par exemple une station de base, pour leur fonctionnement. En général, les liaisons entre les nœuds d'un réseau ad hoc peuvent contenir plusieurs sauts, et par conséquent, on peut qualifier ces types de réseaux de "réseaux sans fil ad hoc multi-sauts". Chaque nœud est en mesure de communiquer directement à un saut radio avec un autre nœud qui se trouve dans sa zone de transmission. Pour communiquer à n sauts avec les nœuds qui se trouvent en dehors de sa zone de transmission, le nœud a besoin d'utiliser des nœuds intermédiaires pour relayer ou router les messages à chaque saut. Ainsi donc, concevoir une méthode d'accès au médium qui doit tenir compte à la fois du contexte multi-saut et du changement dynamique et imprévisible des topologies du réseau, s'avère essentiel et difficile. Le défi majeur d'une méthode d'accès est non seulement de gérer l'accès au médium afin d'éviter des émissions radio concurrentes et simultanées, mais aussi de résoudre le problème, inhérent aux réseaux sans fil, du nœud caché où le nœud émetteur n'entend pas l'émission d'un autre nœud qui n'est pas dans sa portée radio.

Ces dernières années, on constate un progrès indéniable dans le domaine des réseaux locaux sans fil. Leur déploiement se manifeste dans plusieurs domaines d'applications. Cependant, ces applications font face à plusieurs obstacles dus à une gestion difficile à l'accès au médium. Ceci mène le plus souvent à des problèmes tels que : les collisions, la dégradation du débit et l'augmentation des délais de transmission. Pour surmonter ces défis, des travaux de recherches se sont focalisés sur de nouvelles méthodes d'accès multi-canal qui réduisent le risque de collision.

Les méthodes d'accès multi-canal permettent aux différents nœuds de transmettre simultanément dans une même zone de portée sur des canaux distincts. Ce parallélisme augmente le débit et peut potentiellement réduire le délai et la contention. Cependant, le recours à plusieurs canaux ne va pas sans poser de problèmes. La majorité des interfaces de communication sans fil fonctionne en semi-duplex : soit elles sont en mode émission, soit elles sont en mode réception. L'interface radio est capable de commuter dynamiquement sur les canaux, mais elle ne peut transmettre et écouter sur un canal radio en même temps. De plus, quand un nœud est à l'écoute sur un canal particulier, il ne peut

pas entendre la communication qui a lieu sur un autre canal. On fait face également aux défis classiques du protocole d'accès au médium mono-canal à savoir : le problème du terminal caché, le goulot d'étranglement ainsi que d'autres problèmes tels que la surdité et la partition logique.

Nous présenterons donc au début de ce manuscrit les principales méthodes d'accès multi-canal généralement évoquées dans les travaux de la communauté scientifique. Elles se classent en deux catégories principales : (1) le Rendez-vous (RDV) unique comme le *split phase*, le canal dédié et le saut commun, (2) le Rendez-vous parallèle tel que *SSCH (Slotted Seeded Channel Hopping)* et *McMAC (Parallel Rendez-vous Multi-Channel MAC Protocol)*. Nous pourrons ensuite en déduire que pour une topologie multi-saut que nous envisageons de réaliser avec un coût radio très réduit, certaines de ces méthodes ne sont pas adaptées pour le prototypage mono-interface souhaité.

Les réseaux sans fil multi-canal sont des extensions des réseaux mono-canal. Dans un système multi-canal, il existe N canaux indépendants. Une méthode d'accès consiste à diviser la totalité de la bande passante disponible en partitions homogènes fonctionnant avec le même protocole d'accès multiple sur chaque canal. Par exemple, chaque nœud ayant une trame arrivant peut transmettre sa trame sur un canal choisi aléatoirement en utilisant le protocole d'accès aléatoire. Un utilisateur peut transmettre ou recevoir sur n'importe quel de ***N*** canaux selon le protocole d'accès au medium utilisé [58].

Selon les plates formes matérielles, différents protocoles MAC multi-canal peuvent être développés :(i) des couches MAC multi-canal mono-interface : si la préoccupation est basée sur le coût, une plate-forme matérielle intégrant un seul *transceiver* est préférable. Étant donné qu'un seul *transceiver* est disponible, un seul canal est actif à la fois sur chaque nœud de réseau. Cependant, différents nœuds peuvent fonctionner sur différents canaux simultanément. (ii) des couches MAC multi-canal multi-interface : pour ces types de protocoles MAC multi-canal, une plate-forme matérielle intègre plusieurs *transceivers*, ainsi un nœud peut supporter plusieurs canaux simultanément, par exemple écouter sur un canal et émettre sur un autre. Au-dessus de la couche physique, un seul module de couche MAC est nécessaire pour coordonner les fonctions de plusieurs canaux.

Un réseau où chaque nœud peut communiquer directement avec tous les autres nœuds, est appelé réseau mono-saut (*one-hop*). Dans le cas contraire, des répéteurs ou routeurs relayent les trames, créant ainsi un réseau avec de multiples sauts qu'on peut appeler réseau multi-saut (*multi-hop*). Pour des raisons de limitation de la puissance d'émission de chaque nœud, tous les nœuds ne pourront pas entendre une transmission particulière, et la transmission multi-saut devient alors nécessaire. Une trame, de la source à la destination sera généralement relayée sur plusieurs sauts avant d'arriver à destination. Pour le bon fonctionnement du réseau, des nœuds intermédiaires sont fréquemment sollicités afin d'atteindre leurs destinations, et par conséquent assurer la connectivité des nœuds dans le réseau, formant ainsi un réseau à plusieurs sauts.

Un certain nombre de protocoles d'accès ont été proposés pour les réseaux multi-saut, dans le but d'améliorer leurs performances. Pour les réseaux mono-saut, plusieurs protocoles ont été proposés et profondément étudiés. Le problème crucial pour ces protocoles reste toujours l'ordonnancement des trames (messages), afin d'éviter les collisions et améliorer le débit.

Reste à savoir s'il est possible d'appliquer ces protocoles afin de les adapter à des réseaux ad-hoc multi-canal multi-saut ! Ceci pourra bien évidemment poser des problèmes de synchronisation, et rend l'analyse de performance de ces réseaux beaucoup plus complexe, en raison du fait que les réseaux ad-hoc multi-canal multi-saut sont répartis dans l'espace, opérationnellement décentralisés et impliquent un accès multiple sur plusieurs canaux.

Ce travail de thèse est dans la continuité d'un autre travail de thèse de la même équipe, qui s'est focalisée sur la modélisation et l'analyse formelle des principales approches des protocoles MAC

multi-canal mono-saut. Ces modèles analytiques sont basés sur les réseaux de pétri stochastiques. L'objectif principal est de mettre en place des modèles qui décrivent formellement l'architecture des différents composants pour chacune de ces approches en vue d'analyser leurs comportements et de vérifier leurs propriétés attendues. Deux protocoles MAC multi-canal à Rendez-Vous ont été proposés, PSP-MAC (*Parallel Split Phases multi-channel MAC*) et PCD-MAC (*Parallel Control and Data transfer multi-channel MAC*), qui ont été analysés en termes de débit du réseau et d'impact du temps de commutation entre les canaux, et enfin comparés avec une approche MAC mono-canal (IEEE 802.11 DCF).

Une étude et une analyse systématique de l'état de l'art des protocoles MAC multi-canal proposés, ont identifié que l'un des objectifs principaux de l'ensemble de ces protocoles est de réduire le délai et minimiser les collisions, et par conséquent d'améliorer le débit. Pour atteindre cet objectif, le principe le plus souvent utilisé est le Rendez-Vous (RDV) sur un canal de contrôle, afin d'éviter le conflit sur l'usage des canaux. Un couple de nœuds émetteur/récepteur, pour échanger des données, doit s'accorder sur un canal bien déterminé (conclure un RDV). Ils commutent alors leur interface de façon synchrone, au même moment sur le même canal. L'utilisation des RDV permet aussi d'informer les voisins à un saut de ne pas sélectionner le même canal et provoquer des collisions.

Nous avons ensuite constaté que les méthodes d'accès au médium multi-canal souvent proposés, ont fait l'objet d'études soit à travers des modèles analytiques, soit modélisées et simulées sur des simulateurs réseaux. Ces simulateurs ont souvent le défaut de ne pas prendre en compte finement certains paramètres spécifiques liés à certains protocoles et couches radio. L'étude des performances peut s'avérer alors douteuse vis à vis de la spécificité des réseaux sans fil par rapport à une étude sur une plateforme réelle.

Face à ces problématiques que nous avons identifiées, notre contribution se focalise sur deux questions pertinentes :

Nous avons constaté que le RDV, très souvent utilisé dans les méthodes d'accès multi-canal proposées pour les réseaux sans fil, n'assure pas une transmission sans collision, tout particulièrement dans le cas du nœud caché multi-canal. Les collisions se produisent aussi entre les trames des données et les trames d'acquittements, lorsqu'un récepteur n'ayant pas entendu le RDV conclu par son voisin, utilise le même canal pour acquitter. Le second problème des méthodes d'accès multi-canal à RDV se situe au niveau de la gestion des RDV, puisque les RDV évoqués dans ces méthodes d'accès multi-canal, sont conclus sur des topologies mono-saut. Donc, les voisins au-delà d'un saut vont évidemment perturber ces RDV conclus à un seul saut. C'est pourquoi, il faut aussi trouver des méthodes qui permettent de véhiculer les RDV à plusieurs sauts. Diffuser les RDV sur plusieurs sauts, est une tâche complexe et sans grand intérêt pour l'accès multi-canal, puisque la diffusion des RDV sur plusieurs sauts peut engendrer lui aussi un délai important et ainsi ralentir le débit du réseau. On risque donc de s'éloigner de l'objectif principal de l'accès multi-canal, mais également de son intérêt vers lequel on s'oriente. C'est face à cette complexité de la gestion des RDV multi-sauts que s'inscrit notre contribution. Il s'agit pour nous d'utiliser une méthode d'accès multi-canal aléatoire sans RDV multi-saut, qui doit se baser sur la méthode ALOHA slottée (*Slotted-Aloha*), améliorée pour notre contexte multi-canal. Nous avons implémenté cette méthode d'accès multi-canal sur un *testbed* réel constitué de nœuds WiNo (Wireless Node) mono-interface multi-canal, dont nous évaluons les performances en termes de nombres des trames reçues, de nombre des trames perdues, de taux d'erreur trame (*Frame Error Rate*, FER) en fonction de la charge du réseau. Nous comparons ensuite notre solution avec la même méthode d'accès au médium, mais dans un cas mono-canal avec les mêmes paramètres de performance.

Ce manuscrit de thèse est structuré en trois chapitres. Dans le chapitre 1, nous abordons un état de l'art des différentes méthodes d'accès multi-canal citées dans la littérature. Ainsi, nous avons étudié et analysé les principales approches souvent traitées dans ces méthodes d'accès multi-canal. Pour chacune des approches étudiées, après avoir tracé leurs avantages et inconvénients, nous avons également donné notre point de vue en ce qui concerne leurs implémentations réelles dans un contexte multi-saut.

Dans le chapitre 2, après avoir évoqué les inconvénients des méthodes d'accès multi-canal à RDV étudiées pour un contexte multi-saut, dans une approche de prototypage, nous avons illustré avec des exemples bien précis la complexité de gestion des RDV en multi-saut. Nous avons alors donné les raisons pour lesquelles nous nous orientons vers des méthodes d'accès multi-canal sans RDV, c'est-à-dire aléatoires et multi-saut. Nous avons ainsi proposé plusieurs versions de méthodes d'accès multi-canal sans RDV, que nous avons étudiées, développées et analysées.

Finalement, dans le chapitre 3, nous avons implémenté plusieurs de ces protocoles MAC, et effectué une étude comparative de leurs performances, avec une comparaison avec une méthode d'accès aléatoire mono-canal. Cette étude de performance a été confirmée par une proposition d'un modèle analytique, en parallèle au *testbed* réel, réalisé en deux étapes : avec un faible nombre de nœuds, et ensuite avec un plus grand nombre de nœuds.

Chapitre 1

Etat de l'art des méthodes d'accès multi-canal pour les réseaux locaux sans fil

1.1 Etat de l'art et problématique

1.1.1 Les méthodes d'accès mono-canal proposées pour les réseaux locaux sans fil

Récemment, des nombreuses méthodes d'accès MAC ont été proposées par différents travaux pour les réseaux sans fil en général et les réseaux ad hoc en particuliers et qui avaient abordé diverses questions très pertinentes, parmi lesquelles, les problèmes des nœuds cachés, les nœuds exposés, et les problèmes des collisions, etc. Certaines de ces méthodes mono-canales ont été par la suite sérieusement travaillées afin de les adapter aux transmissions multi-canal dans le but de réduire les différents problèmes rencontrés par l'accès au medium mono-canal.

Dans ce chapitre d'état de l'art, nous évoquons brièvement les principales méthodes d'accès mono-canal proposées généralement pour les réseaux sans fil ; ensuite nous abordons les différents problèmes d'accès au médium mono-canal qui ont suscité d'autres méthodes d'accès multi-canal avec des contraintes spécifiques, dont quelques approches de bases ont été proposées afin de résoudre ces problèmes. Nous introduisons aussi le contexte multi-canal multi-saut qui a été déjà étudié par certains travaux précédents et qui fait aussi l'objet d'étude de cette thèse.

1.1.1.1 Contrôle d'accès au médium dans les réseaux sans fil

Dans les réseaux sans fil, pour que tous les nœuds mobiles utilisent le même spectre de fréquence (canal physique), le contrôle d'accès au médium (MAC) joue un rôle primordial dans la coordination de l'accès au canal entre les nœuds afin que l'information passe d'un nœud à un autre. L'objectif principal du protocole MAC est de coordonner l'accès au canal entre plusieurs nœuds pour atteindre une meilleure utilisation du canal. En d'autres termes, la coordination de l'accès au canal devrait réduire au minimum ou d'éliminer l'incidence de collisions et d'optimiser la réutilisation spatiale en même temps.

Nous allons d'abord décrire les protocoles MAC basés sur la contention où les nœuds accèdent au medium de manière aléatoire et compétitive pouvant conduire à de probables collisions puisque, chaque émetteur émet quand il peut, ainsi donc la probabilité pour que deux nœuds émettent au même moment n'est pas à exclure. On peut citer entre autres parmi ces protocoles : Aloha, Aloha slotté, CSMA et CSMA/CA.

Les méthodes d'accès sans contention font recours à des méthodes de multiplexage permettant aux différents nœuds du réseau d'accéder au canal de façon équitable. Les nœuds suivent un certain

ordonnancement qui leurs garanti un temps de transmission sans collision. L'exemple de ces protocoles : le TDMA *(Time Division Multiple Access),* le FDMA *(Frequency Division Multiple Access)*, et le CDMA *(Code Division Multiple Access)*.

Parmi ceux-ci, TDMA est le plus couramment utilisé dans les réseaux ad hoc. Dans la méthode d'accès TDMA, le canal est généralement organisé en trames, où chaque trame prend place dans nombre fixe de slots de temps. Les hôtes mobiles négocient un ensemble de slots de temps TDMA dans lesquels ils transmettent leurs trames. Si un contrôleur centralisé existe, il est en charge de l'attribution des slots pour les nœuds de la zone qu'il contrôle. De cette façon, les transmissions sont sans collisions, et il est possible de planifier des transmissions de nœud selon un critère d'équité ou de qualité de service [18].

En général, les protocoles sans contention fonctionnent en mode centralisé, surtout sur de topologies statiques, ou encore sur certains petits réseaux dédiés à des applications de taille limitée en nombre de nœuds. Dans un environnement dynamique, distribué et multi-saut, elles sont difficiles à mettre en œuvre. Par contre les protocoles basés sur la contention sont plus adaptés au contexte distribué et au concept de l'auto-organisation [12].

1.1.1.2 Les protocoles MAC basés sur la contention

1.1.1.2.1 La méthode d'accès ALOHA ou ALOHA pur *(Pure ALOHA)*

ALOHA est le premier protocole d'accès aléatoire, le plus connu et simple : chaque fois qu'une station a un paquet à transmettre, alors il transmet simplement sans attendre ou vérifier si une autre station est déjà en cours de transmission. Les nœuds ne se préoccupent pas de l'état du canal. Le principe de fonctionnement est le suivant : à chaque fois qu'une station veut envoyer de données, alors elle émet son paquet sans se préoccuper de l'état du canal. Si elle ne reçoit pas un accusé de réception au bout d'un certain délai prédéfini, elle retransmet le paquet de la même manière après un délai d'attente aléatoire. Si au-delà d'un certain nombre de tentatives aucun acquittement n'a pas été reçu, alors il y a échec de la transmission. Au cas où plusieurs stations émettent des trames en même temps, il y aura donc un conflit d'accès et les paquets seront en collision. Cette méthode d'accès n'est pas avantageuse pour un réseau de grande taille et multi-saut.

Un autre protocole a été proposé pour corriger les erreurs engendrées par ALOHA, appelé ALOHA slotté *(Slotted-ALOHA ou SA)*

1.1.1.2.2 ALOHA slotté (Slotted-ALOHA)

Cette méthode apporte une amélioration à ALOHA pur afin de réduire le nombre des collisions. Ainsi le temps est discrétisé en slots de temps d'une durée fixe appelée « slots » et les stations ne vont émettre qu'au début de chaque slot. Même avec ALOHA slotté, des collisions peuvent se produire lorsque deux paquets sont émis dans le même slot de temps. Toutefois, la probabilité de collision est réduite et le débit est amélioré [21] [22].

Dans le système slotté, tous les paquets transmis ont la même longueur et chaque paquet nécessite une unité de temps (appelée slot) pour la transmission. Les stations sont autorisées à commencer la transmission uniquement au début d'un slot et non pas à n'importe quel moment. L'inconvénient est

que, contrairement ALOHA pur, ALOHA slotté nécessite une synchronisation de temps global qui est un problème difficile en soi [22].

1.1.1.2.3 Le protocole CSMA

Contrairement à ALOHA où les stations émettent leurs données quand elles veulent transmettre, pour les protocoles de détection de la porteuse, les stations écoutent le canal avant de transmettre. Chaque station écoute si le canal est libre, elle émet ses paquets de données ; s'il est occupé, elle diffère alors sa transmission selon une probabilité *p* de persistance d'écoute, d'où on distingue trois variantes du CSMA (Table 1) : *persistent CSMA, non-persistent CSMA,* et *p-persistent CSMA*. Ces protocoles diffèrent par l'action dont la station qui a un paquet à transmettre occupe après détection le canal au repos [19].

En CSMA persistent (*persistent CSMA*), le nœud écoute de manière permanente le canal, dès qu'il est libre, il émet ses données. Par contre, le protocole CSMA non persistant (*non-persistent CSMA)* a été proposé afin de corriger les défauts précédents : un nœud qui veut transmettre des données détecte le canal pour déterminer si aucun autre nœud n'a commencé à transmettre. Si le nœud détecte une activité sur le canal, il effectue une opération de temporisation *(backoff)* avant d'essayer de transmettre à nouveau. Lorsque le nœud ne détecte pas d'activité sur le canal, il transmet immédiatement ses données. Pour le protocole *p-persistent CSMA,* le nœud continue à détecter le canal lorsqu'il y a une activité sur ce canal, plutôt que de différer et de vérifier un peu plus tard. Lorsque le nœud ne détecte pas d'activité sur le canal, soit à la première tentative ou à la fin d'une transmission précédente par un autre nœud, il transmet avec une probabilité *p*, et diffère l'émission avec une probabilité *1-p* [19].

Une autre version étendue de CSMA, appelé CSMA avec évitement de collision *(CSMA/CA Carrier Sense Multiple Access with Collision Avoidance)*, ajoute des mécanismes pour limiter le nombre des paquets des données qui seront perdus lorsque les nœuds proches transmettent en même temps.

Les réseaux sans fil cherchent à éviter les collisions plutôt que de les détecter, c'est ainsi que le CSMA/CA tente d'éviter les collisions en utilisant un échange de messages de contrôles pour réserver le canal sans fil avant chaque transmission de données. Le nœud émetteur transmet d'abord un message de contrôle : une demande d'envoi au récepteur *(RTS Request To Send)*. Si le récepteur peut recevoir les données en cours, il répond avec un message de contrôle : prêt à émettre *(CTS Clear To Send)*. Le nœud émetteur réessaie la transmission par la suite, s'il ne reçoit pas un CTS dans un certain délai. Après la bonne réception d'un CTS, l'émetteur transmet ses données.

Les nœuds voisins qui reçoivent un message RTS ou CTS savent alors qu'un transfert de données se fera bientôt et différent leurs émissions des données jusqu'à un certain délai.

Quand un réseau multi-saut distribué est mis en œuvre, une synchronisation précise dans le réseau global est difficile à réaliser. Par conséquent, les systèmes d'accès multiples distribués tels que CSMA/CA sont plus favorables. Cependant, CSMA/CA a une très faible fréquence d'efficacité réutilisation spatiale, ce qui limite considérablement le passage à l'échelle (scalabilité) des réseaux multi-saut basé CSMA/CA [20].

Comme on peut le constater, CSMA/CA introduit volontairement un délai de transmission afin d'éviter une collision. Éviter les collisions augmente l'efficacité du protocole en termes de pourcentage de paquets qui seront transmis avec succès (de débit utile)

Table 1 : Caractéristiques des trois protocoles CSMA de base lorsque le canal est détecté libres ou occupés

Protocol CSMA	Mode de transmission
Non-persistant	Si le médium est libre, transmettre. Si le médium est occupé, attendre un délai aléatoire et écouter à nouveau le canal.
1-persistant	Si le médium est libre, transmettre. Si le médium est occupé, continuer l'écoute jusqu'à ce que le canal sera libre; puis transmettre immédiatement.
p-persistant	Si le médium est libre, transmettre avec une probabilité *p*. Si le médium est occupé, continuer l'écoute jusqu'à ce que le canal sera libre;

1.1.1.3 Les protocoles MAC mono-canal proposés pour le standard IEEE 802.11

Le protocole MAC 802.11 prend en charge deux modes de fonctionnement, à savoir la fonction de coordination distribuée *(Distributed Coordination Function DCF)* et la fonction de coordination par point (*Point Coordination Function, PCF)*. Le mode DCF offre un service best-effort c'est-à-dire sans priorité et sans garantie, tandis que le mode PCF a été conçu pour prendre en charge le trafic en temps réel dans des configurations de réseau sans fil basé sur une infrastructure [23] [24]. Le mode DCF n'utilise aucun type de contrôle centralisé, toutes les stations sont autorisées à concurrencer pour le support partagé simultanément.

Le protocole IEEE 802.11 DCF : le mode DCF est essentiellement basé sur le principe du CSMA/CA. Pour réduire la probabilité de collisions, la DCF applique un mécanisme d'évitement des collisions *(CA Collision Avoidance)*, où les stations effectuent une procédure de temporisation *(backoff)* avant d'engager une transmission. Après avoir détecté le support inactif pendant une durée minimale appelé espace inter trame DCF *(DIFS Interframe)*, les stations continuent à détecter le support pour un temps aléatoire supplémentaire appelé le temps de temporisation *(backoff)*. Une station initie sa transmission que si le medium reste inactif pendant ce temps aléatoire supplémentaire. La durée de ce temps aléatoire est déterminée individuellement par chaque station. Une nouvelle valeur aléatoire indépendante est choisie pour chaque nouvelle tentative de transmission. Il en résulte aucun mécanisme pour différencier entre les stations et leur trafic, et donc pas de support de la QoS *(Quality of service)* dans le DCF. Si le médium est très occupé en raison d'interférences ou d'autres transmissions alors qu'une station décompte son compteur d'attente, la station s'arrête de décompter et diffère l'accès au support jusqu'à ce que le medium devienne inactif pendant un nouveau DIFS. Cela se produit, par exemple, lorsque la durée de temporisation aléatoire d'une station est plus longue que celle d'au moins une autre station. Les stations qui diffèrent leur accès au support parce qu'elles ont détecté que le médium était occupé ne sélectionnent pas une nouvelle durée de temporisation aléatoire, mais continuent à décompter les slots de temps de la temporisation différée après détection à nouveau du support à l'état inactif. La figure 1 illustre le temps de l'accès au canal en mode DCF.

Chaque nœud sélectionne une temporisation aléatoire uniformément distribué dans [0, CW], où CW est la taille actuelle de la fenêtre de contention *(Contention Window CW)*. Il diminue le compteur de temporisation d'une unité pour chaque slot de temps libre (d'inactivité du canal). La transmission commence quand le *timer* de temporisation atteint zéro. Lorsqu'il y a des collisions, lors de la transmission, ou en cas d'échec de la transmission, le nœud double la valeur de CW jusqu'à ce qu'elle atteigne la valeur maximale CW_{max}. Par la suite, le nœud commence le processus de temporisation à nouveau, et retransmet le paquet lorsque la temporisation est terminée. Si la limite maximale d'échec

de transmission est atteinte, la retransmission doit s'arrêter, *CW* est remis à la valeur initiale CW_{min}, et le paquet sera rejeté.

Comme le mode DCF n'utilise aucun type de contrôle centralisé, toutes les stations sont autorisées à concurrencer simultanément pour le support partagé. Par contre, la méthode d'accès PCF dont le fonctionnement est contrôlée par un équipement central *(polling master)*, le coordinateur, qui scrute toutes les stations et donne les droits d'émettre, par un processus appelé "*polling"* [24]. La fonction PCF est implémentée au-dessus de la DCF. Etant donné que, d'une part le coordinateur est à portée de tous les autres nœuds et, d'autre part, utilise l'espace PIFS *(PCF IFS)*, plus court que l'espace DIFS, tout autre trafic asynchrone est exclu une fois qu'il obtient le média lors du *polling* et de la réception des réponses.

Il faut aussi signaler que le protocole MAC 802.11 ne prend pas en charge le concept de la différenciation des trames avec des priorités différentes. Ainsi donc la DCF fournit un accès au canal avec des probabilités égales à toutes les stations qui concurrencent pour l'accès au canal de manière distribuée [24].

L'amendement de la norme IEEE 802.11e introduit la différenciation de trafic et des services sur la couche MAC afin qu'il puisse fournir des flux multimédias avec le respect de leurs exigences de qualité de service et en temps réel en plus de trafic best effort [25].

Ainsi donc, la EDCA *(Enhanced Distributed Channel Access)* autrement appelé EDCF *(Enhanced DCF)* a été proposée pour améliorer la fonction DCF en introduisant la priorisation du trafic [25]. L'IEEE 802.11 EDCA fournit un accès au médium basé sur la contention et distribué qui peut prendre en charge la différenciation de service entre les classes de trafic *AC (Access Categories)* : AC_BK *(background traffic)*, AC_BE *(best effort traffic)*, AC_VI *(video traffic)* et AC_VO *(voice traffic)*, dont chacune a sa propre file d'attente de transmission et ses propres paramètres d'accès au canal tels que :

- ✓ *AIFSN (Arbitrary interframe space number)* : utilisé à la place de DIFS, Chaque *AC (Access Categories)* commence sa procédure de backoff quand le canal est inactif pendant une période de *AIFSN* au lieu de DIFS (Le *SIFSN* est au moins égal à DIFS). Ainsi donc le trafic le moins prioritaire utilisera la plus grande valeur de *AIFSN*.
- ✓ *Contention Window CW* (CW_{min} *et* CW_{max}): Un nombre aléatoire est tiré à partir de cet intervalle pour le mécanisme de *backoff*. Chaque catégorie de trafic concurrence l'accès au canal avec différents paramètre de CW_{min} *et* CW_{max}, ainsi la valeur minimale CW_{min} est utilisée par les classes de trafics de plus haute priorité.
- ✓ *Transmission opportunity (TXOP) limit*: La durée maximale pendant laquelle un nœud peut transmettre un maximum de flux après avoir obtenu l'accès au canal.
  La valeur minimale de CW_{min} pour chaque *AC (Access Categories)* peuvent être différentes les unes des autres. L'attribution de valeurs plus petites de CW_{min} aux classes de haute priorité peut assurer que les classes de haute priorité obtiennent plus *TXOPs* que ceux de faible priorité. L'idée de CW_{min} est similaire pour CW_{max}.

En conclusion, même si l'objectif primordial du protocole est d'assurer la qualité de service dans le réseau, puisque l'accès au medium du protocole EDCA est basé sur la contention, il faut donc qu'il y'ait au préalable un rendez-vous entre les nœuds avant toute transmission ; dès que le réseau est saturé, la QoS sera rapidement remise en cause et surtout quand il s'agit du multi-saut. Peut-être en exploitant tous les canaux dédiés à la norme 802.11 et utilisant un protocole d'accès multi-canal adéquat, que la QoS sera assurée au minimum.

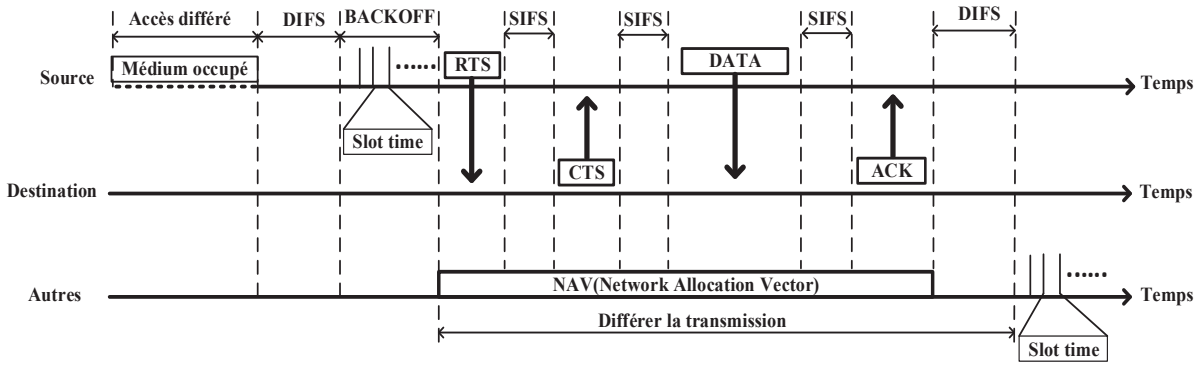

Figure 1: Accès au canal IEEE 802.11 DCF

1.1.2 Les méthodes d'accès multi-canal mono-saut

Le réseau sans fil ad-hoc est composé d'un groupe d'hôtes mobiles équipés chacun d'une interface sans fil et peut être déployé rapidement et sans une infrastructure quelconque établie ou une administration centralisée. Pour des raisons de limitations telles que la puissance radio et l'utilisation de différents canaux, un mobile peut ne pas être en mesure de communiquer directement avec d'autres hôtes dans un mode mono-saut. Dans de tels cas, un scénario multi-saut se produit, où les paquets envoyés par un hôte source doivent être relayés par un ou plusieurs hôtes intermédiaires avant d'atteindre leur destination. Nous rappelons que l'une des principales problématiques de ce type de réseaux est la gestion de la méthode d'accès au medium. Cette dernière a été la cible de plusieurs travaux de recherche mais, le plus souvent dans un contexte mono-canal. Dernièrement, des travaux ont abordé le cas multi-canal et déjà certains résultats pourraient servir de base afin d'étendre les capacités du réseau, les performances de la méthode d'accès au médium, etc. Il est aussi primordial de penser au cas multi-canal et multi-saut, puisque ces protocoles sont souvent utilisés dans un cadre mono-saut.

Plusieurs protocoles MAC multi-canal pour les réseaux ad-hoc sans fil ont été proposés, puisqu'ils permettent aux différents nœuds de transmettre parallèlement sur des canaux distincts sans collision, ce qui permet d'augmenter le débit et potentiellement réduire les délais de transmission. Cependant, la plupart des protocoles proposés sont des protocoles à rendez-vous unique qui sont soumis à la congestion du canal de contrôle. En général, les différents protocoles se distinguent par la manière dont les nœuds du réseau établissent les rendez-vous ou en d'autres termes, comment les nœuds négocient les canaux à utiliser pour la transmission des données.

Le premier protocole MAC multi-canal qui a été présenté dans [1] et [2] est nommée DCA (*Dynamic Channel Assignment*) ; il utilise deux interfaces : une interface pour les échanges des trames de contrôles, l'autre pour les transferts de données. Dans ce protocole, chaque nœud maintient une liste de canaux libre (*Free Channel List FCL*) pour enregistrer les canaux de données libre.

Avec DCA, lorsqu'un nœud source a des données à émettre, il émet une trame RTS *(Request To Send*) incluant la liste des canaux disponibles (FCL) qui ne sont pas utilisés par ses voisins à un saut. Après avoir reçu le RTS, le nœud destinataire compare la FCL reçu avec sa propre FCL et sélectionne un canal libre commun. Ensuite, le nœud destinataire indique au nœud source et à ses voisins, le canal de données sélectionné en envoyant un CTS (*Clear toSend).* En recevant le CTS, chaque nœud informe également ses voisins du canal sélectionné en envoyant une trame RES *(Reservation).* On remarque que par rapport à la norme IEEE 802.11 DCF, le protocole DCA nécessite une trame de contrôle supplémentaire RES pour réserver le canal sélectionné.

Dans [1], [3] et [4] les auteurs classifient les protocoles MAC multi-canal en deux catégories : le rendez-vous unique (à savoir le canal de contrôle dédié), le saut commun, le *Split Phase*, et les protocoles de rendez-vous parallèles comme par exemple le *SSCH* (*Slotted Seeded Channel Hopping*) [5] et *McMAC* (*Parallel Rendezvous Multi-Channel MAC Protocol*) [6]. Les protocoles MAC de rendez-vous uniques ont un canal de contrôle commun appelé également canal de rendez-vous. Les nœuds peuvent échanger les trames de contrôle et négocient les canaux de transmission de données sur ce canal. Ce canal de contrôle, cependant, peut devenir un goulot d'étranglement au fur et à mesure que le trafic de données augmente.

Les protocoles MAC à rendez-vous parallèles, par contre, n'ont pas besoin d'un canal de contrôle commun. L'idée principale de ces protocoles est que les nœuds sautent entre les différents canaux en fonction de leurs propres séquences et les informations de contrôle sont échangées sur différents canaux. Plusieurs rendez-vous peuvent alors s'établir simultanément ; les nœuds arrêtent leurs sauts quand ils concluent des accords et commencent à transmettre des données et, ensuite, reprennent leurs séquences de saut à la fin de la transmission.

Dans [3], Crichigno, J., et *al.* comparent les protocoles à rendez-vous unique et parallèle en terme de nombre de canaux et de débit ; d'après leur étude et en considérant que tous les nœuds sont équipés d'une seule interface radio, ils déduisent que, les protocoles de rendez-vous parallèles tels que *McMAC* et *SSCH* sont plus performant que les protocoles de rendez-vous unique puisqu'ils éliminent le goulot d'étranglement du canal de contrôle.

Dans [7], El Fatni et *al.* proposent deux solutions MAC multi-canal afin de pallier au problème de goulot d'étranglement du canal de contrôle. L'un des protocoles est appelé *PSP-MAC* (*Parallel Split Phase multi-channel MAC*) qui exploite le *split phase* en appliquant le parallélisme pendant la phase de contrôle. L'objectif principal est d'exploiter tous les canaux durant cette phase. Le deuxième protocole proposé est *PCD-MAC* (*Parallel Control and Data transfer multi-channel MAC*), il exploite le concept du rendez-vous multiple et du canal de contrôle dédié. Ce protocole exclut le concept de deux phases par cycle. Malheureusement, ces deux propositions ne prennent pas en compte les topologies multi-saut.

Certains travaux ont également évoqué l'accès multi-canal multi-saut. Nasipuri et al [13] ont mis en œuvre un protocole *CSMA (Carrier Sense Multiple Access)* MAC multi-canal pour les réseaux sans fil multi-saut. L'objectif principal de leur travail est de réduire les collisions qui peuvent se produire pendant les transmissions dans les réseaux sans fil et réduire également l'effet du nœud caché. Ce protocole permet une sélection dynamique de canaux, l'idée est semblable à la méthode d'accès FDMA *(Frequency Division Multiple Access),* mais la différence en est que, pour le CSMA MAC multi-canal, l'affectation des canaux se fait de manière distribuée sans aucune infrastructure centrale à travers le système CSMA.

Chaque nœud surveille les *N* canaux continuellement, à chaque fois qu'il ne transmet pas. Il détecte si oui ou non la puissance totale du signal reçu *(Total Received Rignal Strength TRSS)* dans les canaux sont au-dessus ou au-dessous de son seuil de détection *(Sensing Threshold ST)*. Les canaux pour lesquels la TRSS est en dessous de la *ST*, sont marqués comme *IDLE* (libre). Le temps auquel le TRSS a chuté en dessous du *ST* est noté pour chaque canal. Ces canaux sont mis sur une liste de canaux libres. Le reste des canaux sont marqués comme *BUSY* (occupé).

Lorsque le nombre de canaux N est suffisamment grand, le protocole a tendance à réserver un canal pour la transmission de données pour chaque nœud. Ce système de réservation de canal réduit les occasions où deux transmissions concurrentes choisissent le même canal. Cependant, un grand nombre de canaux peut causer un temps de transmission paquet trop élevé.

Dans [14], les auteurs proposent un protocole MAC multi-canal multi-saut basé sur CSMA qui sépare le canal de contrôle pour l'échange des trames des contrôles et un nombre prédéfini des canaux inférieur à celui des nœuds dans le réseau pour les transmissions des données, afin d'éviter des probables collisions entre les paquets des données et les trames des contrôles. Etant donné que le protocole est basé sur le système *CSMA (Carrier Sense Multiple Access),* en présence d'une forte charge des trafics, il souffre du problème multi-canal du terminal caché. Comme les auteurs étendent une idée de la méthode d'accès 802.11 *DCF (Distributed Coordination Function)* avec quelques structures différentes : un canal de contrôle approprié pour les trames de contrôles séparés des canaux de transmissions des données.

Avant de transmettre une trame *RTS (Request to send),* l'émetteur détecte la porteuse sur tous les canaux de données et crée une liste de canaux de données disponibles pour la transmission, c'est-à-dire les canaux dont la puissance totale reçue est inférieure au seuil de détection de la porteuse. Cette liste est incorporée dans le paquet RTS du nœud émetteur.

Si la liste de canaux libre est vide, le nœud tire un backoff et réessaie de transmettre une autre trame RTS après l'expiration du backoff.

Contrairement au 802.11, les autres nœuds voisins de l'émetteur recevant les RTS sur le canal de contrôle différent leurs transmissions seulement jusqu'à la durée du *CTS (Clear to send)* et pas jusqu'à la durée de *ACK (Acknowledgement)*. C'est parce que les données et ACK sont transmis dans un canal de données et donc ne peuvent pas interférer avec les autres transmissions RTS/CTS. Ainsi donc les nœuds dans le voisinage du nœud destinataire, sur réception du paquet CTS sur le canal de contrôle, s'abstiennent d'émettre sur le canal de données indiqué dans le CTS pour la durée de l'ensemble de transfert (y compris les ACK).

Après avoir reçu la trame RTS et, avant d'émettre le CTS, le nœud destinataire crée sa propre liste de canaux libre par détection de la porteuse sur tous les canaux de données. Il compare ensuite cette liste des canaux libre avec celle contenue dans la trame RTS.

S'il y a des canaux libres en commun, le nœud destinataire choisit un canal commun et envoie ces informations dans la trame CTS.

S'il n'y a pas des canaux libres communs disponibles, le destinataire n'envoie pas un CTS. Puis la source une fois encore tente d'envoyer un autre RTS après un *backoff*.

Lorsque le nœud émetteur reçoit la trame CTS, il transmet le paquet de données sur le canal de données indiqué dans le CTS.

Ensuite les auteurs comparent leur protocole MAC multi-canal en termes de débit et du délai par rapport à la méthode d'accès 802.11 à canal unique. D'après leur résultat, ils concluent qu'il y'ait une nette amélioration du débit et du délai de leur méthode proposé sur la méthode 802.11.

Les auteurs de [17] modifient la couche MAC du 802.11 et l'adaptent en une méthode multi-canal multi-saut, puisque dans cette norme on distingue trois canaux qui peuvent être utilisés simultanément sans interférence. L'objectif de ce travail est de proposer un système de contrôle congestion au niveau des nœuds intermédiaire, afin d'éviter un problème du délai et dégradation du débit dans le réseau, aussi bien de résoudre le problème multi-canal du nœud caché.

Chen et *al* [15] proposent un protocole MAC multi-canal multi-saut appelé AMNP pour *Ad hoc Multichannel Negotiation Protocol for multihop mobile wireless networks* qui utilise un seul canal pour les trames de contrôle et le reste des canaux pour la transmission des données. Ils étendent le concept des trames de contrôles RTS/CTS utilisé dans le standard 802.11 où des champs

supplémentaires ont été ajoutés pour indiquer le canal sélectionné aussi bien pour préciser les canaux libres ou en cours d'utilisation.

Shila et *al* [16] ont mis en œuvre un protocole MAC multi-canal multi-saut appelé CoopMC (*A Cooperative Multi-Channel MAC Protocol for Wireless Networks*) MAC qui permet aux nœuds du réseau de négocier des canaux dynamiquement. Dans CoopMC MAC cinq trames de contrôle sont utilisées, les nœuds voisins à un saut s'échangent des tables d'informations sur les canaux afin de trouver le meilleur canal entre la source et destination. Notons ici que le canal de contrôle sera aussi utilisé pendant la phase de transmission des données. La table 2 recapitule certaines caractéristiques des méthodes d'accès multi-canal discutées dans cette partie de l'état de l'art.

Table 2 : Comparaison des protocoles MAC multi-canal

Protocoles	***Nombre d'interfaces par nœud***	***Nombre de rendez-vous***	***Nombre des sauts***	***Principe de négociation du canal***
DCA [1] [2]	2	Unique	Mono-saut	Canal de contrôle commun
Canal dédié [1] [3]	1	Unique	Mono-saut	Canal de contrôle commun
Split phase [1] [3]	1	Unique	Mono-saut	Période de contrôle commune
Saut commun [1] [3]	1	Unique	Mono-saut	Séquence de saut commune
CoopMC [16]	1	Unique	Mono-saut	Canal de contrôle commun
Saut indépendant (SSCH, McMAC) [1] [3]	1	Multiple	Mono-saut	Séquence de saut indépendant
CSMA multi-canal [13]	1	Unique	Multi-saut	Par détection de la porteuse
CSMA multi-canal à canal de contrôle séparé [14]	2	Unique	Multi-saut	Canal de contrôle commun
AMNP [15]	1	Unique	Multi-saut	Canal de contrôle commun
TMMAC [72]	1	Unique	Mono-saut	Période de contrôle commune
TSCH [42]	1	Multiple	Multi-saut	Séquence de saut indépendant

1.1.2.1 Intérêts de l'approche multi-canal par rapport au mono-canal

Dans une transmission mono-canal, le canal de transmission de données est une ressource partagée entre plusieurs nœuds dans une même zone de portée de communication. Les nœuds vont alors concourir pour accéder à cette ressource ; par conséquent, des collisions pourront parfois se produire, affectant ainsi le débit et le délai. Lorsque plusieurs canaux sont utilisés, les transmissions concurrentes dans la même zone de portée peuvent se dérouler parallèlement sur les différents canaux disponibles, ce qui améliore donc les performances en termes de débit et du délai. Comme le montre la figure 2, les trois transmissions se produisent simultanément sur les trois canaux et dans un seul slot de

temps, ce qui augmente trois fois le débit par rapport au système mono-canal et, réduit aussi le délai de deux slots de temps.

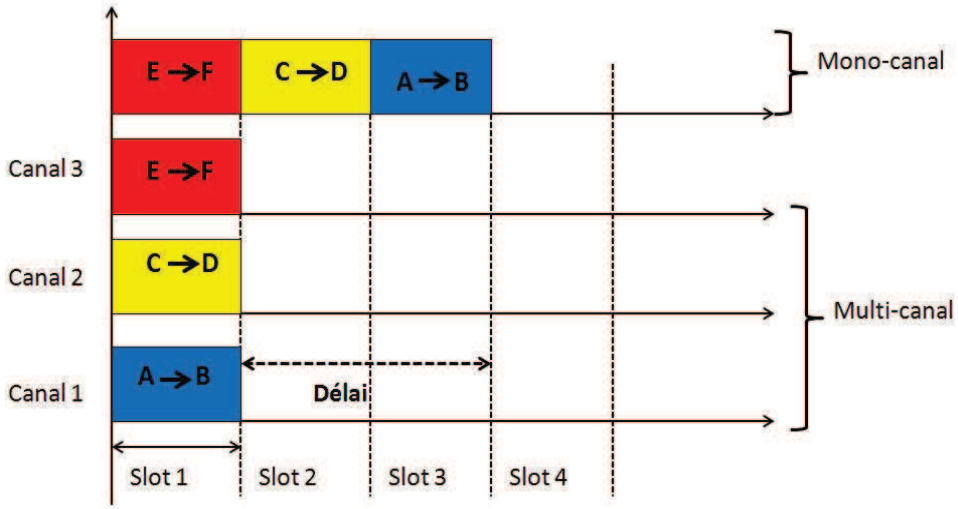

Figure 2: intérêt d'un système multi-canal par rapport à un système mono-canal

1.1.2.2 Les problèmes multi-canal

Les méthodes d'accès multi-canal sont confrontées à plusieurs défis majeurs, dont certains sont quasiment les mêmes que pour leurs homologues mono-canal, comme par exemple le problème du terminal caché, ou le goulot d'étranglement du canal de contrôle ; d'autres, par contre, sont propre au contexte multi-canal, comme la surdité, le problème de diffusion (*broacast*) et la partition logique.

La grande difficulté pour l'accès multi-canal porte sur le choix du canal à utiliser et le partage des canaux disponibles par les nœuds dans un contexte reparti. Dans le contexte multi-canal, pour qu'un nœud transmette des données, il doit nécessairement connaître le canal sur lequel son récepteur est prêt à recevoir les données envoyées [7]. Par conséquent les protocoles MAC multi-canal nécessitent un autre mécanisme qui va se charger de l'allocation des canaux, c'est-à-dire de décider quel canal sera utilisé par tels nœuds est à tel moment. Ce mécanisme a pour rôle principal de mettre en place des méthodes pour le choix d'un canal par les nœuds. Ainsi, l'émetteur et le récepteur doivent se trouver finalement sur le même canal et en même temps pour les transmissions des données. C'est ce que nous appelons l'établissement des rendez-vous par les nœuds.

Le problème du terminal caché [1] [3] [7] [9] se produit très souvent lors que les nœuds sont équipés d'une seule interface radio, ce qui entraîne un manque d'information sur l'état de certains canaux. Ceci provoquera des collisions au niveau des récepteurs.

Comme on peut le remarquer sur la figure 3 (a), après avoir échangé des trames des contrôles RTS et CTS sur le canal 1 (canal de contrôle par défaut), les nœuds K et L décident d'utiliser le canal 2 ; au même moment F et G décident d'utiliser le canal 3. K et L ne sont pas conscients du choix de canal de F et G, et décident d'utiliser le canal 2, provoquent alors une collision au niveau du récepteur G.

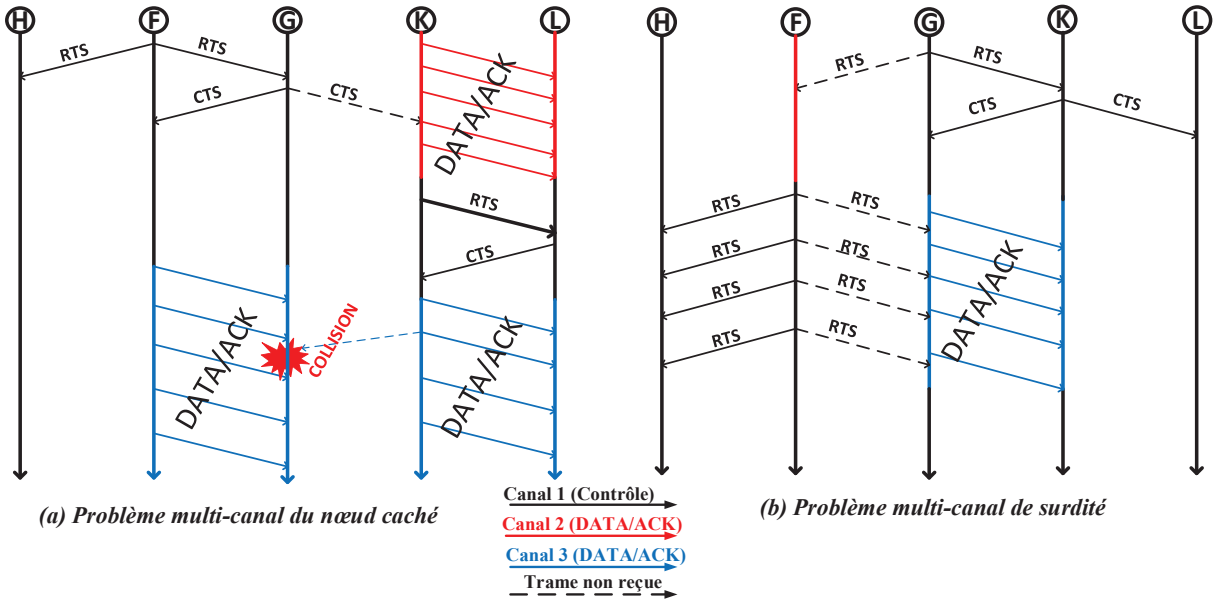

Figure 3: problèmes multi-canal du terminal caché et de surdité

Le problème de surdité [1] [3] [7] [9] survient par manque d'informations du canal sur lequel se trouve le nœud destinataire. Ainsi, la trame de contrôle sur le canal de contrôle rate sa destination, occupée sur un autre canal. Ce problème peut se voir sur la figure 3 (b). Après avoir échangé des trames de contrôle sur le canal 1, G et K commutent sur le canal 2 pour transmettre des données. N'ayant pas d'information sur le fait que le nœud G est en activité sur le canal 2, le nœud F envoie plusieurs trames de contrôle RTS à G sur le canal 1 (canal de contrôle par défaut) mais ne reçoit aucune réponse de la part de G ; par conséquent F conclut à tort que le lien entre F et G est rompu et abandonne la transmission de données par la suite et au même moment empêche son voisin H d'utiliser le canal qu'il vient de réserver. Ces trames émises surchargent le canal de contrôle inutilement.

Le problème de partition logique est un autre cas qui se manifeste lorsqu'une partie du réseau s'isole des autres nœuds par manque d'informations sur l'utilisation des canaux [7] [10].

La diffusion est une activité importante dans les réseaux Ad-hoc [3] [10] [11], surtout lorsqu'il faut diffuser une trame pour coordonner tous les nœuds dans une même zone de portée. Cette activité de diffusion est assez simple dans une méthode d'accès mono-canal puisque tous les nœuds écoutent sur le même canal. Cependant, dans un contexte multi-canal, ce phénomène est souvent complexe du fait que les nœuds commutent sur différent canaux pour transmettre ou recevoir des données, par conséquent, ils peuvent manquer facilement une trame de diffusion (qui généralement n'est pas acquittée donc non sécurisée et ainsi définitivement perdue pour eux).

Dans [11], pour résoudre ce problème, les auteurs utilisent une technique de diffusion d'une balise sur le canal de contrôle. Tous les nœuds qui ont reçu cette balise doivent attendre sur ce canal pour recevoir une trame de diffusion, même si le nœud a déjà négocié un autre canal (rendez-vous) pour transmettre des données.

Pour trouver des solutions aux différents problèmes multi-canal que nous venons d'évoquer, la plupart des recherches ont proposé quatre approches principales, mais plusieurs n'ont abordé le problème que dans un contexte mono-saut.

1.1.2.3 Les différents protocoles multi-canal proposés

1.1.2.3.1 L'accès multi-canal mono interface

1.1.2.3.1.1 Le protocole du canal de contrôle dédié

Le canal de contrôle dédié [1] [3] [4] [12] [38] [39] est un protocole de rendez-vous unique, chaque nœud est muni d'une interface de contrôle et d'une interface de données. L'interface de contrôle est fixée de façon permanente sur un canal commun (appelé canal de contrôle) pour l'échange des trames de contrôle. L'interface de données peut basculer entre les canaux restants (appelés canaux de données) pour la transmission de données. L'idée principale du protocole est d'isoler les trames de contrôle de celles de données en affectant un canal fixe pour échanger des trames de contrôle RTS et CTS, et pour éviter ainsi les interférences entre les trames de contrôle et les paquets de données. Plusieurs travaux considèrent le protocole multi-interface, alors qu'El Fatni et al. [12] le considèrent parmi les protocoles mono-interfaces multi-canal.

Le principe de fonctionnement du protocole est le suivant : lorsqu'une paire de nœuds A et B veut échanger des données, l'émetteur A envoie une trame RTS qui contient une liste des canaux libres dans sa zone de portée sur le canal de contrôle. Le récepteur B choisit un canal libre commun parmi les canaux de la liste envoyés par A en répondant par une trame CTS, qui comprend le canal sélectionné pour le transfert de données. A et B commutent alors leurs interfaces sur le canal sélectionné et commencent à transmettre des données. Les trames RTS et CTS incluent également le NAV*(Network Allocation Vector)* pour informer les voisins de A et B de la durée pendant laquelle le canal sera occupé. Dans [12], les auteurs utilisent une troisième trame de contrôle supplémentaire RES *(Reservation)* aux deux trames RTS et CTS pour confirmer la réservation du canal. La figure 4 explique le principe de canal de contrôle dédié. L'intérêt de ce protocole est qu'il simplifie la diffusion d'une trame, puisqu'il y a une interface radio fixée de manière permanente sur le canal de contrôle, ainsi la diffusion *(broadcast)* sera réalisée sur ce canal. L'inconvénient de ce protocole est qu'il n'est pas une solution du problème majeur multi-canal du terminal caché et de la surdité. Comme le canal de contrôle est unique, si plusieurs nœuds tentent de conclure des accords pour transmettre des données, le canal de contrôle deviendra un goulot d'étranglement. Ainsi pendant la période des transmissions des données les nœuds n'ont aucune information concernant les rendez-vous établis sur le canal de contrôle et, ainsi donc ignorent les informations sur les canaux de données réservés pendant cette période. Ce manque d'information conduisant aux problèmes suivants : 1) le problème multi-canal de surdité où un nœud cherche à établir un rendez-vous avec un autre nœud se trouvant sur autre canal en émission ou en réception des données. 2) le problème multi-canal du nœud caché survient dans le cas où deux établissent un rendez-vous et commutent sur un canal qui est déjà occupé par un autre pair de nœud. Nous remarquons aussi que pendant le transfert des données, le canal de contrôle n'est pas utilisé, ceci prouve que, l'approche du canal de contrôle dédié gaspille aussi de la bande passante. 3) dans le cas multi-saut, il faudrait trouver une solution pour diffuser à plus que 1 saut, les trames RTS, CTS (et RES éventuellement) aux nœuds qui peuvent gêner les échanges DATA-ACK.

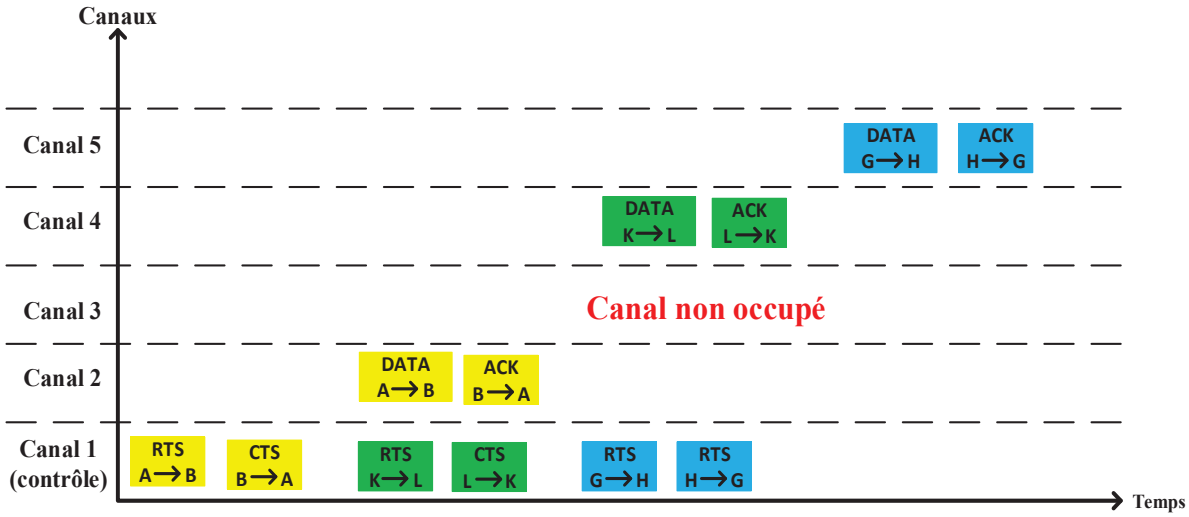

Figure 4: principe du canal de contrôle dédié

1.1.2.3.1.2 Le protocole Split phase

Pour ce protocole de ***Split phase*** [1] [3] [4] [12], les nœuds utilisent une seule interface et le temps est divisé en une séquence alternée de phases de contrôle et d'échange de données. Pendant la phase de contrôle, tous les nœuds commutent leurs interfaces sur le canal de contrôle et tentent de conclure des accords pour les canaux qui seront utilisés lors de la phase d'échange de données suivante : tous les nœuds périodiquement prennent rendez-vous sur un canal commun dans la phase de contrôle. Le principe de fonctionnement de cette approche est illustré par la figure 5.

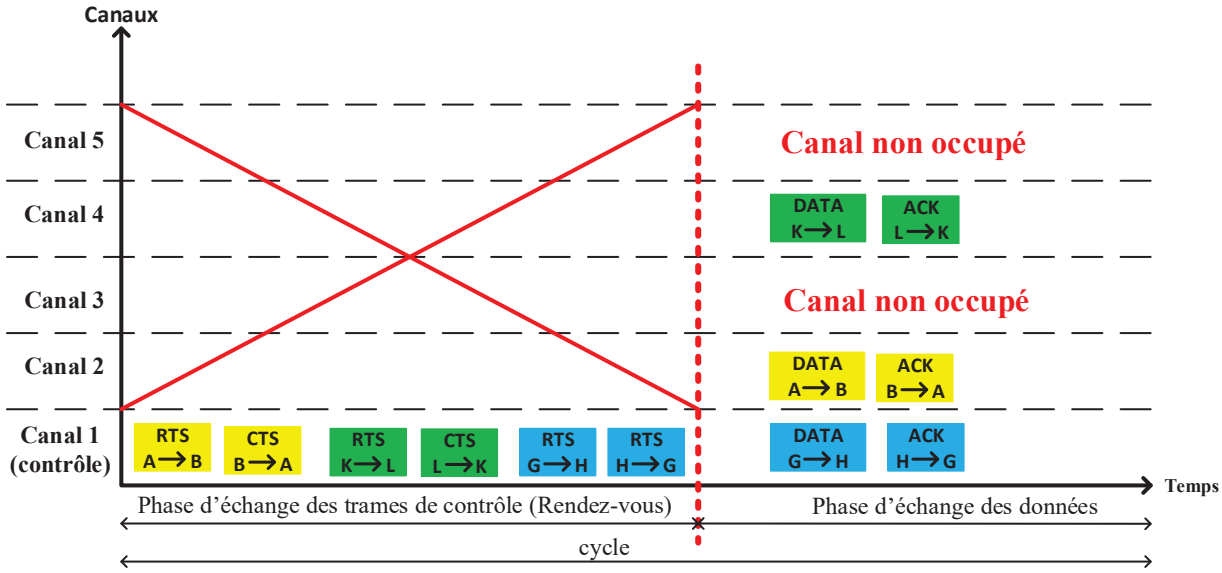

Figure 5: principe de fonctionnement de *split phase*

Au début de chaque cycle qui commence par une phase de contrôle, tous les nœuds commutent sur un canal commun, le canal de contrôle ou canal de rendez-vous. On remarque ici que les nœuds F/G et

K/L tentent de conclure des rendez-vous en échangeant des trames de contrôle RTC/CTS pendant la phase de contrôle sur le canal CH1 (le canal de contrôle par défaut). Les nœuds F et K incluent une liste des canaux préférés ou PCL (*Preferable Channel List*) quand ils envoient les trames de contrôle RTS. Les nœuds G et L sélectionnent chacun un canal de la liste de leur émetteur en renvoyant un CTS. D'après la figure 5 par exemple, à partir des trames RTS et CTS, le voisin de G, soit le nœud K, sait que CH1 sera occupé pendant la phase d'échange de données suivante. Par conséquent, lorsque le nœud K envoie un RTS au nœud L, il n'inclut pas le canal CH1 dans sa liste des canaux préférés, mais plutôt, il sélectionne un autre canal disponible, comme on le voit sur la figure, le canal CH4. Dans le cas où l'émetteur et le récepteur ne trouvent pas un canal commun, la négociation d'un canal sera alors reportée au prochain cycle [12]. Lorsqu'un nœud est inactif pendant la phase de contrôle, il restera inactif pendant la seconde phase de données [12].

L'avantage de cette approche, puisque les nœuds échangent des listes des canaux, est de permettre d'atténuer le problème multi-canal du terminal caché et de la surdité. Par comparaison au protocole du canal de contrôle dédié, ce protocole exploite tous les canaux y compris le canal de contrôle pendant la phase de données. Mais son principal inconvénient est qu'une synchronisation entre les nœuds est nécessaire. De plus, le protocole n'exploite pas tous les canaux disponibles pendant la phase de contrôle, un seul canal de contrôle est utilisé pendant cette phase, donc en cas de forte charge de trafic, il devient un goulot d'étranglement. On remarque aussi qu'au cours de cette phase, une bande passante importante est gaspillée. Dans [12], le pourcentage de la bande passante gaspillé est calculé comme suit : soient L_{cycle} : la longueur du cycle ; L_{cp} : la longueur de la phase de contrôle ; N : le nombre de canaux disponibles ; P_{rc} : le pourcentage de la bande passante gaspillée pendant chaque cycle, est alors (équation 1) :

$$Prc = \frac{N-1}{N} * \frac{Lcp}{Lcycle} * 100 \qquad (1)$$

Il est également complexe d'estimer la longueur appropriée de la phase de contrôle, par contre celle de la phase de donnée dépend du nombre de négociations établies dans la phase précédente. Une petite longueur est la source de goulot d'étranglement, bien évidemment une longueur plus large est un gaspillage de la bande passante [12]. Ainsi, la longueur de la phase de contrôle reste principalement le paramètre le plus délicat de cette approche. Le split phase est bien adapté au cas mono-saut, il faudrait le modifier pour le cas multi-saut afin de faire connaître aux voisins à plus de 1 saut les listes PCL.

Le protocole TMMAC (*TDMA Multi-channel MAC*) [72] mono-interface, se base sur le *split phase* avec certains principes de fonctionnement un peu différent de ce dernier. Pour TMMAC, la phase de contrôle est ajustée de façon dynamique en fonction des différents types de trafic afin d'atteindre un débit plus élevé et une faible consommation d'énergie par rapport aux autres protocoles de même type cités dans les littératures. La phase de données dans TMMAC est basée sur le protocole *TDMA* (*Time Division Multiple Access*), ainsi lors de la phase de contrôle les nœuds négocient les canaux et les slots de temps à la fois. Les performances de ce protocole sont évaluées par simulations et à travers des modèles analytiques, qui prouvent que le protocole TMMAC atteint un débit de communication et faible consommation d'énergie plus meilleurs que les autres protocoles de même types (*split phase*) étudiés précédemment.

1.1.2.3.1.3 Le protocole saut commun

C'est un protocole basé sur le rendez-vous unique [1] [3] [4] [12] [35] [36] [37] les nœuds sont équipés d'une seule interface et le temps est divisé en intervalles de temps ou slots. Chaque slot est égal au moins à la durée nécessaire pour échanger une trame de contrôle. Tous les nœuds suivent une séquence commune de saut à travers tous les canaux et de manière synchrone. Le but principal de cette approche est d'exploiter tous les canaux des données. Ainsi, les nœuds qui veulent échanger des données arrêtent de sauter de canal en canal et restent sur le même pour transmettre après l'échange des trames de contrôles RTS/CTS ; tandis que les autres nœuds continuent de suivre la séquence de saut. Après avoir fini leur transmission, les nœuds se resynchronisent avec les autres et continuent de suivre la séquence de saut commune.

On voit ici sur la figure 6, les nœuds A, B, C, D…, à l'instant t1 commutent sur le canal1, mais à t2, on remarque que A et B après avoir échanger des trames des contrôles avec succès, restent sur le canal2 pour échanger des données et acquittement *ACK*. Tandis que les autres nœuds inactifs continuent à suivre la séquence des sauts commune. D'après la figure, à l'instant t3, comme les nœuds A et B sont en activité sur le canal2, c'est pourquoi ils sont absents sur le canal3. À l'instant t4, F et G échangent des données sur le canal1, ils sont aussi absents à l'instant t5 quand les nœuds commutent sur le canal2. à t6 on remarque que les nœuds A et B ont fini leur transmission et resynchronisent avec les autres nœuds en suivant le saut commun.

Par comparaison avec les approches précédentes, le protocole de saut commun permet d'exploiter tous les canaux des données d'où son avantage mais, l'inconvénient majeur est qu'il nécessite un strict mécanisme de synchronisation et souffre également d'une grande fréquence de commutation entre les canaux et des problèmes multi-canal du nœud caché et de la surdité (dans un contexte multi-saut).

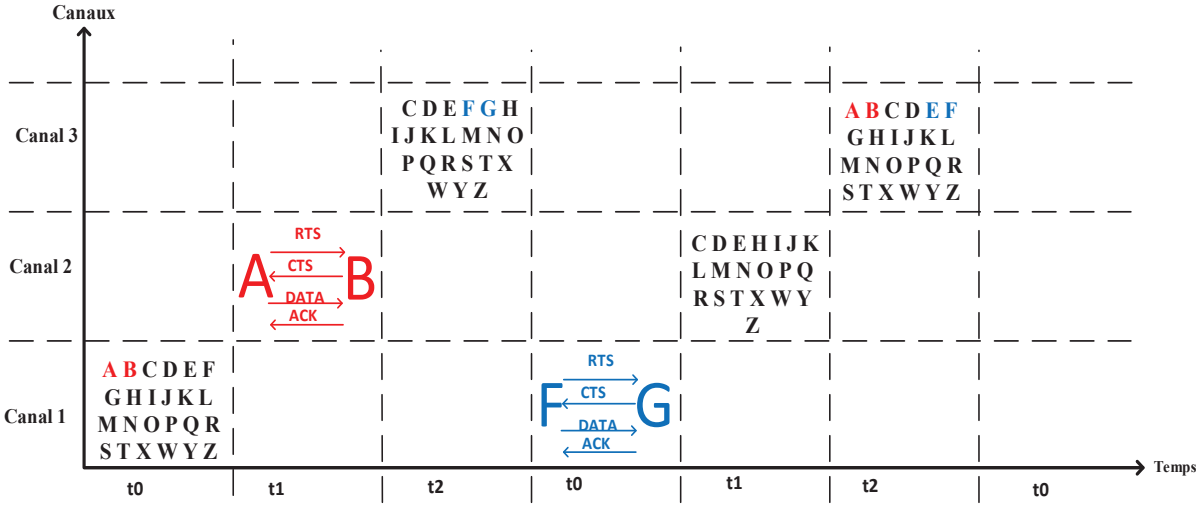

Figure 6: principe de fonctionnement du protocole du saut commun

1.1.2.3.1.4 Le protocole saut indépendant

Contrairement au saut commun, le protocole de ***saut indépendant*** permet plusieurs rendez-vous simultanément sur les différents canaux. Les nœuds sont équipés d'une seule interface et commutent sur les canaux en fonction de leurs propres séquences. Le temps est composé en séquences de cycle et chaque cycle est divisé en plusieurs slots de temps. Les nœuds itèrent alors sur leurs propres séquences

de saut et se chevauchent au moins pendant un slot de temps par cycle, ce qui leur permet d'échanger et d'apprendre leurs séquences les uns des autres. Dans [5], pour éviter la partition du réseau, on exige que les nœuds sautent sur un canal prédéterminé après avoir itéré par tous les canaux de leurs propres séquences. Tel n'est pas le cas du protocole proposé dans [6] pour lequel les nœuds se chevauchent au cours de leurs séquences de sauts où chaque nœud annonce sa séquence de saut.

Pour le protocole *SSCH* (*Slotted Seeded Channel Hopping*) [1] [3] [4] [5], lorsqu'un nœud veut émettre, il attend jusqu'à ce que sa séquence corresponde à celle de son récepteur, le transfert sera effectué alors sur des sauts successifs à la séquence du récepteur. Comme on le voit sur la figure 7, les nœuds F et G suivent chacun leur propre séquence, indiquée sur le cercle en pointillé noir. Sur le cercle en pointillé vert, on voit que les deux nœuds F et G sautent sur le canal CH3. À l'instant t6, le nœud G commence à suivre les séquences de F pour lui transmettre des données. Le transfert de données sera alors effectué sur des sauts successifs de la séquence du récepteur F. Pour parvenir à une topologie multi-saut il faut faire en sorte qu'une fois que tous les nœuds en se commutant sur le canal prédéterminé (par exemple le canal CH3 sur la Figure 7), trouvent le moyen de diffuser leurs séquences au-delà d'un saut, surtout s'il y a des couples de nœuds qui souhaitent échanger des données.

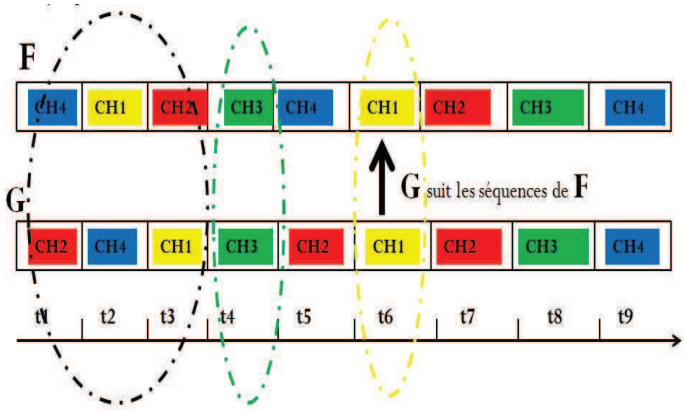

Figure 7: principe de fonctionnement du protocole de saut indépendant (exemple du SSCH)

Le protocole *McMAC (Parallel Rendez-vous Multi-Channel MAC Protocol)* [1] [3] [4] [6], apporte quelques corrections sur le principe de fonctionnement de *SSCH* afin d'éliminer le délai d'attente causé par le nœud émetteur. Pour *McMAC*, comme le nœud connaît déjà les séquences du récepteur, il saute sur le canal de la séquence de sauts du récepteur, et le transfert de données est entièrement réalisé sur ce canal.

1.1.2.3.1.4.1 Principe de fonctionnement du protocole de saut indépendant

Dans *SSCH* chaque nœud adopte une ou plusieurs paires *(canal, germe)* d'initialisation (4 au maximum), chaque nœud saute entre les différents canaux disponibles en utilisant ses propres séquences de saut de canal. Les séquences de sauts de canal sont conçues de telle sorte que les nœuds se chevauchent les uns avec les autres au moins une fois dans un cycle. Plus précisément, l'ordonnancement de saut de chaque nœud peut être déterminé par un ensemble de paires *(canal, germe)* que nous pouvons noter autrement (Ch_i, S_i). S'il y a n canaux disponibles dans le système, le canal est un nombre entier appartenant à l'intervalle $Ch_i \in [0, n-1]$ et les germes est un nombre entier tirer aléatoirement dans l'intervalle $S_i \in [1, n-1]$, ainsi Ch_i constituera un cycle périodique entre $[0, n-1]$. Le nœud ainsi incrémente chacun des canaux dans sa liste d'ordonnancement en utilisant le germe et le processus se répète continuellement suivant l'équation : $\boldsymbol{Ch_i \leftarrow (Ch_i + S_i)\ modulo\ n}$, qui se traduit de la manière suivante :

Nouveau_Canal= (Ancien_Canal + germe) modulo Nombre_de_Canaux

Les nœuds apprennent les ordonnancements des uns et des autres en diffusant périodiquement leurs paires *(canal, germe)*, ainsi chaque nœud diffuse sa paire *(canal, germe)* une fois par slot. En considérant la couple *(canal, germe)* pour le choix d'un prochain canal, quatre cas possibles peuvent se présenter entre deux nœuds :

1) ayant les mêmes canaux et mêmes germes
2) ayant à la fois différent canaux et différentes germes
3) ayant les mêmes canaux mais différent germes
4) ayant les mêmes germes mais différents canaux

Les nœuds ne se chevauchent jamais pendant un cycle que dans le quatrième cas lequel aboutit au problème de partition logique, mais ils le seront au moins une fois dans les autres cas. Ainsi le quatrième cas ne sera résolu que lorsqu'un slot de parité est ajouté à la fin du cycle *($Canal_{parité}$ = germe)*.

Un slot de parité à la fin d'un cycle est également introduit pour éviter la situation que deux nœuds ne commutent jamais sur le même canal en même temps (problème de partition logique). Si *n* canaux et *k* paires *(canal, germe)* sont utilisés, un cycle peut contenir alors *k*n+1* slots.

Sur la figure 8, on peut voir, comment un nœud quelconque du réseau en utilisant la relation : ***Nouveau_Canal= (Ancien_Canal + germe) modulo Nombre_de_Canaux*** et lorsque trois canaux sont disponibles peut choisir son prochain canal.

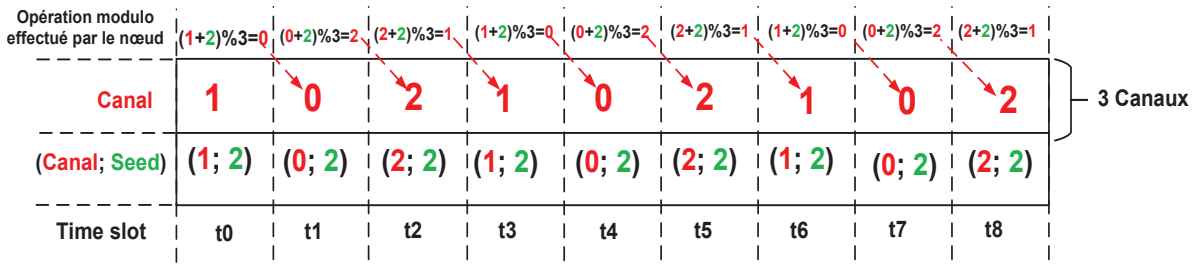

Figure 8: Opération effectué par le nœud pour accéder au prochain canal

La figure 9 représente le cas où chacun de deux nœuds A et B suit son propre ordonnancement et se chevauchent au moins une fois par cycle (c.-à-d. se trouvent sur le même canal), qui est le canal 2 dans notre exemple représenté par le cercle pointillé.

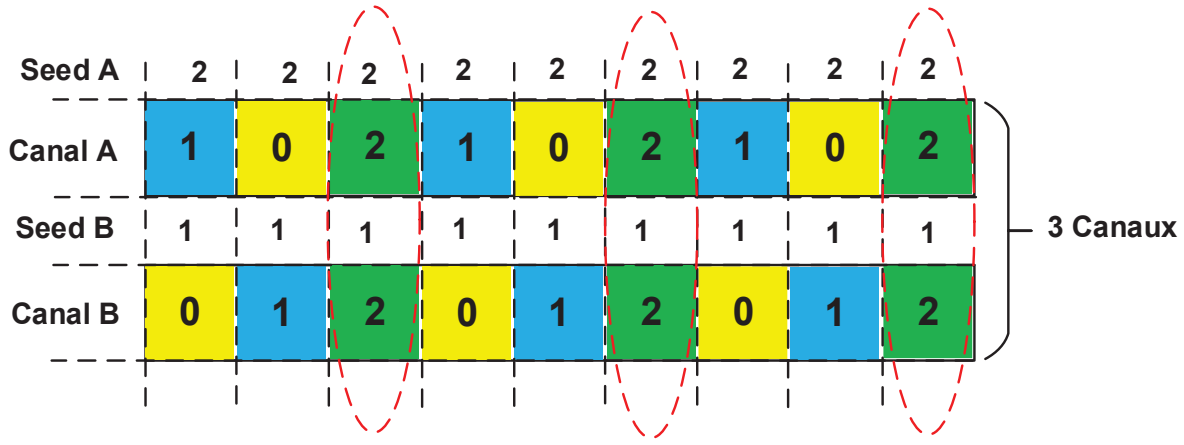

Figure 9: Opération effectué par les nœuds A et B ayant des germes (*Seed*) différents pour accéder au même canal

Comme les nœuds se chevauchent au moins une fois par cycle et échangent leurs ordonnancement *(canal germe)*, un nœud qui a des données à émettre, peut changer son ordonnancement pour correspondre à celui de son récepteur destiné qu'on peut voir sur la Figure 10 en cercle pointillé dans lequel le nœud B change son ordonnancement *(canal, germe)* et suit celui du nœud A, mais ceci peut aboutir au cas de la réception manqué, puisque ce dernier (par exemple le nœud A) peut aussi changer son ordonnancement pour adopter celui d'un autre nœud.

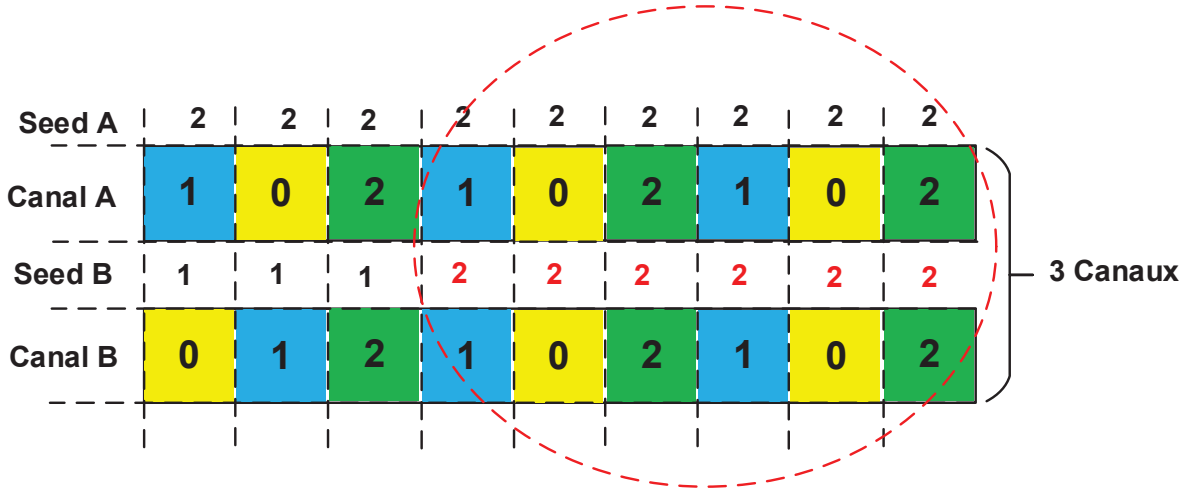

Figure 10: Cas d'un nœud qui suit l'ordonnancement d'un autre

On peut voir sur la figure 11 (a), le cas où deux nœuds ayant les mêmes germes mais différents canaux et par conséquent ne se retrouvent jamais sur un même canal afin d'échanger leurs ordonnancement *(canal germe)* et ensuite des données, celui-ci ne sera résolu que par l'ajout d'un slot de parité à la fin de chaque cycle, l'exemple de Figure 11 (b) représente ce dernier cas sur laquelle on voit en cercle pointillé un slot de parité ajouté à la fin de chaque cycle.

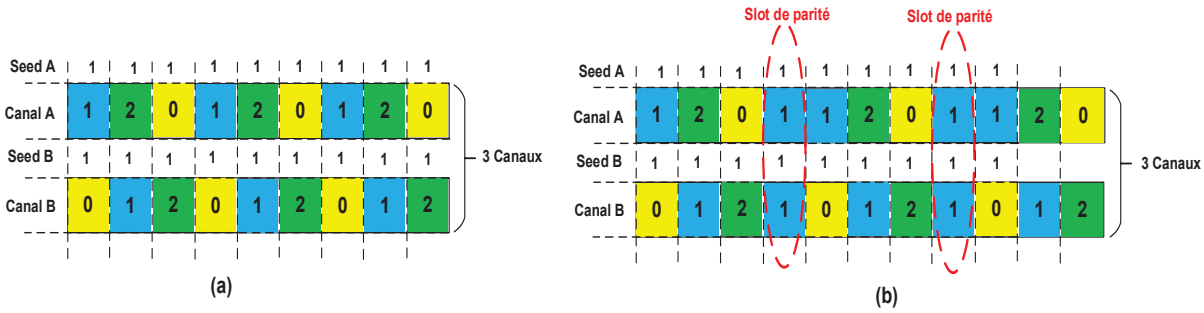

Figure 11: Cas des nœuds qui ne se chevauchent pas

Le fonctionnement de McMAC est très simple par rapport au SSCH puisqu'un émetteur adopte un modèle de sauts de manière aléatoire et sans besoin de le modifier, Le générateur de l'ordonnancement est moins complexe : il est différent en termes de paramètres d'initialisation afin de ne garder qu'une seule séquence de saut avec *n* slots et dans laquelle tous les nœuds peuvent se retrouver au moins une fois par cycle. Chaque nœud saute à travers tous les *k* canaux disponibles d'une manière pseudo-aléatoire en utilisant sa propre adresse MAC comme germe. Les séquences de saut des récepteurs de chaque nœud destiné peuvent facilement être obtenues si leurs adresses MAC sont connues.

En outre, un nœud peut dévier de sa séquence de sauts constante et sauter sur le canal du récepteur pour émettre, l'échange et le transfert de données se passent sur le canal sur lequel se trouve le récepteur.

Ainsi, pour réduire les collisions, McMAC effectue une détection de porteuse et un *backoff* (fenêtre de contention). Lorsqu'un nœud visite un nouveau canal, il détecte d'abord la porteuse pour vérifier s'il y a une transmission en cours.

Lorsqu'un nœud a une trame de données à envoyer à un nœud récepteur, il commute au début du slot sur le canal où se trouve le récepteur et écoute si le canal est libre, il attend une durée aléatoire de *backoff CW* (*entre 1 et* CW_{max}). Si aucune trame n'est détectée pendant cette période aléatoire de *CW*, l'émetteur commence la transmission de ses trames de données vers le récepteur après échange des trames de contrôles *RTS/CTS*, les deux nœuds restent sur ce canal pendant toute la durée du transfert de données *(DATA/ACK)*. Si le début d'une autre transmission de trame est détecté, l'émetteur arrête son *backoff* et retourne à sa séquence initiale. Le transfert des données des données peut durer pendant plusieurs slots de temps, à la fin de la transmission, l'émetteur et le récepteur retournent à leurs séquences initiales.

La solution apportée par le protocole McMAC surgit un autre inconvénient, puisque le récepteur destiné pourra aussi dévier de sa séquence et commuter sur le canal d'un autre nœud pour transmettre des trames de données. Ceci se traduit par le phénomène de la réception manqué et aussi une occupation inutile du canal par le nœud émetteur pour empêcher ses voisins d'utiliser ce canal.

Les protocoles de rendez-vous parallèles ont l'avantage d'éliminer le problème potentiel de goulot d'étranglement des approches précédentes avec une seule interface radio en permettant plusieurs rendez-vous sur différents canaux disponibles. Mais le principal inconvénient de ces protocoles est le délai de commutation pour le saut du canal. De plus, chaque nœud nécessite des mécanismes de synchronisation pour suivre la séquence de sauts des autres, il y a une fréquence importante des diffusions d'ordonnancement *(canal, germe)* dont la gestion sera difficile dans un mode ad-hoc avec une grande taille du réseau. La plupart des protocoles étudiés et présentés ne prennent pas en compte

les aspects multi-saut et fonctionnement correctement pour la plupart uniquement dans une topologie très théorique où tout nœud est à portée de tout autre nœud.

1.1.2.3.1.5 La method d'accès multi-canal 802.15.4e TSCH (*Time Slotted Channel Hopping*)

1.1.2.3.1.5.1 Introduction TSCH

La méthode d'accès IEEE 802.15.4e a été publiée en 2012 comme un amendement au protocole de contrôle d'accès au médium défini par la norme IEEE 802.15.4 (2011) et qui a pour but d'améliorer les performances de la norme IEEE 802.15.4 en termes de latence et de fiabilité en exploitant plusieurs canaux simultanément. Elle adopte deux méthodes d'accès principales : l'accès multiple à répartition dans le temps (TDMA) et l'accès multi-canal [5]. Le concept initial de ce protocole, provient du protocole TSMP (Time synchronized mesh protocol) qui a été étudié en 2008 [9]. L'un des objectifs principaux de la méthode d'accès TSCH est d'étendre les applications de la norme IEEE 802.15.4 au-delà du domaine industriel, par exemple les maisons intelligentes, la localisation…, mais aussi améliorer la conservation de l'énergie des nœuds dans les réseaux. A cet effet, il est classiquement prévu d'utiliser des calendriers d'ordonnancement (*schedule*), des *slotframe*s ajustables, … L'une des principales caractéristiques de TSCH est la communication multi-canal, basée sur le saut de canal.

IEEE 802.15.4e TSCH ne modifie pas la couche physique initialement prévue, il peut donc fonctionner sur tout matériel qui est conforme à la norme IEEE 802.15.4. Il se focalise seulement sur la couche MAC. Comme dans la norme IEEE 802.15.4 initiale, un PAN est formé par un coordinateur de PAN en charge de la gestion de l'ensemble du réseau, et, éventuellement, un ou plusieurs coordinateurs qui sont responsables d'un sous-ensemble de nœuds. La méthode d'accès TSCH peut être utilisée sur des topologies de réseau en étoile, arbre, maillée partielle ou totale [42].

Plusieurs travaux essaient d'améliorer la performance du TSCH dans différents contextes, dans [43], les auteurs à travers des méthodes analytiques et par simulations tentent d'analyser l'influence de nombre des canaux sur le délai pris par un nouveau nœud afin de rejoindre le réseau. Puis que le nouveau nœud qui cherche à rejoindre le réseau doit garder la radio toujours allumée pour recevoir le beacon diffusé par le PAN. Un délai long est source de gaspillage de l'énergie.

1.1.2.3.1.5.2 Principe de fonctionnement de la méthode d'accès multi-canal 802.15.4e TSCH

1.1.2.3.1.5.2.1 La structure de la slotframe

Dans un PAN TSCH, le concept de supertrame est remplacé par une trame de slots (*slotframe*). C'est un ensemble des slots temporels qui se répètent continuellement au cours du temps [42]. La principale différence entre le *slotframe* et la *supertrame* est que les nœuds du réseau sont supposées partager une notion commune du temps, ainsi le *slotframe* se répète automatiquement sans la nécessité des trames *beacons* afin d'initier des communications entre les différents nœuds du réseau. Le nombre de slots de temps dans une *slotframe* donnée (influençant ainsi la taille du *slotframe*) détermine combien de fois chaque slot de temps se répète, en établissant ainsi un ordonnancement (*schedule*) de communication [42]. La durée d'un slot de temps n'est pas définie dans la norme. Un slot de temps est suffisamment grand [42], [45] pour permettre l'émission d'une trame de taille maximale de 127 octets et le renvoi de son acquittement justifiant la bonne réception de cette trame. Contrairement à la supertrame classique

de 802.15.4, les *slotframes,* et les slots de temps affectés à un nœud dans le *slotframe* peuvent être initialement communiqués par *beacon*, mais sont généralement configurés par une couche supérieure du nœud qui rejoint le réseau [42] [48]. TSCH n'impose pas une taille de *slotframe*. Elle est variable en fonction des besoins de l'application. Plus une *slotframe* est courte, plus souvent un slot de temps se répète, entraînant ainsi plus de bande passante disponible pour le nœud qui s'en sert, mais aussi une consommation d'énergie plus importante. Pour conserver l'énergie, des longues *slotframes* sont préférables [46].

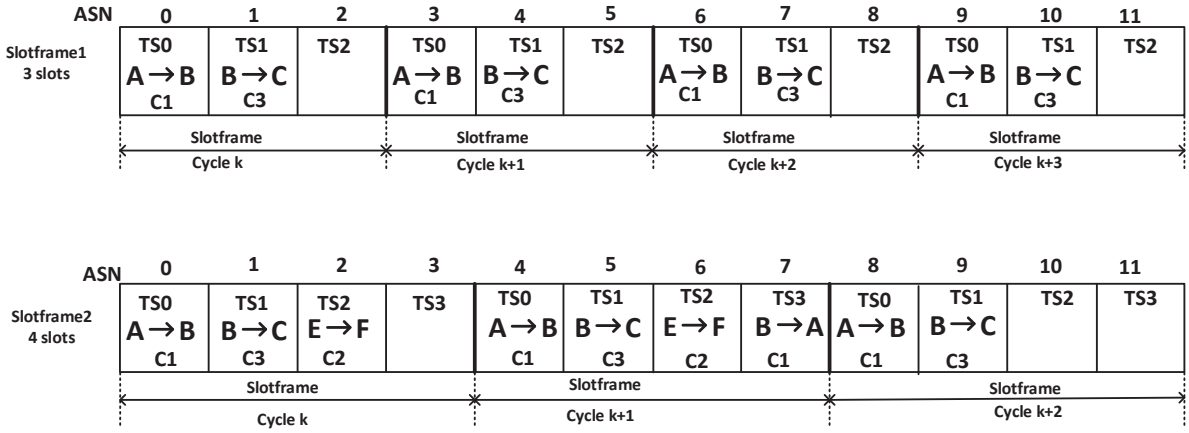

Figure 12: Structure d'une *slotframe* TSCH

On peut voir sur la figure 12 et l'exemple du haut où sont utilisés des *slotframes* de 3 slots, que le nœud A émet vers B sur le canal C1 pendant le slot de temps TS0. Le nœud B émet vers C sur le canal C3 pendant le slot de temps TS1. Par contre, le slot de temps TS2 n'a pas été utilisé.

Un réseau utilisant l'accès par slot de temps peut contenir plusieurs *slotframes* simultanées de différentes tailles pour divers groupes de nœuds, afin de faire fonctionner l'ensemble du réseau à différents cycles de service. Les *slotframes* peuvent être ajoutées, supprimées et modifiées alors que le réseau est opérationnel [42]. On peut aussi voir sur la figure 1 vers le bas que la deuxième *slotframe2* contient quatre slots de temps à l'inverse de la *slotframe1*.

1.1.2.3.1.5.2.2 Le numéro de slots absolu (absolute slot number ASN)

Le nombre total d'intervalles de temps écoulés depuis le début de vie du réseau ou un instant de départ arbitraire déterminé par le coordinateur de PAN est appelé le numéro de slot absolu (*Absolute Slot NumberASN*) [42]. C'est TSCH qui définit ce compteur d'intervalles de temps *ASN.* Quand un nouveau réseau est créé, son ASN est initialisé à 0. A partir de là, il s'incrémente de 1 à chaque slot de temps et il est partagé par tous les nœuds dans le réseau. Voici de façon explicite comment cela évolue :

ASN = (k*S+t) où ***k*** est le cycle de *slotframe* (c.-à-d. le nombre de répétitions de *slotframes* depuis que le réseau a été déployé), ***S*** est la taille de la *slotframe*, et ***t*** le *slotOffset*. Un nouveau nœud apprend l'ASN courant quand il rejoint le réseau. Comme les nœuds sont synchronisés (cf plus tard 2.4), ils connaissent tous la valeur actuelle de l'ASN, à tout moment. L'ASN est utilisé pour calculer le canal sur lequel communiquer.

1.1.2.3.1.5.2.3 L'ordonnancement *TSCH*

L'idée primordiale de TSCH est l'ordonnancement des liens de transmissions des données aux nœuds dans le réseau. Chaque nœud accède au médium suivant un ordonnancement (calendrier) de communication, ainsi les nœuds voisins dont les transmissions peuvent interférer, par conséquent ne seront pas ordonnancés à émettre sur le même slot de temps et même décalage de canal (*ChannelOffset*). L'ordonnancement est une matrice dont chaque cellule est indexée par un décalage de slots (*slot offset*) et un décalage de canal (*ChannelOffset*). Chaque cellule peut être attribuée à une liaison définie. Une cellule programmée peut être soit dédiée à un seul lien, soit partagé entre plusieurs liens. La norme MAC IEEE 802.15.4e définit un système simple de *backoff* pour les cellules partagées en cas de collision [42] [47]. Les nœuds dans un PAN TSCH utilisent également le système CSMA/CA slottée pour les slots partagés [42]. Cependant, la norme IEEE 802.15.4e TSCH ne définit pas le mécanisme pour savoir comment l'ordonnancement (*schedule)* est construit, mis à jour, maintenu, et adapté à l'exigence de trafic du réseau, mais explique seulement comment la couche MAC s'exécute [45] [49] [52].

Le *ChannelOffset* est un nombre utilisé dans le calcul du canal dans un système TSCH (*slottedchannelhopping*) pour permettre à différents canaux d'être utilisés dans le même slot.

L'ordonnancement *TSCH* indique à chaque nœud quoi faire dans chaque slot de temps : transmettre vers quel nœud, recevoir de quel nœud, ou se mettre en mode sommeil. L'ordonnancement indique, pour chaque cellule programmée (en émission ou réception), un *ChannelOffset* et l'adresse du voisin avec lequel communiquer.

Une fois qu'un nœud obtient (fournis par le PAN coordinateur) son ordonnancement, il l'exécute :

- ✓ Pour chaque cellule d'émission, le nœud vérifie s'il y a une trame dans la mémoire tampon de sortie qui correspond au voisin écrit dans les informations du *schedule* pour ce slot de temps. S'il n'y en a pas, le nœud éteint sa radio pendant la durée du slot de temps. S'il y a une trame, il l'envoie et reste en écoute pour l'acquittement de la trame émise.
- ✓ Pour chaque cellule de réception, le nœud écoute pour les trames entrantes possibles. Si aucune n'est reçue après une certaine période d'écoute, il éteint sa radio. Si une trame est reçue, adressée au nœud, le nœud peut renvoyer un accusé de réception.

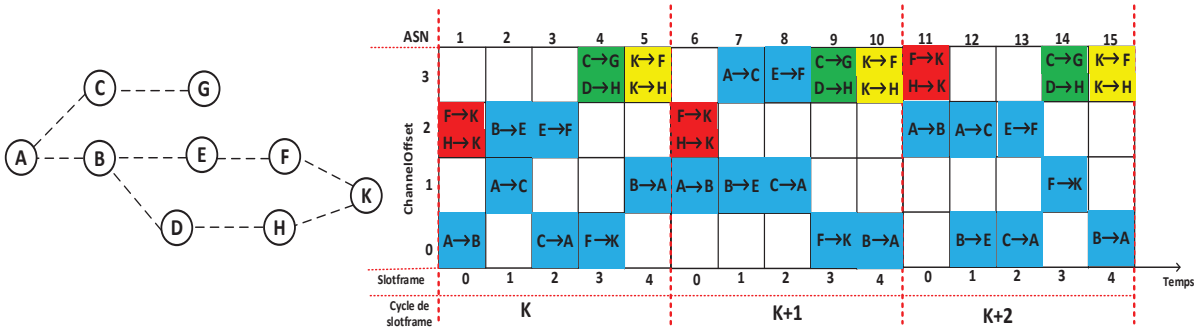

Figure 13: Structure d'un *schedule* TSCH avec des slots dédiés et partagés

Si une transmission échoue ou l'ACK n'est pas reçu dans un délai prédéfini, la trame de données est reportée au prochain slot de temps affectée à la même paire (source, destination) de nœuds. Le nœud peut réémettre sa trame de données et attend l'ACK jusqu'à un certain nombre de fois [48].

On voit sur la figure 13, la structure d'un ordonnancement (*schedule)* TSCH, tous les slots de temps en bleus sont dédiés tandis que les autres sont partagés.

Dans ce *schedule*, si le nœud B est programmé de transmettre au nœud E au *slotOffset1* et *channelOffset 2*, le nœud E sera programmé pour recevoir à partir du nœud B au même *slotOffset*et *channelOffset*. On remarque aussi que, la taille de cette *slotframe* (Figure 13) est de 5 slots de temps et 4 *channelOffset*. Chaque nœud dans le réseau ne s'intéresse qu'aux cellules auxquelles il participe, et il restera éteint pour les autres cellules.

Le *channelOffset* est traduit par les deux nœuds en fréquence en utilisant la fonction suivante [42] [43] [47] :

$$frequency = F\{(ASN + channelOffset)\ \%\ Nb\_ch\},$$

Où *%* est l'opération modulo ; les valeurs de *t* et *channelOffset* sont prises respectivement dans :

$0 \leq t \leq S - 1$, et $0 \leq channelOffset \leq Nb\_ch - 1$.

La fonction *F* contient l'ensemble des canaux disponibles (liste de séquence de saut globale).

La valeur *Nb_ch* est le nombre de fréquences physiques disponibles. Dans un réseau IEEE 802.15.4e, 16 canaux sont disponibles et chaque canal est identifié par un *channeloffset* c'est-à-dire une valeur entière comprise dans l'intervalle [0, 15]. En outre, une liste noire peut être utilisée pour restreindre l'ensemble des canaux autorisés pour des raisons de coexistence ou faible qualité de communication [43] [47].

1.1.2.3.1.5.2.4 Le mode de synchronisation en utilisant l'accès TSCH

Pour assurer la communication multi-canal slottée, dans un réseau TSCH, les nœuds doivent maintenir une synchronisation fine. Tous les nœuds sont supposés être équipés d'une horloge locale qui permet de maintenir une trace du temps. Cependant, puisque les horloges dans les différents nœuds dérivent les unes par rapport aux autres, les nœuds voisins doivent se resynchroniser périodiquement entre eux. Chaque nœud doit synchroniser périodiquement son horloge réseau à au moins un de ses voisins, et il doit fournir également son horloge réseau à ses voisins. Il appartient donc à l'entité (le coordinateur de PAN) qui gère le calendrier de connaître et contrôler une base de temps commune à chaque nœud.

A l'inverse d'autres protocoles de synchronisations, TSCH n'utilise pas de trames balises ou *beacons* pour la synchronisation d'horloge. Lorsqu'un nœud a reçu une trame de données, il renvoie une trame ACK avec des informations de synchronisation pour corriger la dérive de l'horloge. Cela signifie que les nœuds voisins peuvent se resynchroniser les uns aux autres quand ils échangent des trames de données.

Deux types de méthodes sont définies dans IEEE 802.15.4e (2012) pour permettre à un nœud de se synchroniser dans un réseau TSCH : (i) la synchronisation basée sur l'acquittement ; (ii) et la synchronisation basée sur les trames de données. Dans les deux cas, le récepteur de la trame émise calcule la différence de temps entre l'instant d'arrivée prévue de la trame et son arrivée effective [42] [49].

Pour la synchronisation basée sur l'acquittement, le récepteur fournit des informations de synchronisation au nœud émetteur dans son accusé de réception. Dans ce cas, c'est le nœud émetteur de la trame initiale de data qui se synchronise à l'horloge du récepteur de cette même data.

Pour la synchronisation basée sur des trames de data, le récepteur utilise le delta calculé (par le récepteur) pour ajuster sa propre horloge. Dans ce cas, c'est le nœud récepteur qui se synchronise sur l'horloge de l'émetteur.

Quand il y a du trafic dans le réseau, les nœuds qui communiquent implicitement se resynchronisent en utilisant les trames de données qu'ils échangent. En l'absence de tout trafic de données, s'ils n'ont pas communiqué pendant un certain temps, les nœuds peuvent échanger des trames de données fictives (vides) et ainsi utiliser la trame d'acquittement pour se resynchroniser. Les nœuds doivent se synchroniser périodiquement sur la source de temps de leurs voisins afin d'éviter la dérive de leur horloge.

En bilan, différentes politiques de synchronisation sont possibles dans TSCH. Les nœuds peuvent aussi garder la synchronisation exclusivement en échangeant des *beacons* améliorés (*Enhanced Beacons EB*) [42] [49]. Un réseau PAN TSCH est formé lorsqu'un nœud, généralement le coordinateur du PAN, annonce la présence du réseau en envoyant ces EB. Les nœuds déjà dans le réseau envoient périodiquement des EBs pour annoncer la présence du réseau. Lorsqu'un nouveau nœud rejoint le réseau, il écoute les EBs afin de synchroniser au réseau TSCH. On trouve dans les EBs les informations sur les slots de temps qui permettent aux nouveaux nœuds de se synchroniser au réseau. Pour rester synchronisés, les nœuds doivent avoir la même notion du début et de la fin de chaque slot de temps, les informations sur les sauts de canal et la taille du *slotframe* [42] [49].

Les nœuds peuvent également rester synchronisée en émettant périodiquement des trames valides à une source de temps voisine et utiliser l'acquittement pour se resynchroniser.

Cependant la norme 802.15.4e TSCH définit seulement les mécanismes pour qu'un nœud se synchronise à la source du temps de son voisin, malheureusement elle ne fournit des informations, pour une synchronisation complète multi-saut du réseau, elle précise simplement que cette fonctionnalité est à la charge de la couche supérieure [12].

Finalement, plusieurs méthodes de synchronisation ont été imaginées et proposées dans le standard, la bonne méthode à utiliser dépend des exigences du réseau.

Notons aussi que certains travaux ont abordés l'ordonnancement et la synchronisation TSCH dans un mode distribué [45], et plus encore dans un contexte multi-canal multi-saut [53].

1.1.2.3.1.5.3 Conclusion TSCH

Nous venons d'étudier la nouvelle méthode d'accès multi-canal 802.15.4e TSCH, qui est une première approche déterministe puisqu'elle combine l'accès multi-canal et le TDMA (*Time Division Multiple Access*) en affectant à chaque lien (émetteur/récepteur) un slot de temps sur un canal déterminé parmi les canaux disponibles. On remarque que la méthode d'accès multi-canal 802.15.4e TSCH favorise la gestion d'énergie des dispositifs dans le réseau puisque la synchronisation entre les nœuds dans un réseau s'effectue sans l'utilisation permanente de *beacons* comme dans les normes précédentes, cette synchronisation est réalisée par les trames échangées (données et acquittement) entre les nœuds du réseau. Les nœuds non concernés par des slots pouvant dormir alors.

L'absence de trames de contrôle dans la méthode d'accès multi-canal 802.15.4e TSCH peut être une solution et favoriser les transmissions multi-sauts, qui exigent la propagation de ces trames de contrôle à plusieurs sauts, tâche qui n'est pas facile à réaliser dans certains réseaux de grandes tailles. Ces

trames de contrôle, en particulier pour annoncer l'ordonnancement à tous les nœuds, représentent un trafic important. La gestion de cet ordonnancement est laissée libre à l'utilisateur de la norme TSCH.

Les nœuds dans le réseau TSCH étant configurés avec un ordonnancement afin d'effectuer une opération d'émission ou de réception, les problèmes inhérents du terminal caché multi-canal et de surdité sont ainsi résolus.

Le grand désavantage de la méthode d'accès multi-canal 802.15.4e TSCH, vient de son principe de fonctionnement centralisé, c'est-à-dire la gestion du réseau via un coordinateur de PAN qui a la charge de construire le calendrier (*schedule*) pour tous ou une partie du réseau pour que ce dernier fonctionne. Le moindre défaut de celui-ci conduit à un disfonctionnement ou au blocage total du réseau.

Certains principes de fonctionnement de la méthode sont encore implicites, surtout la façon dont le coordinateur construit le calendrier afin d'affecter les slots de temps en fonction des besoins de chaque nœud. Il faut configurer et prévoir chaque paire de nœuds, l'un en émission et l'autre en réception, dans le même slot de temps et sur le même canal. Il est par exemple important d'éviter de programmer l'émetteur et le récepteur en même temps en émission. Il faut aussi que le coordinateur affecte les slots de temps uniquement aux nœuds qui ont des données à envoyer pour n'est pas gaspiller de la bande passante. Il reste aussi à savoir de quelle façon le coordinateur gère le contexte multi-saut, en d'autres termes, lorsque le coordinateur construit le calendrier, comment va-t-il tenir compte des trafics qui sera envoyé au-delà d'un seul saut ? Serait-il aussi simple dans une topologie dynamique et multi-saut avec un nombre important des nœuds de construire un ordonnancement optimal ? Ces questions restent pour nous un défi que la norme devra s'impliquer. On peut dire que la méthode d'accès multi-canal 802.15.4e TSCH a aussi apporté des idées et des pistes à exploiter pour améliorer les méthodes d'accès multi-canal multi-saut, mais il reste à trouver les moyens pour s'orienter vers une méthode d'accès multi-canal 802.15.4e TSCH distribuée et non centralisé. Le problème de la diffusion des slots à n sauts (qui sont en fait des RDV déguisés) reste entier.

1.1.2.3.2 L'accès multi-canal multi-interface

Puisque les protocoles MAC multi-canal mono-interface posent plus souvent des problèmes pour la gestion des canaux (multi-canal du nœud caché, surdité…), et aussi que les interfaces radios sont en mode semi-duplex, c'est-à-dire qu'en un instant donné un nœud du réseau peut soit émettre ou recevoir, si bien que plusieurs canaux sont disponible ; ce qui a conduit certains travaux [3] [27] [29] [30] [31] à exploiter plusieurs interfaces à la fois afin de mieux contrôler l'utilisation des différents canaux disponible dans le réseau et les exploiter efficacement, moyennant un coût matériel.

1.1.2.3.2.1 L'intérêt des multi-interfaces

Dans la plupart des réseaux multi-sauts, si un seul canal est disponible, et donc souvent une seule interface est aussi utilisée. Cependant, lorsque plusieurs canaux sont disponibles, le recours à plusieurs interfaces est nécessaire [54] [55] [56] [57].

Lorsqu'on utilise une seule interface, si les interfaces des deux nœuds sont commutées sur des canaux différents, alors ils ne peuvent pas communiquer entre eux. Par conséquent, si les trames traversent des trajets multi-saut, ces trames peuvent être retardées à chaque saut, à moins que le prochain saut soit également sur le même canal.

Ainsi, quand une seule interface est utilisée, il y a une augmentation de la latence de bout en bout si les différents sauts traversés sont effectués sur des canaux différents. Par contre si plusieurs sauts se trouvent sur le même canal, les transmissions sur des sauts successifs des nœuds voisins peuvent interférer réduisant ainsi la capacité maximale du réseau.

Un autre avantage du multi-interface est la possibilité de recevoir et de transmettre des données simultanément (en parallèle). Les interfaces sans fil semi-duplex *(half-duplex)* ne peuvent pas transmettre et recevoir des données simultanément. Cependant, lorsque plusieurs interfaces et plusieurs canaux sont disponibles, alors une interface reçoit des données sur un canal *C1* par exemple, et la seconde interface peut transmettre simultanément des données sur un autre canal *C2*. Cela peut donner la possibilité de doubler le débit maximal possible sur un trajet multi-saut.

Par exemple, considérons le trajet **B→A→D** sur la figure 14. Soit ***L*** le débit maximal d'émission possible sur un saut (i.e. **B→A**). Avec une seule interface radio, le nœud A passe à peu près la moitié du temps de réception du nœud **B**, et l'autre moitié à transmettre au nœud **D**. Par conséquent, si le débit source (nœud **B**) est ***L bps***, le débit moyenne de réception au nœud **D** est approximativement de ***L/2 bps***.

Avec deux interfaces radios sur le nœud A et deux canaux, l'interface radio 1 sera commuté sur le canal 1 et l'interface radio 2 sera commuté sur le canal 2, dans ce cas le débit de réception au nœud **D** sera théoriquement égale à ***L bps***.

Maintenant, considérons le cas lorsque le nœud **A** ne possède qu'une seule interface radio et il y a une transmission simultanée sur les liens **B→A→D** et **C→A→E**. Dans ce cas, le nœud **A** doit passer un quart de son temps à recevoir des nœuds **B** et **C** et ensuite transmettre aux nœuds **D** et **E**. Le débit moyen de réception au niveau des nœuds **D** et **E** dans ce cas est ***L/4 bps***.

Par contre, si nous considérons le cas où le nœud **A** est équipé de deux interfaces radios et il y a deux canaux non recouverts disponibles. Dans ce cas, les interfaces radios 1 et 2 peuvent être commutées sur les deux canaux *C1* et *C2*, respectivement. Si deux interfaces radios sont utilisées en mode semi-duplex *(half-duplex)* pour supporter les liens **B→A→D** et **C→A→E**, le débit moyen de réception pour chaque flux va doubler de ***L/2 bps***, le même que si chaque flux aurait été reçu s'ils ont été ordonnancés à des instants différents.

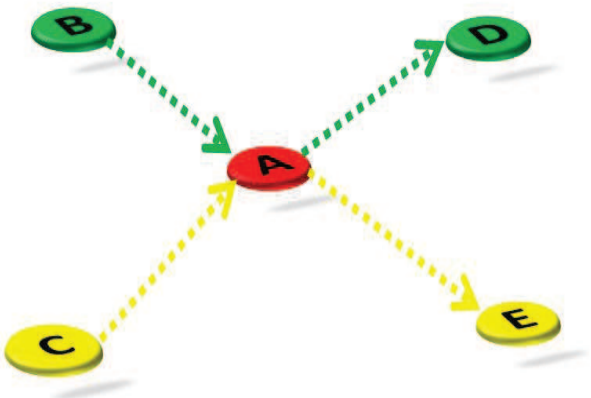

Figure 14: Avantage du multi-interface par rapport à la mono-interface

Généralement dans l'accès multi-canal multi-interface on peut rencontrer trois types d'approches selon les modes d'allocation 1) une approche basée sur le principe de rendez-vous unique : le canal dédié et 2) deux approches sont basées sur le principe de rendez-vous multiples : l'interface dédié à la réception (mixte) et l'autre fixe sans commutation [12] [27] [32].

1.1.2.3.2.2 Le protocole de canal dédié multi-interface

Le canal dédié multi-interface [12] [32], à la différence du canal dédié mono-interface, les nœuds ici sont équipés de deux interface conduisant à un mode de fonctionnement différent du contexte mono-interface. Ainsi donc une interface est dédiée aux trames de contrôle, tandis que l'autre interface servira aux transmissions des données. L'interface de contrôle est fixée de façon permanente sur le canal de contrôle dont le rôle est d'affecter un canal des données après négociation entre les deux nœuds. Par contre l'autre l'interface dédiée aux canaux peut commuter dynamiquement entre les canaux des données pour l'émission ou la réception des trames des données, ce principe de fonctionnement est illustré sur la figure 15. Pendant le transfert des données un nœud à travers son interface de contrôle continue à écouter le canal de contrôle et maintient donc une liste d'information sur l'usage des canaux de données et les rendez-vous établis entre les nœuds. Pour émettre une trame de données, l'émetteur vérifie d'après sa liste deux conditions : si le récepteur n'utilise son interface de données, et il y a un canal de données libre ou il sera libre pendant une durée égale à la durée d'échange d'une trame de contrôle. Cette approche, par rapport au contexte mono-interface, réduit les problèmes multi-canal du terminal caché grâce à son interface fixée de façon permanente sur le canal de contrôle, au coût radio non négligeable.

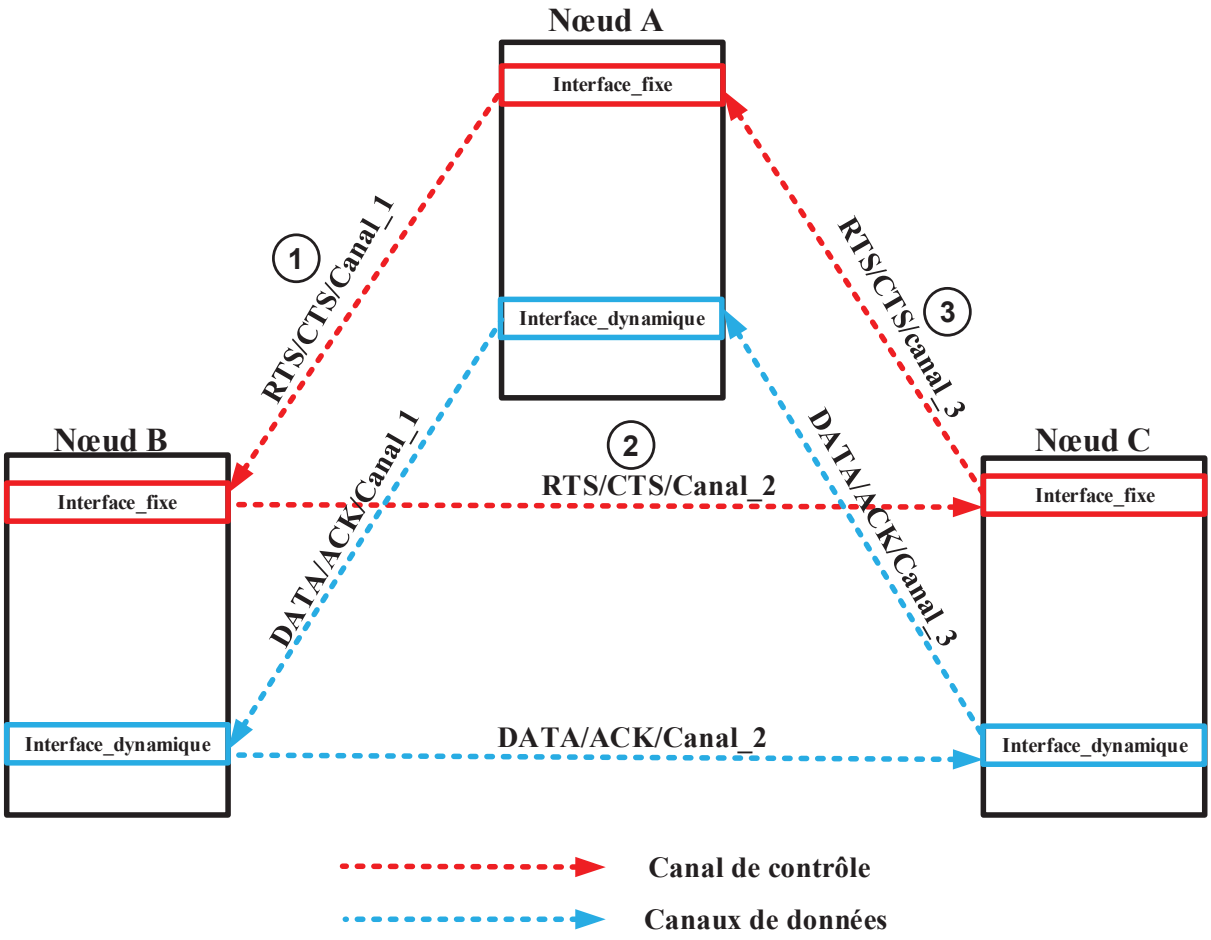

Figure 15: Canal dédié multi-interface

1.1.2.3.2.3 Le protocole de l'interface dédié

Pour le protocole l'interface dédié [12] [32], chaque nœud est équipé de n_i $(n_i \geq 2)$ interfaces, dont la valeur de n_i peut être différente pour les différents nœuds [32], dont k $(1 \leq k \leq n_i)$ sont fixées à k canaux différentes de façon permanente ou pour une longue durée. Le reste des interfaces $n_i - k$ peuvent commuter dynamiquement entre les autres canaux restants (les canaux fixes ne sont pas les mêmes pour tous les nœuds). Le principe de fonctionnement de ce protocole est illustré par la figure 16. L'objectif de ce protocole est pour chaque nœud qui veut émettre des trames données, utilisera une interface dynamique et envoyer sur un canal fixe de l'interface fixe du récepteur. Dans le cas où deux nœuds ont en commun un canal fixe, alors l'émetteur doit utiliser ce canal fixe commun pour émettre des trames données. Puis qu'il y a des canaux fixe pour tous les nœuds et à l'aide des tables de voisinages qui contiennent les informations des canaux fixes des voisins, on peut dire que cette approche peut résoudre certains problèmes rencontrés dans le cas mono-interface multi-canal bien évidemment au coût radio imposé par cette dernière.

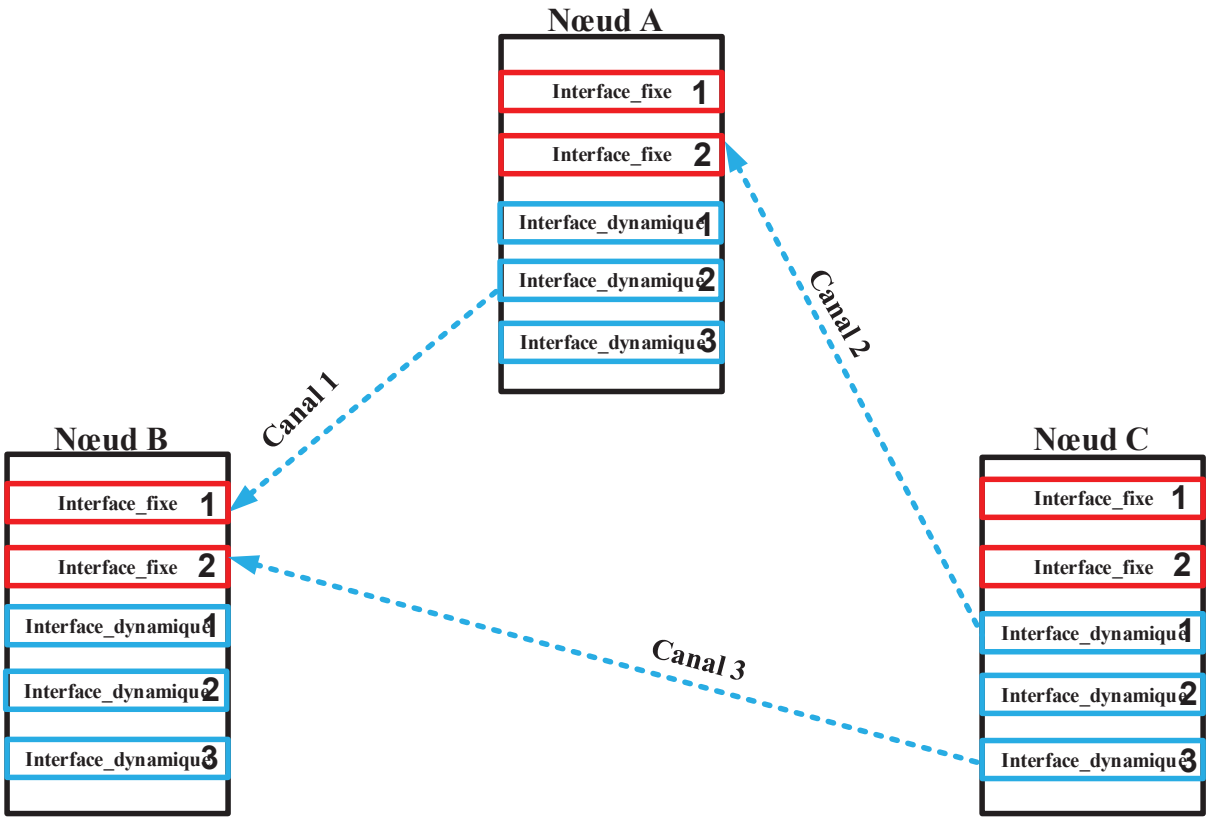

Figure 16: Principe de l'interface dédié

1.1.2.3.2.4 Le protocole d'affectation statique (sans commutation)

L'affectation statique (sans commutation) [12] [32] **:** les interfaces sont fixées de façon permanentes quand le système est initialisé sur les canaux qu'elles utilisent (figure 17) et peuvent communiquer à la fois. La présence d'une entité [27], qui se charge de coordonner les différentes interfaces et de sélectionner l'interface appropriée pour communiquer. L'affectation statique peut être classée en deux types :

1) ***affectation des canaux communs*** [33] : Dans cette approche, les interfaces de tous les nœuds sont affectées à un ensemble de canaux commun. Par exemple, si deux interfaces sont utilisées sur chaque nœud, les deux interfaces sont affectées aux mêmes deux canaux pour chaque nœud. L'avantage de cette approche est que la connectivité du réseau est la même que celle d'une approche mono-canal. Cette stratégie est semblable à celle de mono-canal sauf que plusieurs canaux sont exploités à la fois. On peut remarquer que l'approche mono-canal et lorsqu'une seule interface est utilisée est un cas particulier de la stratégie d'affectation statique du canal commun.
2) ***affectation des canaux différents*** [34] : Dans cette approche, les interfaces des différents nœuds peuvent être affectées à un ensemble de canaux différent. Avec cette approche, il est probable que les partitions logiques du réseau pourraient surgir si les affectations d'interfaces ne sont pas reparties convenablement.
   On peut aussi créer des sauts des transmissions inutilement entre deux nœuds qui sont à porté, mais vu qu'ils n'ont pas un canal commun, l'émetteur doit passer par d'autres nœuds afin d'atteindre sa destination.

Pour éviter un délai important de liaison (nombre des sauts) et une éventuelle partition du réseau, il faut au moins que chaque nœud dans le réseau doit avoir une interface commutée sur un canal commun à ses voisins immédiat.

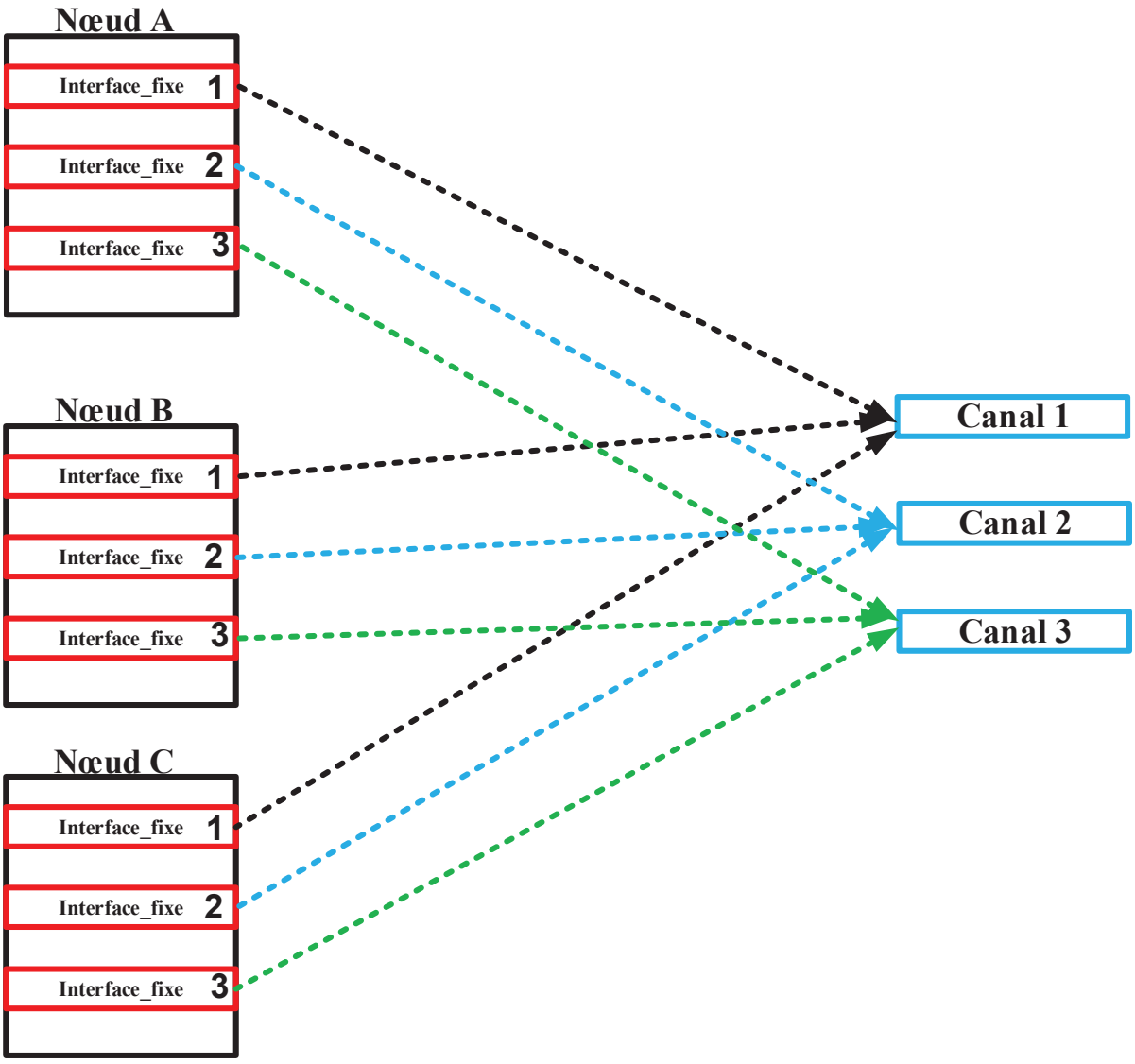

Figure 17: Principe de l'affectation statique (approche sans commutation)

Cette approche résout certains problèmes multi-canal, mais nécessite une entité coordinatrice et souffre également d'un coût matériel très élevé lié au nombre d'interface.

1.1.2.3.2.5 Le protocole d'affectation dynamique

Dans les stratégies d'affectation dynamiques, une interface peut être affectée à un canal quelconque parmi les canaux disponibles, et les interfaces peuvent souvent commuter d'un canal à un autre. Dans ce contexte, pour que deux nœuds se communiquent entre eux, un mécanisme de coordination est nécessaire afin d'assurer qu'ils seront sur un canal commun à un moment donné. Par exemple, le mécanisme de coordination peut exiger que tous les nœuds dans le réseau de visiter périodiquement un canal commun (canal de rendez-vous). L'avantage de l'affectation dynamique est la possibilité de

commuter une interface sur n'importe quel canal, permettant ainsi la possibilité de couvrir tous les canaux disponibles avec quelques interfaces.

1.1.3 Méthode d'accès multi-canal multi-saut

1.1.3.1 L'accès multi-canal multi-saut aléatoire (sans rendez-vous)

Certains travaux ont également évoqués l'accès multi-canal aléatoire (sans RDV) [40] et multi-saut [13]. Nasipuri et *al* [13] ont mis en œuvre un protocole *CSMA (Carrier Sense Multiple Access)* MAC multi-canal pour les réseaux sans fil multi-saut. L'objectif principal de leur travail est de réduire les collisions qui se produisent pendant les transmissions dans les réseaux sans fil et réduire également l'effet du nœud caché. Ce protocole permet une sélection dynamique de canaux, l'idée est semblable à la méthode d'accès FDMA *(Frequency Division Multiple Access),* mais la différence en est que, pour le CSMA MAC multi-canal, l'affectation des canaux se fait de manière distribuée sans aucune infrastructure centrale à travers le système CSMA.
Chaque nœud surveille les *N* canaux continuellement, à chaque fois qu'il ne transmet pas. Il détecte si oui ou non la puissance totale du signal reçu *(Total Received Rignal Strength TRSS)* dans les canaux sont au-dessus ou en dessous de son seuil de détection *(Sensing Threshold ST)* Les canaux pour lesquels la TRSS est en dessous de la *ST*, sont marqués comme *IDLE* (libre). Le temps auquel le TRSS a chuté en dessous du *ST* est noté pour chaque canal. Ces canaux sont mis sur une liste de canaux libres. Le reste des canaux sont marqués comme *BUSY* (occupé).

Lorsque le nombre de canaux, N est suffisamment grand, le protocole a tendance à réserver un canal pour la transmission de données pour chaque nœud. Ce système de réservation de canal réduit les occasions où deux transmissions concurrentes choisissent le même canal. Cependant, un grand nombre de canaux peut causer un temps de transmission paquet trop élevé.

1.1.3.2 L'accès multi-canal multi-saut avec rendez-vous

Dans [14], les auteurs proposent un protocole MAC multi-canal multi-saut basé sur CSMA qui sépare le canal de contrôle pour l'échange des trames des contrôles et un nombre prédéfini des canaux inférieur à celui des nœuds dans le réseau pour les transmissions des données, afin d'éviter des probables collisions entre les trames des données et les trames des contrôles. Etant donné que le protocole est basé sur le système *CSMA (Carrier Sense Multiple Access),* en présence d'une forte charge des trafics, il souffre du problème multi-canal du terminal caché. Comme les auteurs étendent une idée de la méthode d'accès 802.11 *DCF (Distributed Coordination Function)* avec quelques structures différentes : un canal de contrôle approprié pour les trames de contrôles séparés des canaux de transmissions des données.

Avant de transmettre une trame *RTS (Request to send),* l'émetteur détecte la porteuse sur tous les canaux de données et crée une liste de canaux de données disponibles pour la transmission, c'est-à-dire les canaux dont la puissance totale reçue est inférieure au seuil de détection de la porteuse. Cette liste est incorporée dans le paquet RTS du nœud émetteur.

Si la liste de canaux libre est vide, le nœud tire un *backoff* et réessaie de transmettre une autre trame RTS après l'expiration du *backoff*.

Contrairement au 802.11, les autres nœuds voisins de l'émetteur recevant les RTS sur le canal de contrôle différent leurs transmissions seulement jusqu'à la durée du *CTS (Clear to send)* et pas jusqu'à

la durée de *ACK (Acknowledgement)*. C'est parce que les données et ACK sont transmis dans un canal de données et donc ne peuvent pas interférer avec les autres transmissions RTS/CTS. Ainsi donc les nœuds dans le voisinage du nœud destinataire, sur réception du paquet CTS sur le canal de contrôle, s'abstiennent d'émettre sur le canal de données indiqué dans le CTS pour la durée de l'ensemble de transfert (y compris les ACK).

Après avoir reçu la trame RTS et, avant d'émettre le CTS, le nœud destinataire crée sa propre liste de canaux libre par détection de la porteuse sur tous les canaux de données. Il compare ensuite cette liste des canaux libre avec celle contenue dans la trame RTS.

S'il y a des canaux libres en commun, le nœud destinataire choisit un canal commun et envoie ces informations dans la trame CTS.

S'il n'y a pas des canaux libres communs disponibles, le destinataire n'envoie pas un CTS. Puis la source une fois encore tente d'envoyer un autre RTS après un *backoff*.

Lorsque le nœud émetteur reçoit la trame CTS, il transmet le paquet de données sur le canal de données indiqué dans le CTS.

Ensuite les auteurs comparent leur protocole MAC multi-canal en termes de débit et du délai par rapport à la méthode d'accès 802.11 à canal unique. D'après leur résultat, ils concluent qu'il y'ait une nette amélioration du débit et du délai de leur méthode proposé sur la méthode 802.11.

Le protocole du [41] basé sur le CSMA/CA n'utilise l'accès multi-canal aléatoire que pour l'échange des trames de contrôles RTS/CTS afin de réduire les collisions entre les trames de contrôles qui à leur point de vu dégradent les performances du réseau, et utilise la totalité du canal pour l'échange des trames de données.

Les auteurs de [17] modifient la couche MAC du 802.11 et l'adaptent en une méthode multi-canal multi-saut, puisque dans cette norme on distingue trois canaux qui peuvent être utilisés simultanément sans interférence. L'objectif de ce travail est de proposer un système de contrôle congestion au niveau des nœuds intermédiaire, afin d'éviter un problème du délai et dégradation du débit dans le réseau, aussi bien de résoudre le problème multi-canal du nœud caché.

Chen et *al* [15] proposent un protocole MAC multi-canal multi-saut appelé AMNP pour *Ad hoc Multi-channelNegotiation Protocol for multi-hop mobile wireless networks* qui utilise un seul canal pour les trames de contrôle et les restes de canaux pour la transmission des données. Ils étendent le concept des trames de contrôles RTS/CTS utilisé dans le standard 802.11 où des champs supplémentaires ont été ajoutés pour indiquer le canal sélectionné aussi bien pour préciser les canaux libres ou en cours d'utilisation.

Shila, et *al* [16] ont mis en œuvre un protocole MAC multi-canal multi-saut appelé CoopMC *(A Cooperative Multi-Channel MAC Protocol for Wireless Networks)* MAC qui permet aux nœuds du réseau de négocier des canaux dynamiquement. Dans CoopMC MAC cinq trames de contrôle sont utilisées, les nœuds voisins à un saut s'échangent des tables d'informations sur les canaux afin de trouver le meilleur canal entre la source et destination. Notons ici que le canal de contrôle sera aussi utilisé pendant la phase de transmission des données.

1.1.4 Intérêt des approches étudiées dans le contexte multi-saut

En général, les réseaux mobiles ad hoc sont formés de façon dynamique par un système autonome de nœuds mobiles qui sont connectés via des liaisons sans fil sans l'aide de l'infrastructure réseau existante ou administration centralisée. Les nœuds sont libres de se déplacer de façon aléatoire et s'organiser de façon arbitraire ; ainsi, la topologie du réseau sans fil peut changer rapidement et de manière imprévisible. En effet, les liaisons entre les nœuds d'un réseau ad hoc peuvent inclure plusieurs sauts, pour communiquer avec les nœuds qui se trouvent au-delà de leurs portées, les nœuds ont besoin donc d'utiliser des nœuds intermédiaires pour relayer les messages à chaque saut. Partant de ces contraintes très complexes de ces types des réseaux et vue les exigences de l'approche du saut commun qui, d'abord nécessite une synchronisation stricte entre les nœuds du réseau, ce qui n'est pas du tout évident dans un réseau multi-saut de par ses caractéristiques. Cette approche souffre aussi des problèmes multi-canal du terminal caché et exposé ce qui complique d'avantage l'activité multi-saut. Aussi, la resynchronisation de deux nœuds en activité sur un canal à un moment donnée au reste du réseau reste un peu douteuse avec le changement constant de la topologie du réseau.

Bien que les protocoles de rendez-vous parallèles éliminent le problème de goulot d'étranglement en exploitant plusieurs canaux à la fois, mais tel n'est pas le problème crucial d'un réseau ad-hoc multi-saut. En effet dans ces types des réseaux, l'émission et la réception des données ne sont pas seulement affectée par un des nœuds dans un saut mais plutôt au-delà d'un saut. Ceci laisse comprendre que les problèmes des nœuds cachés et les nœuds exposés sont plus fréquents dans les réseaux multi-saut. Notons aussi que les protocoles de *sauts indépendant* exigent également à chaque nœud du réseau des mécanismes de synchronisation afin de suivre les séquencés de sauts des autres nœuds ; en plus de cette activité, dans le contexte ad-hoc multi-saut, la mobilité modifie dynamiquement la configuration du réseau, afin de s'adapter à la mobilité, les nœuds du réseau doivent échanger des informations sur la topologie constamment. Concevoir un protocole MAC multi-canal qui va tenir compte de tous ces paramètres et de toutes ces contraintes s'avère difficile.

Partant de ces remarques, nous pouvons déduire que les protocoles de rendez-vous parallèles permettent d'exploiter efficacement tous les canaux du réseau, mais ces derniers imposent aussi des paramètres et des procédures qui sont difficile à exploiter dans un contexte multi-saut.

Par contre les deux autres approches à savoir : le canal de contrôle dédié et le *split phase* donnent la possibilité de les exploiter dans un contexte multi-saut. Puis que ces derniers utilisent en commun un seul canal appelé le canal de contrôle pour tous les nœuds du réseau à un moment donné, où chaque nœud inactif écoute ce canal. Etant donné que l'activité de la diffusion *(broadcast)* est très importante dans les réseaux ad-hoc multi-saut pour échanger des informations sur la topologie dynamique du réseau, et la synchronisation est plus simple par rapport au saut commun, ces approches simplifient le contexte multi-saut.

1.2 Conclusion

Nous avons effectué une première étude bibliographique des protocoles d'accès multi-canal existants, ce qui nous a permis d'identifier les lacunes des uns et les avantages des autres, sur lesquelles, nous pourrons nous baser dans nos futurs travaux.
Nous remarquons que certes, les protocoles des méthodes d'accès MAC multi-canal ont considérablement amélioré le débit et réduit le délai, mais ces derniers ont aussi suscité d'autres problèmes. Certains sont classiques aux méthodes d'accès MAC mono-canal, à savoir le problème multi-canal du nœud caché, le goulot d'étranglement… D'autres par contre sont inhérents aux MAC multi-canal, tel que le problème de surdité, le délai de commutation du canal et la diffusion

(broadcast). Le problème de goulot d'étranglement a été en quelque sorte résolu mais avec un surcout radio à ne pas négliger.

Nous avons aussi remarqué que la plupart des méthodes d'accès multi-canal qui ont été proposées traitent principalement uniquement le cas des réseaux mono-saut.

Il est donc primordial pour nous de proposer une méthode d'accès multi-canal adaptée à une topologie multi-saut, qui passe à l'échelle, que l'on pourra également prototyper et simuler afin de vérifier ses performances.

Chapitre 2

Proposition d'une méthode d'accès multi-canal multi-saut

2.1 Objectifs et justification des choix

2.1.1 Introduction

De nos jours, les réseaux locaux et réseaux personnels sans fil sont devenus de plus en plus utilisés dans presque tous les domaines, en particulier dans les communications humaines et industrielles. En plus des simples services de données traditionnels, il y a aussi une demande croissante des services multimédias en temps réel tels que la voix et la vidéo. Cette demande se traduit par une demande croissante de bande passante du canal. Les différents groupes de standardisation abordent ces questions de différentes manières, par exemple, en utilisant des techniques OFDM pour atteindre des débits très importants de l'ordre des quelques centaines de mégabits par seconde.

2.1.2 Augmentation des débits et réduction des délais d'attente en émission

Très rapidement, on constate qu'il y a de plus en plus d'exigence dans les domaines sans fil mais, l'accès mono-canal classique pose en même temps des problèmes majeurs. Étant donné que le canal est unique, donc une ressource partagée par tous les nœuds dans le réseau, on remarque d'abord que l'accès à cette ressource est en forte concurrence. Ainsi, une forte charge du réseau augmente rapidement la probabilité des collisions des trames de données et les contentions. Quand un nœud transmet ou reçoit, ses voisins ne peuvent exécuter aucune opération (cf. Figure.18). Ces contraintes ralentissent le débit et augmentent aussi le délai, par conséquent la performance du réseau est dégradée. Par exemple, N9 ne peut recevoir de trames de la part de N12 sur le même canal que celui de N2, car un risque de collision se présente si N9 veut acquitter vers N12 (tout en diffusant sa réponse qui risque de perturber N2).

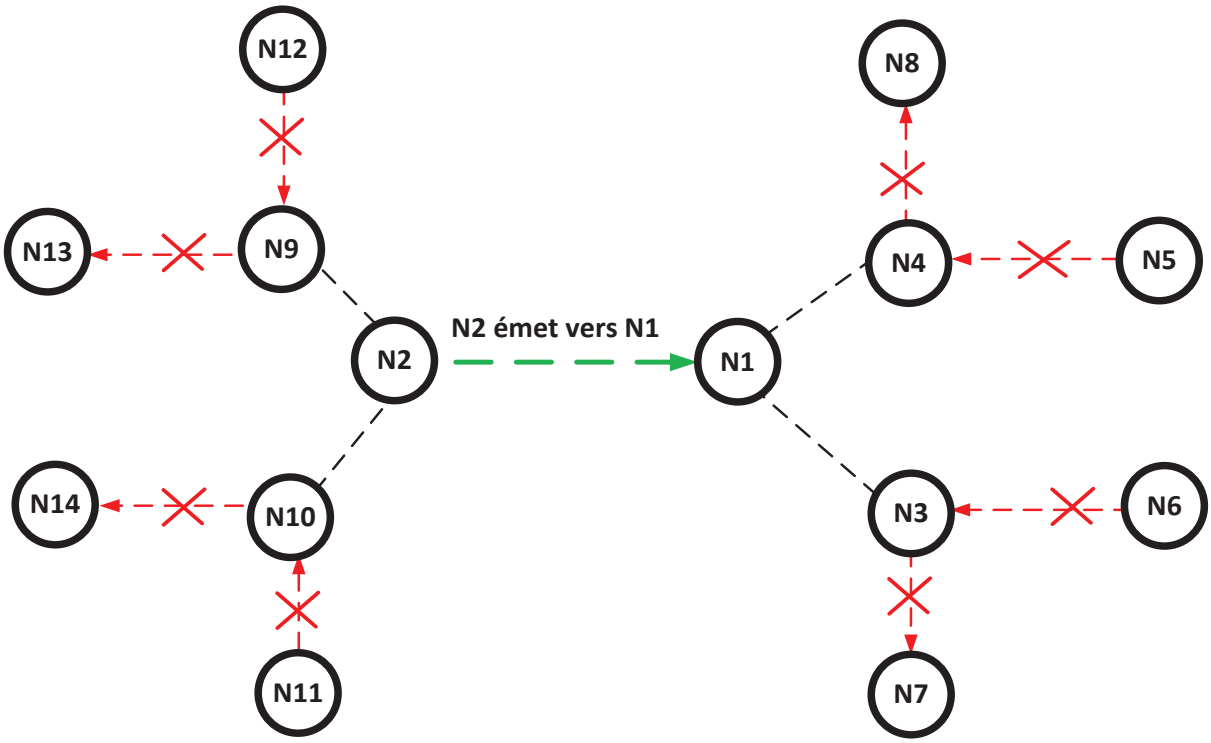

Figure 18: Défaut de l'accès mono-canal

2.1.3 Topologie multi-sauts

Comme on le voit sur la figure 18, une topologie multi-saut peut être présente, où les trames envoyées par un nœud source doivent être relayées ou routées par plusieurs nœuds intermédiaires avant d'atteindre leur destination. Ce cas-là peut très souvent poser des problèmes dans le système mono-canal. En effet, les nœuds, pour relayer leurs trames, peuvent être gênés par leurs voisins immédiats et doivent attendre longtemps pour accéder au canal. Une solution multi-canal est alors envisagée. Les solutions proposées dans la littérature prennent souvent comme hypothèse une topologie mono-saut où il est facile de définir des RDV entre les nœuds. Par contre, cela est plus complexe en multi-saut. La figure 19 montre les différentes zones de propagations des RDV nécessaires afin d'établir une communication multi-saut.

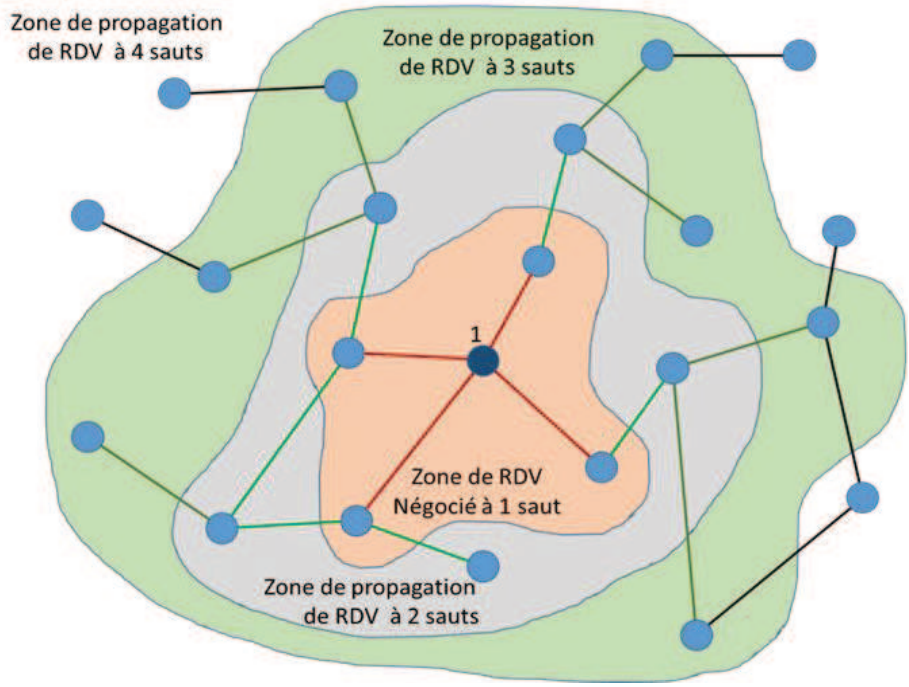

Figure 19: Zones de propagations de RDV pour une transmission multi-saut

Un nœud se trouvant dans une autre zone (au-delà de 2 sauts), hors de la zone du RDV pris en local entre deux nœuds, peut gêner la transmission de ces derniers (provoquer une collision de leurs trames de données), si leurs RDV n'ont pas été propagés dans les autres zones (au-delà de 2 sauts).

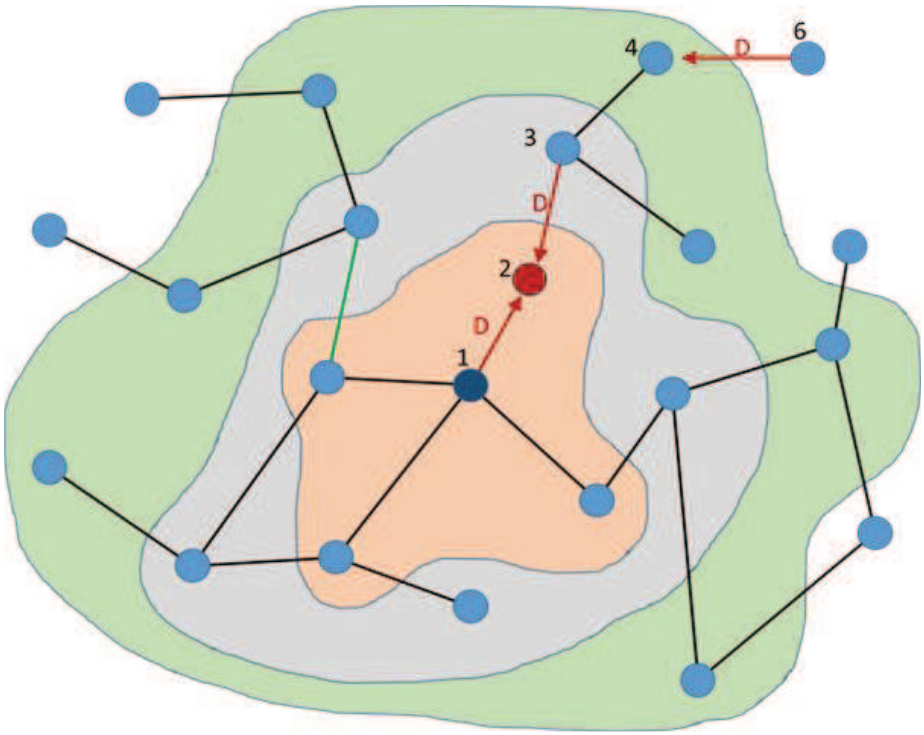

Figure 20: Etablissement d'un RDV mono-canal multi-saut

Sur cette figure (Figure 20, le RDV pris par le nœud 1 avec le nœud 2 doit être propagé à 2 sauts du nœud 1 jusqu'au nœud 3 pour éviter une collision au niveau du nœud 2 si le nœud 1 émet une trame de data en même temps que le nœud 3, sur le même canal. Mais, ceci ne suffit pas pour réellement échapper à une éventuelle collision au niveau du nœud 2, comme on le voit sur la figure 21, où, pour effectuer une transmission fiable entre les nœuds, il faut nécessairement propager le RDV au-delà de 2 sauts.

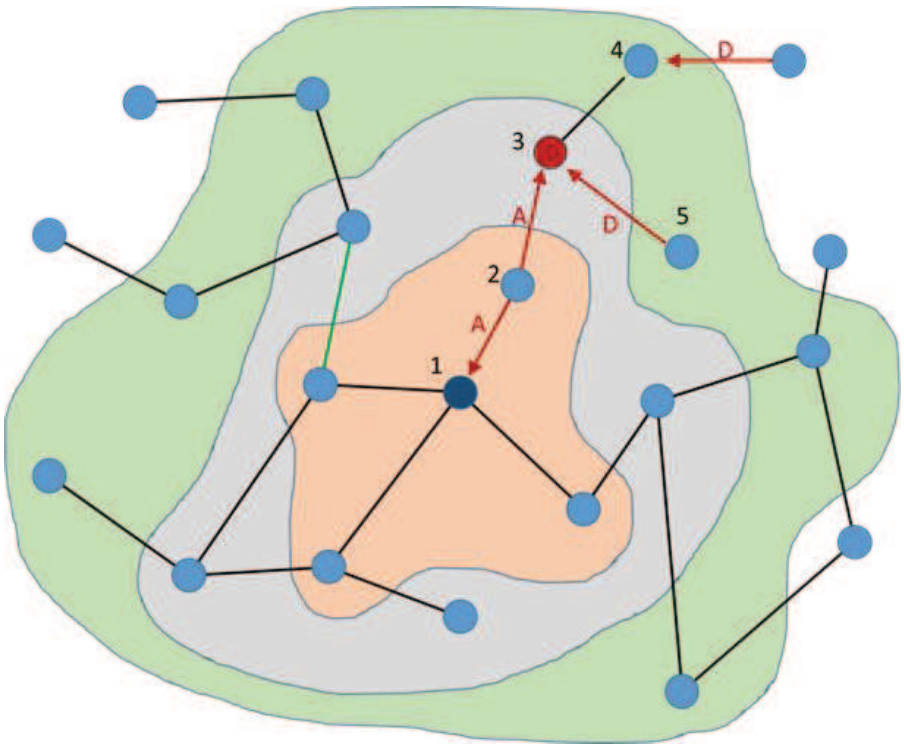

Figure 21 : Etablissement d'un RDV mono-canal multi-saut

Il faut aussi propager le RDV pris par le nœud 1, à 3 sauts vers le nœud 5 pour éviter une collision au niveau du nœud 3, si le nœud 2 acquitte la data du nœud 1 précédemment émise, en même temps que le nœud 5 qui émet sur le même canal une trame, par exemple vers le nœud 3.

Il faut même propager le RDV pris par le nœud 1 jusqu'au nœud 6, donc à 4 sauts pour éviter que le nœud 6 émette une data vers le nœud 4, en même temps que le nœud 1 émette une data vers le nœud 2 (cf. Figure 22). Il risque en effet d'y avoir une collision au niveau du nœud 3 si le nœud 4 acquitte cette data du nœud 6 en même que le nœud 2 acquitte la data du nœud 1.

En conclusion, les RDV en multi-sauts doivent être propagés à 4 sauts, ce qui entraine un important trafic de signalisation, qu'on évite avec une MAC aléatoire comme celle que nous proposons.

Remarque : la propagation d'un RDV n'est pas juste une notification qu'un RDV est pris, c'est aussi une négociation à 4 sauts avec tous les nœuds concurrents à la prise de RDV sur un canal pour un slot (donc un très important trafic et des délais de négociations très longs).

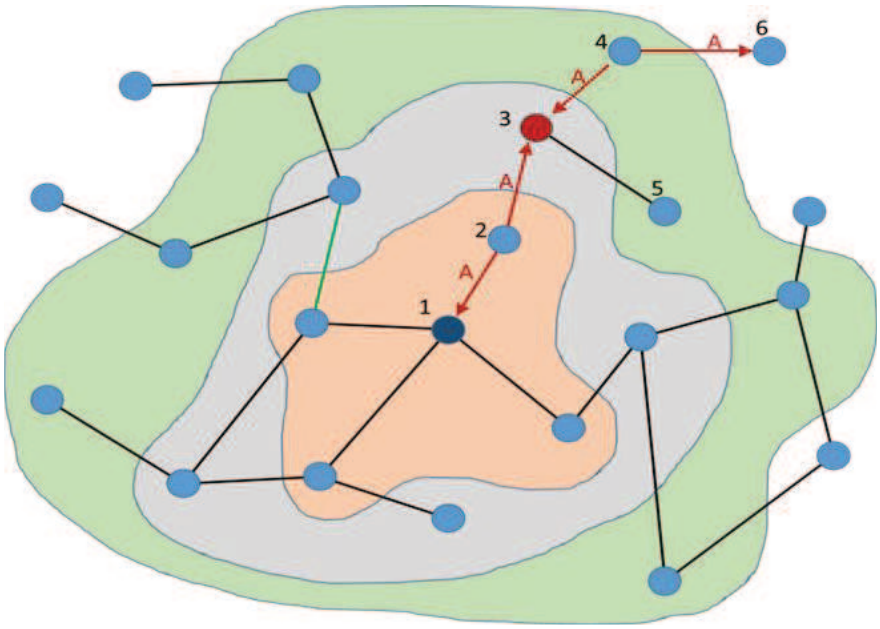

Figure 22 : Etablissement d'un RDV mono-canal multi-saut

Face aux inconvénients des protocoles d'accès au médium mono-canal, la génération actuelle des réseaux sans fil peut tirer avantage de la division du spectre disponible radio en plusieurs canaux. Le nombre des canaux et la capacité du canal dépend de chaque technologie. Par exemple, la norme IEEE 802.11b spécifie 14 canaux fonctionnant dans la bande 2,4 GHz, la norme IEEE 802.11a propose 13 canaux et enfin, la norme IEEE 802.15.4 définit 16 canaux dans la bande 2.4 GHz industrielle, scientifique et médicale (*ISM Industrial, Scientific and Medical*) [44]. En exploitant plusieurs canaux à la fois, on peut résoudre certains problèmes évoqués dans le cas mono-canal et ceci présente les avantages suivants : puisque les trames transmises sur les différents canaux n'interférent pas les uns avec les autres, plusieurs transmissions concurrentes peuvent donc avoir lieu dans la même zone en même temps. Cela conduit à beaucoup moins de collisions, réduit considérablement le délai d'émission et, par conséquent, utiliser plusieurs canaux de manière appropriée, peut potentiellement augmenter le débit (cf. Figure 24).

L'utilisation de plusieurs canaux différents peut faciliter la transmission multi-saut, puisque deux nœuds se trouvant dans une même zone de portée pourront facilement relayer leurs trames en utilisant deux canaux différents. Ainsi, en exploitant la réutilisation spatiale, on peut relayer le maximum des trames possible en même temps.

Les deux cas de figure nous montrent clairement l'inconvénient d'un accès mono-canal et, l'intérêt de l'accès multi-canal. Sur la figure 23, il s'agit d'une transmission mono-canal, les émetteurs N1, N5, N9 peuvent émettre au même moment (même slot de temps), tandis que tous les autres dans la topologie considérée n'exécutent aucune opération (émission ou réception). Sur les quatre liens potentiels de transmissions, une seule transmission est possible, limitée par la zone d'interférence (estimée à deux sauts par rapport au récepteur). On remarque ici que les émetteurs N1, N5, N9 monopolisent une partie importante de la bande passante. Pendant chaque intervalle de temps, on a une proportion de 1/4 (un lien sur quatre est activé) des nœuds en émission, donc un débit avoisinant le 1/4 du débit disponible. Si on se trouve dans un réseau multi-saut, les émetteurs N1, N5, N9 pénalisent les autres nœuds se trouvant dans la zone d'interférence pour relayer les trames reçues vers leurs destinations et, on se trouve finalement dans un réseau encombré, c'est un grand défaut de l'accès mono-canal.

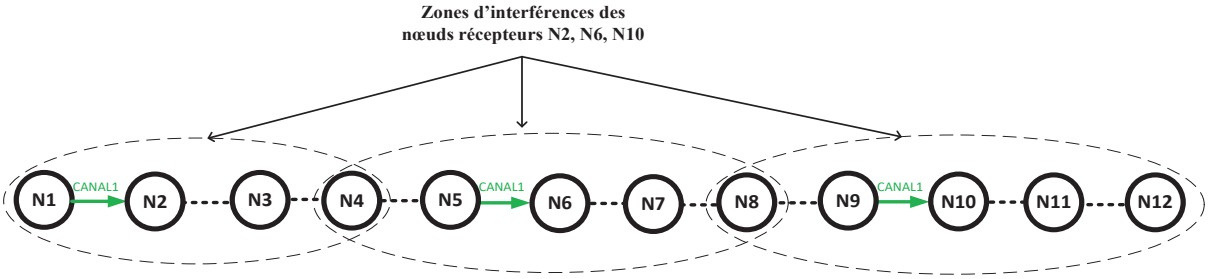

Figure 23: un seul canal disponible pour toute la topologie

En utilisant plusieurs canaux à la fois (deux canaux sur la topologie de la Figure.24), deux communications peuvent s'établir simultanément dans une même zone (deux liens sur quatre sont actifs), ce qui n'est pas possible lorsqu'on utilise un seul canal, ainsi donc le débit est doublé de 1/4 à 1/2.

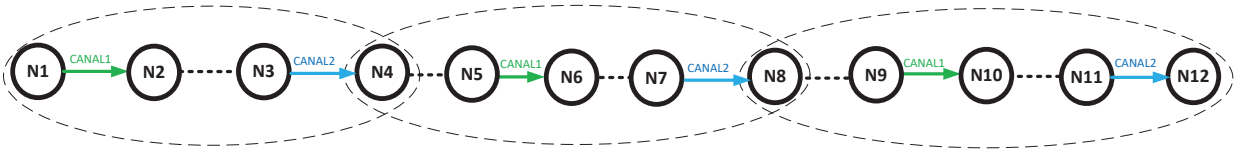

Figure 24: deux canaux disponibles pour toute la topologie

On peut ainsi dire que l'accès multi-canal est bien adapté pour les réseaux multi-saut, puisque les nœuds dans le réseau ne seront pas pénalisés par la zone d'interférence pour relayer les trames reçues vers leurs destinations. D'après la figure 24, les nœuds N3, N7, N9, N11 utilisent le CANAL2 pour émettre leurs trames et, dans le même intervalle de temps que les émetteurs N1, N5, N9.

2.1.4 Autres problèmes de RDV en multi-saut liés aux chemins

Les méthodes d'accès multi-canal qui ont été proposées pour gérer l'attribution des canaux aux différents nœuds dans le réseau, abordent très souvent l'accès multi-canal dans un réseau mono-saut simpliste. Dans ces types des réseaux, on considère toujours que les RDV établis par chaque paire des nœuds sont signalés à leurs voisins immédiats donc à un seul saut. Mais, il peut y avoir des cas où l'émetteur sera limité par sa portée radio. Dans ce contexte précis, les trames nécessitent d'être relayées ou routées par des nœuds intermédiaires jusqu'à leur destination (cf. Figure 25). Si le nœud R émet des trames vers le nœud G, ces trames seront relayées par les nœuds A, B, D avant d'atteindre leur destination G. Dans ce cas précis, le RDV établi par chaque paire de nœuds du trajet doit être absolument propagé par ces derniers à leurs voisins au-delà d'un saut avant leurs transmissions à travers des trames des contrôles. Si un canal unique est utilisé par le réseau, alors, les voisins à deux sauts de toutes les paires des nœuds du trajet entre R et G (pointillé vert) qui sont déjà informés de l'occupation du canal à partir des trames de contrôles reçues seront pénalisés par les zones d'interférences n'émettant pas sur le canal durant toute la transmission.

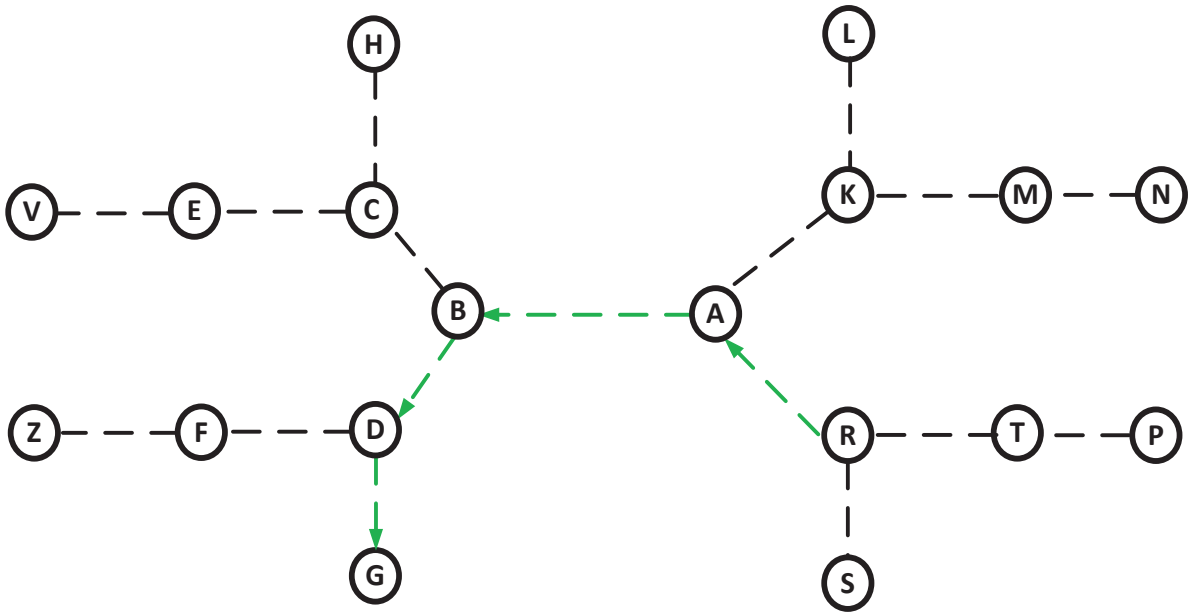

Figure 25: Topologie d'un réseau multi-saut

Mais lorsque deux ou plusieurs canaux sont utilisés, les voisins à deux sauts (sans ACK) ou 4 sauts (avec ACK) de ces nœuds du trajet peuvent sélectionner des canaux différents pour envoyer leurs trames. Par exemple, si le nœud R a choisi le canal 1 pour émettre vers le nœud A, à partir des trames de contrôles échangées par les nœuds A et R et qui seront reçues par leurs voisins, le nœud T sera donc informé du RDV établie par la paire des nœuds A et R et du canal sélectionné et peut alors utiliser le canal 2 pour émettre vers le nœud P.

Si nous appliquons les différentes méthodes d'accès multi-canal avec RDV que nous avons étudiées, telles que le canal de contrôle dédié, le *split phase*, le saut commun… à la topologie de la figure 25, il est nécessaire que le RDV établi par les nœuds R et A soit connu par leurs voisins immédiats (S et T voisins de R, B et K voisins de A) et qu'ils le diffusent à leurs tour aux voisins. On risque d'être en présence d'un réseau inondé par des petites trames de contrôles qui vont aussi provoquer des collisions avec les autres trames de données, ralentir le débit et, par conséquent dégrader la performance globale du réseau.

2.1.5 Evolution multi-canal de la méthode d'accès aléatoire Aloha slottée : simplicité et réduction du trafic de service par rapport à un accès contrôlé

Nous avons donc constaté que les méthodes d'accès multi-canal avec RDV ne sont pas une solution simple et optimale aux problèmes classiques rencontrés dans les méthodes d'accès mono-canal (problème du nœud caché…), puisque ces derniers suscitent encore d'autres problèmes (surdité…), surtout quand il s'agit d'un réseau multi-saut où il existe très peu des solutions optimales car faisant souvent recours à l'utilisation des trames de contrôles afin d'établir le RDV.

En multi-saut, les nœuds se trouvant au-delà d'un saut ne sont pas souvent conscients des RDV pris en local et, ainsi peuvent perturber les transmissions qui se déroulent suite aux RDV.

Pour des telles raisons, nous nous orientons vers une solution protocolaire MAC simple, en proposant une méthode d'accès multi-canal sans RDV basée sur l'Aloha slottée afin d'éviter la propagation et la

négociation des RDV, et ainsi réduire le trafic de services par rapport à un accès contrôlé. Nous espérons que cette réduction permettra à combler les lacunes liées à la faible performance native d'Aloha de 18% (tout de même doublée dans sa version slottée comme nous allons le rappeler plus bas).

Selon la topologie de la figure 25, en utilisant la méthode d'accès multi-canal multi-saut sans RDV, la transmission de A vers R n'empêche pas leurs voisins de choisir des canaux différents pour transmettre leurs données figure 26. Ainsi plusieurs transmissions concurrentes se déroulent au même moment sur les canaux disponibles dans une même zone d'interférence mono-canal.

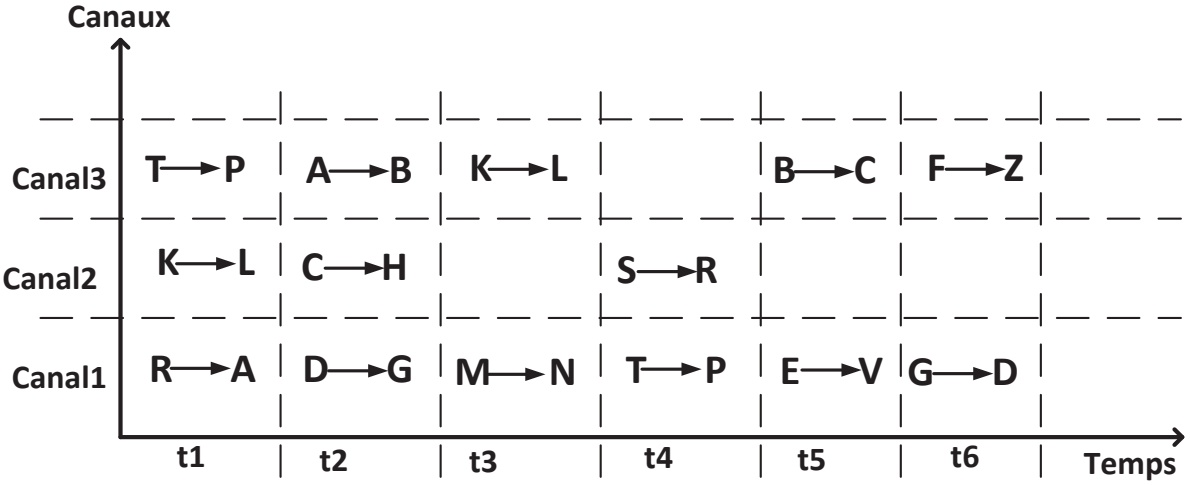

Figure 26 : Accès multi-canal sans RDV

2.2 Rappel de la méthode d'accès Aloha slottée mono-canal

La méthode d'accès aléatoire utilise deux principes essentiels de transmissions : le protocole Aloha pure (*unslotted Aloha*) et le protocole Aloha slotté (*slotted Aloha*) ou Aloha en tranches.

2.2.1 Principe de fonctionnement de l'Aloha pur

Il s'agit de l'un des premiers protocoles d'accès aléatoire décrit dans la littérature [21][22][59]. Dans ce mode de transmission, il n'y a pas écoute du médium avant la transmission. Quand une trame arrive à la file d'attente d'envoi d'un terminal mobile, il transmet immédiatement la trame sur la ressource sans fil partagée entre tous les utilisateurs mobiles se trouvant dans le réseau. Le terminal attend ensuite pendant un délai déterminé (chien de garde) un acquittement de la trame envoyée par le destinataire de cette trame de données. Si la trame est acquittée dans le délai prévu, on suppose que la transmission s'est effectuée correctement. Dans le cas contraire, c'est sans doute que plusieurs trames ont été envoyées simultanément et ont donc créé une collision, c'est-à-dire une superposition des signaux de plusieurs utilisateurs. Ces trames ont besoin d'être retransmises plus tard après un délai aléatoire tiré par chaque nœud initial, pour désynchroniser les collisions. Ceci peut être observé sur la figure 27.

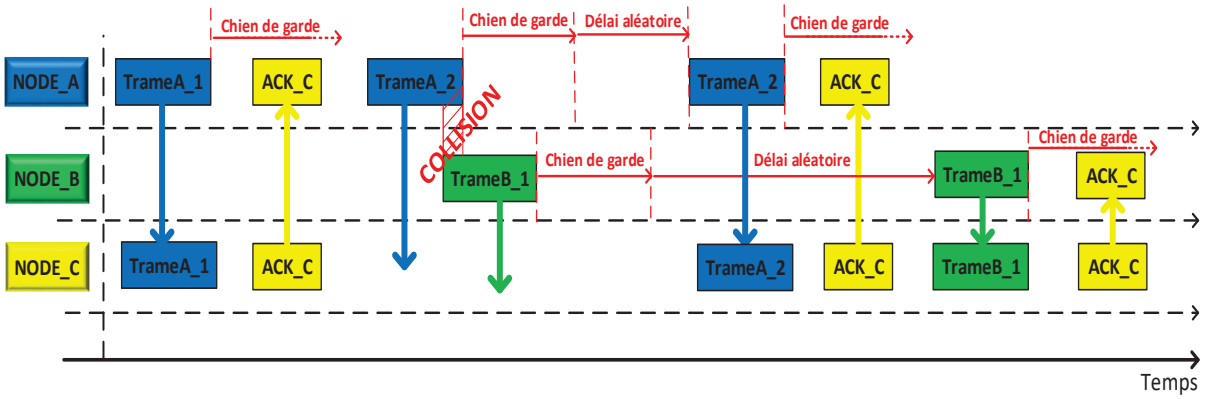

Figure 27: Principe de transmission de l'Aloha pur

Pour éviter que les mêmes trames n'entrent encore en collision, il est effectivement nécessaire que chaque émetteur tire un délai d'attente aléatoire avant de retransmettre sa trame. Ce protocole est particulièrement simple à mettre en œuvre, mais en présence d'une forte charge de trafic (nombreux utilisateurs avec des trames à émettre), il souffre d'un débit utile très faible. Ceci s'explique avec la modélisation mathématique associé à l'Aloha pur :

On suppose que le nombre de trames générées et retransmis dans le réseau est une distribution de Poisson.

La probabilité pour que *n* trames arrivent en deux périodes différentes de trames est [60] [61] :

$$P(n) = \frac{(2G)^n \exp(-2G)}{n!} \tag{2}$$

G est la charge du trafic, et *exp()* est une exponentiel.

Nous pouvons ainsi déterminer la probabilité de succès *P(0),* c'est-à-dire qu'aucune autre trame soit transmise pendant la période de vulnérabilité d'une trame comme suit [60] [61]:

$$P_{succ}(0) = \frac{(2G)^0 \exp(-2G)}{0!} = \exp(-2G) \tag{3}$$

La probabilité pour qu'une trame arrive au cours de la période vulnérable est [Kleinrock 1976]:

$$P_{vul} = 1 - \exp(-2G) \tag{4}$$

Ainsi, le débit *D* des trames transmises avec succès est :

$$D = G * P_{succ}(0) = G * \exp(-2G) \tag{5}$$

Et donc le débit maximal possible qu'on puisse espérer avec cette méthode est :

$D = \frac{1}{2\exp(1)} = 0.18$, pour une charge de trafic offert de $G = 0.5$. Ainsi, le débit maximum est limité à 18% de la capacité totale du canal.

2.2.2 Principe de fonctionnement de l'Aloha slotté mono-canal

L'exploitation maximale du canal très limitée dans la méthode d'accès Aloha pur a évidemment amenée certaines recherches pour des améliorations possible de la méthode ; d'où la proposition de l'Aloha slotté [21] [22] [59]. La piste qui a suscité le plus d'intérêt est de discrétiser le temps en slots de temps. Chaque nœud synchronisé du réseau n'émet sa trame qu'au début du slot. Dans cette méthode, la synchronisation entre les différents nœuds est effectivement nécessaire. Ainsi, avec Aloha slotté, les collisions se produisent sur l'ensemble de la trame émise, il n'y a pas une collision partielle des trames, la collision se produit ou pas sur toutes les trames (si on suppose qu'elles sont toutes de même taille, inférieure à celle du slot). La collision est totale, réduisant ainsi la période de vulnérabilité par rapport à Aloha pur. Ainsi, en procédant de la même façon comme pour la méthode d'accès Aloha pur, excepté pour la période de vulnérabilité qui est réduite à 1, on peut représenter le principe de transmission de l'Aloha slotté sur la figure 28.

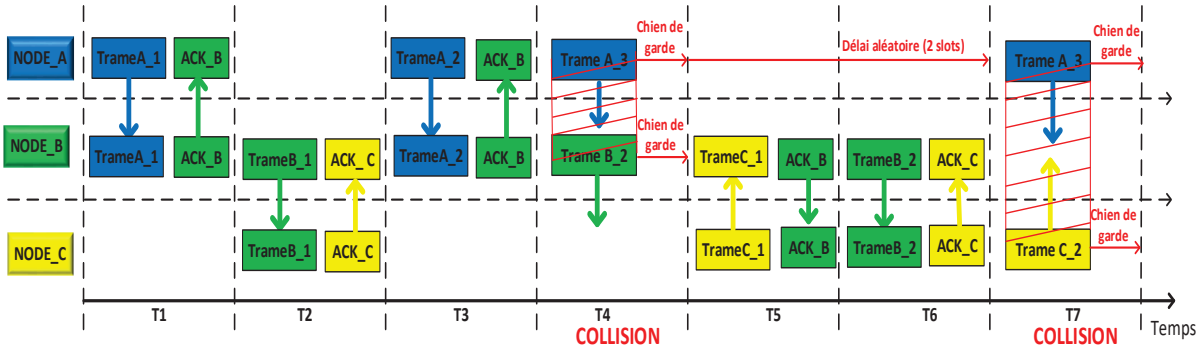

Figure 28: Principe de transmission de l'Aloha slotté

La probabilité pour qu'une trame n'ait pas de collision (probabilité de succès *P(0)*) est [60] [61]:

$$P(0) = \exp(-G) \tag{6}$$

La probabilité pour qu'une collision se produise est :

$$P(collision) = 1 - P(0) = 1 - \exp(-G) \tag{7}$$

Le débit sera alors :

$$D = G * P(0) = G * \exp(-G) \tag{8}$$

Le débit maximal possible est : $D = \frac{1}{\exp(1)} = 0.36$

Cet avantage de l'Aloha slotté sur Aloha pur peut être observé sur la figure 29 qui représente d'un point de vue mathématique, le débit maximum utile en fonction de la montée en charge imposée par l'ensemble des nœuds via leurs émissions de trames.

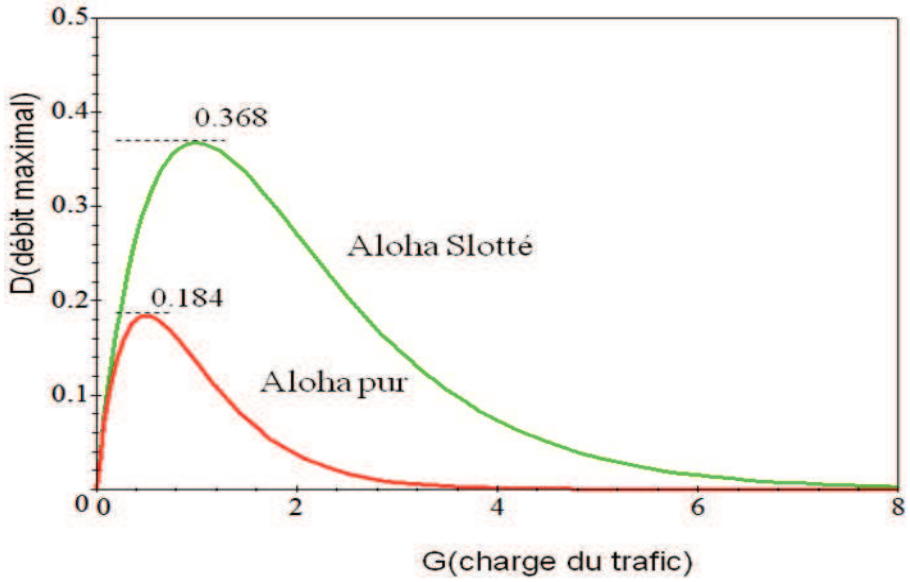

Figure 29: Débit de l'Aloha pur et Aloha slottée (mono-canal)

Nous allons donc proposer une MAC basée sur l'Aloha slottée en l'adaptant au multi-canal. Il nous faut commencer par synchroniser les nœuds, en utilisant SiSP [62], développé auparavant au laboratoire.

2.3 Rappel sur le protocole SiSP de synchronisation des nœuds

2.3.1 Diffusion des horloges et consensus

On vient de le voir, la synchronisation des nœuds est un mécanisme très important, plus particulièrement dans un réseau de capteurs sans fil, permettant aux nœuds capteurs dans le réseau d'avoir une différence entre leurs horloges subjectives la plus faible possible. Pour parvenir à une synchronisation globale dans un réseau sans fil ad hoc, un protocole appelé *SiSP* (*Simple Synchronisation Protocol*) pour la synchronisation des nœuds dans le réseau a été imaginé dans [62]. C'est un protocole scalable qui n'introduit pas de contraintes de hiérarchie dans la topologie du réseau, à l'inverse d'autres comme le très connu RBS [63]. Le protocole de synchronisation SiSP est basé sur un algorithme simple permettant aux nœuds du réseau en quelques itérations d'obtenir par consensus une horloge partagée (SCLK) obtenue par la méthode de la moyenne.

Le principe d'échange des horloges est simplement basé sur un seul type de message : SYNC dont la charge utile contient la valeur de SCLK (horloge partagée) du nœud au moment de la création du message SYNC. Le message est diffusé de manière simple (sans MAC) comme dans le mode diffusion des trames beacons de 802.15.4. Ainsi par exemple, le message SYNC peut être simplement encapsulé par exemple dans la charge utile des beacons 802.15.4. On peut aussi imaginer que cette horloge est émise dans un champ supplémentaire de toute trame, qu'elle soit diffusée (cas idéal) ou pas (dans ce cas-là, il faut compter sur un trafic important du réseau entre tous les nœuds. Nous en reparlerons plus tard…

Chaque nœud maintient deux variables principales, à savoir : le SCLK (horloge partagée) et le LCLK (horloge locale). Le nœud écoute durant 99 tops d'horloge (si on se base sur 1 émission toutes les 100 incrémentations) tandis qu'il calcule la moyenne de toutes les horloges reçues (RCLK) de ses voisins avec son propre SCLK. Pendant ce temps-là, son SCLK est modifiée par la valeur de chaque moyenne.

Au 100^{ième} top d'horloge, le nœud diffuse sa valeur SCLK et les voisins appliquent le même algorithme en effectuant le calcul de la moyenne (SCLK= (SCLK+RCLK / 2)). Chaque fois qu'on atteint le top d'horloge 100, le message SYNC est diffusé. Après un certain nombre de cycles, en fonction du nombre de nœuds et de la topologie, une horloge de consensus globale est obtenue dans tous les nœuds du réseau, car le résultat de la division est arrondi puisque le SCLK est une valeur entière.

L'horloge (SCLK) de consensus est obtenue de manière décentralisée, sans aucune prérogative hiérarchique. La figure 30, montre un exemple de principe de fonctionnement du protocole SiSP où, on observe des échanges des messages SYNC entre deux nœuds A et B.

On voit ici que le nœud A est mis en marche avant le nœud B jusqu'à ce que son horloge locale atteigne la valeur 380. A un moment donné, les nœuds partageront la même valeur de SCLK. Le principe simple du protocole SiSP montre que, lorsqu'un message SYNC est reçu, chaque nœud calcule la moyenne entre les valeurs de RCLK et SCLK, et le résultat de ce calcul est considéré comme la nouvelle valeur de SCLK. Si RCLK et SCLK sont égaux (à l'arrondi prés), cela signifie que l'horloge SCKL du nœud est la même que celle de son voisin.

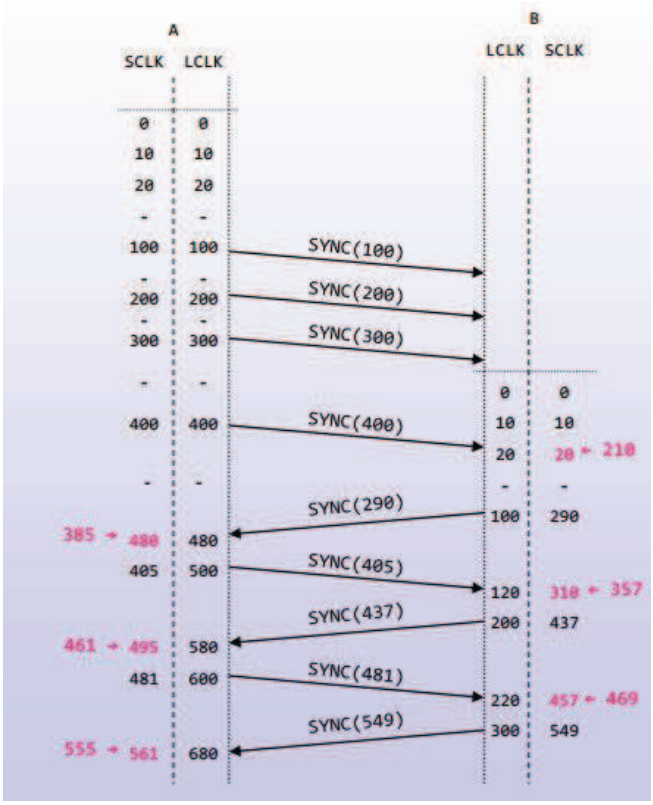

Figure 30: Principe de synchronisation et d'échange des horloges dans SISP [62]

2.3.2 Exemples de résultats de SiSP

Considérons une topologie d'un réseau à quatre nœuds tous à portée (*full-mesh*) comme en figure 31. Les nœuds sont mis en marche indépendamment (de façon aléatoire), le message SYNC est diffusé chaque 100 ms (ceci afin d'éviter d'émettre des messages de synchronisation trop souvent et ainsi risquer d'encombrer trop le médium), alors que le LCLK est incrémenté chaque µs. Sur le graphique, l'axe des abscisses représente le temps et l'axe des ordonnées représente les valeurs de SCLK de chaque nœud. Chaque point du graphique représente le message SYNC. Dans le cas où tous les points sont alignés sur le graphique, ceci veut dire que les nœuds sont synchronisés, ils ont convergé (et continuent normalement de le faire) alors vers une même horloge partagée, le consensus est donc déjà obtenu.

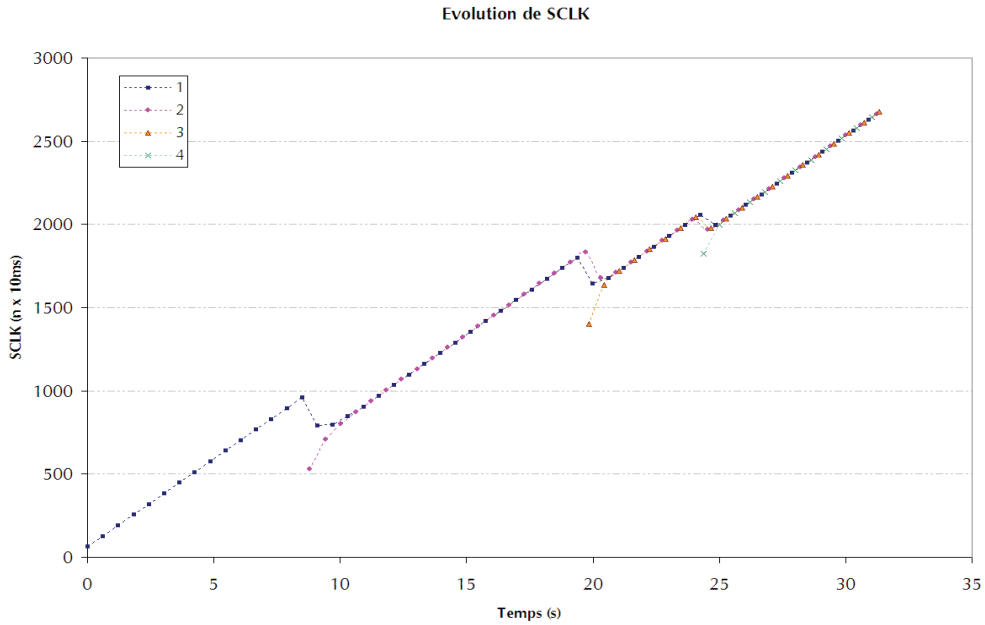

Figure 31: Synchronisation (consensus) entre les nœuds [62]

Comme on peut l'observer sur la figure, après 10 secondes, les deux premiers nœuds 1 et 2 mis en marche commencent à se synchroniser, on voit que leurs courbes se rapprochent ; après 20 secondes écoulées, un troisième nœud est aussi mis en marche et commence ensuite à écouter le médium ; il reçoit deux messages SYNC, il calcule ainsi la moyenne de SCLK deux fois et se rapproche alors ainsi des 2 autres.

Comme on peut le voir sur le graphique, le premier message SYNC transmis par le troisième nœud comprend une valeur de SCLK qui n'est pas très éloignée de celles précédemment diffusées par les deux autres nœuds. Ceci montre que, plus le nombre de nœuds synchronisés est important, plus la synchronisation d'un nouveau nœud s'établit rapidement.

C'est ce protocole SiSP qui sera utilisé dans notre cas afin de définir des slots communs entre les nœuds du réseau pour la méthode d'accès multi-canal slottée.

2.4 Méthode d'accès MAC multi-canal sans RDV proposée

Nous allons dans cette partie, présenter la méthode d'accès multi-canal aléatoire, donc sans RDV, adaptée ainsi aux topologies multi-sauts, que nous proposons dans un premier temps. Cette méthode d'accès de base, sera ensuite enrichie par diverses options permettant de l'améliorer, en fonction des cas d'études et d'applications.

2.4.1 Hypothèses pour la mise en œuvre de la méthode d'accès multi-canal de base

Dans la méthode d'accès multi-canal que nous proposons, nous supposons un nombre des nœuds Net un nombre des canaux C. On suppose aussi que le nombre des nœuds est supérieur au nombre de canaux (donc $N>C$), ce qui est le cas le plus courant et le plus complexe à régler. Le réseau est composé classiquement de nœuds (dans notre prototypage, ce seront des nœuds WiNo) qui sont initialement mono-interface (une seule radio par nœud), donc à un instant donné, un nœud peut soit émettre soit recevoir sur un canal $C_k(k=1, 2, 3...)$ parmi les canaux $C_1, C_2, C_3...$ disponibles mais, le

nœud ne peut pas effectuer les deux opérations à la fois (c'est-à-dire émission et réception sur un même canal en même temps, ou 2 émission ou 2 réceptions en même temps).

Nous considérons aussi une topologie à plat, sans hiérarchie, avec une disposition des nœuds nécessitant généralement un certain nombre de sauts (que nous pouvons noter ***NS : Nombre de Saut avec NS=1, 2, 3…***). Si par exemple ***NS=1,*** ceci veut dire que le nœud a fait un saut pour atteindre sa destination, si ***NS=2*** il a fait deux sauts, etc. On considère aussi qu'il n'y a pas de nœuds isolés puisqu'il existe toujours au moins un lien entre un nœud et le reste du réseau.

Pour la synchronisation des nœuds dans le réseau, nous utilisons comme vu ci-dessus le protocole SiSP développé par l'équipe IRT à l'IUT de Blagnac, qui permet une discrétisation de l'échelle du temps en tranche ou slots. Chaque slot a une durée qui équivaut à l'émission d'une trame d'une longueur égale par exemple lors de notre implémentation à 57 octets (type de la trame de données, adresse source, destination, numéro de séquence et, charge utile), et son acquittement (type de la trame d'acquittement, adresse source, destination, numéro de séquence). Chaque cycle commence par un slot réservé pour les émissions et les réceptions des beacons qui sont utilisés pour émettre les informations utiles à SiSP et la MAC (horloges, informations de voisinage et de connaissance sur l'utilisation de canaux que nous verrons plus tard).

Nous supposons que dans chaque cycle, les slots pairs sont réservés uniquement pour les émissions et les réceptions des beacons sur les différents canaux disponibles. Le premier slot (dont le numéro est 0) du cycle est donc un slot pair de beacons, le second slot est un slot de DATA/ACK.

En début de slot pair, chaque nœud commence par décider aléatoirement s'il émet un beacon dans ce slot ou s'il se met en écoute pour recevoir le beacon d'un autre nœud.

Soit le paramètre *P* la probabilité d'écouter ou d'émettre le beacon : si par exemple *P=0.1*, le nœud émet en moyenne 1 beacon sur 10 slots (*P=1/10*) et peut donc ainsi recevoir pendant les 9 autres slots.

Dans nos hypothèses, le médium radio est supposé réaliste, on estime que le taux d'erreur binaire *BER (Bit Error Rate)* typique est de 10^{-5}.

Chaque trame dispose d'un CRC 16 bits (CCITT) pour la détection des erreurs. Dans cette méthode d'accès multi-canal où les diffusions sont nombreuses, il n'y a pas d'acquittement négatif (NACK). Nous considérons uniquement des acquittements positifs (ACK) quand une trame de données (toujours en unicast) est bien reçue. Les beacons qui sont diffusés ne sont pas acquittés. Les ACK considérés ici sont des ACK de niveau MAC, donc à un seul saut.

Nous considérons aussi que le trafic applicatif n'engendre pas de débordement. La couche 7 est juste présente pour évaluer la couche 2 multi-canal dans notre testbed présenté au chapitre suivant.

La couche 3 permettant d'acheminer une donnée d'un nœud émetteur vers un nœud récepteur est basée dans un premier temps sur un routage aléatoire : quand un nœud reçoit une trame de donnée, il tire au sort un voisin de destination pour lui propager cette donnée si cette trame n'est pas pour lui (elle ne porte pas son adresse de destination). La table de voisinage de niveau 2 maintenue par chaque nœud est ici utilisée également par cette couche 3 (*cross-layer*). Un historique est géré afin d'éviter les boucles : un couple {adresse globale source, numéro de paquet} est utilisé pour différencier les paquets. Un numéro de paquet ou numéro de séquence est utilisé (par exemple sur 8 bits).

2.4.2 Principe de fonctionnement de la méthode d'accès multi-canal basée sur l'Aloha slotté

Le principe de fonctionnement de la méthode d'accès multi-canal proposée peut être décrit de la manière suivante :

Lorsqu'un nœud dans le réseau a une trame à émettre vers un nœud de destination, il tire aléatoirement un prochain slot (***PS***) de données entre 1 et ***PS*** (valeurs typiques de ***PS*** : 1, 2, 5 et 10 pour nos tests). Il tire au sort également un canal d'émission ***CE*** (canal d'émission de données), sauf si un canal d'un succès précédent est mémorisé pour ce voisin (cf. améliorations possibles), et émet sa trame de donnée dans ce slot en attendant l'ACK à la suite de la trame de donnée dans le même slot (et donc sur le même canal). Si le nœud émetteur de la data ne reçoit pas d'ACK, il recommence le même processus en tirant aléatoirement un prochain slot de données et un prochain canal d'émission (notion de *Backoff* simple non exponentiel, pouvant ensuite être améliorée).

Quand un nœud n'a rien à émettre, il tire aléatoirement un canal de réception ***CR*** (canal de réception des données) et y reste un certain nombre de slots ***NSR*** (en fonction des tests et analyses de performances lors de notre évaluation sur testbed, typiquement, ***NSR*** = 1, 2, 5, 10 ou infini).

En début de chaque slot, le beacon est émis ou écouté en tirant aléatoirement un canal d'émission des beacons ***CB*** (canal d'émission/réception des beacons). Une seconde version plus évoluée pourra utiliser le canal 0 dédié uniquement pour les beacons (en comparant l'efficacité de la synchronisation et la réactivité des mises à jour des données de voisinages échangées), comme nous le verrons dans les options possibles.

Chaque nœud maintient à jour en local une table de voisinage regroupant différentes informations :

- La liste des nœuds connus, voisins à 1 ou plusieurs (typiquement 2) sauts.
- Des informations liées à SiSP pour chacun de ces voisins (leur propre vision de l'horloge globale)
- Pour chaque nœud connu, le dernier canal utilisé qui a permis un échange de données avec lui. Dans une version améliorée de cette MAC, quand un nœud doit émettre une trame vers un nœud connu, il commence par choisir le dernier canal utilisé (si succès) plutôt que de le tirer aléatoirement. Il faut donc obligatoirement qu'un nœud en réception reste au moins 2 slots sur le même canal de réception (donc ***NSR*** ici est strictement supérieur à 1).

Cette table (Table 3) est envoyée dans le beacon pour permettre aux autres nœuds de connaître ces informations et mettre à jour eux-mêmes leur propre table. Ces informations vont petit à petit se propager dans tout le réseau.

Table 3 : Caractéristique de la table voisinage

NODE_ID	*CANAL*	*NB_HOP_RX*	*SCLK*

NODE_ID : Identifiant du nœud voisin
CANAL : dernier canal à succès du nœud voisin
NB_HOP_RX : Nombre de saut se trouve le nœud voisin
SCLK : Valeur de l'horloge locale du nœud

Cette proposition de couche MAC est à implémenter sur des vrais nœuds (WiNo Testbed). Nous devrons la comparer en fonction des paramètres cités ci-dessus, à une même couche MAC ALOHA mono-canal slottée de base (travail initial de prototypage réalisé en 2ième année de thèse et présenté au début du chapitre 3). Les métriques sont les délais d'acheminement des trames entre émetteur et récepteur de bout en bout, et les débits de bout en bout offerts en fonction du nombre de canaux

disponibles. Il sera également intéressant d'étudier les travaux de testbed et d'analyse de la communauté scientifique sur des MAC aléatoires, comme ceux de [64].
La figure suivante (Figure 32) donne un exemple de répartition temporelle par canaux de cette MAC en se basant sur la topologie (Figure 33) de test suivante.

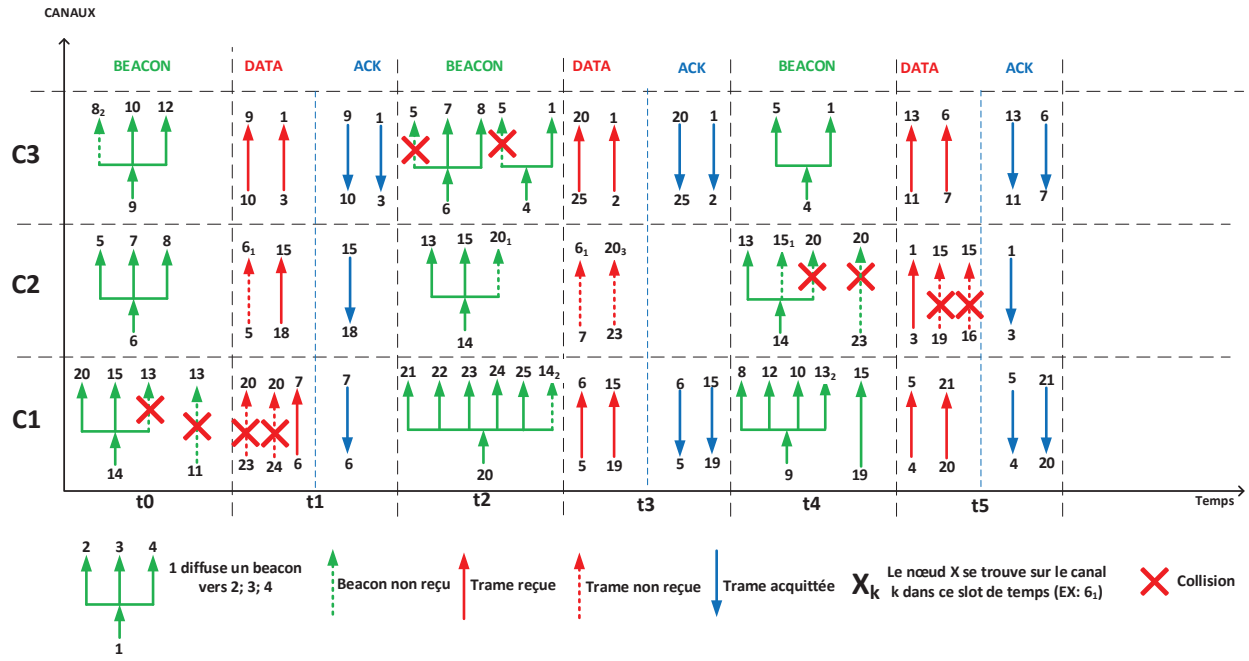

Figure 32 : Méthode d'accès multi-canal basée sur l'Aloha slotté

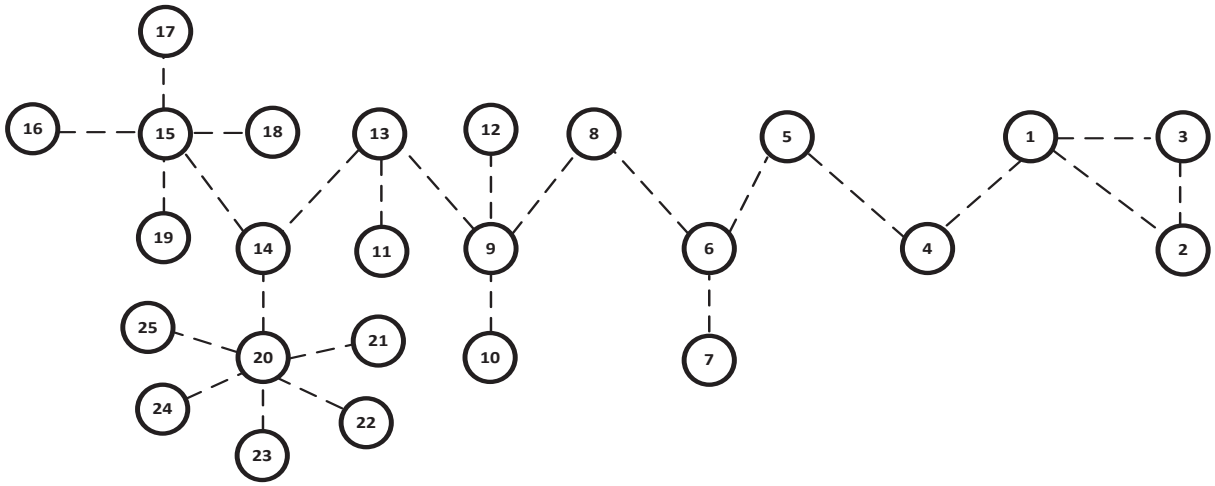

Figure 33: Exemple de topologie du réseau

Nous pouvons observer le principe de fonctionnement sur la figure 32 en fonction de la topologie du réseau figure 33, où on utilise 3 canaux dans le réseau, les slots pairs sont réservés pour les émissions et réceptions des beacons, les slots impairs sont réservés pour les transmissions des données et leurs acquittements (DATA et ACK). Les flèches vertes représentent les beacons, les rouges représentent les données (DATA) et, les bleus représentent les acquittements (ACK). Les flèches orientées vers le haut représentent les émissions (nœuds émetteurs) et celles vers le bas représentent les réceptions (nœuds récepteurs). Les flèches en pointillé représentent les trames émises et non reçues, le numéro du nœud au bout de flèche en pointillé indique qu'il n'a pas reçu la trame émise par le nœud noté au début de la

flèche, dont il n'y a pas d'acquittement (s'il s'agissait d'une trame de données, car les beacons ne sont jamais acquittés). Les nœuds qui n'ont pas reçu les trames, sont sur la figure notés avec des indices indiquant l'autre canal sur lequel ils se trouvent.

On peut aussi voir sur la figure 32, des nœuds qui, se trouvant dans des zones d'interférences différentes, peuvent émettre sur le même canal pendant le même slot de temps (réutilisation spatiale), les émissions entre les paires des nœuds 10 vers 9 et 3 vers 1 qui se déroulent pendant le même slot de temps (t1) sur le canal 3 (C3).

Par exemple, en observant la figure 32, à t0 (slot pair de beacons), le nœud 1 a émis un beacon sur le canal 1, les nœuds 2, 3 et 4 l'ont reçu correctement. Les nœuds qui n'ont pas reçu le beacon peuvent être en émission ou en réception d'un beacon sur un autre canal pendant ce même slot de temps ou trop éloignés.

Ce principe s'observe sur le slot t1 où suivent les données et leurs acquittements : il y a deux émetteurs sur le canal 2 : les nœuds 5 et 18. La trame du nœud 5 n'ayant pas été reçue par 6, elle n'est donc pas acquittée puisque le nœud 6 est en émission sur le canal 1(le nœud 6 est en indice 1, qui indique que le nœud 6 est sur le canal 1 pendant ce slot de temps) pendant ce même slot de temps.

Le nœud 5 procédera à d'autres tentatives pour envoyer sa trame au nœud 6 (jusqu'à un crédit de répétition épuisé, par exemple de 5). S'il reçoit un ACK de cette trame data, il enregistre alors ce canal à succès (par exemple pendant le slot t3 sur le canal 1), et lors de sa prochaine émission avec le même nœud 6, il sélectionnera d'abord ce canal plutôt que de tirer au sort un autre canal d'émission. Comme on peut le voir à t3, le nœud 5 tire au sort le canal 1 et émet sa trame au nœud 6, cette trame a été bien reçue puis est enfin acquittée. Le nœud 5 enregistre alors le canal 1 comme canal de succès et pour sa prochaine communication vers le nœud 6, il commencera d'abord par utiliser le canal 1.

Si le nœud 6 n'a pas de données à émettre, il sélectionne alors le canal 1 comme canal de réception (car un récepteur reste un certain temps sur son dernier canal utilisé) et passera donc quelques slots de temps sur ce canal pour recevoir des éventuelles émissions du nœud 5 ou d'autres nœuds.

Après avoir présenté notre proposition de méthode d'accès de base, nous allons détailler les diverses options d'améliorations et d'optimisation de paramétrages possibles que nous avons imaginées.

2.4.3 Options d'améliorations et paramétrages optimaux

2.4.3.1 Rémanence sur le précédent canal de réception

Un paramètre important dont il faut trouver la valeur la plus adéquate, est la période de persistance ou rémanence qu'un nœud va passer sur le dernier canal de réception *CR*.

Si le nœud de réception reste longtemps sur le même canal, alors à un moment donné, il y aura un nombre important d'émetteurs qui vont partager le même canal et on fait face à nouveau à une transmission mono-canal avec de nombreuses collisions, donc sans succès de transmission, on perd ainsi l'intérêt du multi-canal vers lequel on s'oriente.

Si le délai de rémanence sur le dernier canal de réception est petit, par exemple un nœud ne reste qu'un seul slot sur le canal de réception, ceci réduit la probabilité pour que l'émetteur qui a enregistré le canal de réception du nœud récepteur réussisse à le joindre lors de sa prochaine émission, puisque ce délai étant trop court, le nœud récepteur peut rapidement commuter sur un autre canal. Néanmoins, ce court délai de rémanence permet aux nœuds lors du slot suivant d'émettre ou recevoir le beacon des

autres nœuds, il est aussi possible d'éviter que plusieurs émetteurs partagent le même canal en envoyant simultanément leurs trames de données.

Mais, si le délai de rémanence sur le dernier canal de réception n'est pas très grand, par exemple lorsqu'un nœud reste juste 2 ou 3 slots sur le précédent canal de réception, ceci permettra à l'émetteur qui a déjà enregistré le canal de réception de ce nœud de le joindre lors de sa prochaine émission, même si l'émetteur se trouve sur un autre canal au premier slot, il a la possibilité de commuter sur le canal du récepteur pendant le seconde ou le troisième slot de temps. Un nombre de slot de temps adéquat permet aussi d'éviter à plusieurs émetteurs de partager le même canal.

2.4.3.2 Historique des succès

Nous avons constaté que le tirage aléatoire de canal d'émission *CE* et de réception *CR* engendre des retards pour que l'émetteur et le récepteur se retrouvent sur le même canal au même moment, surtout lorsqu'on a un nombre important de canaux. Afin d'y remédier en partie, nous considérons qu'à chaque fois qu'une transmission a réussie (succès), l'émetteur doit enregistrer ce canal des trames transmises avec succès pour ce nœud (comme prochain canal d'émission *CE*). Il en va de même pour le récepteur des trames qui va sélectionner le dernier canal de réception *CR* sur lequel il passera quelques slots de temps. Ainsi, lors des prochaines émissions vers ce nœud, l'émetteur privilégie le dernier canal d'émission des trames émises avec succès avant toute autre tentative sur un autre canal tiré au sort. On peut le constater sur la figure 35 et en fonction de la topologie du réseau représentée sur la figure 34, sur laquelle (Figure 35) les nœuds A et C sont des émetteurs, alors que B et D sont des récepteurs. Chacun de ces émetteurs, après avoir choisi aléatoirement un slot et un canal, a réussi à émettre ses trames avec succès sur le canal sélectionné vers la destination prévue (A vers B et C vers D). Ainsi, chaque émetteur enregistre le canal de succès ; A enregistre le canal C2 pour B et C enregistre le canal C3 pour D. Le nœud B se met en réception sur le canal C2 et le nœud D sélectionne le canal C3 comme canal de réception, dans cet exemple illustré par la figure 35, ainsi au niveau de chaque nœud, on peut avoir la table de voisinage représentée par la table 4.

Table 4: Caractéristique de la table voisinage de A

ND_ID	*CANAL*
B	*C2*
F	*Φ*
R	*C1*
C	*Φ*

ND_ID : Identifiant du nœud voisin
*C*ANAL : dernier canal à succès du nœud voisin
Φ : aucune émission vers ce voisin

Bien que l'historique des canaux utilisés avec succès présente plusieurs avantages, ce dernier va également susciter quelques inconvénients et la question suivante : combien de temps ou de slots faut-il garder en mémoire les canaux de succès pour chaque récepteur avec qui on a réussi à transmettre des données ?
Mémorisation longue : ceci permet de garder en mémoire le plus longtemps possible le canal de succès vers un nœud récepteur, et va ainsi permettre à l'émetteur de libérer rapidement sa file d'attente des trames des données, par conséquent cela réduit le délai des transmissions et donne la possibilité de gagner en débit, c'est l'un des besoins les plus importants en réseau maillé multi-saut.

Mémorisation courte : ceci a l'inconvénient de devoir chercher à nouveau un récepteur qu'on vient de joindre récemment, il faut parfois essayer tous les canaux de réceptions disponibles dans le pire des cas. Ceci introduit ici aussi un délai important de transmission.

Dans les deux cas de mémorisation que nous avons évoqués, il s'agit de trouver le meilleur compromis et donc le paramètre le plus juste possible pour qu'une transmission multi-canal multi-saut sans RDV soit effectuée convenablement, c'est-à-dire, libérer à un moment donné certains canaux de succès abandonnés par leurs récepteurs correspondants et, ne garder que les canaux de succès dont les nœuds (récepteurs) correspondants persistent encore.

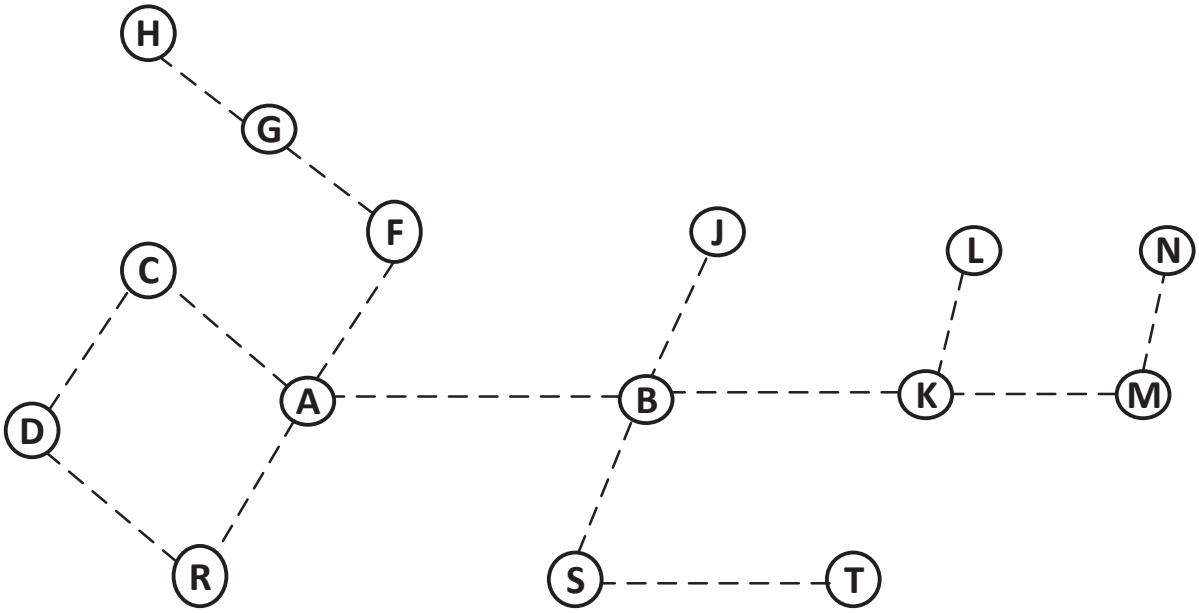

Figure 34: Topologie associée au chronogramme de la figure 15

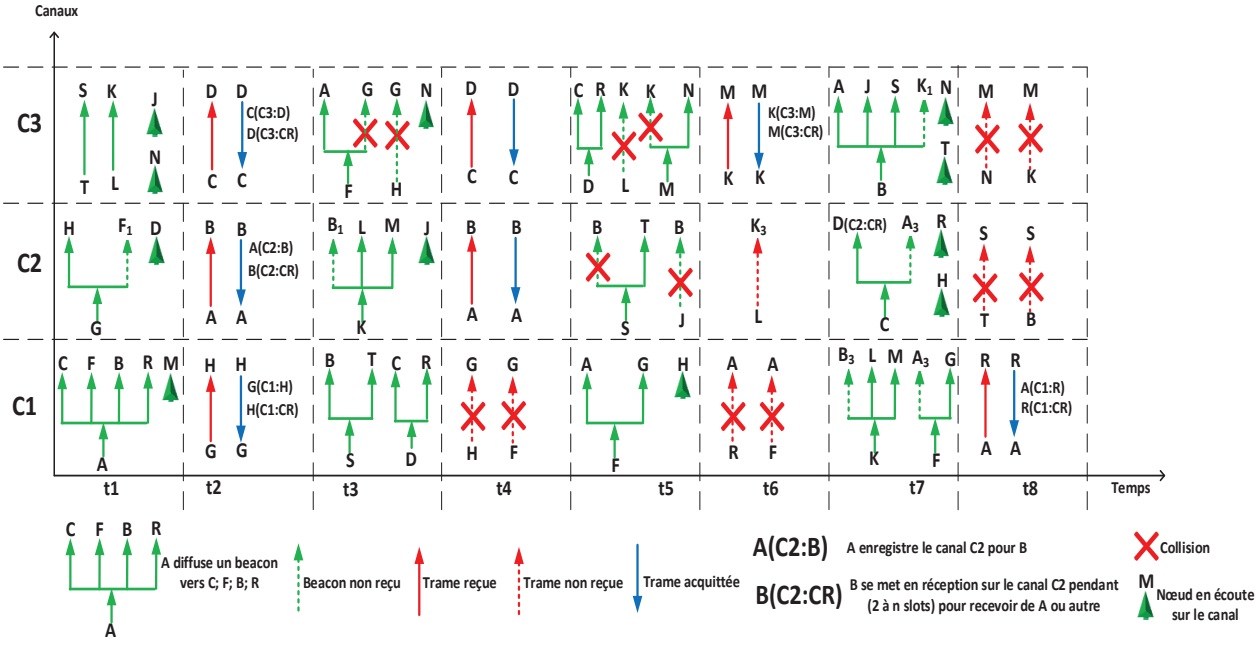

Figure 35: Chronogramme présentant le principe lié aux canaux à succès

Dans tous les cas, nous supposons qu'un nœud quelconque dans le réseau, il est dans l'une des états suivants pendant chaque slot de temps : en émission, réception, écoute, ou éteint.

2.4.3.3 Canal dédié SiSP

La seconde version de la méthode d'accès multi-canal que nous avons imaginé est de considérer un canal dédié uniquement pour les beacons, parmi les canaux disponibles. Le principe de cette méthode est le suivant : si un nœud a des données à émettre, il tire au sort un canal d'émission *CE (parmi les canaux de données, donc tous sauf le canal dédié)* puis émet sa trame. S'il n'a pas des données à

émettre, lors du prochain slot, il va soit émettre ou se mettre en réception d'un beacon sur le canal dédié aux beacons, et ensuite tire aléatoirement un canal de réception *CR* et se met en réception pour recevoir des éventuelles trames des données, sauf s'il y'a un dernier canal de succès précédemment enregistré. Ce principe est présenté en Figure 37 (en fonction de la topologie du réseau Figure.36) sur laquelle trois canaux sont utilisés en tout : le canal C1 est dédié uniquement pour les échanges des trames beacons et les deux canaux C2 et C3 pour les E/R de données et acquittement.

Bien évidemment certains problèmes peuvent apparaitre avec cette méthode :

1. Un grand nombre des canaux dans le réseau rend leur gestion difficile, surtout quand il s'agit d'un choix aléatoire du canal d'émission et de réception. Le choix d'un canal d'émission/réception de façon aléatoire introduit un délai important lorsqu'on cherche à joindre un nœud, puisqu'il faut plusieurs tentatives pour réussir à joindre le destinataire.
2. Le second problème est le suivant : le canal dédié aux beacons va manquer aux E/R de DATA/ACK. Une idée serait qu'il puisse aussi servir pour les transmissions de données afin d'élargir la bande passante disponible puisqu'il est utilisé pour des échanges des petites trames de contrôles moins fréquemment (les beacons).

L'avantage de cette méthode, puisqu'on isole les trames beacons sur un canal dédié, est qu'il permettra de moins encombrer les canaux de données et d'éviter aussi les collisions des trames de données avec des petites trames de contrôles. Ainsi, l'intérêt du canal dédié est d'être sûr que la synchronisation se fait avec tous les nœuds sur le même canal au même moment, les nœuds qui veulent émettre ou recevoir des beacons seront tous sur le même canal.

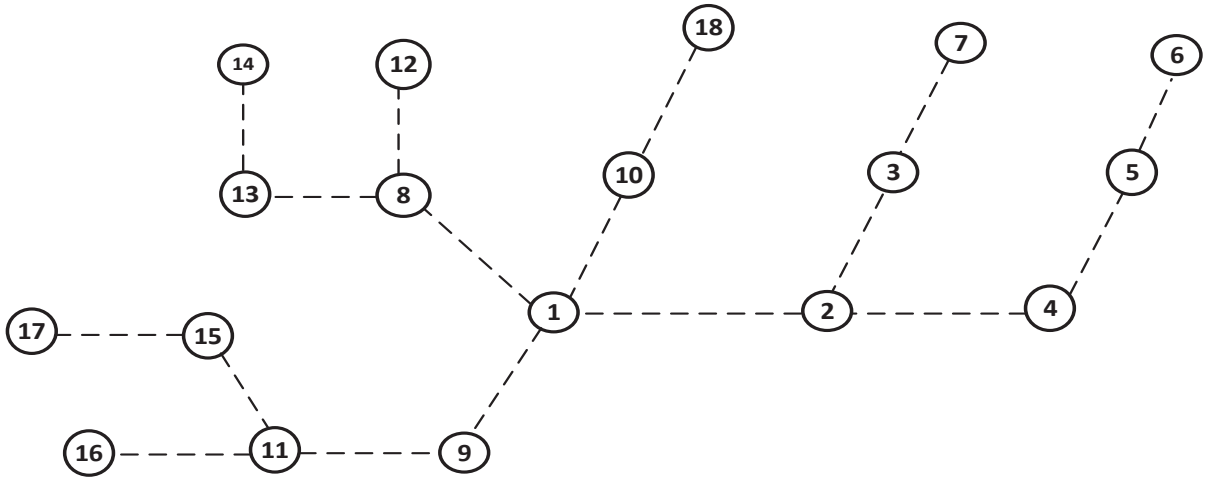

Figure 36 : Topologie du réseau associée au canal dédié aux beacons

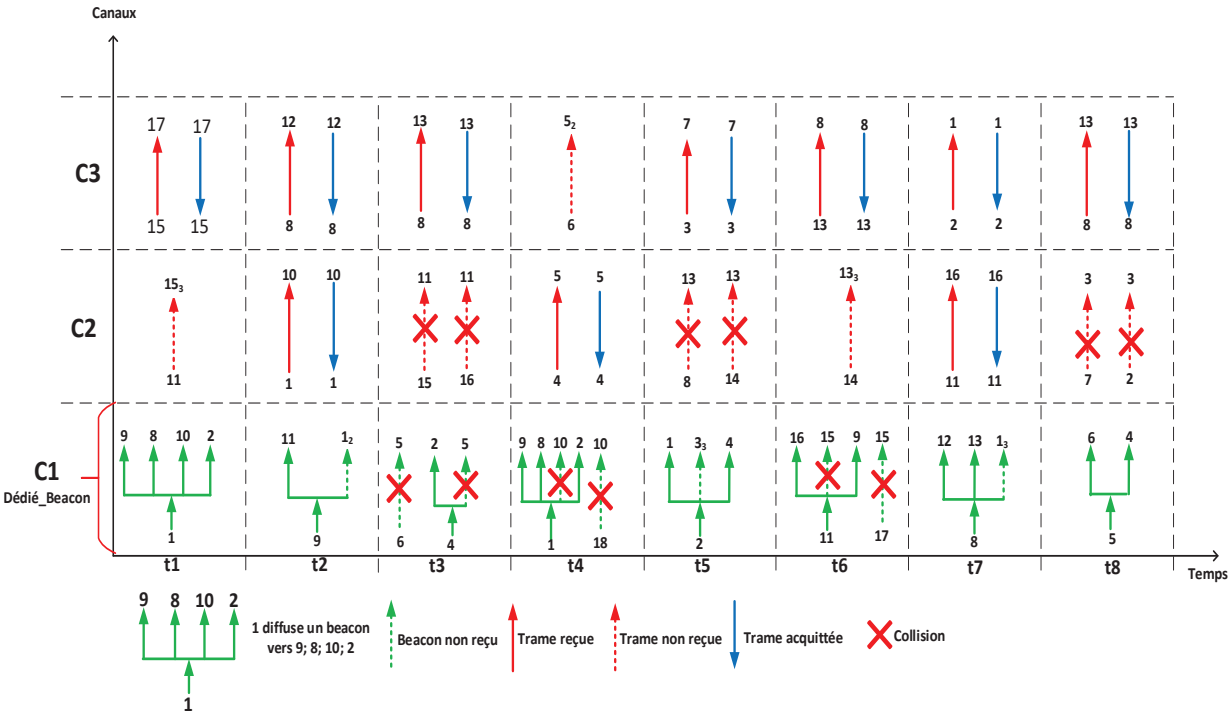

Figure 37 : le principe du canal dédié aux beacons

IL faut s'assurer que chaque nœud émet et reçoit suffisamment de beacons, et peut émettre et recevoir suffisamment de data et ACK. A chaque slot nouveau, le nœud tire au sort pour savoir s'il va traiter des DATA/ACK ou des beacons. Il tire ensuite au sort s'il va émettre ou recevoir.

2.4.3.4 Méthode sans canal dédié et sans beacon

Dans une autre version, nous n'utilisons pas un canal et des slots dédiés aux beacons, il n'y aura pas de trames beacons pour le transport des informations de voisinage, ces informations seront transportées dans des nouveaux champs de deux trames principales des données (DATA) et acquittement (ACK). On peut ainsi réduire la charge en termes de nombre des trames utilisées dans le réseau.

Ainsi, lorsqu'un nœud a une trame à émettre, il tire au sort un slot et un canal puis émet sa trame ; s'il reçoit un ACK de sa trame, alors le nœud émetteur enregistre le canal de succès qu'il vient d'utiliser avec ce nœud récepteur pour sa prochaine émission, le nœud récepteur va utiliser ce canal comme son canal de réception où il restera au moins deux slots. Lorsqu'un nœud n'a pas émis ni reçu une trame pendant un certain slot de temps (3 slots par exemple), il va diffuser une trame sans charge utile, et vers n'importe quel nœud, ceci permet à ce nœud de se synchroniser aux autres nœuds du réseau. Puisque les nœuds émettent et reçoivent des données et acquittement, à un certains moments les trafics vont se propager dans le réseau permettant l'échange des informations de voisinage entre les nœuds. Il faut juste assurer qu'il y a assez de trafic DATA/ACK sur tous les canaux en forçant les nœuds n'ayant rien à dire à diffuser une trame vide de données, mais comportant au moins les infos de SiSP et de voisinage comme dans les versions précédentes.

2.5 Conclusion

Dans ce chapitre, nous avons présenté notre proposition de nouvelle méthode d'accès multi-canal multi-saut sans RDV qui utilise un protocole de synchronisation simple (SiSP) car basée sur l'Aloha slotté. L'intérêt de cette méthode est l'absence de RDV qui pose des problèmes de RDV quand il s'agit de réseaux multi-saut. Dans les méthodes d'accès multi-canal existantes étudiées mono-saut, pour que deux nœuds communiquent, ils doivent généralement établir un RDV au préalable et ceci avant toute communication, tâche qui n'est pas facile à réaliser dans les réseaux maillés sans fil et multi-saut, puisqu'il faut nécessairement négocier et propager le RDV au-delà d'un saut afin d'éviter les collisions avec des nœuds ne connaissant pas le RDV pris à plus qu'un saut. On risque d'avoir alors une communication perturbée. L'intérêt du multi-canal est alors remis en cause. Avec la méthode que nous avons proposée, il est possible qu'on arrive à résoudre les défis des méthodes existantes étudiées, mais, notre MAC présente l'inconvénient qu'il faut évaluer le délai le meilleur qu'un nœud doit rester en réception sur un canal, et ceci dans un compromis adéquat entre trop de collision sur un même canal où le récepteur reste longtemps, et temps de rencontre entre émetteur et récepteur trop long. Le prototypage que nous présenterons au chapitre suivant visera à choisir ce type de paramétrage optimal.

Chapitre 3

Implémentation du protocole MAC multi-canal sans RDV et analyse de performance

3.1 Introduction

Pour effectuer l'analyse de performance de la méthode d'accès multi-canal sans rendez-vous (RDV) que nous proposons, nous avons tout d'abord réalisé un prototypage ou testbed en utilisant des nœuds WiNo, avec comme référence initiale, la méthode d'accès mono-canal Aloha slotté (*Slotted Aloha*) [21] [22] [59]. Cette MAC basique va nous servir de référence et nous aidera à construire petit à petit notre MAC, en comparant les différentes versions. Pour le découpage en slots, ici, les slots de temps sont indiqués à tous les autres nœuds par un nœud synchronisateur, qui émet chaque seconde un *beacon* (balise pour indiquer le début du slot de temps) qui ne contient que le type de message de ce dernier avec une puissance plus forte que celle utilisée par les autres nœuds pour les émissions des données et acquittements. Ceci permet de synchroniser facilement (ceci n'étant pas le but de la thèse et pouvant être réalisé ensuite facilement avec SiSP [62]) tous les nœuds du réseau, même ceux assez éloignés. Afin d'évaluer la portée entre deux nœuds normaux, il faut trouver la juste valeur de la puissance du signal. Nous avons ensuite procédé de la même façon que pour cet accès mono-canal pour aboutir au prototypage de la méthode d'accès multi-canal Aloha slotté.

3.2 Présentation des nœuds WiNo utilisé pour l'implémentation et testbed de la couche MAC multi-canal multi-saut sans RDV

La raison pour laquelle nous avons choisi d'utiliser le nœud WiNo (*Wireless Node*), est simplement que WiNo est une plate-forme matérielle ouverte (*Open Hardware Platform*), ceci permet une grande polyvalence sur le matériel; Il est très simple, par exemple, de changer la couche physique d'un WiNo: il s'agit seulement de remplacer le *tranceiver* et la bibliothèque associé (Table 5), ce qui simplifie le processus d'ingénierie des protocoles tels que : le prototypage rapide et l'évaluation pragmatique, en déploiement réel, des performances des protocoles de niveau MAC, pour le réseau de collecte dans l'Internet des Objets et les réseaux de capteurs sans fil. S'appuyant sur l'environnement Arduino et couplés au logiciel OpenWiNo [65], les nœuds WiNo permettent le déploiement d'un réseau maillé (mesh) auto-organisé et également le prototypage rapide de systèmes complets tels que des objets connectés, incluant capteurs, actionneurs, protocoles de collecte, etc. bien au-delà des aspects purement réseaux. L'implémentation réaliste permet le prototypage de solutions suffisamment intégrées permettant la preuve de concept et le test du prototype de l'objet connecté sous l'angle des usages, par de vrais utilisateurs. Les WiNos ont été développés de façon à offrir un accès de bas niveau pour un développeur exigeant qui souhaite non seulement maîtriser précisément les temps d'accès au médium, la mise en veille et le réveil des nœuds, mais aussi les temps CPU et la gestion de la mémoire restreinte. Que ce soit dans le but de piloter des politiques drastiques d'économie d'énergie

ou pour le respect de contraintes temps réel, une telle maîtrise de l'ensemble des composants du nœud est nécessaire ; WiNo est une plate-forme matérielle candidate à l'accueil de protocoles à fortes contraintes temporelles visant plusieurs mois de fonctionnement avec deux piles AAA [66].

Table 5 : Caractéristique du nœud WiNo

	WiNoRF22	**TeensyWiNo**	**DecaWiNo**	**WiNoVW**
CPU/RAM/Flash	ARM Cortex M4 (32bit) 72MHz, 64kB RAM, 256kB Flash (PJRC Teensy 3.1)			
Transceiver (*librairie Arduino*)	HopeRF RFM22b : 200-900MHz, 1-125kbps, GFSK/FSK/OOK, +20dBm *RadioHead*		DWM1000 : UWB IEEE 802.15.4 *DecaDuino*	Variés *VirtualWire*
Capteurs	température, luminosité	idem + pression, accélération, compas, gyro	température, luminosité	
Usage	WSN, IoT		IoT avec Ranging, localisation en intérieur	Communication très bas débit sur médiums non conventionnels
Disponibilité	DIY	snootlab.com	DIY (*Do It Yourself*, à assembler soi-même)	

La figure 38 représente certains types des nœuds *WiNos* : *WiNoRF22* (a), *TeensyWiNo* (b) tous le deux sont basés sur HopeRF RFM22b radio et *DecaWiNo* (c). Les *WiNos* sont intégrés dans l'écosystème *Arduino*, ce qui permet aux chercheurs et développeurs d'intégrer facilement des composants matérielles et/ou logicielles (capteurs et actionneurs, algorithmes avancés de traitement, dispositifs d'interaction…) et prototyper des solutions très complètes sur le plan applicatif, permettant d'aller jusqu'à des tests d'usage.

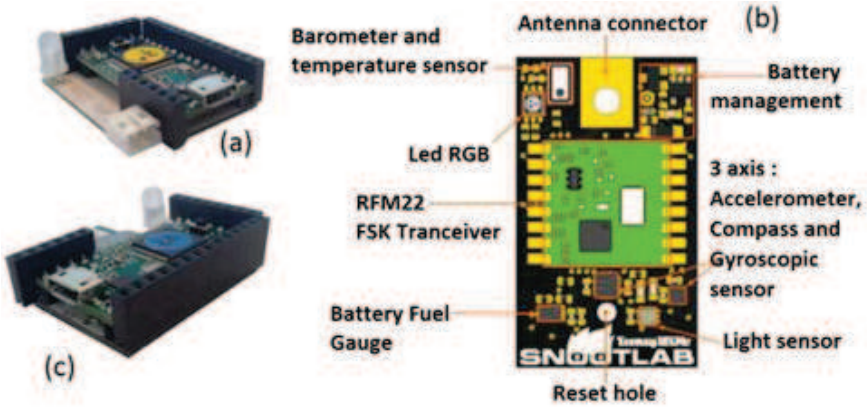

Figure 38: Image des nœuds WiNos

3.3 Métriques pour l'étude des performances

Nous étudions les performances de notre méthode d'accès multi-canal en considérant trois métriques essentielles : le nombre des trames reçues, le nombre de trames perdues et le taux d'erreur trame qui sont représentés en ordonnée sur nos courbes, par rapport à la charge du trafic réseau qu'on augmente progressivement, ce que l'on peut voir en abscisse. Ces mêmes paramètres sont évalués en mono-canal, pour que nous puissions faire la comparaison avec l'accès multi-canal. Le but est aussi d'observer le nombre des sauts séparant un récepteur et un émetteur et évaluer lorsque qu'une collision se produit.

Les données sont recueillies en temps réel et selon les équations suivantes :

$$LOST \leftarrow LOST + SQNT - RXSQN - 1 \tag{9}$$

Où *LOST* est le nombre des trames perdues, *SQNT* est le numéro de séquence de la trame qui vient d'être émise et donc reçue, et RXSQN est le numéro de séquence de la trame précédemment reçue. Ce principe est illustré par le diagramme de séquence suivant (Figure 39).

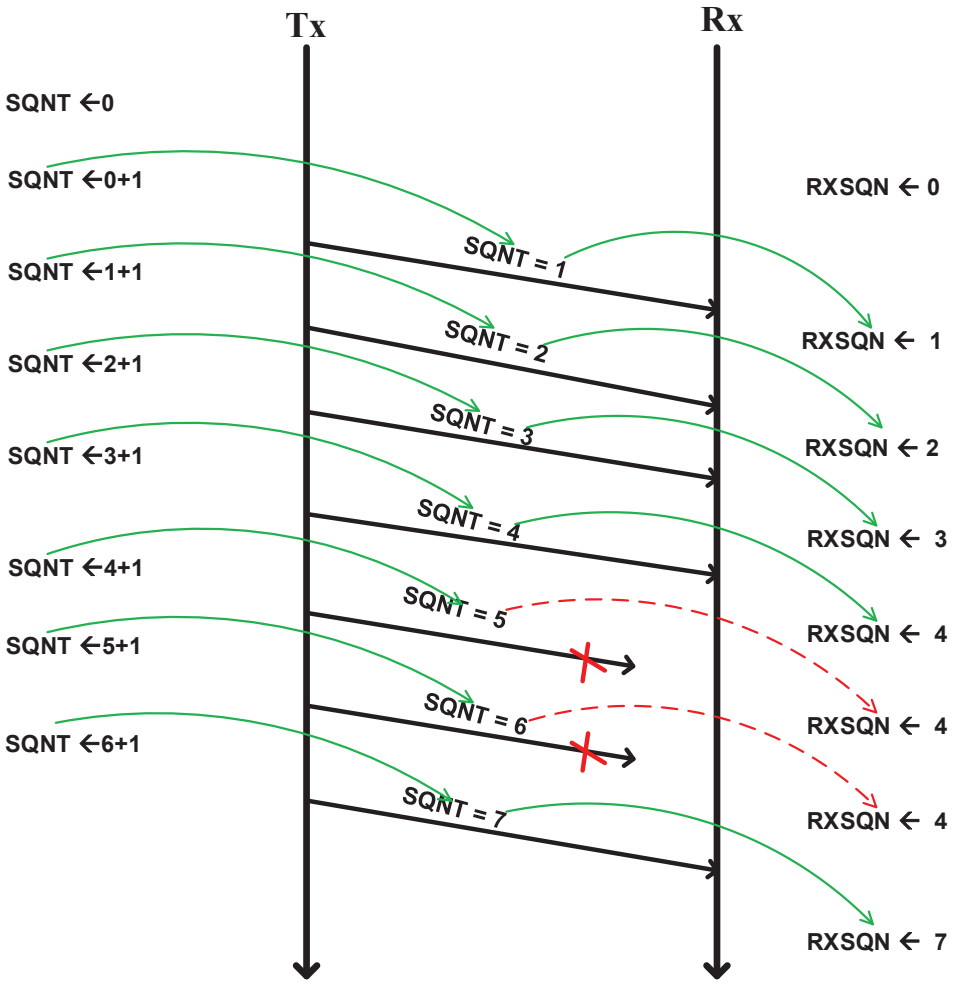

Figure 39 : Diagramme de séquences précisant le nombre des trames perdues

Sur ce diagramme de séquence, on voit que l'émetteur TX a envoyé sept trames. Le récepteur RX reçoit les trames 1 à 4, mais n'a pas reçu les trames 5 et 6. Il reçoit la trame suivante portant le numéro de séquence SQNT = 7. En utilisant l'équation (1) :

$LOST \leftarrow 0 + 7 - 4 - 1 = 2$, il identifie que deux trames sont perdues.

FER (Frame Error Rate) est le taux d'erreur trames, *REC* étant le nombre des trames reçues.

Lorsqu'un nœud reçoit des trames de données de différents émetteurs, dans ce cas précis, chaque variable illustrant sa métrique est divisée par le nombre des nœuds à partir desquels le récepteur a reçu des trames, ainsi, nous calculons les valeurs moyennes des REC, LOST, FER :

$$REC\_Moyen = \frac{REC}{Nombres\_des\_émetteurs} \quad (10)$$

$$LOST\_Moyen = \frac{LOST}{Nombres\_des\_émetteurs} \quad (11)$$

$$FER\_Moyen = \frac{LOST\_Moyen}{LOST\_Moyen + REC\_Moyen} \quad (12)$$

3.4 L'accès mono-canal

Nous avons en 1er implémenté un accès mono-canal entre Tx et Rx.

3.4.1 Les automates mono-canal de transmissions des trames de données

3.4.1.1 Automate mono-canal du nœud émetteur (Tx)

L'émetteur Tx (automate Tx, Figure 40) met sa radio en réception et attend de recevoir le beacon. A chaque fois, après avoir reçu le beacon, il décrémente Nb_BC (nombre de beacons reçus) jusqu'à atteindre la valeur zéro, si précédemment une collision s'est produite pour pouvoir rentrer d'émettre une trame (procédure de backoff), sinon il émet directement sa trame après réception du *beacon*. Dans le cas où le nombre de beacons n'a pas atteint zéro, il laisse toujours la radio en mode réception et attend de recevoir le prochain beacon. Après avoir émis sa trame de données, un chien de garde (WatchDog) est armé pour borner le temps pour recevoir l'acquittement (ACK) de la trame émise. La radio est mise en réception en attendant l'acquittement. S'il reçoit un acquittement dont il n'est pas destinataire, ou s'il reçoit une autre trame type autre que ACK, alors il la néglige (il ne la traite pas et l'absorbe), et reste donc toujours en attente d'ACK. Si le chien de garde expire et qu'il n'a pas reçu l'ACK avant, alors, il affiche « échec », ce qui veut dire sans doute que la trame de données émise n'a

pas été reçue à cause d'une possible collision (il est aussi possible que ça soit l'ACK qui est perdu, mais en moindre mesure car de taille plus réduite). Il tire alors un nombre de slots (beacons) aléatoire et met finalement sa radio en mode réception. Les numéros de séquences sont initialisés à 0, tandis que Nb_BC (nombre de beacon) est tiré aléatoirement en utilisant différents intervalles (selon les tests pour faire varier le trafic demandé) de (1, 5) ; (1, 10) ; (1, 20) ; (1, 30) ; (1, 40), il est donc initialisé à 1.

Par contre, si l'émetteur Tx reçoit un acquittement qui porte bien son adresse que nous appelons "NODE ADRESS" et avec le numéro de séquence de la trame qu'il vient d'envoyer, alors il affiche que la trame est bien reçue et Nb_BC (nombre de beacon) sera réinitialiser à 1.

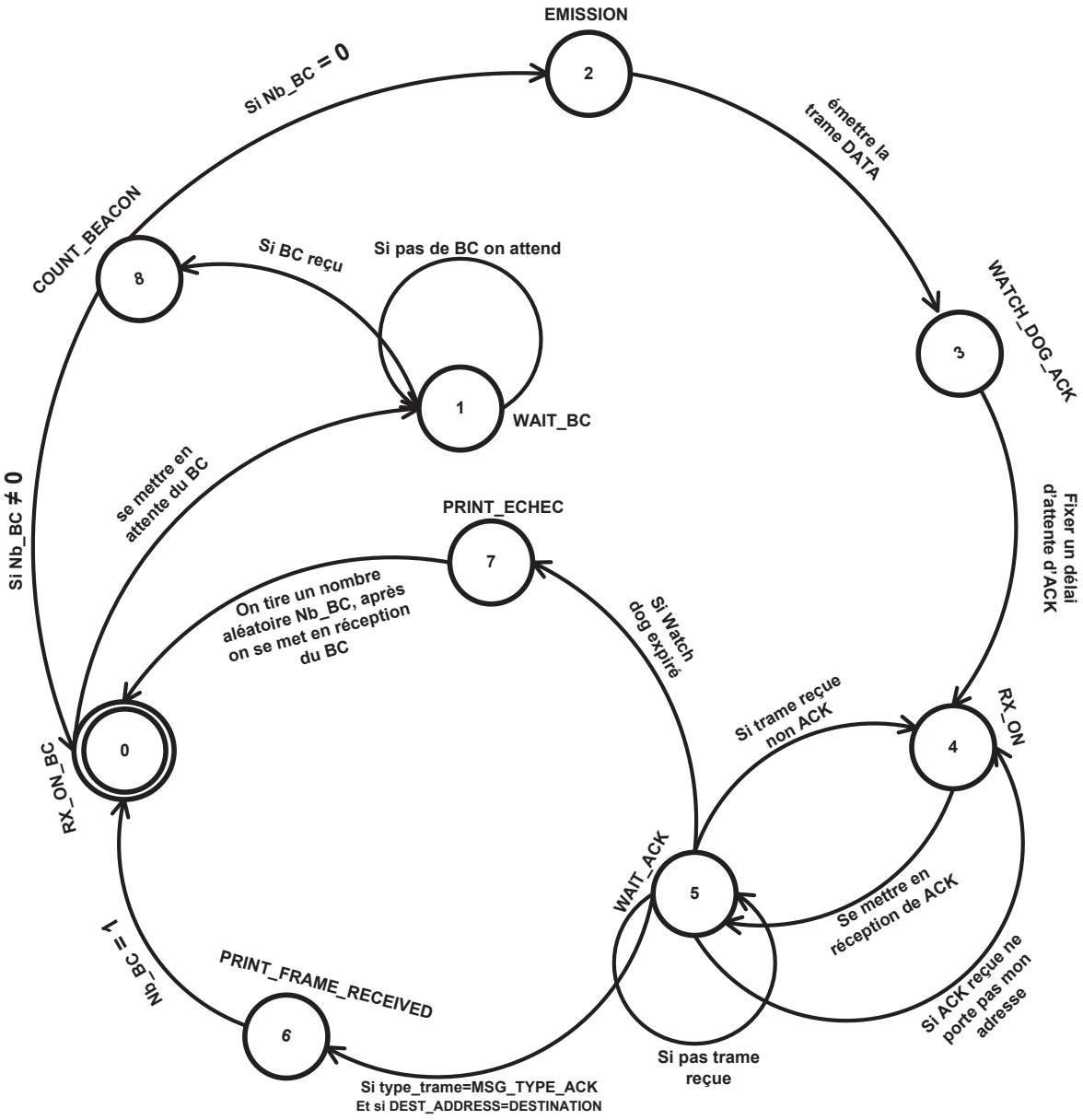

Figure 40 : Automate mono-canal du nœud émetteur (Tx)

3.4.1.2 Automate mono-canal du nœud récepteur (Rx)

Le récepteur Rx (automate Rx, Figure 41) met sa radio en réception et attend de recevoir le beacon. Après avoir reçu le beacon, il met sa radio en réception et vérifie s'il y a un message disponible à recevoir. S'il reçoit une trame dont le type de message n'est pas le type « Data », alors il affiche "pas de trame de données". Si par contre la trame reçue est de type « data » et porte l'adresse de destination du récepteur Rx, alors il affiche la trame reçue et, renvoie une trame d'acquittement ACK à l'émetteur de la trame reçue qui contient le type de message (type ACK), l'adresse de destination qui est effectivement l'adresse source de la trame de data, l'adresse source qui est l'adresse de destination de la trame de data et le numéro de séquence de la trame de data reçue. Dans le cas où la trame de données porte une adresse de destination autre que celle du récepteur Rx, alors celui-ci n'affiche que la trame reçue "ne lui est pas destinée".

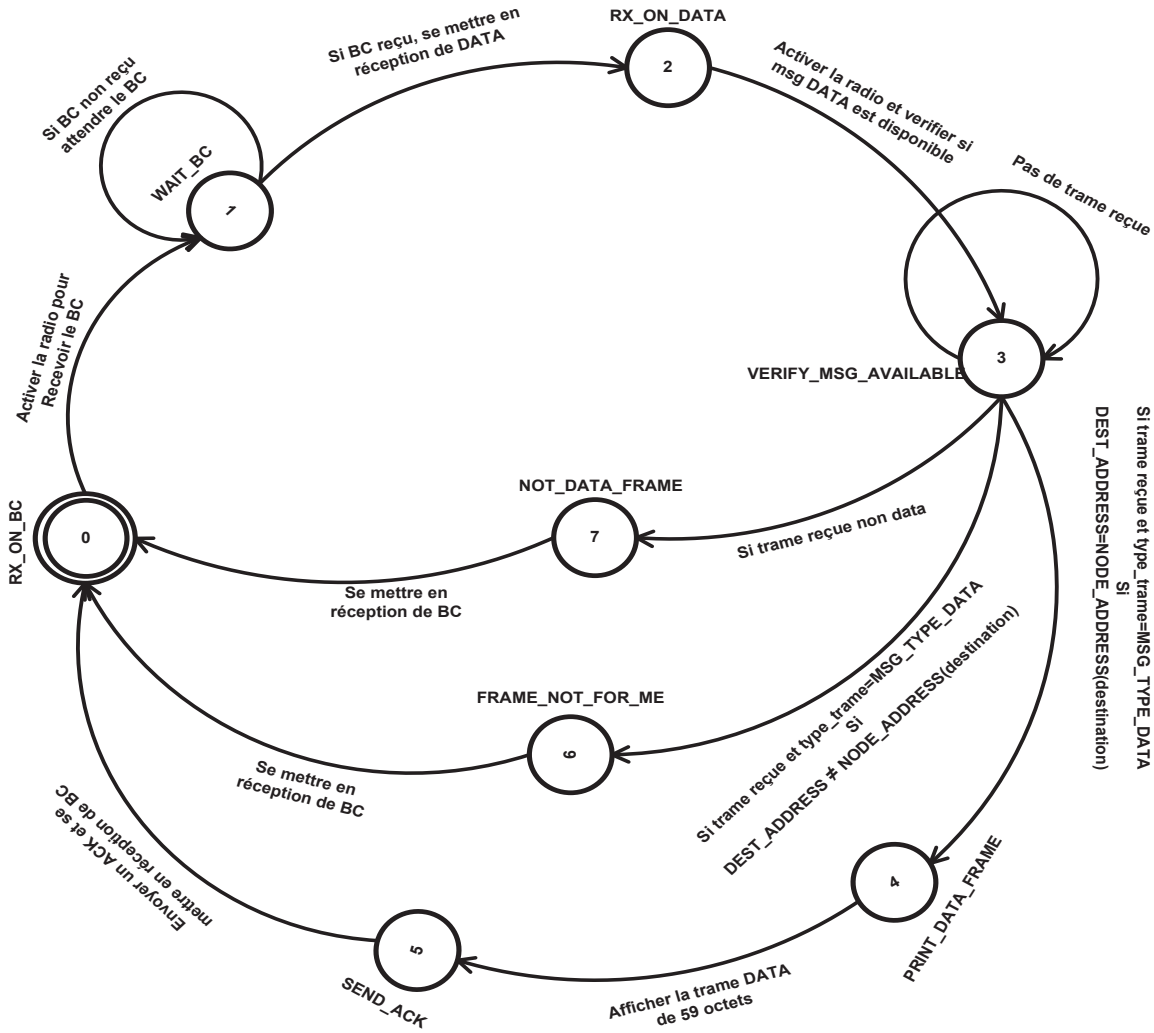

Figure 41 : Automate mono-canal du nœud récepteur (Rx)

3.4.2 Les formats des trames

3.4.2.1 Trame des données (DATA)

Nous utilisons trois types des trames : trame de données (DATA), trame d'acquittement (ACK), et une trame balise (BEACON).

La trame de données que l'on peut l'observer sur la figure 42 a la structure suivante : le premier octet indique le type de message (01 pour les données), les deuxième et troisième octets indiquant respectivement l'adresse de destination et la source de la trame. Le quatrième octet, après l'adresse source, désigne le numéro de séquence de la trame. Les derniers octets sont réservés à la charge utile (limitée ici pour simplifier à 2 octets). La figure ci-dessous illustre le format de la trame de données.

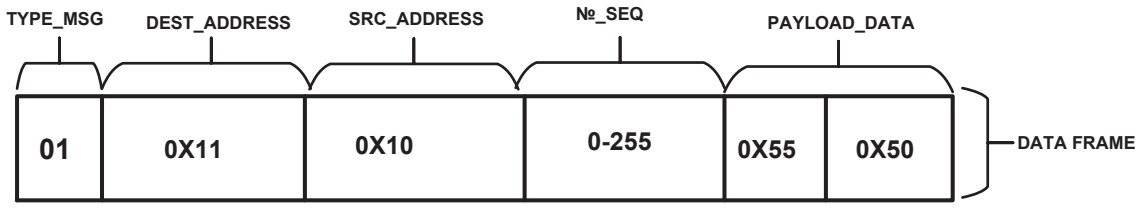

Figure 42 : Format de la trame de données

3.4.2.2 Trame d'acquittement (ACK)

La structure de la trame d'acquittement (Figure 43) diffère de celle de Data par l'absence de charge utile dans la trame ACK et le type de message qui est ici 02. L'ACK acquitte la trame de data reçue à travers les informations suivantes : l'adresse source et destination de la trame et, le numéro de séquence de trame reçue comme on peut le voir sur la figure ci-dessous.

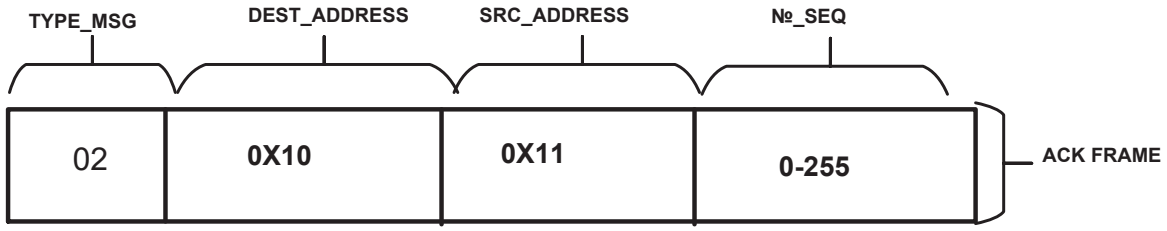

Figure 43 : Format de la trame d'acquittement(ACK)

3.4.2.3 Le beacon

La trame balise (*beacon*) qu'on peut voir sur la figure 44 est la plus petite trame que nous utilisons pour envoyer un signal de coordination (synchronisation) des nœuds. Le *beacon* ne porte comme information que le type de message ici, dans notre cas le 03. Tout nœud en attente recevant le *beacon* envoie une trame de data ou se met en réception d'une trame data. La figure 44 montre le format du *beacon*.

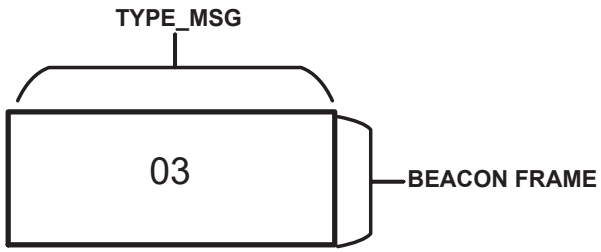

Figure 44: Format de la trame beacon

3.4.3 Les séquences des trames

Nous avons construit plusieurs diagrammes de séquences de trames pour illustrer les différents cas qui peuvent se produire lors des émissions et réceptions des trames en mono-canal/multi-canal.

La figure 45 nous montre qu'après réception d'un *beacon* diffusé par le nœud synchronisateur et bien reçu par tous les nœuds, l'émetteur Tx émet une trame « DATA » contenant le type de message, l'adresse destination, l'adresse source, le numéro de séquence de la trame et la charge utile (payload). Après avoir reçu la trame des données (DATA), le récepteur Rx retourne un acquittement ACK avec l'adresse source, l'adresse destination et le même numéro de séquence que celui de la trame reçue. Un délai d'attente (chien de garde) est lancé pendant lequel l'acquittement devrait être reçu… ainsi de suite.

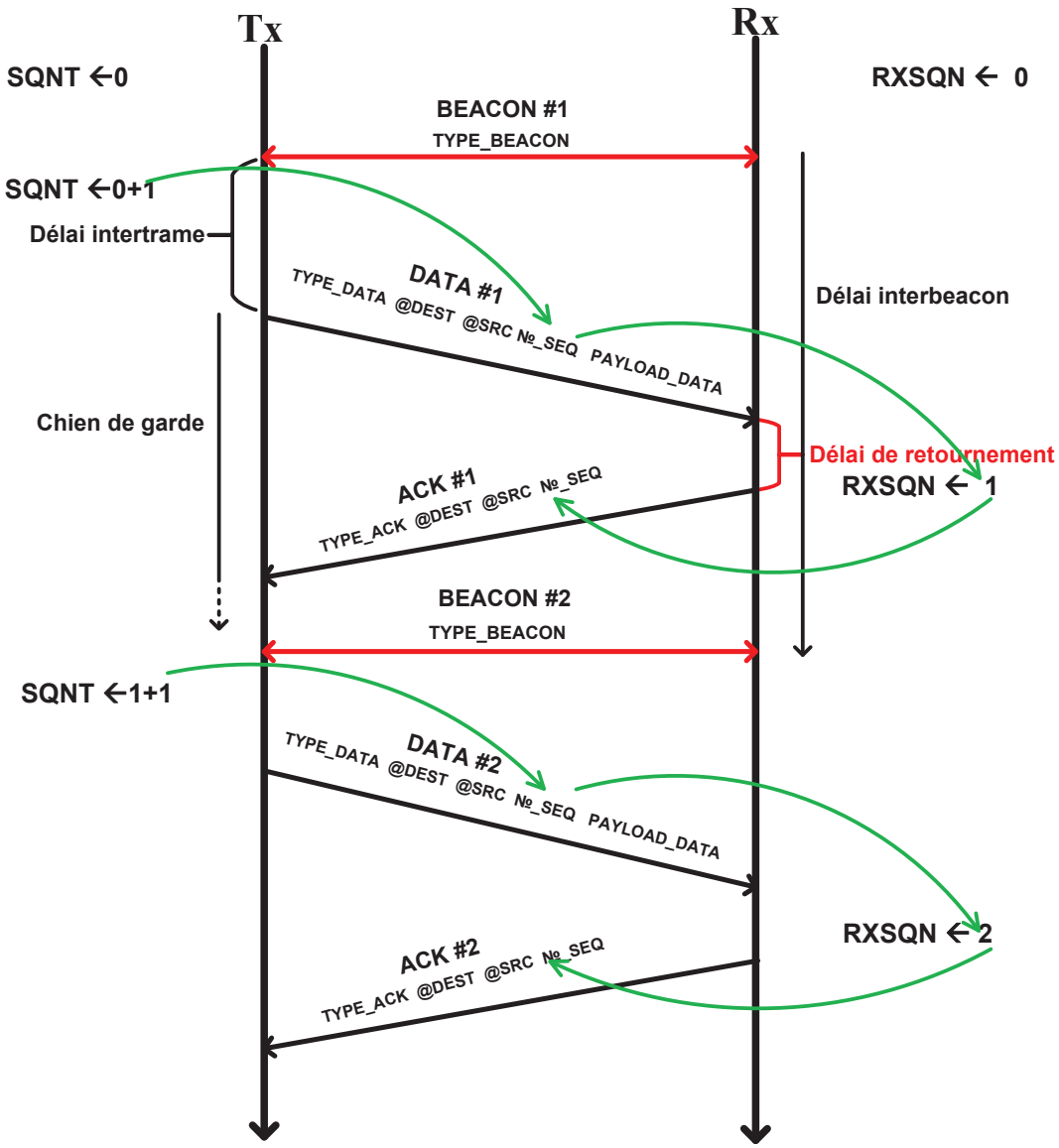

Figure 45: Diagramme de séquence BEACON/DATA/ACK

Il peut arriver que le chien de garde expire avant la réception de l'acquittement (ACK), quand l'ACK met trop de temps à arriver, ou quand surtout l'ACK est perdu (erreur de transmission ou collision), ou quand la trame de data est perdue (erreur de transmission ou collision). Alors, Tx attend le beacon suivant avant d'émettre une trame (la même normalement si la couche 2 LLC était implémentée complètement. Dans cette implémentation, on ne souhaite évaluer que la MAC et pas la couche LLC, donc, il n'y a pas de notion de réémission, mais juste des émissions de trames de données (DATA) qui ont comme rôle de charger le réseau. Ceci peut être observé sur la figure 46.

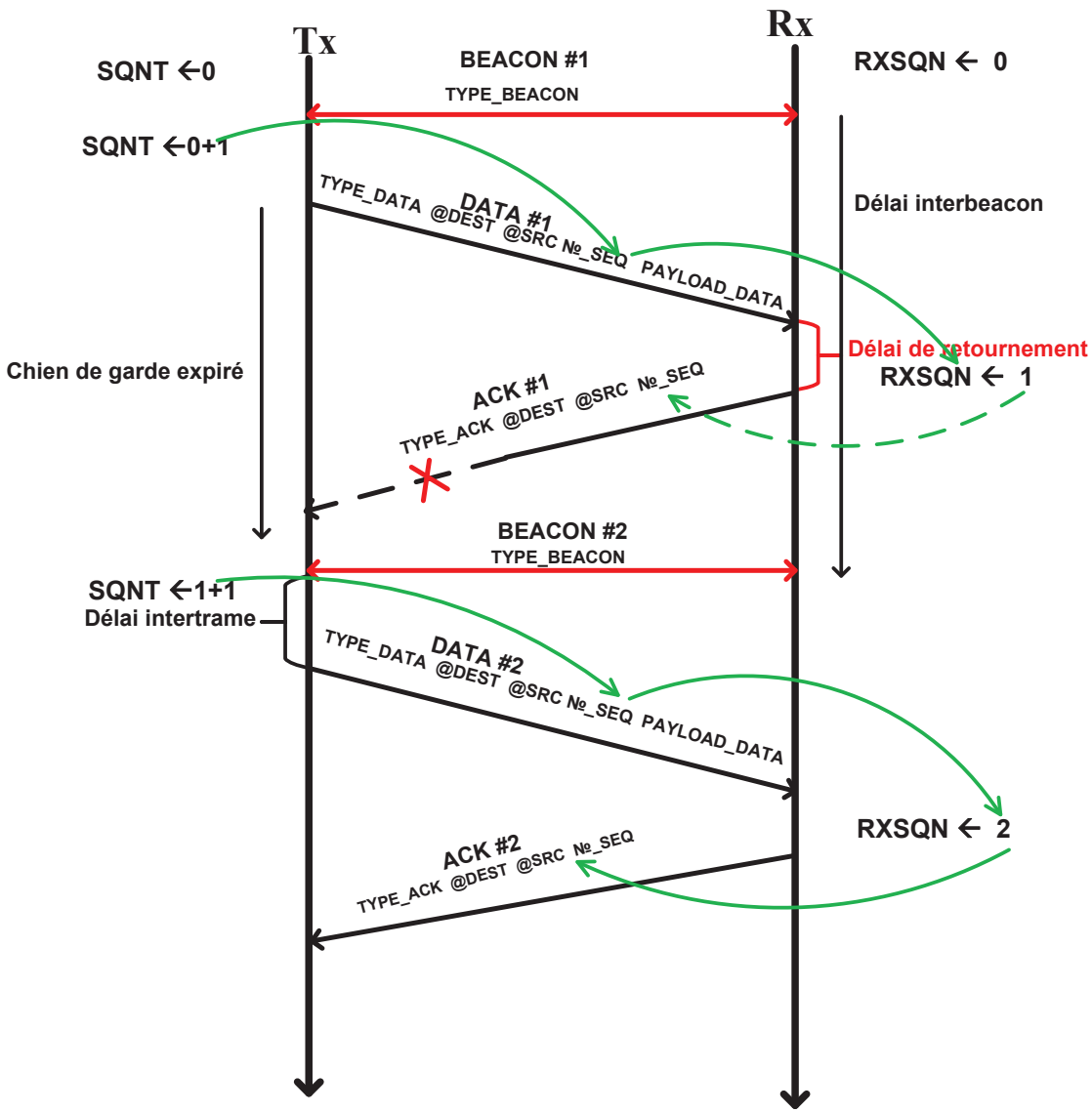

Figure 46 : Diagramme de séquence avec renvoi de la trame Data après expiration du chien de garde (sur perte d'ACK)

L'autre cas qui peut apparaitre pendant les transmissions est quand la trame est perdue complètement. Elle n'est donc pas acquittée, conduisant à l'expiration du chien de garde. Ceci peut être observé sur la figure 47.

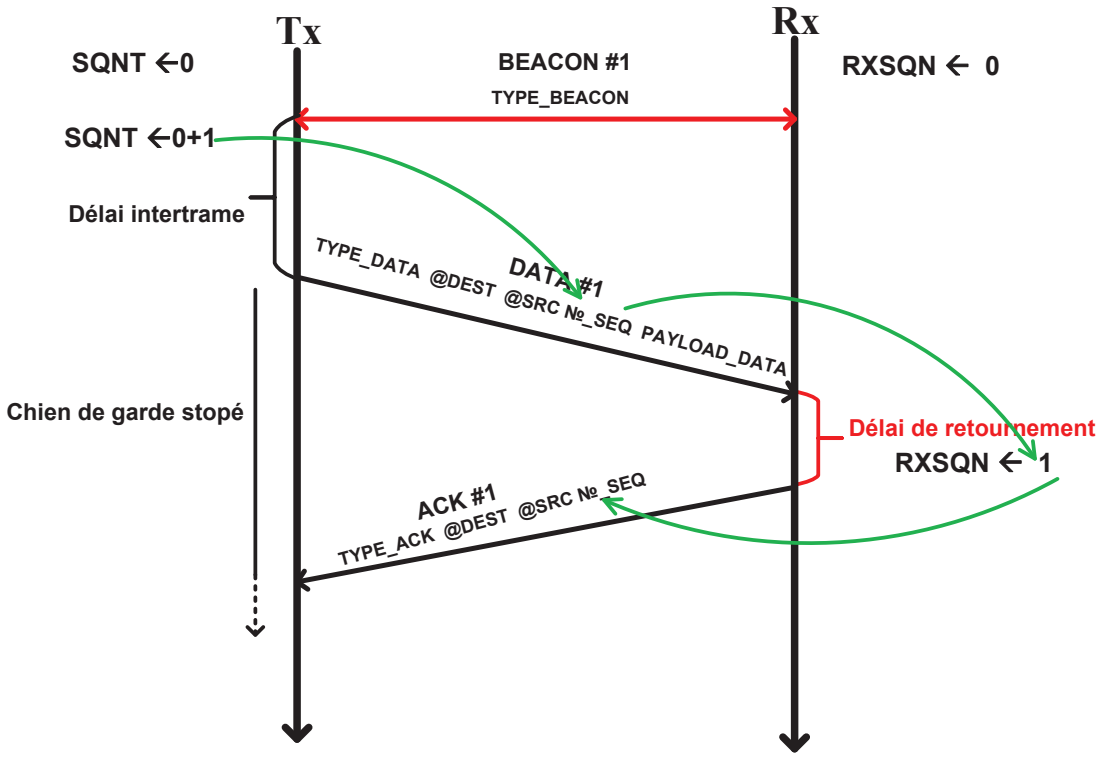

Figure 47 : Diagramme de séquence quand la Data est perdue, il n'y a pas d'acquittement

Autre cas possible qui peut se produire et que l'on peut observer sur la figure 48 est la réception d'un acquittement avec une adresse de destination différente de la trame émise.

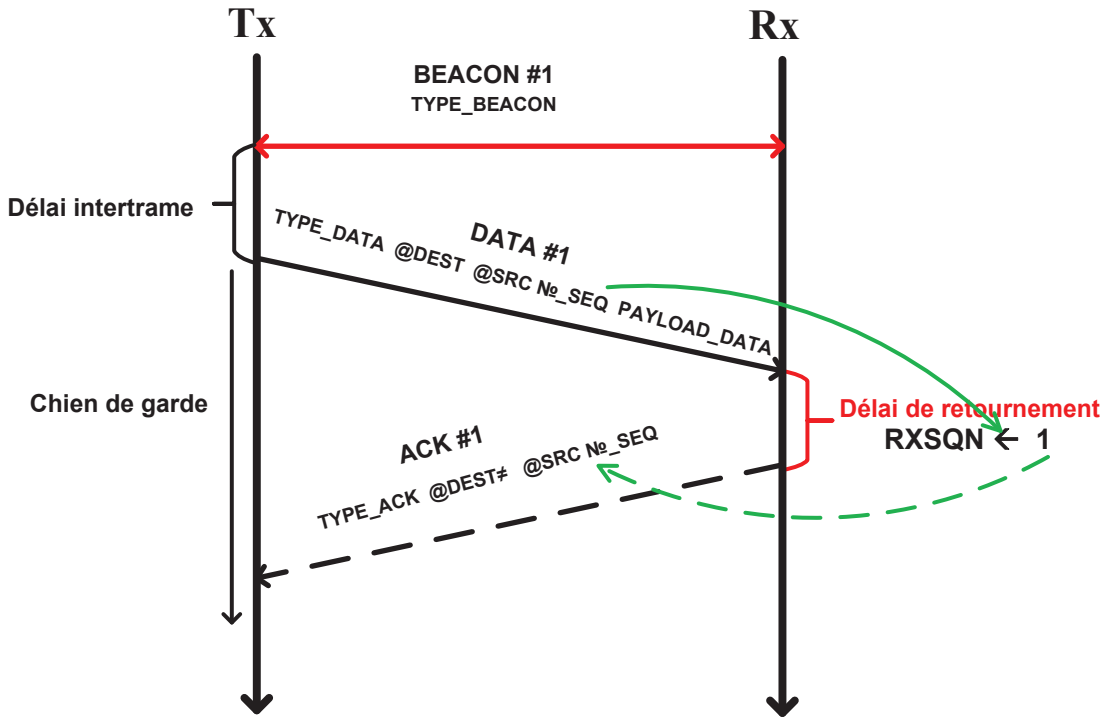

Figure 48 : Diagramme de séquence, l'ACK porte une adresse de destination différente de la trame émise.

Enfin le dernier cas possible qui parfois peut se présenter et que l'on peut observer sur la figure 49, est lorsqu'une trame quitte bien la source, mais n'atteint pas la destination souhaitée.

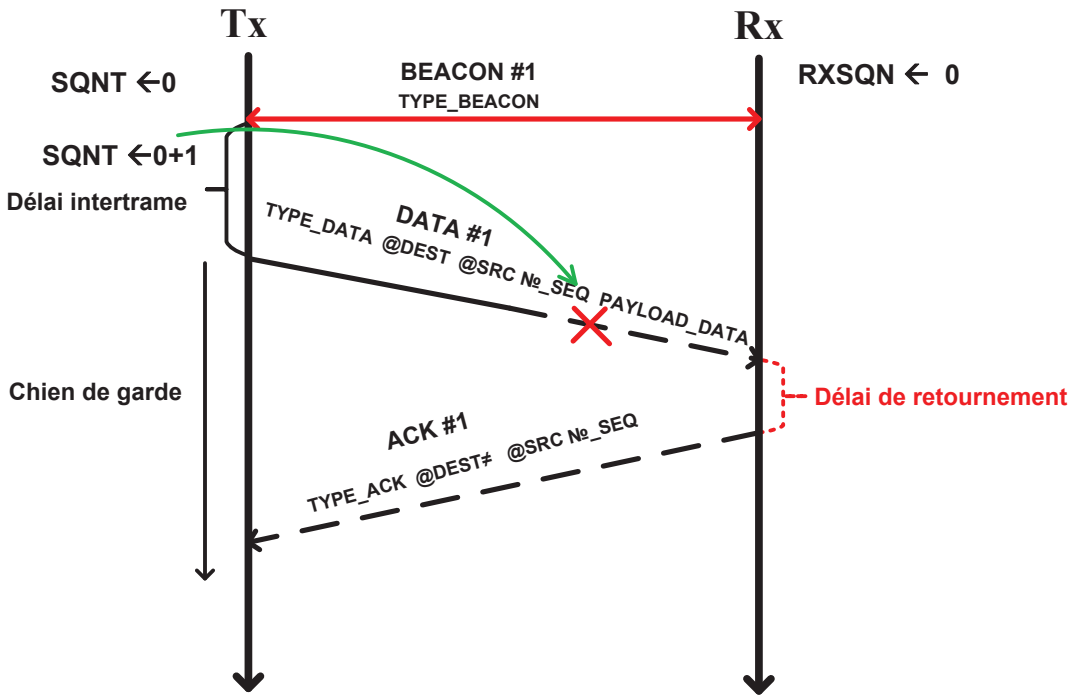

Figure 49 : Diagramme de séquence, où la trame n'atteint pas sa destination

3.4.4 Topologie de test pour accès mono-canal avec récepteur à portée d'un seul émetteur

Nous avons tout d'abord déployé un réseau de test constitué de 4 nœuds comme présenté sur la figure 50 suivante. DATA_@DEST_RXi désigne la trame de données émise vers RXi sur C0 (le canal 0), ainsi ACK_@DEST_TXi désigne l'acquittement de RXi vers TXi. RX1 (cercle pointillé vert) est à portée de TX1 et va recevoir des trames de cet émetteur. RX2 (cercle pointillé rouge) est à portée radio de TX2 dont il reçoit des trames. TX1 et TX2 sont à portées radio également.

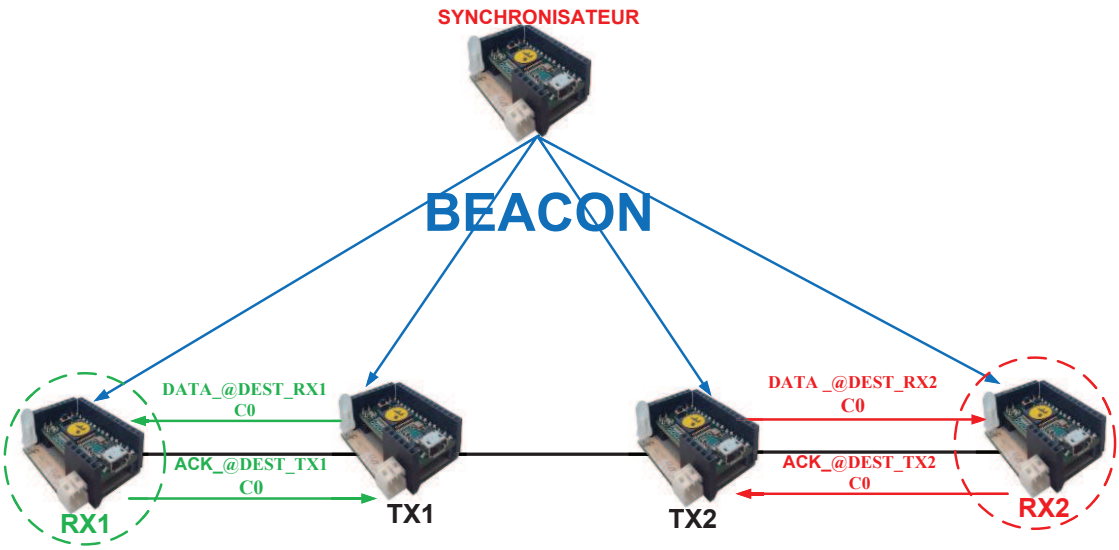

Figure 50 : Topologie d'un réseau où chaque récepteur se trouve à portée d'un seul émetteur

Pour augmenter ou diminuer la charge du réseau, nous avons fait varié la durée inter trame en utilisant la fonction ***RANDOM*** de l'Arduino avec une plage de délais aléatoires croissante, ce qui, permet de générer plus ou moins des trafics. On peut observer ceci en abscisse. En cas de collision par exemple, lorsque l'émetteur ne reçoit pas l'acquittement de la trame qu'il vient d'envoyer, celui-ci tire un nombre aléatoire de slots de temps avant d'émettre la trame suivante. Ce nombre aléatoire sera tiré entre (1 ; max) donc par l'instruction ***RANDOM(1, max)***.Plus le max est petit, plus la charge est élevée, ainsi par exemple, pour tirer aléatoirement un nombre de slot de temps entre 1 et 4, nous utilisons ***RANDOM(1, 5).*** Plus l'intervalle pour générer le nombre aléatoire est petit, plus le trafic généré est important et plus il y a une forte probabilité des collisions des trames. Les émetteurs peuvent bien sûr tirer le même nombre de slots de temps avant d'émettre, ce qui va créer une collision. Lorsque l'intervalle pour générer le nombre aléatoire est grand, la probabilité pour qu'il y ait des collisions des trames sera très faible, c'est-à-dire la probabilité pour que les émetteurs tirent le même nombre de slots de temps pour émettre est également faible.

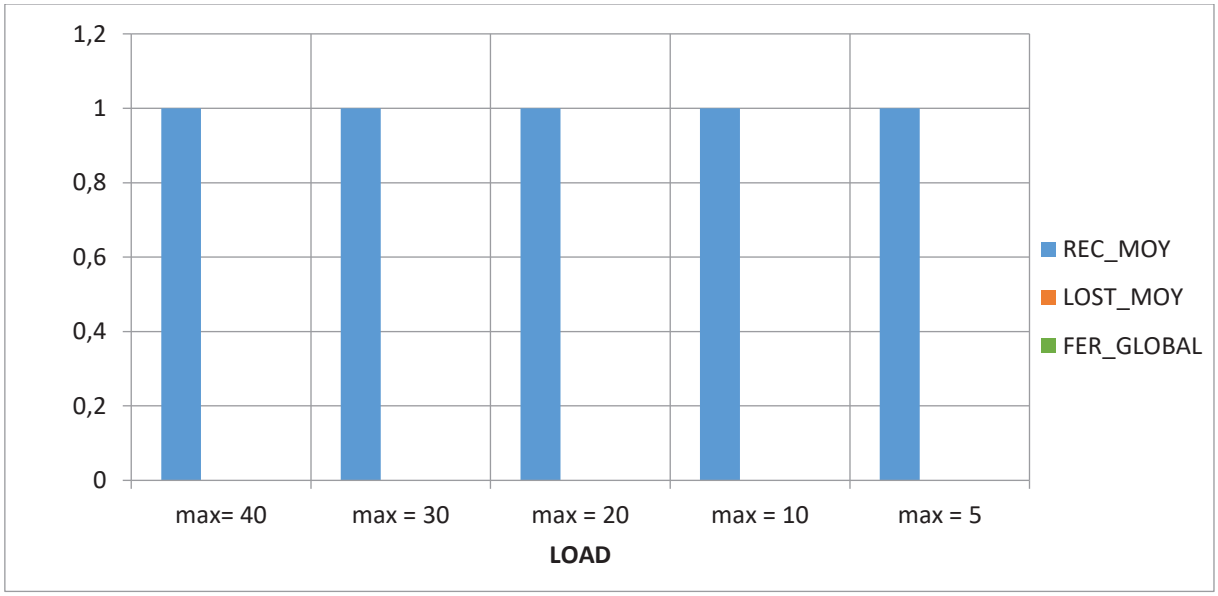

Figure 51 : Analyse de performances par rapport à la charge réseau au niveau de RX1

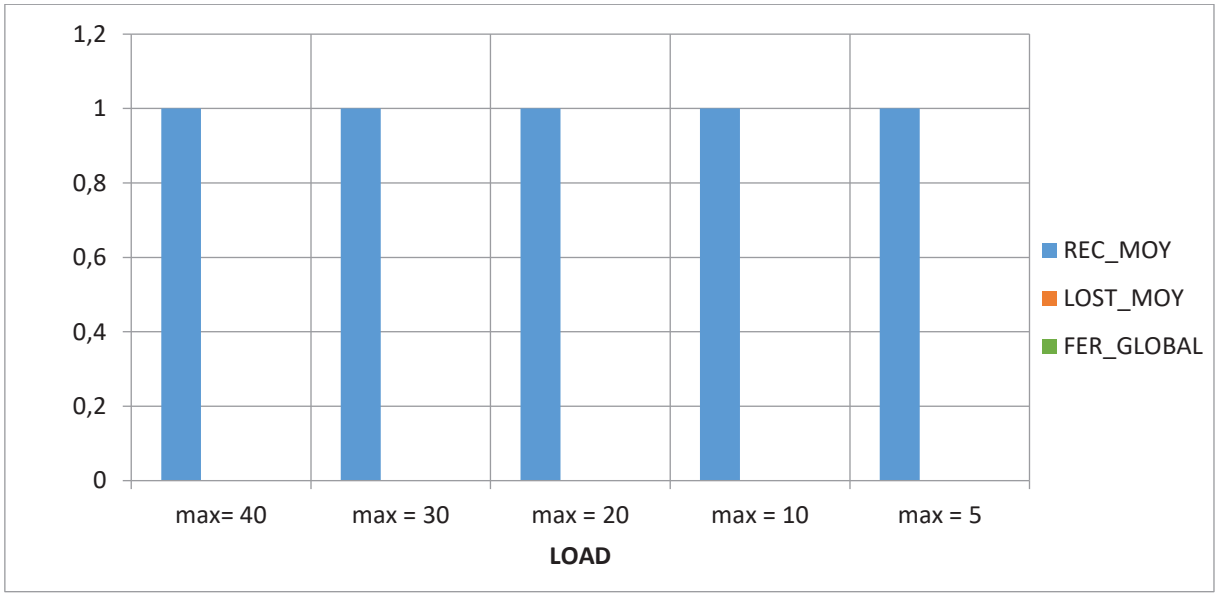

Figure 52 : Analyse de performances par rapport à la charge réseau au niveau de RX2

Les graphiques pour les deux récepteurs RX1 (Figure 51) et RX2 (Figure 52) montrent qu'il n'y a aucune perte des données, ce qui s'explique simplement par le fait qu'aucun des récepteurs ne se trouve dans la portée de plusieurs émetteurs (chaque récepteur se trouve dans la portée d'un seul émetteur), alors que tous les nœuds utilisent le même canal (cas de réutilisation spatiale du même canal à deux zones différentes), les données de l'émetteur TX1 n'atteignent pas le récepteur RX2, qui est en dehors de sa portée, il en est de même pour celles de TX2 qui n'atteignent pas non plus le récepteur RX1. On néglige bien sûr les erreurs de transmission, liées au taux d'erreur du médium qui est ici très bon sur de petites portées. Sur ces tests, sur mille (1000) trames émises, aucune perte de trame n'est remarquée à cause de ceci. Toutes les émissions et réceptions des trames (données, acquittements, beacons) sont effectuées en visibilité indirect (Non-Line-Of-Sight NLOS) dans une portée entre deux nœuds voisins de dix mètres (10m) en utilisant la puissance d'émission de 4dBm. La trame de données émise chaque 900 milliseconde est d'une longueur de 59 octets.

Pendant ce test, nous avons utilisé un sniffer (nœud radio espion programmée à partir d'un nœud WiNo) pour vérifier jusqu'à quelle portée les trames émises sont reçues. Influence des autres phénomènes externes sur l'analyse de performance

Il nous est arrivé de constater quelquefois en effectuant un test selon la topologie de la figure 50, quelques pertes des trames (Figure 53) qui surviennent à cause des mouvements des personnelles dans le laboratoire, qui effectuent aussi des tests sur des *testbeds* constitués de nœuds WiNo mais n'utilisent pas la même fréquence que celle que nous utilisons et qui de temps en temps perturbent les transmissions de nos données. Ce cas particulier influence aussi nos analyses de performance, comme on le voit figure suivante.

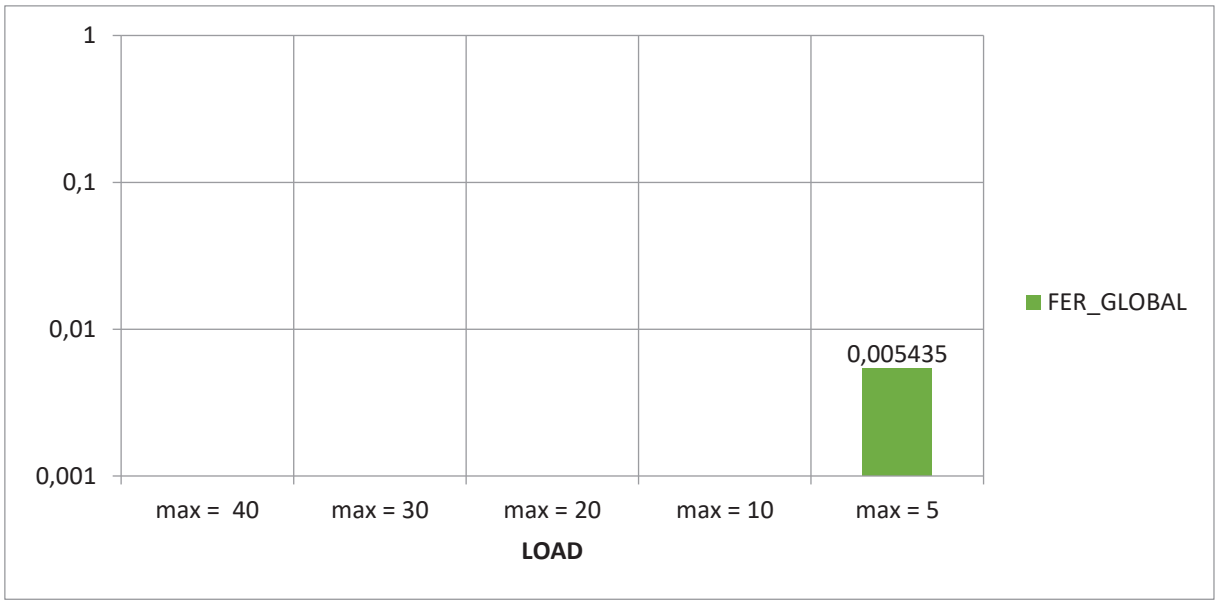

Figure 53 : FER lié au bruit externe et non causé par des collisions Entre nos nœuds

3.4.5 Analyse de performances d'un récepteur à portée radio de 2 émetteurs

Nous avons ensuite réalisé un testbed basé sur cette topologie (Figure 54) de 4 nœuds où RX2 est à portée radio des deux émetteurs, TX1 et TX2. DATA_@DEST_RXi désigne la trame de données émise vers RXi sur C0 (le canal 0), ainsi ACK_@DEST_TXi désigne l'acquittement de RXi vers TXi.

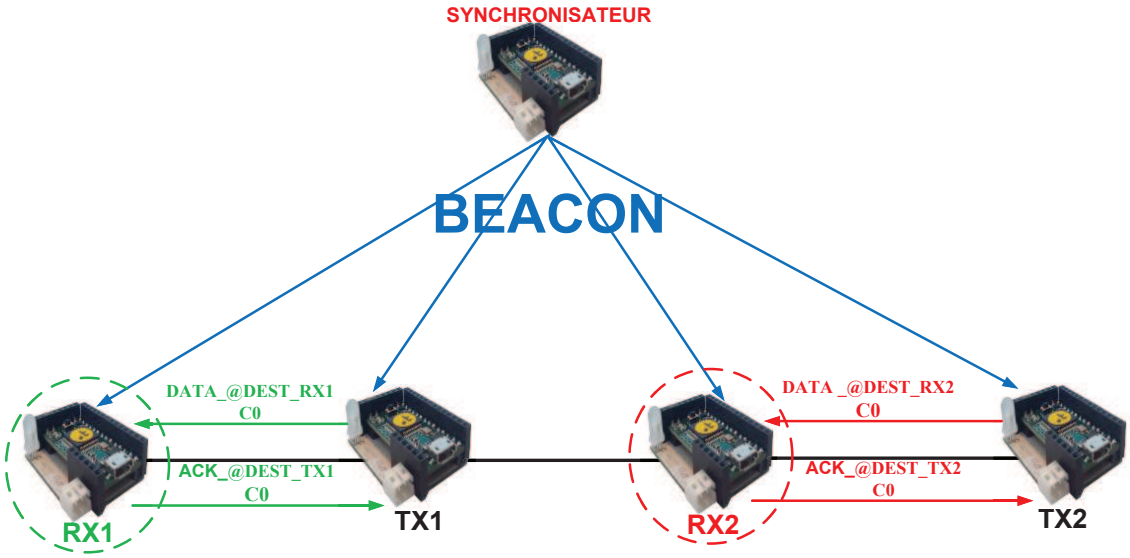

Figure 54 : Topologie mono-canal d'un récepteur qui se trouve à portée de deux émetteurs

L'émetteur TX2 émet vers RX2 et ne porte que vers lui. TX1 qui se trouve dans la zone de portée de RX1 et RX2 diffuse en radio vers les 2 récepteurs, mais n'émet que vers l'adresse de RX1. Nous observons sur la figure 55 en augmentant progressivement la charge du réseau, que le taux d'erreur trame est quasiment nul au niveau du récepteur RX1 qui est à la portée du seul émetteur TX1.

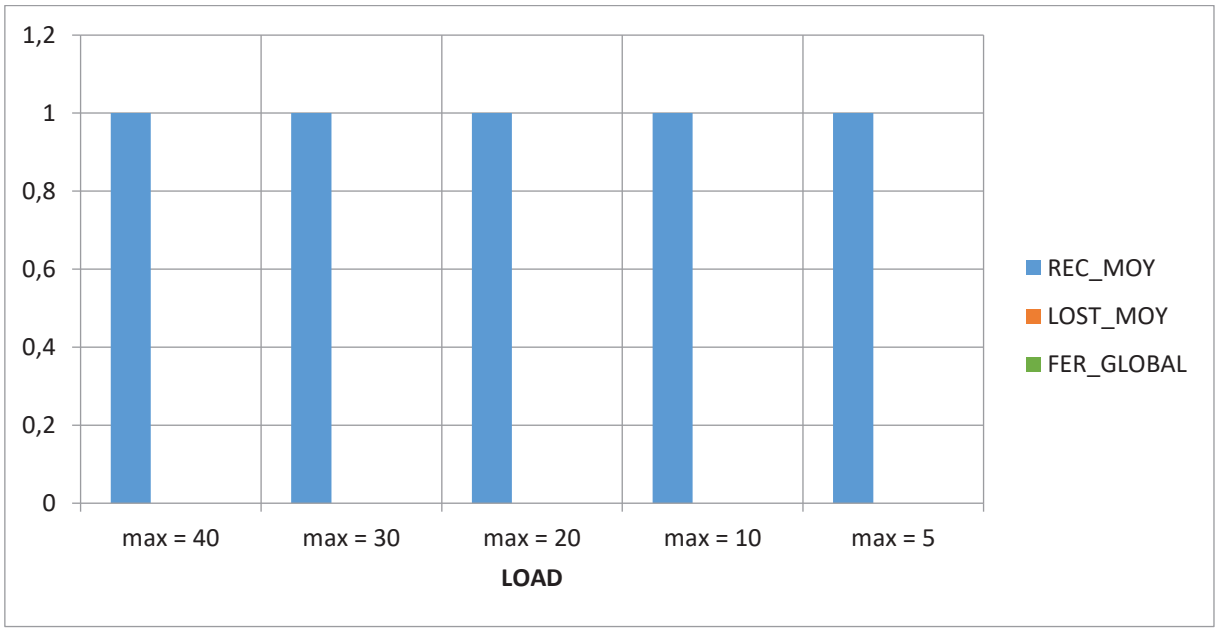

Figure 55 : Analyse de performances par rapport à la charge réseau au niveau de RX1

A l'inverse, au niveau du récepteur RX2, lorsque les deux émetteurs TX1 et TX2 émettent dans un même slot de temps, les données entrent en collision au niveau du récepteur RX2, ceci peut se voir sur la figure 56 **:** plus la charge du réseau est élevée, plus il y a de pertes des trames (à cause principalement des collisions), on peut voir ceci sur le max = 5. Le taux d'erreur trame croit progressivement lorsqu'on augmente la charge du réseau, à cause des collisions.

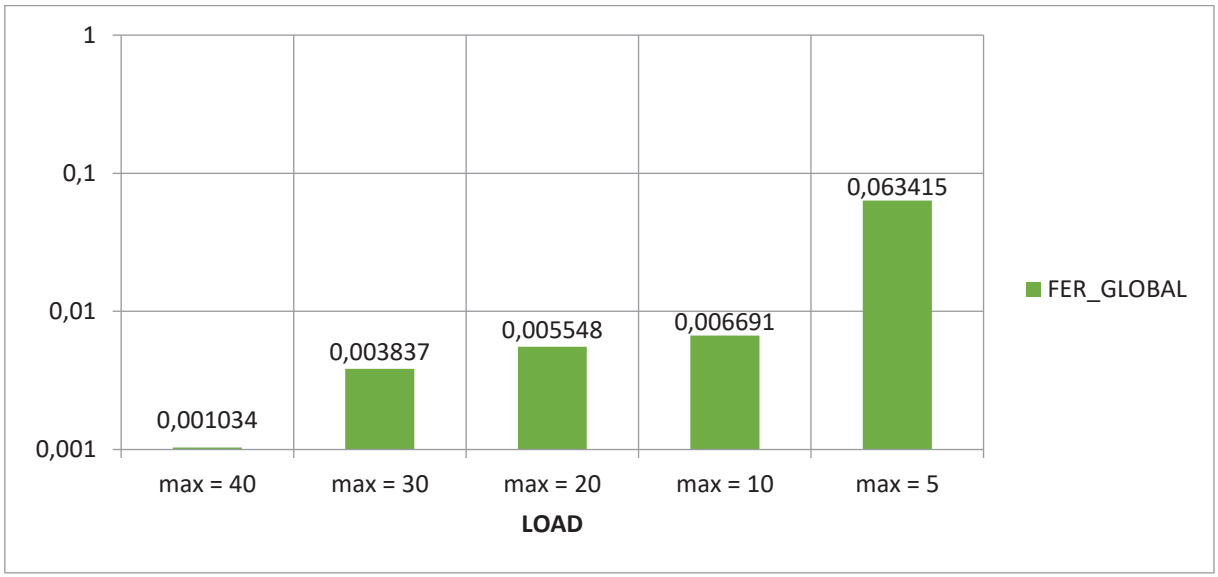

Figure 56 : Analyse de performances par rapport à la charge réseau au niveau de RX2

Avec la même topologie de la figure 34 et en utilisant la même puissance du signal d'émission des données et acquittements, un test nous fournis les mêmes résultats que ceux observés sur la figure 51 où le taux d'erreur trame est nul au niveau du récepteur RX1, c'est le cas idéal. Mais il est aussi quasiment nul au niveau du récepteur RX2 (Figure 57) qui est dans la portée de deux émetteurs qui peuvent potentiellement émettre dans un même slot de temps et provoquer une collision au niveau du récepteur RX2, mais tel n'est pas le cas attendu. Nous pouvons expliquer simplement ce dernier cas que parfois un nœud de destination peut entendre seulement la source qui émet vers lui et ignorer les autres voisins à portée qui émettent vers d'autres destinations (phénomène de capture radio) [70].

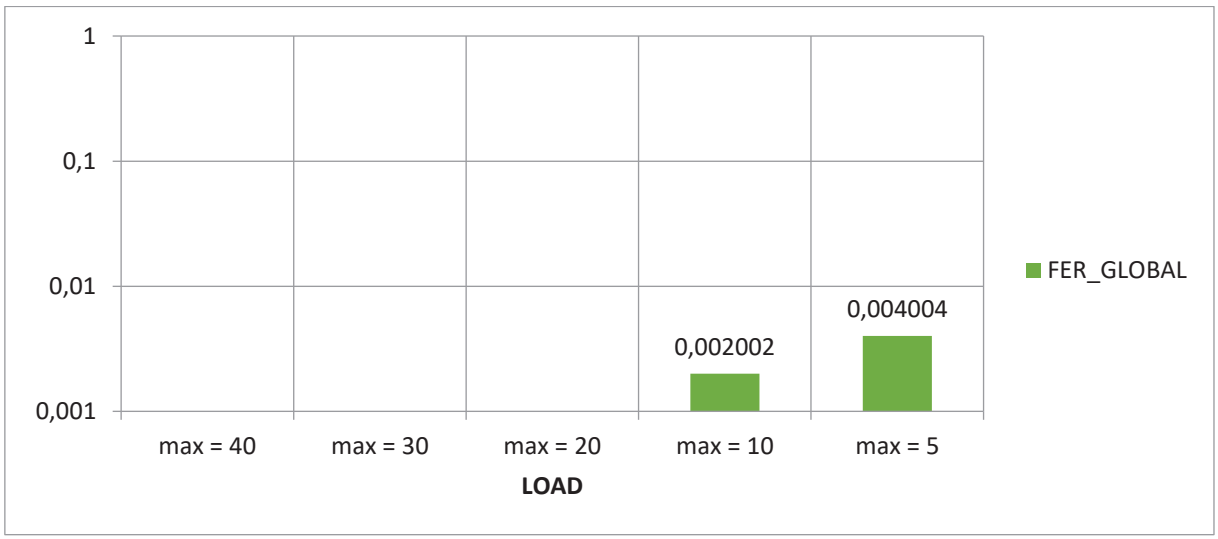

Figure 57 : Analyse de performances par rapport à la charge réseau au niveau de RX2

3.4.6 Accès mono-canal sans RDV, topologie avec émetteur générant des trafics vers 2 récepteurs

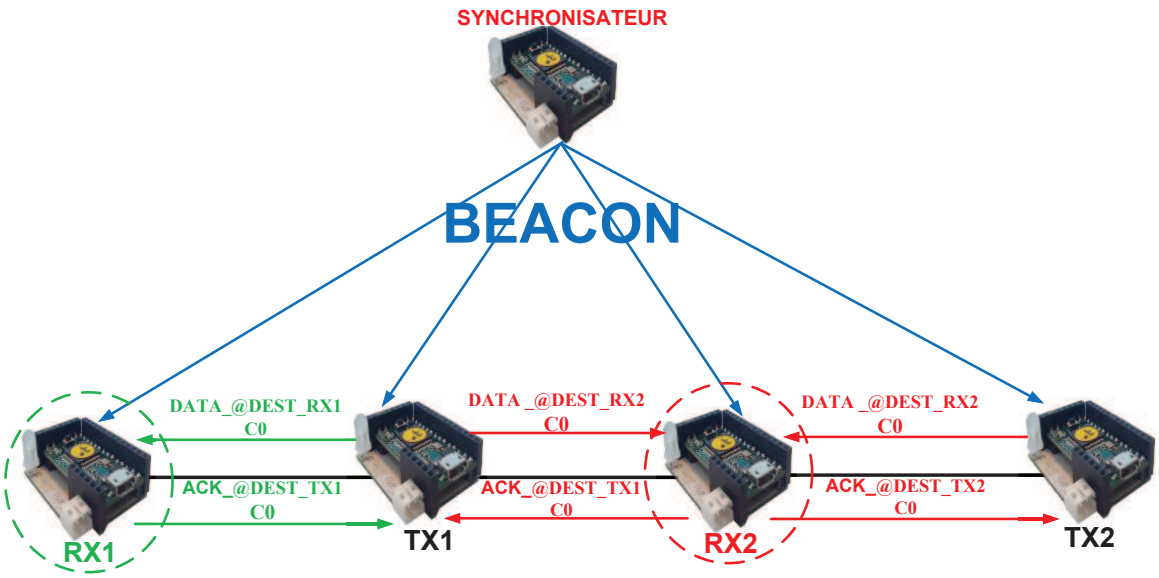

Figure 58: TX1 émet soit vers RX1, soit RX2

Dans cette topologie (Figure 58) DATA_@DEST_RXi désigne la trame de données émise vers RXi sur C0 (le canal 0), ainsi ACK_@DEST_TXi désigne l'acquittement de RXi vers TXi.

Nous étudions les performances dans le contexte multi-saut, le récepteur RX2 se trouve dans la zone de portée de deux émetteurs, chaque Txi se trouve à deux sauts de l'autre. TX1 peut tirer un slot de temps et décider aléatoirement aussi d'émettre soit vers RX1, soit vers RX2.Dans ce cas précis, il est évident que le récepteur RX2 ne peut pas recevoir des trames dans un même slot de temps des deux émetteurs TX1 et TX2, les trames se chevauchent ce qui produit une collision par conséquent RX2 ne reçoit aucune trame, dont nous observons les résultats sur les figures suivantes (Figure 59 et Figure 60).

Sur le graphique du récepteur RX1 (Figure 57), nous observons que le taux d'erreur trames augmente lorsque la charge du réseau diminue, autrement dit lorsque le récepteur a moins de voisins qui lui envoient des trames. D'après l'équation (1), le récepteur RX1 reçoit une séquence de trame de l'émetteur TX1 après un certain nombre de slot de temps.

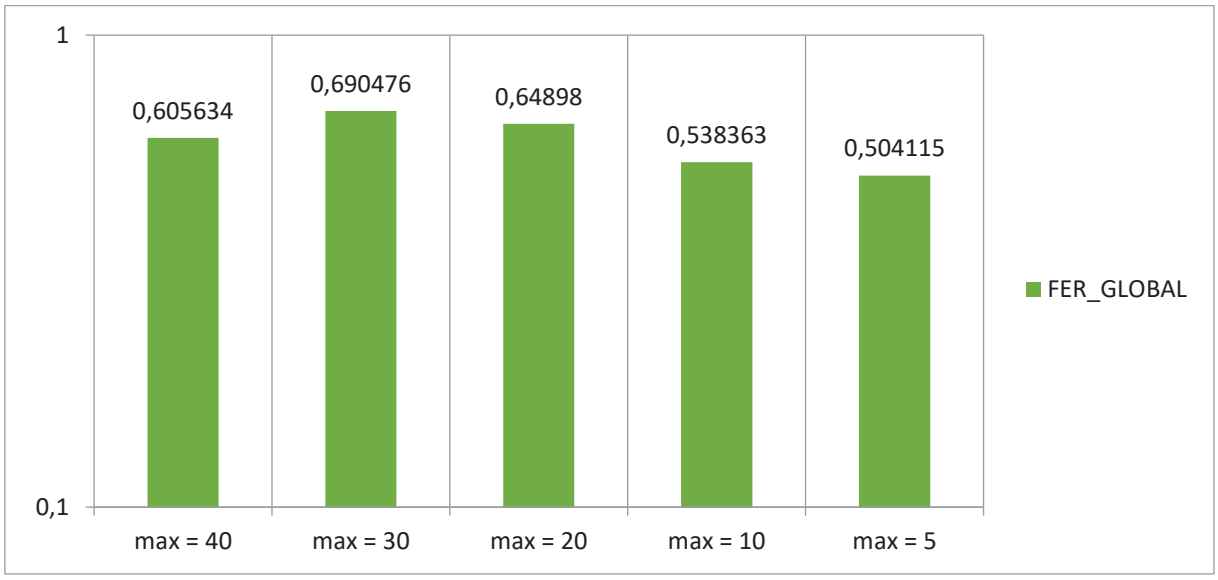

Figure 59 : Analyse de performances par rapport à la charge réseau au niveau de RX1

Puisque le récepteur RX2 peut recevoir de deux émetteurs TX1 et TX2, on voit sur la figure 60 que lorsque la charge du réseau diminue, le taux d'erreur trame décroît ; lorsqu'une collision se produit, les deux émetteurs TX1 et TX2 se désynchronisent en tirant un nombre aléatoire de slots de temps, que nous avons indiqué en abscisse pour représenter la charge du réseau. Plus ce nombre est grand, moins le taux d'erreur trame est important. La probabilité que les deux émetteurs TX1 et TX2 tirent le même slot de temps pour émettre est très faible.

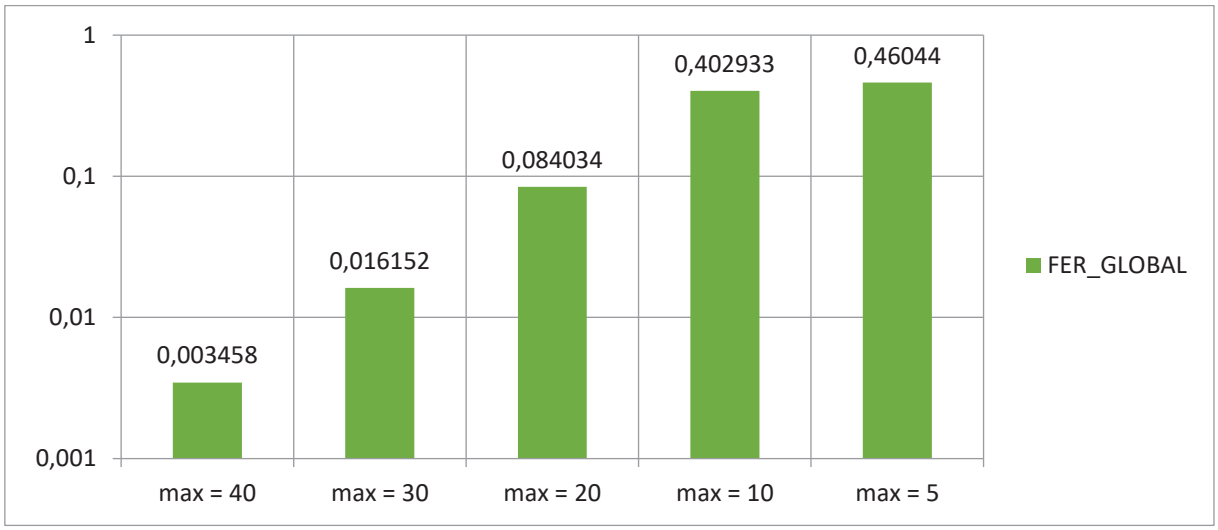

Figure 60 : Analyse de performances par rapport à la charge réseau au niveau de RX2

Après avoir analysé le comportement et les performances en mono-canal, nous allons mettre en place le multi-canal, et ainsi se servir de ces précédents résultats comme éléments de comparaison.

3.5 L'accès multi-canal

Nous disposons dans la bande ISM 433 MHz de plusieurs canaux. Nous allons ici utiliser 2 canaux possibles pour les DATA-ACK (0 et 1) et un même canal pour les beacons (0) émis à d'autres instants.

3.5.1 Automates de transmissions multi-canal

3.5.1.1 Automate multi-canal du nœud émetteur (Tx)

L'émetteur Tx (automate Tx Figure 61) met sa radio en réception sur le canal 0 et attend de recevoir le *beacon*. A chaque fois, après avoir reçu le *beacon*, il décrémente Nb_BC (nombre de *beacons reçus*) jusqu'à atteindre la valeur zéro, si précédemment, une collision s'est produite pour émettre une trame, sinon il émet directement sa trame après réception du *beacon*. Dans le cas où le nombre de *beacons* n'a pas atteint zéro, il laisse toujours la radio en mode réception et attend de recevoir le prochain *beacon*. Après avoir reçu le *beacon*, il tire aléatoirement un canal entre 0 et 1 et émet la trame de données dessus, un chien de garde (*Watchdog*) est armé pour borner le temps pour recevoir l'acquittement (ACK) de la trame émise sur le même canal, la radio est mise en réception en attendant l'acquittement. Si il reçoit un acquittement dont il n'est pas destinataire, ou si il reçoit une autre trame du type autre que ACK, alors il la néglige (il ne la traite pas et l'absorbe), et reste donc toujours en attente d'ACK. Si le chien de garde expire et qu'il n'a pas reçu l'ACK alors, il affiche « échec », ce qui veut dire que la trame de données émise n'a pas été reçue (à cause sans doute d'une collision ou d'une erreur sur l'ACK), il tire alors un nombre de slots (*beacons*) aléatoire et met finalement sa radio en mode réception. Les numéros de séquences sont initialisés à 0, tandis que Nb_BC (nombre de *beacon*) est tiré aléatoirement en utilisant différents intervalles selon les tests : (1, 5) ; (1, 10) ; (1, 20) ; (1, 30) ; (1, 40), il est donc initialisé à 1.

Par contre, si l'émetteur TX reçoit un acquittement qui porte bien son adresse que nous appelons "NODE ADRESS" et le numéro de séquence de la trame qu'il vient d'envoyer, alors il affiche que la trame est bien reçue et le Nb_BC (nombre de *beacon*s) sera réinitialiser à 1.

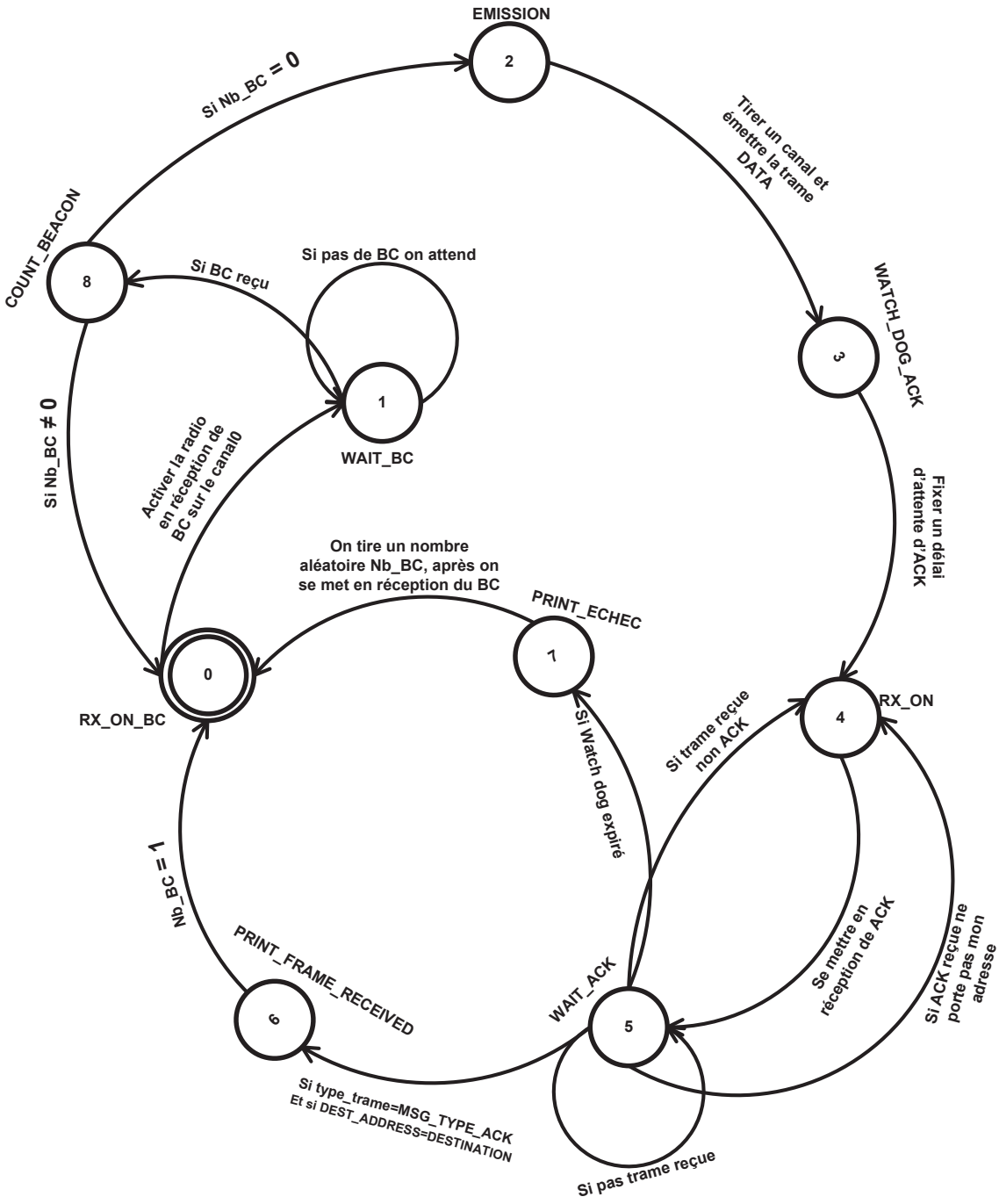

Figure 61 : Automate multi-canal du nœud émetteur Tx

3.5.1.2 Automate multi-canal du nœud récepteur (Rx)

Le récepteur Rx (automate Rx Figure 62) met sa radio en réception sur le canal 0 et attend de recevoir le *beacon*. Après avoir reçu le *beacon*, le récepteur Rx tire aléatoirement un canal de DATA-ACK entre 0 et 1, et met sa radio en réception, et vérifie s'il y a un message disponible à récupérer dans le tranceiver radio RF22. La trame de données contient le type de message (DATA), adresse source, adresse destination, le numéro de séquence et la charge utile *(payload)*, au total elle contient 59 octets.

S'il reçoit une trame dont le type de message n'est pas le type « Data », alors il affiche "pas de trame de données". Si par contre la trame reçue est de type « Data » et porte l'adresse de destination du récepteur Rx, alors il affiche la trame reçue et, renvoie une trame d'acquittement ACK à l'émetteur de la trame reçue sur le même canal qui contient le type de message (type ACK), l'adresse de destination qui est effectivement l'adresse source de la trame de data, l'adresse source qui est l'adresse de destination de la trame de data et le numéro de séquence de la trame de data reçue. Dans le cas où la trame de données porte une adresse de destination autre que celle du récepteur Rx, alors celui-ci affiche que la trame reçue "ne lui est pas destinée".

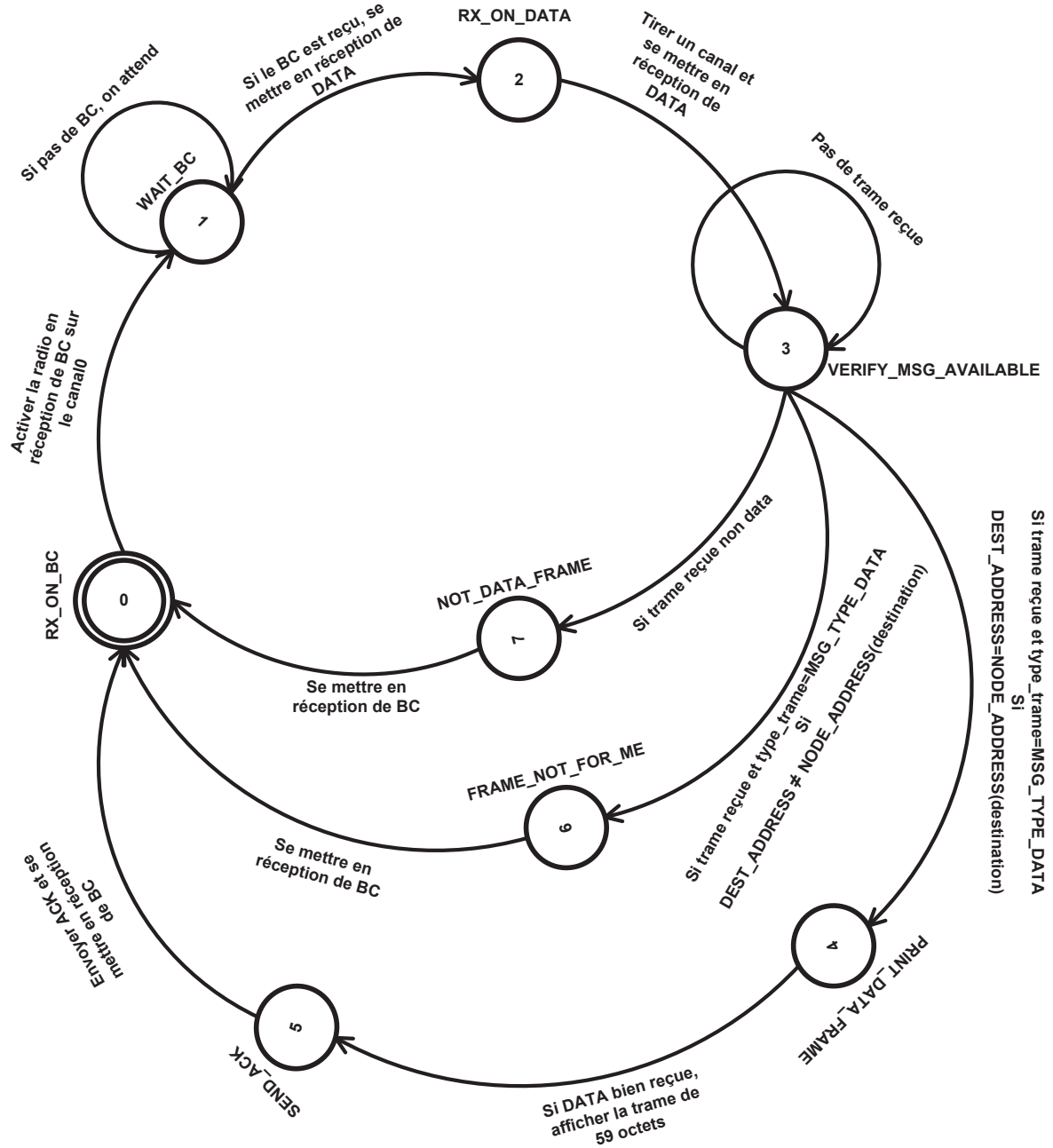

Figure 62 : Automate multi-canal du nœud récepteur Rx

3.5.2 Analyse de performances d'un récepteur à portée radio de 2 émetteurs

Nous analysons maintenant les mêmes types de topologies précédentes, mais dans un contexte multi-canal (deux canaux) ; et nous utilisons les mêmes formats et séquences de trames (cf. paragraphe **3.4.2** et **3.4.3**). On peut voir sur la figure 21, DATA_@DEST_RXi désigne la trame de données émise vers RXi sur C0/C1 (sur le canal 0 ou le canal 1), de même ACK_@DEST_TXi désigne l'acquittement de RXi vers TXi sur le même canal de la trame reçue, chacun de deux récepteurs (RX1 et RX2), à chaque slot de temps, tire aléatoirement un canal et se met en réception des trames des données. Le récepteur RX1qui se trouve dans la zone de portée de l'émetteur TX1, et ne reçoit des trames que de celui-ci, à son tour, de la même manière, tire aléatoirement un canal et émet vers le récepteur RX1. Par contre, comme on peut constater sur la figure 21, que le récepteur RX2 est à portée radio de deux émetteurs (TX1 et TX2), mais l'émetteur TX2 émet vers le récepteur RX2, ainsi donc des collisions pourront se produire au niveau du récepteur RX2 au cas où le deux émetteurs (TX1 et TX2) tirent le même slot de temps et le même canal.

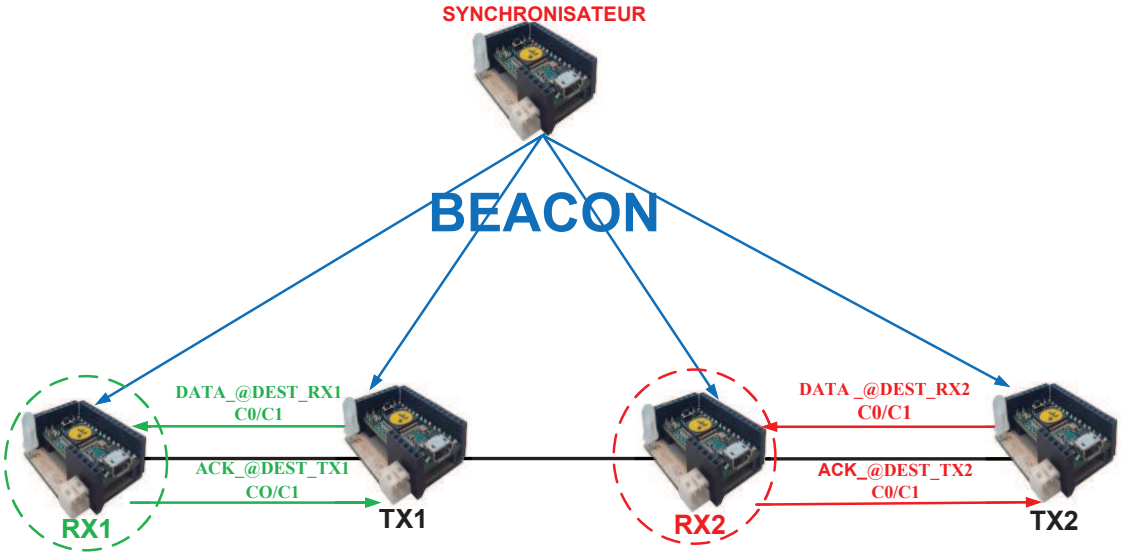

Figure 63 : Topologie multi-canal d'un récepteur qui se trouve à portée de deux émetteurs

D'après la topologie de la figure 63, DATA_@DEST_RXi désigne la trame de données émise vers RXi sur C0/C1 (sur le canal 0 ou le canal 1), de même ACK_@DEST_TXi désigne l'acquittement de RXi vers TXi sur le même canal de la trame reçue, les analyse de performances au niveau du récepteur RX1, nous fournissent les résultats qu'on peut observer sur le graphique de la figure 64; nous rappelons que RX1 est dans la portée radio d'un seul émetteur TX1 et ne reçoit des trames que de ce dernier, on observe un taux d'erreur trame important par rapport au mono-canal dû au changement fréquent des canaux entre l'émetteur et le récepteur, il faut aussi noter que l'émetteur peut souvent manquer le canal de réception. On constate que le taux d'erreur trame démunie progressivement avec la réduction de la charge réseau.

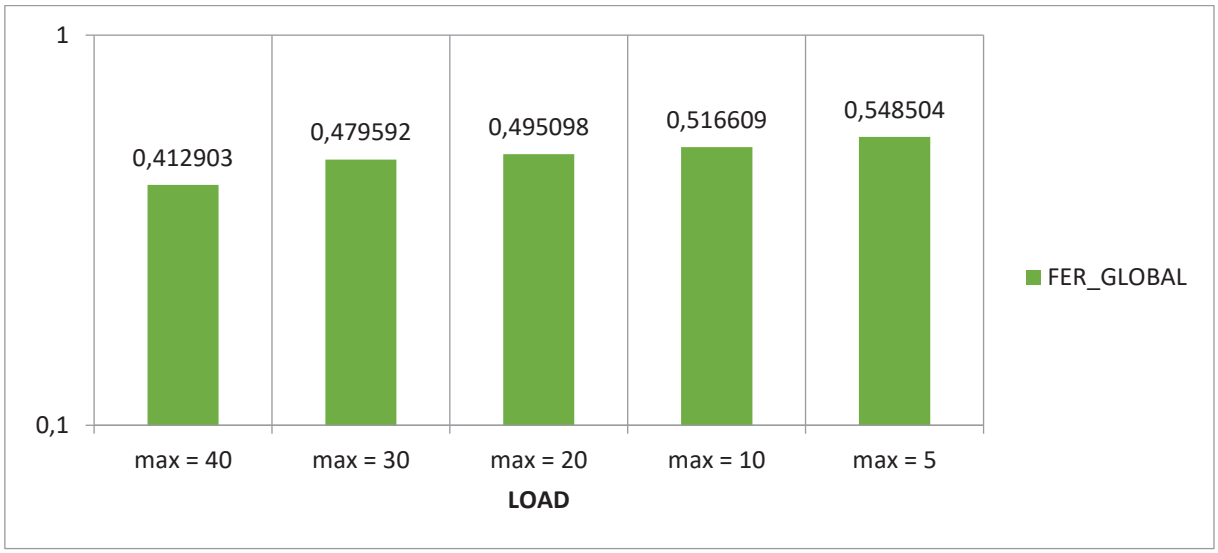

Figure 64 : Analyse de performances par rapport à la charge réseau au niveau de RX1

Par contre au niveau du récepteur RX2 (figure 65), on peut observer presque le même résultat avec un taux d'erreur trame qui diminue progressivement avec la réduction de la charge réseau, mais de façon plus importante que celui au niveau du récepteur RX1, puisque le récepteur RX2 est dans la zone de portée de l'émetteur TX1 et TX2 qui peuvent tirer le même slot de temps et le même canal provocant des collisions au niveau du récepteur RX2. Les pertes des trames surviennent aussi lorsque l'émetteur envoie une trame sur le canal dont le récepteur n'est pas en écoute pendant ce slot de temps. Il faut en effet que Tx et Rx se trouvent sur le même canal.

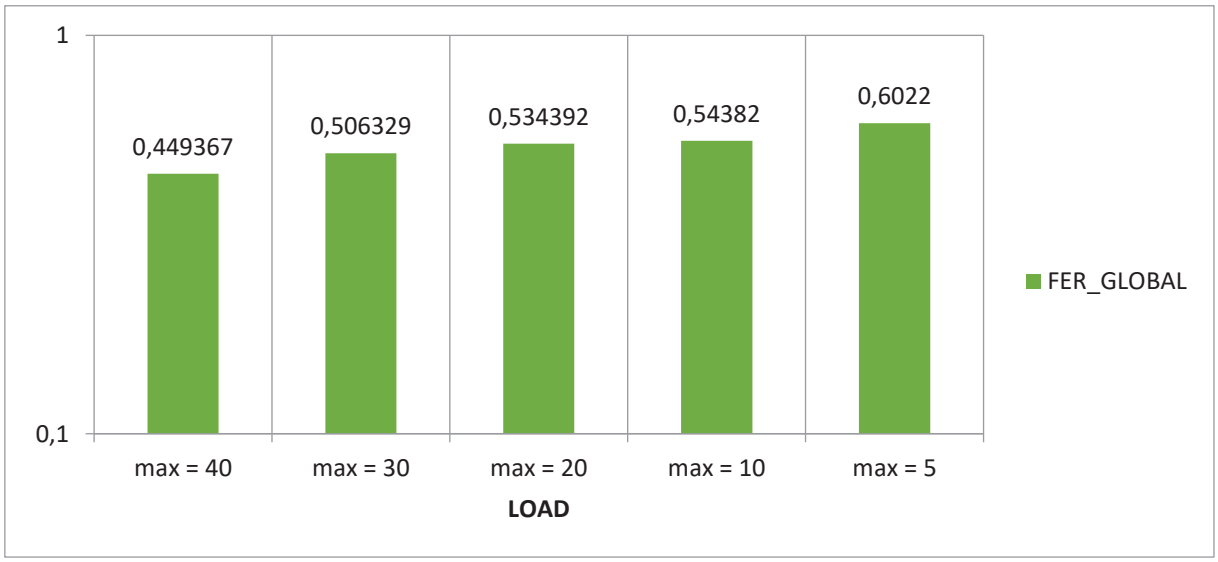

Figure 65 : Analyse de performances par rapport à la charge réseau au niveau de RX2

Nous remarquons d'après les résultats fournis par nos analyses de performances dans le contexte où un récepteur est à portée radio de deux émetteurs, le débit (faible FER) offert par l'accès mono-canal est meilleur qu'en multi-canal (résultats que nous n'attendions pas) pour la simple raison qu'en multi-canal, excepté les collisions des trames (lorsque les deux Tx tirent le même slot et même canal), il faut tenir compte aussi du nombre des trames perdues (trames émises et non reçues par les récepteurs), dû au manque de canal de réception qui influence considérablement sur l'analyse de performance, ce que

nous pensons corriger par la stratégie de rémanence sur le dernier canal de succès (voire paragraphe 6).

3.5.3 Accès multi-canal sans RDV, cas d'un émetteur générant des trafics vers 2 récepteurs

Contrairement à la topologie mono-canal (Figure 63), dans cette topologie multi-canal (Figure 66), DATA_@DEST_RXi désigne la trame de données émise vers RXi sur C0/C1 (sur le canal 0 ou le canal 1), de même ACK_@DEST_TXi désigne l'acquittement de RXi vers TXi sur le même canal de la trame reçue, les récepteurs (RX1 et RX2), comme les émetteurs (TX1 et TX2) tirent aléatoirement un canal de réception ou d'émission. On peut le voir sur la topologie de la figure 23 : le récepteur RX1 peut recevoir seulement de l'émetteur TX1, tandis que le récepteur RX2 peut recevoir soit de l'émetteur TX1 ou de l'émetteur TX2.

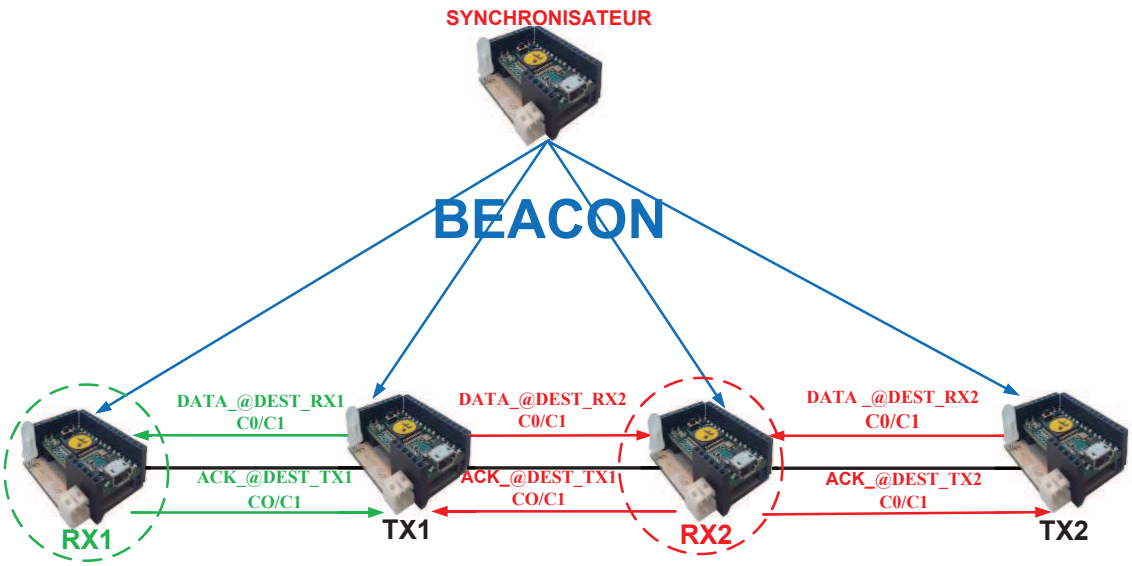

Figure 66 : Topologie 2 (TX1 émet soit vers RX1, soit RX2)

Nous pouvons voir au niveau du récepteur RX1 d'après le graphique de la figure 67 un taux d'erreur trame important et presque constant vis-à-vis de la charge réseau (quelle que soit la charge), ceci s'explique par le manquement du canal de réception du nœud récepteur RX1.Comme l'émetteur TX1 tire aléatoirement une adresse de destination, en calculant le nombre des trames perdues (équation 1), on se trouve avec un numéro de séquence reçu (SQNT) très grand. Pour corriger ce défaut lié au manquement du canal de réception, et qui génère plus de taux d'erreur trames, nous utilisons une stratégie de rémanence sur le dernier canal utilisé avec succès qui sera détaillé au paragraphe suivant.

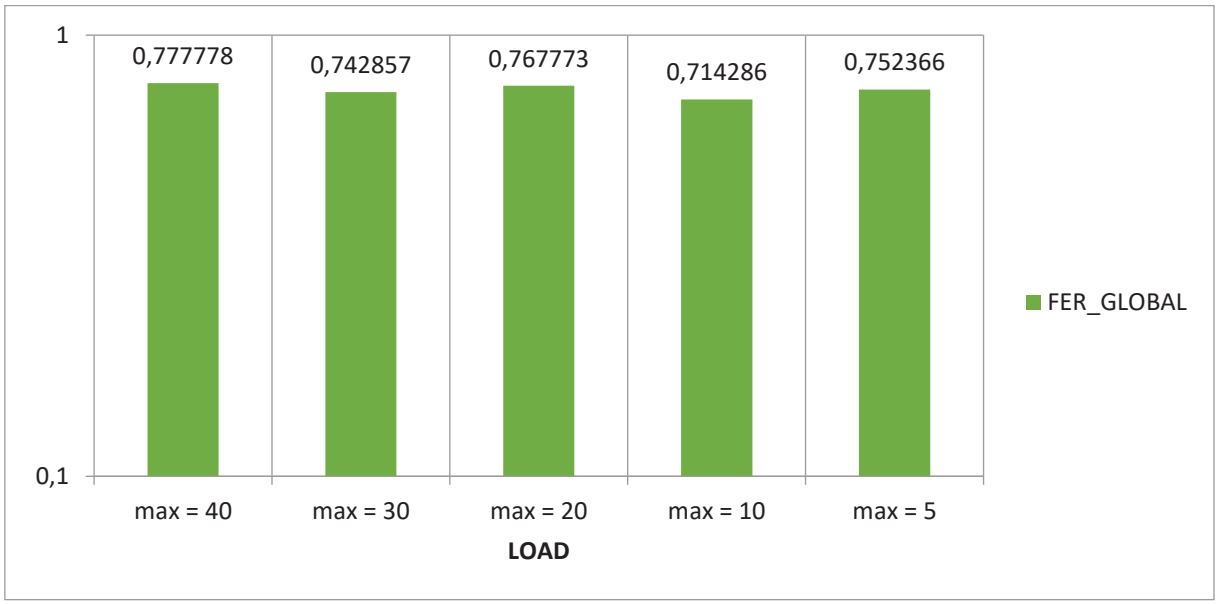

Figure 67 : Analyse de performances par rapport à la charge réseau au niveau de RX1

Nous pouvons observer sur le graphique de la figure 68 correspondant aux données du récepteur RX2, un taux d'erreur trame important, qui décroît progressivement avec la réduction de la charge et se stabilise à un moment donné. Ce dernier se produit par le manquement du canal de réception du nœud récepteur RX2 par les deux émetteurs mais aussi par les collisions, puisque les émetteurs TX1 et TX2 peuvent tirer le même slot de temps et le même canal et envoyer vers la même destination RX2. En comparant les résultats des deux récepteurs (RX1 et RX2), on constate que, même si le récepteur RX2 subit des collisions des trames, auquel il faut rajouter le manquement du canal de réception, les pertes des trames se réduisent lorsque la charge réseau diminue. Par contre, le taux d'erreur trame est important et reste constant au niveau du récepteur RX1 même si la charge du réseau baisse.

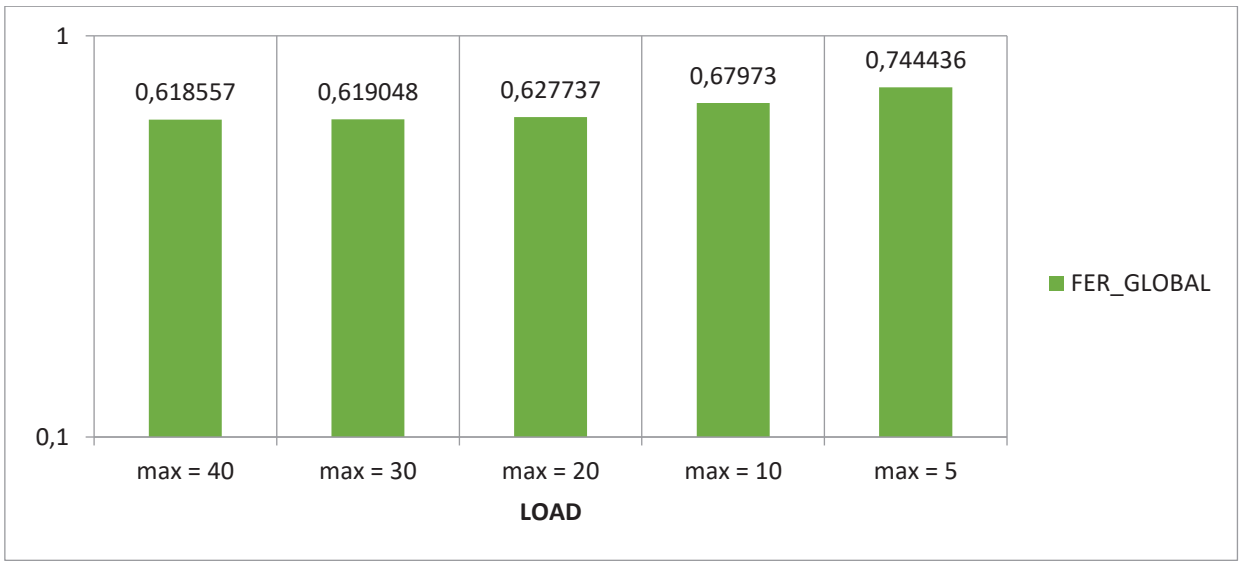

Figure 68 : Analyse de performances par rapport à la charge réseau au niveau de RX2

Nous allons améliorer ces lacunes en mettant en place une rémanence sur le dernier canal qui a entrainé un succès, ainsi, une fois que les deux nœuds se sont retrouvés sur le même canal, ils vont y rester un moment s'ils doivent continuer de parler tous les deux.

3.6 L'accès multi-canal avec stratégie de rémanence sur le précédent canal d'émission et de réception

3.6.1 Analyse de performances

Nous avons comme objectif principale l'implémentation d'une méthode d'accès multi-canal multi-saut sans RDV tout en optimisant le taux de succès de réception sans collision [69], malgré l'absence de RDV. Pour y aboutir, nous avons pensé à adopter une stratégie de rémanence sur le dernier canal de succès, ceci va compenser le taux de non correspondance entre le choix du canal d'émission et du canal de réception des deux nœuds communicants identifié dans les pré-versions précédentes.

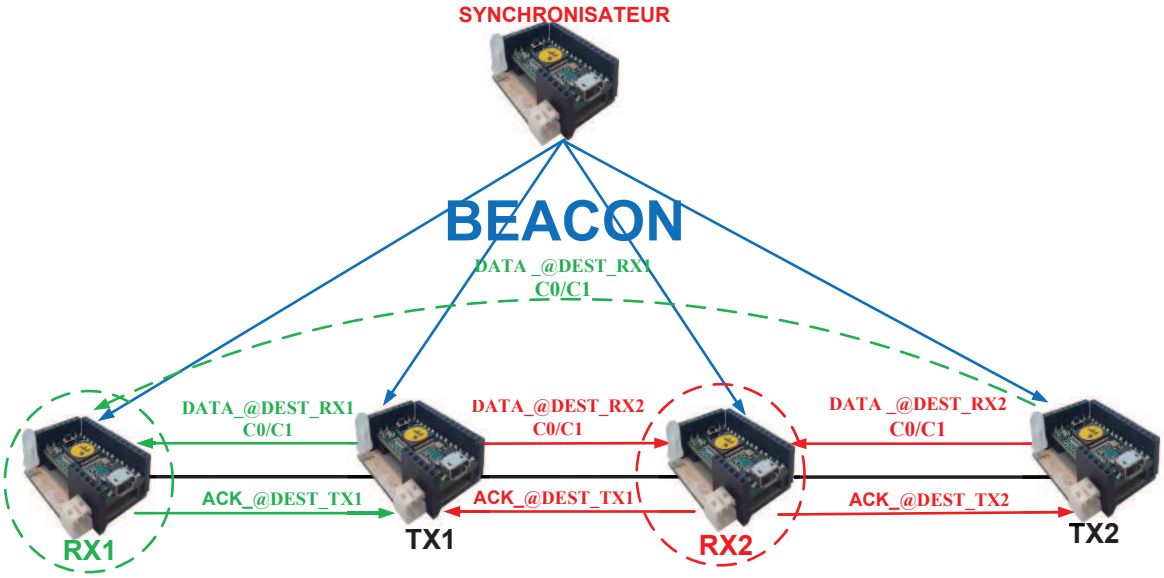

Figure 69 : Topologie de la rémanence sur le dernier canal de succès

Considérons la topologie de la figure 69 sur laquelle le nœud synchronisateur (**SYNCHRONISATEUR**) qui diffuse chaque 900 ms un **BEACON** (balise) ; DATA_@DEST_RXi désigne la trame de données émise vers RXi sur C0/C1 (sur le canal 0 ou le canal 1), de même ACK_@DEST_TXi désigne l'acquittement de RXi vers TXi sur le même canal de la trame reçue, les récepteurs (RX1 et RX2), comme les émetteurs (TX1 et TX2) tirent aléatoirement un canal de réception ou d'émission.

Sur cette topologie multi-sauts (Figure 69), chacun des émetteurs TX1 et TX2 génère progressivement de plus en plus des trafics vers tous les récepteurs. Les trames des données et d'acquittements sont émises avec la même puissance d'émission. Les trames des données portent l'adresse source, l'adresse destination, le numéro de séquence et la charge utile et contiennent donc en tout 59 octets. Par contre la trame d'acquittement ACK est émise uniquement avec l'adresse source, destination et le numéro de séquence.

Sur cette topologie, on peut observer que TX1 est à portée radio des deux émetteurs RX1 et RX2, donc, si on exclut tout obstacle, la trame émise par TX1 sera reçue par chacun des deux récepteurs ; contrairement à TX2 qui n'est à portée que de RX2, donc, la trame émise par TX2 ne sera reçue que par RX2.

Si l'un des émetteurs choisi un canal et émet une trame sur ce canal, et s'il reçoit un acquittement de celle-ci, alors l'émetteur mémorise ce canal de succès pour la prochaine émission vers ce nœud, et va sélectionner ce même canal plus tard si il désire encore parler à la même destination. Sinon, en cas

d'échec, il choisira aléatoirement un autre canal. Un peu de la même façon (mais sans condition bien sûr sur la source qui n'est pas prédictible), le récepteur reste aussi en réception sur le dernier canal de réception pendant ***K*** slot de temps (***K*>=*4*** pour ce testbed). Par contre, au-delà de ce temps (***K*** slot de temps expirés), il tire aléatoirement un autre canal de réception. Ceci évite tout phénomène de famine, car on pourrait être tenté de rester sur ce même canal de réception tant que des trames arrivent de la part d'un émetteur bavard, empêchant les autres émetteurs potentiels sur d'autres canaux de parler avec ce récepteur.

En utilisant cette méthode de rémanence avec le même nombre de canaux (deux canaux sur notre exemple de test), on peut observer sur le graphique de la figure 70 que le FER au niveau de RX1 qui est à portée d'un seul émetteur TX1 croît progressivement lorsque la charge réseau augmente, cependant le FER est moins important que celui observé au niveau de RX2 (Figure 71). Mais si nous le comparons par rapport au FER observé sur le testbed sans rémanence sur le canal de succès (Figure 67) il est encore moins important avec des valeurs différentes très significatives. Ceci s'explique simplement par le fait que la probabilité pour que l'émetteur TX1 et le récepteur RX1 se trouvent sur le même canal est nettement améliorée, toutefois les perturbations externes telle que : le réseau WiFi, les autres testbeds dans notre environnement qui s'exécutent au même moment que les nôtres, perturbent toujours nos analyses de performances.

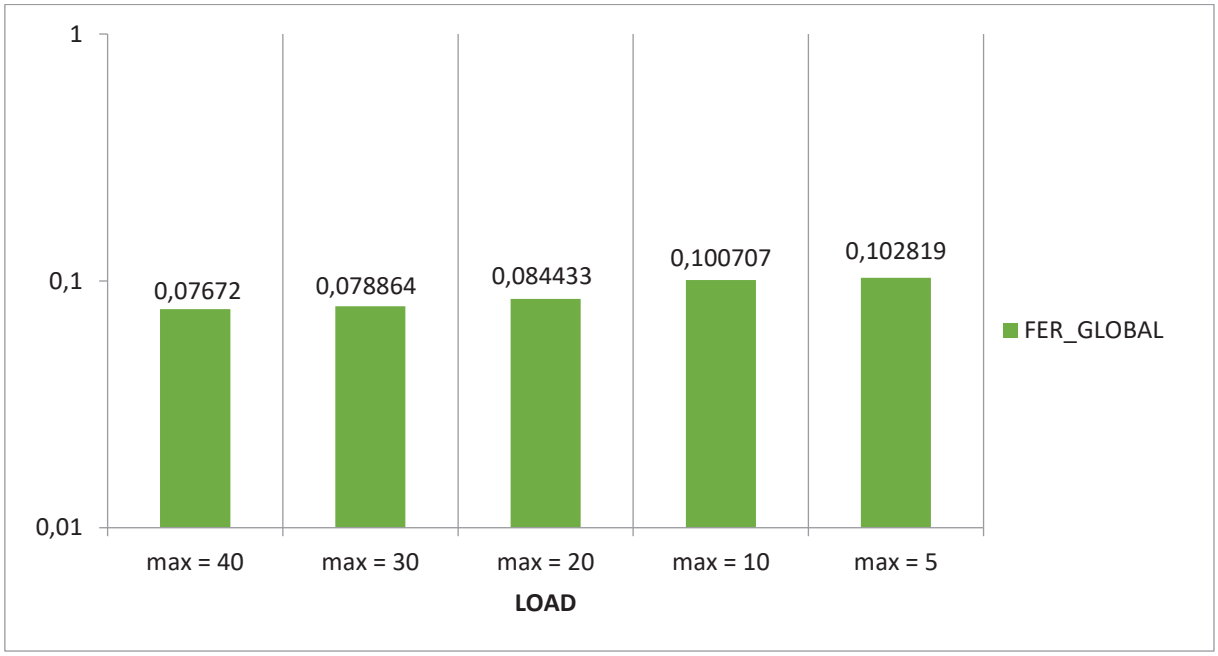

Figure 70: Analyse de performances par rapport à la charge réseau au niveau de RX1 avec rémanence sur le dernier canal de succès

Au niveau de RX2 (Figure 71), on voit que le FER croît progressivement lorsque la charge réseau augmente. On peut aussi simplement remarquer que le FER au niveau du récepteur RX2 est plus important par rapport à celui qu'on vient d'observer au niveau de RX1 (Figure 70), ceci s'explique simplement par le fait que le récepteur RX2 est soumis à plusieurs facteurs à la fois, en plus de ceux qu'on a cité au niveau de RX1, s'ajoute le problème de collision de trames. Comme nous l'avons signalé tout au début, le récepteur RX2 se trouve dans la portée radio de deux émetteurs TX1 et TX2, ces derniers peuvent parfois tirer le même slot et le même canal, par conséquent les signaux de leurs trames se chevauchent et deviennent indéchiffrables au niveau de RX2. Tout de même, le FER au niveau de RX2 est beaucoup plus réduit par comparaison à celui observé sur le testbed sans rémanence sur le canal de succès (Figure 68).

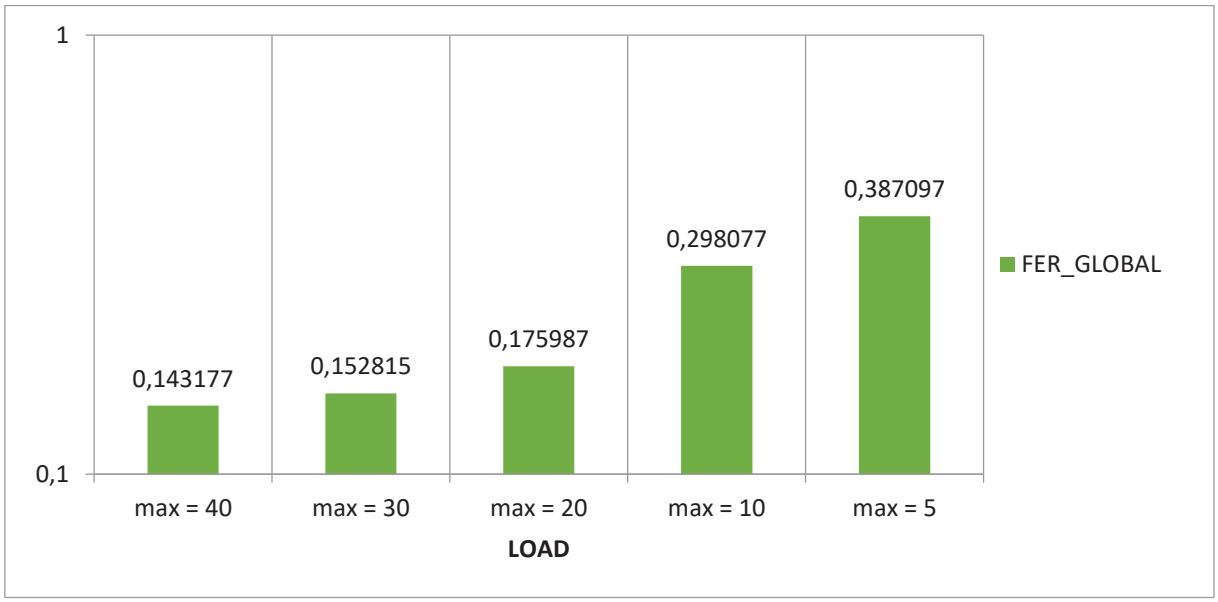

Figure 71: Analyse de performances par rapport à la charge réseau au niveau de RX2 avec rémanence sur le dernier canal de succès

3.7 Conclusion

Notre objectif primordial est de proposer une méthode d'accès multi-canal adaptée à une topologie multi-saut, qui passe à l'échelle. Cette étude nous a menée à prototyper premièrement une méthode d'accès sans fil MAC mono-canal basée sur *Slotted-Alloha*, ensuite nous avons étendu notre étude pour prototyper l'accès multi-canal multi-saut sans RDV qui a été améliorée par une stratégie de rémanence sur le dernier canal de succès.
D'après l'étude de performance que nous avons réalisée, même si l'accès mono-canal offre un FER meilleur que le multi-canal pour ce testbed avec seulement 4 nœuds WiNo et 2 canaux. La méthode d'accès multi-canal sans RDV que nous avons proposé sera une bonne solution pour les réseaux distribués multi-saut de grande taille et mieux encore lorsqu'une stratégie de rémanence sur le dernier canal de succès est utilisée, mais elle n'a pas un grand intérêt pour les réseaux de petite taille (problèmes de canal de réception), nous allons tester très prochainement la méthode proposée avec plusieurs nœuds et nous allons apporter évidement des améliorations à notre méthode d'accès multi-canal multi-saut sans RDV dans le futur.

D'après nos analyses de performances en comparant les deux méthodes d'accès et en se référant sur les différentes topologies que nous avons présentées, nous remarquons que la méthode d'accès mono-canal présente moins de taux d'erreur trame (à cause des collisions) par rapport à la méthode d'accès multi-canal dont le FER provient en grande partie du manque de canal de réception entre l'émetteur et le récepteur, lorsque la taille du réseau est très petite. Mais en présence d'un réseau de grande taille, la méthode d'accès mono-canal devient inefficace et surtout très pénalisante pour les transmissions multi-saut.

Par contre le taux d'erreur trame de la méthode d'accès multi-canal que nous avons étudié découle en grande partie par le manque du canal de réception et non principalement à cause des collisions des trames comme le cas de l'accès mono-canal. Ainsi donc nous pouvons dire que cette méthode d'accès multi-canal peut être très efficace en termes de taux d'erreur trame et surtout plus favorable aux les transmissions multi-saut.

3.8 Modèle analytique de la méthode d'accès multi-canal aléatoire sans Rendez-vous

Dans cette section, nous décrivons et analysons des modèles simplifiés du protocole Aloha slottée (*Slotted-Aloha*). L'objectif des modèles est de nous permettre de faire des comparaisons numériques des caractéristiques de performance du protocole dans des conditions de fonctionnement idéal (pas des perturbations des canaux) et toute émission est supposée reçue. La loi de Poisson semble bien adaptée pour modéliser ces types des problèmes à caractère aléatoire comme le nôtre. Elle est utilisée dans plusieurs travaux pour modéliser l'accès multi-canal [40] [67] [68], où chacun l'adapte selon l'objectif du contexte recherché.

Nous allons passer par deux étapes essentielles pour ce modèle considéré. Nous considérons premièrement un modèle à un saut où tous les nœuds sont à portée les uns des autres et dont nous cherchons la probabilité de succès et de collision globales pour tous les nœuds du réseau, ainsi que le débit global dans le réseau.

3.8.1 Modèle analytique mono-saut

On suppose que les canaux sont indépendants, les trames des données accèdent sur chaque canal de façon aléatoire avec la même probabilité.

$\mathbf{G_c}$ (la charge d'un canal) est le processus du taux d'arrivée sur chaque canal qui suit une distribution de Poisson avec une moyenne $\left(\frac{G}{C}\right)$ où **G** est la charge globale offerte à l'ensemble des canaux disponible pendant un slot de temps (durée d'une trame) ; ***C*** est le nombre des canaux disponible. Pour commencer notre étude analytique du protocole Aloha slottée multi-canal, on suppose que les trames sont générées par tous les nœuds du réseau à un taux fini total de **Gc** trames par slot de temps, nous considérons aussi que chaque nœud dans le réseau a toujours une trame à émettre pendant chaque slot de temps. Le nombre d'arrivé des trames par slot de temps est supposé obéir à la distribution de Poisson.

$$P(N=k)=\frac{(G_C)^k \exp(-G_C)}{k!} \tag{13}$$

Sous la condition de $N = k$, N est le nombre des k trames de données, qui ont été générées pendant une durée de slot de temps sur les ***C*** différents canaux de données disponibles. Notons aussi que pour des raisons de simplicité, on suppose qu'une trame sera reçue avec succès par le destinateur s'il n'y a pas de collisions pour cette trame.

En se référant au même modèle de l'Aloha slottée mono-canal (équation 13) et en supposant que le nombre de trames générées dans le réseau est une distribution de Poisson, nous considérons plusieurs canaux dans le réseau *(Nombre de canaux C < Nombre de nœuds N)*.

Ainsi, s'il y a ***C*** canaux dans le système réseau d'accès aléatoire multi-canal basé sur Slotted-Aloha, les sélections de canaux sont réalisées de façon aléatoire pour chaque transmission de trames. Puisque

la probabilité pour un nœud de choisir un canal est égale, chaque canal est sélectionné avec une probabilité qui est égale $\left(P_{canal} = \frac{1}{C}\right)$, le nombre d'arrivées des trames dans un slot de temps sur un canal obéit également à la loi de Poisson, ainsi, si la charge de trafic du réseau considérée est *G*, il n'y a pas des retransmissions des trames dans notre cas, alors chaque canal aura une charge moyenne offerte de G_C $\left(G_C = G * p_{canal} = \frac{G}{C}\right)$.

La probabilité pour qu'il n'y ait aucune collision (probabilité de succès) est en d'autre terme la probabilité qu'un nœud émet sur un canal pendant un slot de temps P_{succ} *(P_{succ} = P(N = 1). La* probabilité de l'ensemble des nœuds sur le système multi-canal dans un slot de temps selon l'équation 13 est :

$$p_{succ} = P(N = 1)$$

$$P_{succ} = G_C * \exp(-G_C) = \frac{G}{C} * \exp\left(-\frac{G}{C}\right) \quad (14)$$

Par conséquent, la probabilité qu'il y ait une collision ou échec de réception d'une trame émise avec succès sur le canal, mais non acquittée par le destinataire d'un nœud sur les ***C*** canaux dans un slot de temps est :

$$P_{succ} + P_{echec} = 1$$

$$P_{echec} = 1 - P_{succ}$$

$$P_{echec} = 1 - \frac{G}{C} * \exp\left(-\frac{G}{C}\right) \quad (15)$$

On peut alors calculer le débit utile d'un *canal i* unique D_i du système Slotted-Aloha (S-Aloha) est :

$$D_i = G_C * \exp(-G_C) = \frac{G}{C} * \exp\left(-\frac{G}{C}\right) \quad (16)$$

Il est à noter que, puisque tous les canaux sont indépendants, si l'émetteur a sélectionné le canal C1, il en est de même pour le récepteur qui doit se mettre en réception sur ce même canal C1. Ainsi donc, le débit du S-ALOHA multi-canal D_T (débit total) est la somme des débits pour chacun des canaux dans le réseau, ainsi :

$$D_T = \sum_{i=1}^{C} D_i = C * D_i = C * \frac{G}{C} \exp\left(-\frac{G}{C}\right) = G \exp\left(-\frac{G}{C}\right) \quad (17)$$

C'est le nombre moyen des trames transmises et reçues avec succès sur l'ensemble des canaux disponible pendant le slot de temps (durée d'une trame).

A partir de cette équation (17), nous allons effectuer un test pour observer le comportement du multi-canal à un saut en essayant de faire varier la charge et le nombre de canaux (Figure 72), nous avons fait varier le nombre des canaux jusqu'à 16 pour tenir compte du nombre utilisés dans certains normes

ISM *(Indutrial Scientific and Medical)*. Pour tracer ce graphique, nous avons écrit un code sous Python en exploitant ses deux modules essentiels : *Numpy* et *Matplotlib*, le module *Numpy* permet d'effectuer des opérations des calculs sur des vecteurs ou des matrices, il contient aussi certaines fonctions des bases pour manipuler de l'algèbre linéaire ou encore des tirages des nombres aléatoires avec le module *random*, tandis que le module *matplotlib* est un outil complémentaire de *Numpy* permettant de tracer des graphes interactifs depuis Python.

Dans notre code pour tracer le graphique nous avons utilisé la fonction *linspace( )* à laquelle on donne des bornes et un nombre des valeurs attendues, ce qui nous permet de spécifier le nombre des points en abscisse. Avec cette fonction, nous pouvons choisir la quantité de charge que nous désirons. Ensuite, l'appel de la fonction*plt.plot( )* va créer automatiquement la courbe avec les axes nécessaires afin d'obtenir les graphiques attendus. Enfin, l'appel de la fonction *plt.legend()* sans arguments récupère automatiquement les descripteurs des légendes et leurs étiquettes associées, et la fonction *plt.show()* affiche la figure à l'écran.

On peut voir sur le graphique que lorsqu'on a un seul canal, le débit utile se dégrade très rapidement avec l'augmentation de la charge. Mais on peut remarquer également que le débit s'améliore lorsqu'on augmente petit à petit le nombre de canaux ; bien évidemment il décroît dans tous les cas avec l'augmentation de la charge. Par exemple, quand on observe le débit lorsque la charge varie entre 10 et 20, en utilisant 16 canaux, on remarque que le débit croit jusqu'à la valeur de la charge égale à 20, puis commence à chuter légèrement lorsque la charge augmente. Ceci s'explique par la simple raison que dans cet intervalle de charge, le nombre de canaux est important pour écouler suffisamment les trafics. Au-delà de cet intervalle, les canaux disponibles ne pourront pas suffire pour supporter toute la charge du réseau et par conséquent des collisions pourront se produire.

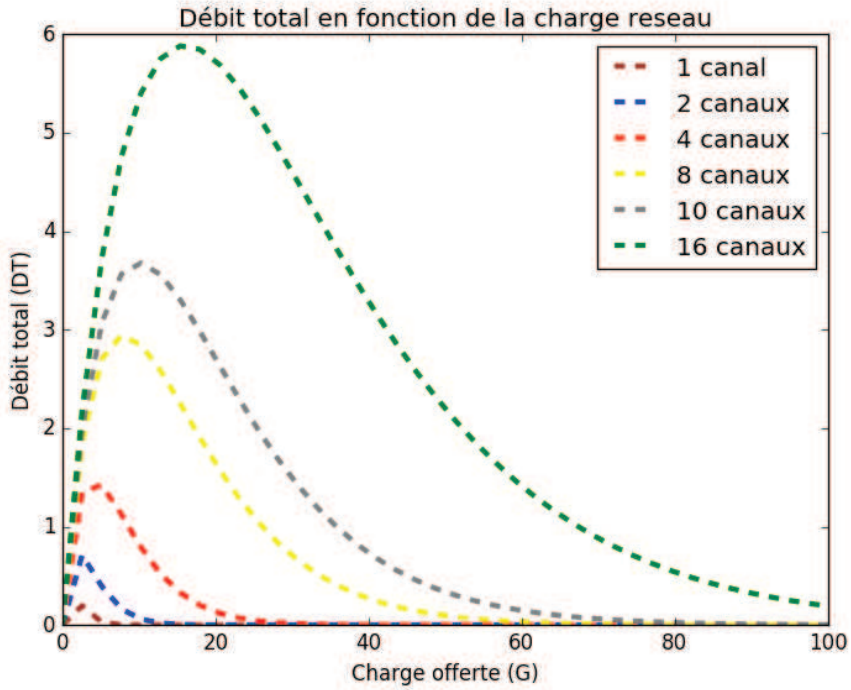

Figure 72 : Variation du débit en fonction du nombre des canaux et de la charge

Considérons la figure 73 suivante sur laquelle nous analysons le comportement analytique de 4 nœuds qui représente le contexte de notre modèle d'étude réel précédent réalisé avec les nœuds WiNo. À la différence du *testbed* étudié, dans ce modèle analytique, la topologie des nœuds est supposée mono-

sauts. Il est facilement remarquable sur la figure 73 que le débit est toujours meilleur en multi-canal (2 canaux) qu'en mono-canal. Comme on peut aussi facilement remarquer sur cette figure que ce débit s'affaiblit avec l'augmentation de la charge dans le deux cas mono-canal et multi-canal.

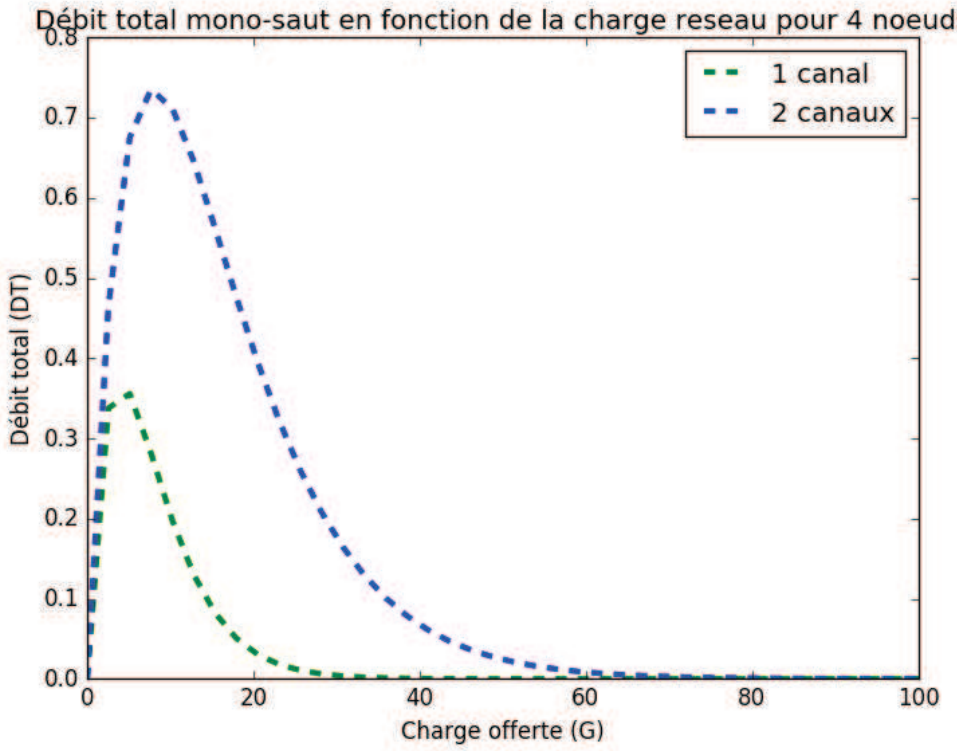

Figure 73 : Variation du débit en fonction du nombre des canaux et de la charge pour 4 nœuds

3.8.2 Modèle analytique multi-saut

Pour modéliser le contexte multi-canal multi-saut, nous pensons à organiser notre réseau par groupe des nœuds qui sont interconnectés par quelques autres nœuds appartenant aux deux groupes voisins l'un de l'autre.

Contrairement au contexte mono-saut et en utilisant la même équation (13), nous allons chercher la probabilité que tous les canaux soient selectionnés sauf le canal utilisé par le récepteur (qu'aucune de ces k trames des données ne prenne le même canal que le nœud en réception) , puisqu'il y a *C* canaux de données dans le réseau et comme chaque nœud sélectionne son canal aléatoirement et indépendamment. Ainsi la charge offerte sur le *canal i* (C_i) sera alors $G_i = G * \left(\frac{C-1}{C}\right)$, la probabilité qu'une trame soit bien reçue, en d'autres termes la probabilité que le canal de réception et non utilisé soit définie par :

$$P(N=k) = \frac{\left[G\left(\frac{C-1}{C}\right)\right]^k \exp\left(-G\left(\frac{C-1}{C}\right)\right)}{k!} \qquad (18)$$

Ainsi, la probabilité de succès sur le canal utilisé par le nœud récepteur, mais où il peut y avoir des succès ou échec sur l'ensemble d'autres canaux restant peut être écrite par l'équation suivante (19) :

$$P_{succ\_canal} = P(N=1) = G\left(\frac{C-1}{C}\right)\exp\left(-G\left(\frac{C-1}{C}\right)\right) \qquad (19)$$

Nous considérons la topologie de la figure suivante (Figure 74) pour réprèsenter le contexte multi-saut simple. Tous les nœuds à portée (nœuds voisins) forment un groupe g_i (vert ou rouge), et chacun des groupes est interconnecté à l'autre par au moins un nœud commun (nœud rouge) donc se trouvant dans la portée de deux groupes. Bien évidemment, la probabilité de succès dépend du nombre des voisins du nœud, ceci laisse comprendre que plus le nombre des voisins du nœud est important, plus la probabilité de succès est peu probable.

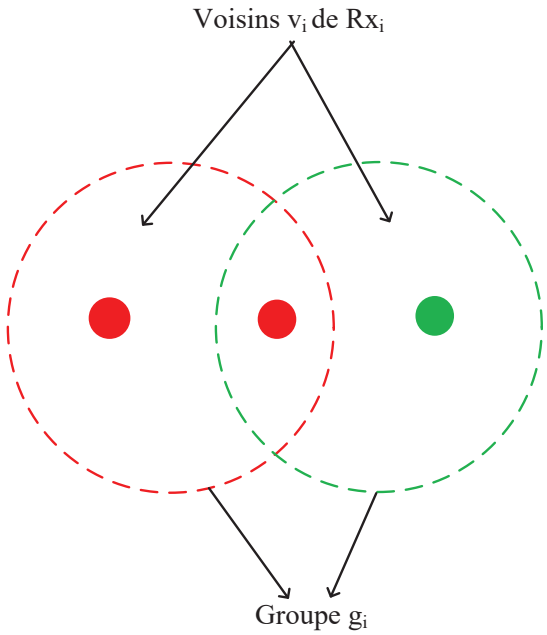

Figure 74 : Topologie du modèle analytique multi-saut

Nous pouvons ainsi déduire la probabilité de succès multi-saut à partir de l'équation (19). Nous notons v_i pour désigner le nombre des voisins de chaque nœud, on peut ainsi formuler cette probabilité de la maniére suivante :

$$P_{succ\_n\_sauts} = P_{succ\_canal}\left(1-P_{succ\_canal}\right)^{v_i-1}$$

$$P_{succ\_n\_sauts} = G\left(\frac{C-1}{C}\right)\exp\left(-G\left(\frac{C-1}{C}\right)\right) * \left[1-G\left(\frac{C-1}{C}\right)\exp\left(-G\left(\frac{C-1}{C}\right)\right)\right]^{v_i-1} \qquad (20)$$

Pour cette probabilité à n sauts avec $v_i - 1$, nous considérons que l'émetteur n_i émet sa trame avec succès ; et aucun nœud voisin dans la zone de portée de la destination de cette trame dans le groupe g_i ne transmet sur ce canal.

Ainsi donc d'après l'équation (17) nous pouvons déduire le débit total par l'équation suivante (21)

$$D_T = \sum_{i=1}^{C} D_i = C * G\left(\frac{C-1}{C}\right)\exp\left(-G\left(\frac{C-1}{C}\right)\right) * \left[1 - G\left(\frac{C-1}{C}\right)\exp\left(-G\left(\frac{C-1}{C}\right)\right)\right]^{v_i - 1}$$

$$D_T = G(C-1)\exp\left(-G\left(\frac{C-1}{C}\right)\right) * \left[1 - G\left(\frac{C-1}{C}\right)\exp\left(-G\left(\frac{C-1}{C}\right)\right)\right]^{v_i - 1} \quad (21)$$

Contrairement à la figure précédente (Figure 73), dans ce modèle de la figure suivante (Figure 75), nous tenons compte du nombre des voisins d'un potentiel récepteur, ou encore du nombre du saut (Figure 74) en se référant sur nos topologie du *testbed* précédemment réalisées avec 4 nœuds où 2 émetteurs se trouvent à 2 sauts l'un de l'autre et un récepteur qui est voisin de chacun de deux émetteurs, il apparait clairement que le débit est toujours meilleur en multi-canal qu'en mono-canal, mais on peut aussi bien remarquer que, lorsqu'il s'agit du multi-saut alors on remarque que le débit se dégrade par rapport au contexte mono-saut observé précédemment (Figure 73). Ce dernier sera bien évidemment impacté par le nombre des voisins du nœud sur chaque saut à effectuer par les trames afin d'atteindre leurs destinations.

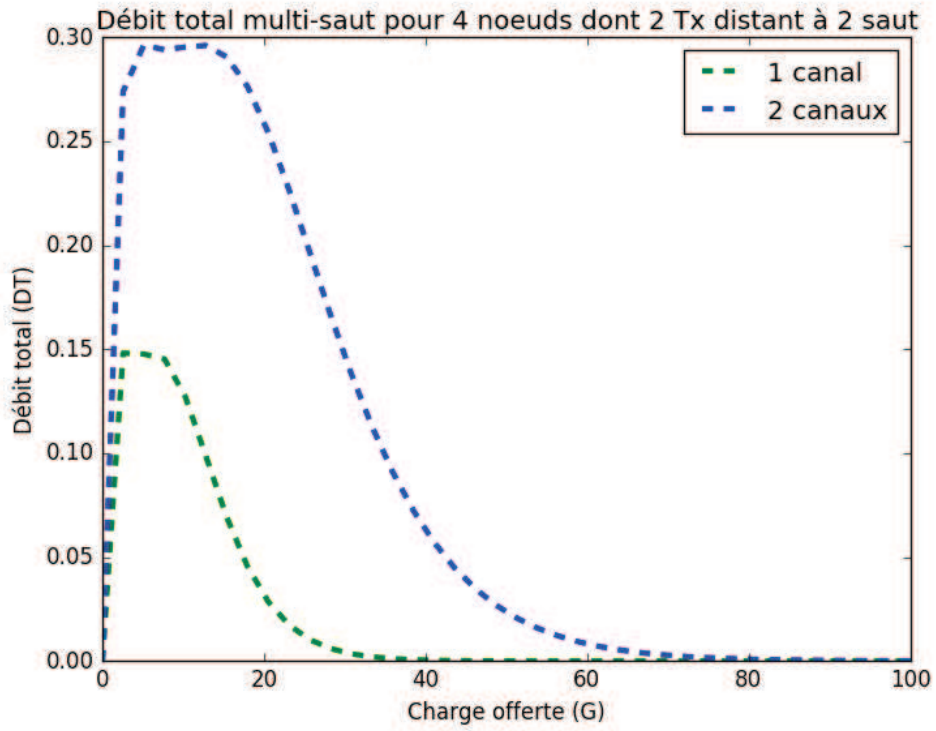

Figure 75 : Variation du débit en fonction du nombre des canaux et de la charge pour 4 nœuds, cas d'un récepteur voisin de deux potentiels émetteurs

Nous souhaitons, finalement, comparer nos résultats analytiques aux résultats que nous obtenons avec notre testbed. Nous observons la figure 76 correspondant au *testbed* avec 4 nœuds pour une topologie à 2 sauts dont le récepteur Rx2 est voisin de deux émetteurs distants de 2 sauts l'un de l'autre. Rx2 peut potentiellement recevoir de chacun des deux émetteurs s'ils ne sélectionnent pas le même canal d'émission en même temps. Par comparaison au modèle analytique, sur cette figure 76, le débit est une fonction du taux d'utilisation des canaux. Plus les canaux sont sollicités, plus la quantité de données reçues est importante, alors que le débit régresse. Tel est aussi le cas du modèle analytique, lorsque la charge du réseau augmente le débit s'affaibli. Nous pouvons également remarquer que dans les deux contextes mono-canal et multi-canal, le débit est toujours influencé par le nombre des voisins du nœud récepteur, ceci peut s'observer sur la courbe bleu (Figure 76) du récepteur Rx2 voisin de deux nœuds émetteurs par rapport au récepteur Rx1 qui est voisin d'un seul émetteur. Ceci est toujours prouvé par le modèle analytique, même si le multi-canal permet d'améliorer le débit, nous avons constaté à travers les différents cas des modèles analysés, que le nombre des voisinages et les transmissions multi-sauts affectent le débit espéré.

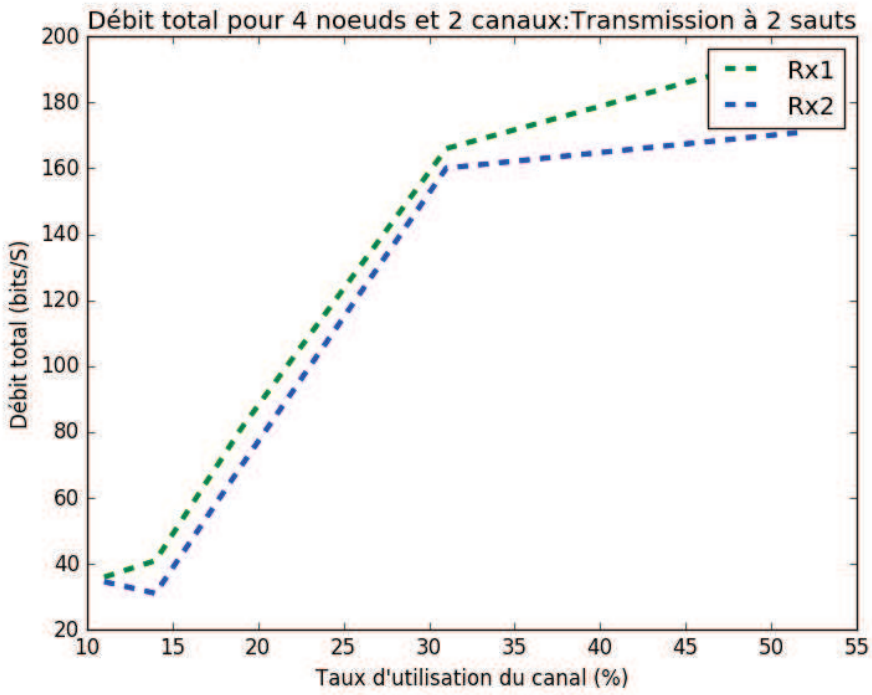

Figure 76 : Variation du débit du *testbed* par rapport au taux d'utilisation de 2 canaux pour 4 nœuds, cas d'un récepteur voisin de deux potentiels émetteurs

Pour conclure, nous pouvons dire que les modèles analytiques (comme les simulateurs) ne reflètent pas exactement les comportements réels des méthodes d'accès pour les réseaux locaux et personnels sans fil, surtout lorsqu'il s'agit des réseaux ad hoc (*mesh*, distribué…). La raison est que les méthodes analytiques ne tiennent pas compte de beaucoup de facteurs qui sont très importants pour déterminer les performances réels d'un réseau quelconque sans fil. Nous pouvons par exemple citer quelques-unes de ces raisons face auxquelles nous avons rencontré des difficultés pour réaliser nos *testbed* : le problème d'asymétrie entre les nœuds, les perturbations des ondes radio par les phénomènes externes, des difficultés provenant des positions des antennes radio… Bien évidemment, il existe autres problèmes liés à la réalité. Les modèles analytiques nous permettent de fixer les idées sur le comportement attendu du phénomène à étudier, ainsi donc, étudier et tester un modèle analytiquement est initialement primordiale et complémentaire d'un futur testbed.

3.9 Méthode d'accès multi-canal avec stratégie de rémanence testbed Ophelia

Le projet Ophelia est une plateforme permettant le déploiement d'un grand nombre des nœuds et aussi des outils de programmation à distance. Ceci simplifie le testbed, mais permet également d'effectuer plusieurs tests sans mettre assez du temps. Les nœuds sont connectés en réseau sans fil via des Raspberry Pi qui nous permettent des récupérer nos données et la mise à disposition des résultats d'expérimentations sous forme des fichiers textes.

Pour finaliser l'étude de performance de notre méthode d'accès multi-canal, nous avons augmenté le nombre des nœuds du réseau qui est passé de 4 à 11 nœuds, dont un nœud émetteur des *beacons*. Les nœuds sont disposés aléatoirement dans le bâtiment de recherche. La position du nœud *synchronisateur* a été choisie à partir de plusieurs expérimentations de telle sorte qu'il puisse atteindre

tous les nœuds du bâtiment en utilisant une forte puissance. La puissance d'émission des données et des acquittements a été sélectionnée d'après plusieurs expérimentations en parcourant les valeurs de la puissance de 2 à 20 dBm. A cet effet, nous avons établi une matrice d'adjacence pour tous les nœuds, où chaque nœud émet un certain nombre des trames déterminé vers tous les autres, afin de voir quels nœuds seront atteints. Finalement, nous avons opté pour la valeur 2 dBm afin d'éviter la topologie d'un réseau totalement maillé (full *mesh*) qui ne permet pas l'évaluation des performances d'un réseau multi-saut que nous voulons étudier.

La disposition des nœuds pour le testbed est représentée sur la figure 77 qui est le schéma du plan du bâtiment des recherches sur lequel sont fixés les nœuds WiNo, pilotés à distance via le réseau sans fil. Les numéros en rouge représentent les nœuds WiNo, les numéros en forme de rectangle identifient les bureaux dans le bâtiment. La figure 78 montre les images, les positions pour illustrer comment les nœuds WiNo sont fixés.

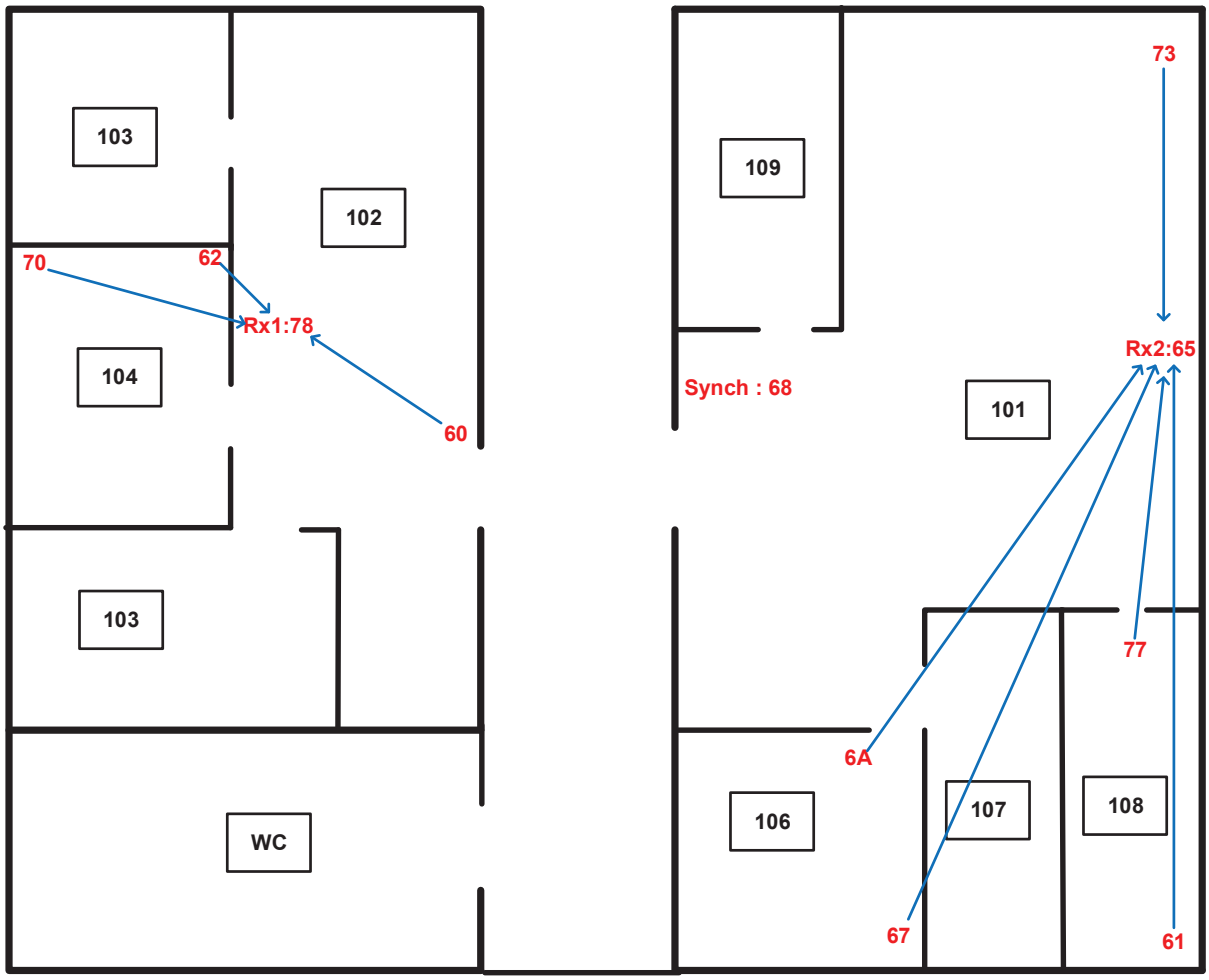

Figure 77 : Répartition des nœuds WiNo dans le bâtiment de la recherche

Figure 78: Image montrant la disposition des nœuds

3.9.1 Intérêt de l'accès multi-canal par rapport au mono-canal

3.9.1.1 Cas d'un récepteur unique et de plusieurs émetteurs

Pour cette expérimentation, nous utilisons un récepteur (1 Rx), et neuf émetteurs (9 Tx), qui émettent vers ce même récepteur. Nous avons également un nœud qui émet les *beacon* (Figure 77 ; **Synch : 68**) pour indiquer le début du slot, toutes les 900 ms. A tour de rôle, chacun des 10 nœuds devient récepteur alors que les autres sont des émetteurs. Il faut noter que, même si tous les nœuds envoient vers le même récepteur, certains ne portent pas directement vers ce récepteur, limités par la puissance

du signal ou les obstacles (les murs, les armoires… dans notre testbed). Lorsqu'une collision se produis, nous utilisons le même *backoff* (en utilisant la fonction *Random* entre deux valeurs) non exponentiel précédemment utilisé, qui est le paramètre important de notre étude de performance et nous permet de charger progressivement le réseau. Ainsi donc, plus l'intervalle entre les deux valeurs du *Random* est petit, plus le réseau est chargé, et par conséquent plus les collisions sont importantes. Nous calculons la valeur moyenne du taux d'erreur trame (*FER*) de l'ensemble des nœuds. Nous procédons ensuite de la même façon, mais en utilisant deux récepteurs. Chaque émetteur choisi l'un des récepteurs et émet sa trame. Dans chacun de ces deux contextes (un récepteur et deux récepteurs), nous réalisons des tests en utilisant 1 canal, ensuite 2, et enfin 3 canaux.

L'objectif pour lequel nous avons utilisé un seul récepteur et plusieurs émetteurs, dont certains ne portent pas nécessairement vers le récepteur, a deux raisons essentielles : la première raison est pour montrer l'intérêt de la transmission multi-saut, puisque certains émetteurs tentent de joindre un destinateur qui n'est pas à leur portée. Ces trames sont perdues car non reçues par le récepteur, donc ne seront pas acquittées. La deuxième raison est pour observer également l'impact du nombre de voisins. Plus ce nombre est important, plus il y aura de collisions. Généralement, on observe ainsi un taux d'erreur trame (*FER*) plus important, ce dernier sera comparé en fonction du nombre de canaux. On considère comme négligeable le FER lié à de simples perturbations radio, ce qui a été constaté dans des tests précédents sans collision. Enfin, nous ajoutons un deuxième récepteur pour identifier la nécessité du multi-saut et de la même façon comparer le *FER* en fonction du nombre de canaux et le comparer avec le cas lorsqu'on a un seul récepteur.

On peut voir sur la Figure 79 que lorsqu'un seul canal est disponible pour 9 Tx qui envoient vers la même destination, le *FER* augmente progressivement en fonction du taux d'utilisation du canal (augmentation de la charge du réseau). On remarque que, plus le canal est sollicité, plus le *FER* est important et donc dégradé.

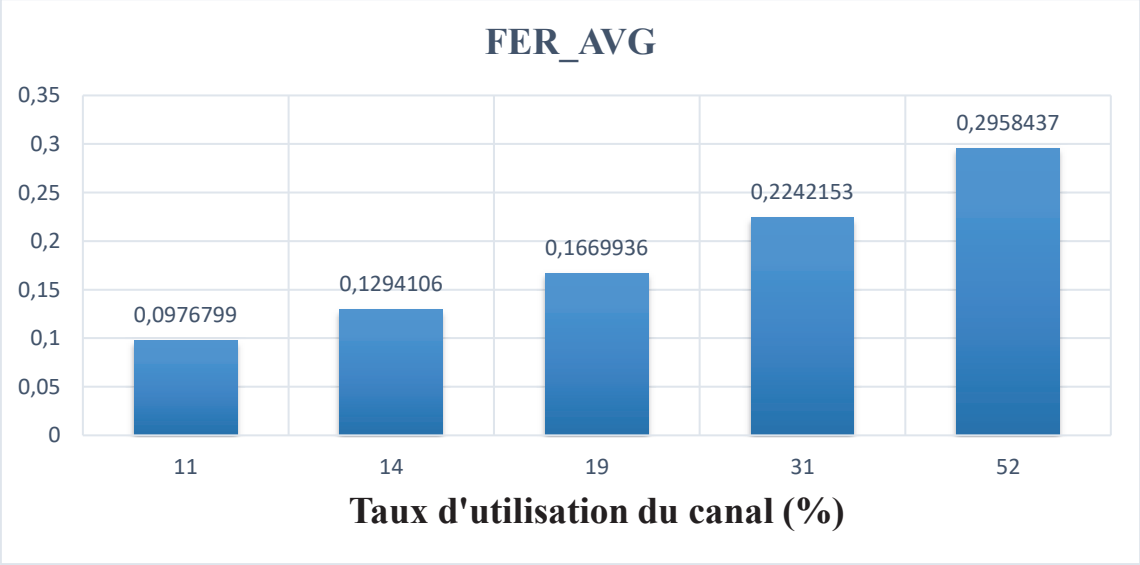

Figure 79 : Taux d'erreur trame moyen (*FER_AVG*) en utilisant 1 canal, 9 Tx et 1 Rx

Mais, le *FER* avec un seul canal est aussi plus important que celui que l'on observe sur la figure 80 avec 2 canaux, quel que soit le taux d'utilisation des canaux, puisqu'il y a le facteur probabiliste qu'un nœud sélectionne ou pas un canal libre du nœud récepteur. Sa trame sera ainsi reçue avec succès sans collision.

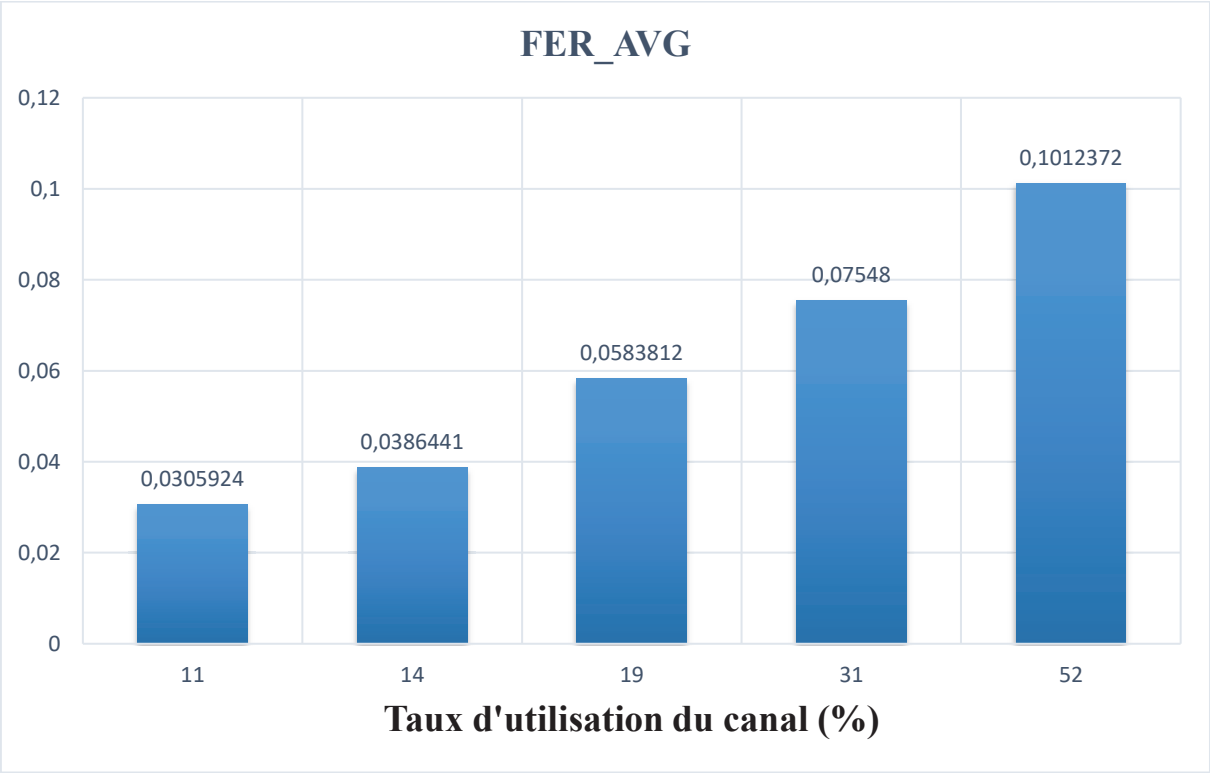

Figure 80 : Taux d'erreur trame moyen (*FER_AVG*) en utilisant 2 canaux, 9 Tx et 1 Rx

On peut observer une amélioration du *FER* lorsqu'on ajoute un canal supplémentaire (figure 81). On dispose alors de 3 canaux. Si nous comparons les résultats de la figure 81 (3 canaux) par rapport à ceux de la figure 80 (2 canaux), nous observons donc une amélioration même si le réseau est très surchargé.

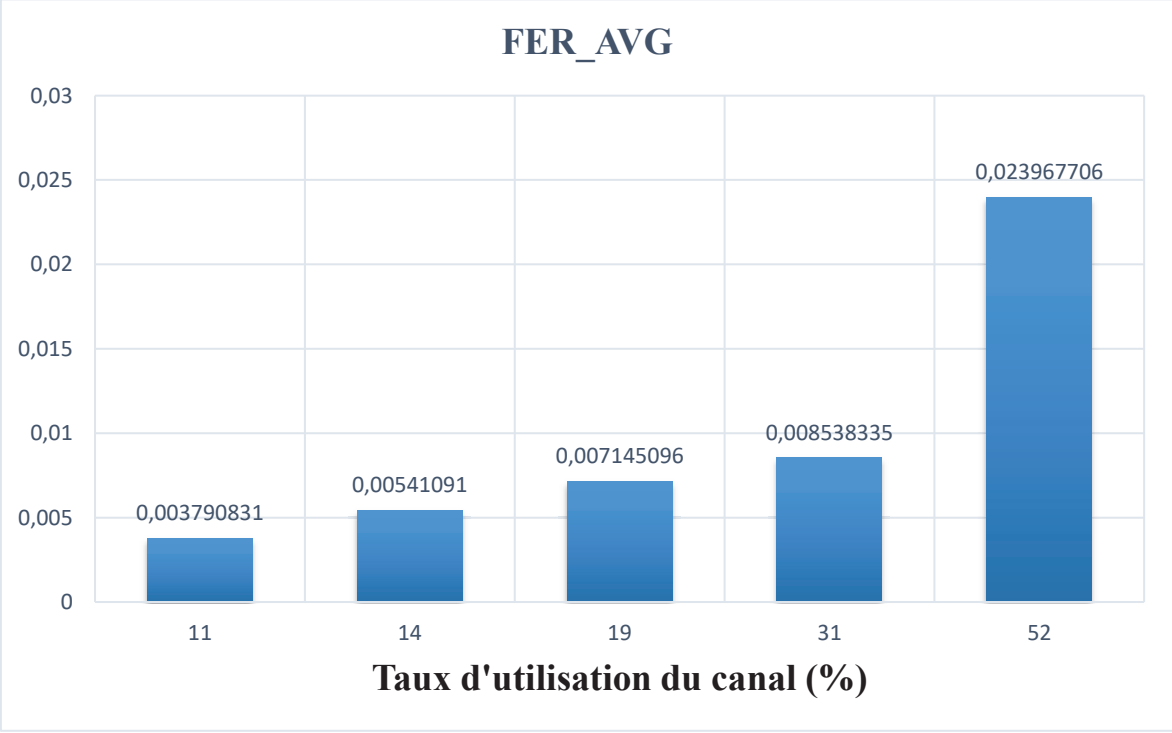

Figure 81 : Taux d'erreur trame moyen (*FER_AVG*) en utilisant 3 canaux, 9 Tx et 1 Rx

3.9.2 Cas de deux récepteurs et plusieurs émetteurs (contexte multi-saut)

Nous avons vu que, plusieurs transmissions simultanées sur un seul même canal vers une même destination (cas de 9 Tx et 1 Rx) provoquent une collision au niveau de la destination et par conséquent, aucune trame valide provenant des émetteurs ne sera correctement reçue par le récepteur. Nous ajoutons donc un autre Rx. Chaque Tx sélectionne une des 2 Rx puis émet sa trame. Nous procédons à des tests avec 1 canal, puis 2 et 3 canaux. Dans le contexte mono-canal, comme nous l'avons déjà constaté (chapitre 3), lorsque les couples Tx/Rx ne se trouvent pas dans une même zone de portée, il n'y a généralement pas de conflits de transmissions. Le conflit survient, lorsqu'un Rx se trouve dans la portée de plusieurs émetteurs potentiels. Dans le contexte multi-canal, on peut avoir deux émissions simultanées sur deux canaux différents vers les deux Rx. Les Tx et Rx peuvent partager la même zone de portée, et leurs trames seront reçues avec succès sur des canaux différents. Il est aussi possible que le couple Tx/Rx puissent partager le même canal sans collision s'ils sont hors de portée (réutilisation spatiale), d'où l'intérêt du multi-canal pour l'approche multi-saut.

Nous remarquons que, en utilisant deux Rx et 1 canal (Figure 82 et Figure 83), on obtient un *FER* qui croit progressivement en fonction de l'augmentation de la charge réseau, mais nous observons que le *FER* est bien réduit par rapport au cas avec 1 canal et un Rx. Ceci peut être observé sur les deux graphiques des deux Rx. Si nous comparons les deux graphiques, on observe que le *FER* au niveau de la figure 83 est plus important que celui de la figure 82, ceci s'explique simplement par le fait que le Rx représenté dans la figure 83 a plus de voisins par rapport à celui utilisé pour la figure 82.

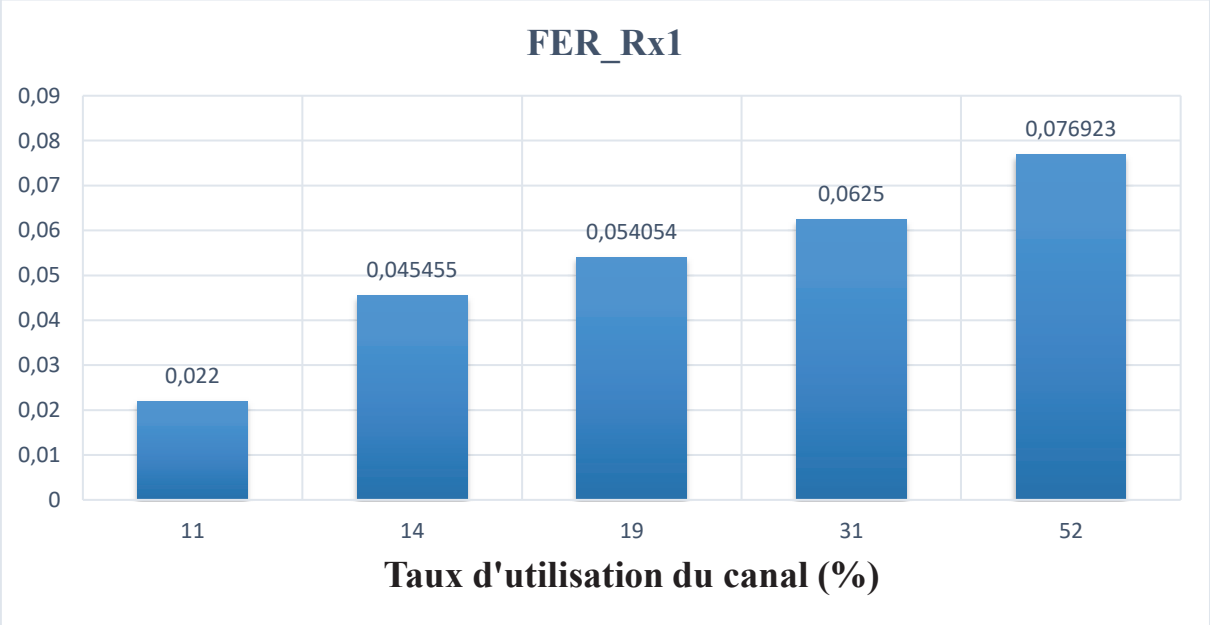

Figure 82 : Taux d'erreur trame en utilisant 1 canal, 8 Tx et 2 Rx (Rx1 : 78)

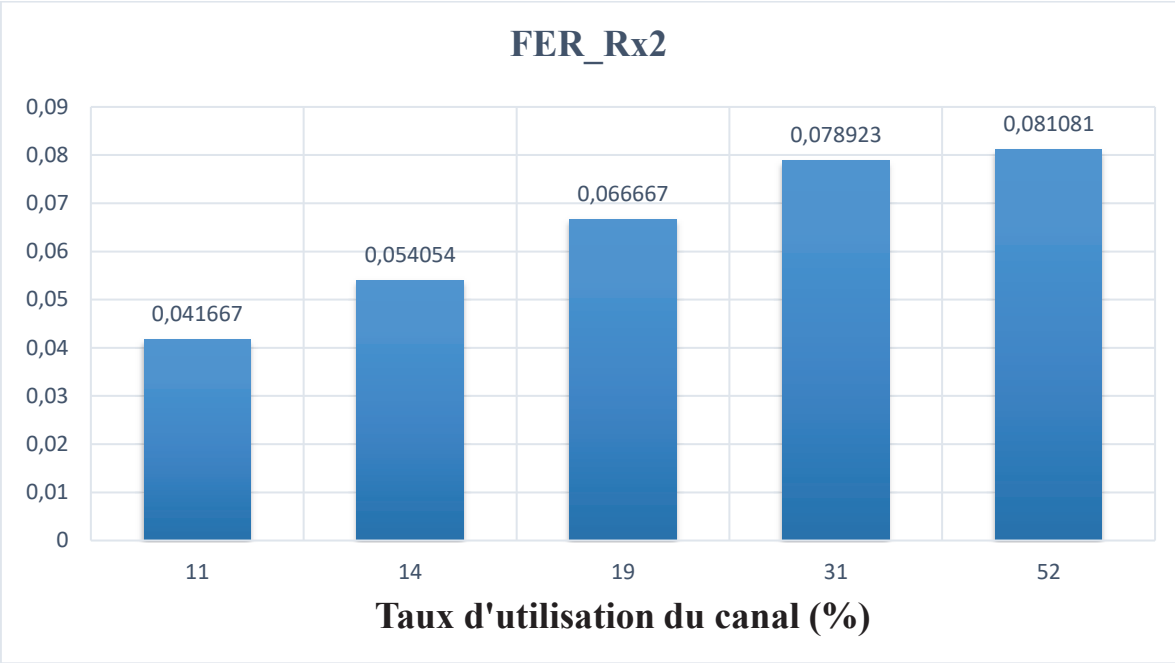

Figure 83 : Taux d'erreur trame en utilisant 1 canal, 8 Tx et 2 Rx (RX2 : 65)

Sur la figure 84**,** le FER moyen en utilisant 1 canal et 2 Rx, est meilleur que le FER moyen avec 1 canal et 1 Rx (Figure 79).

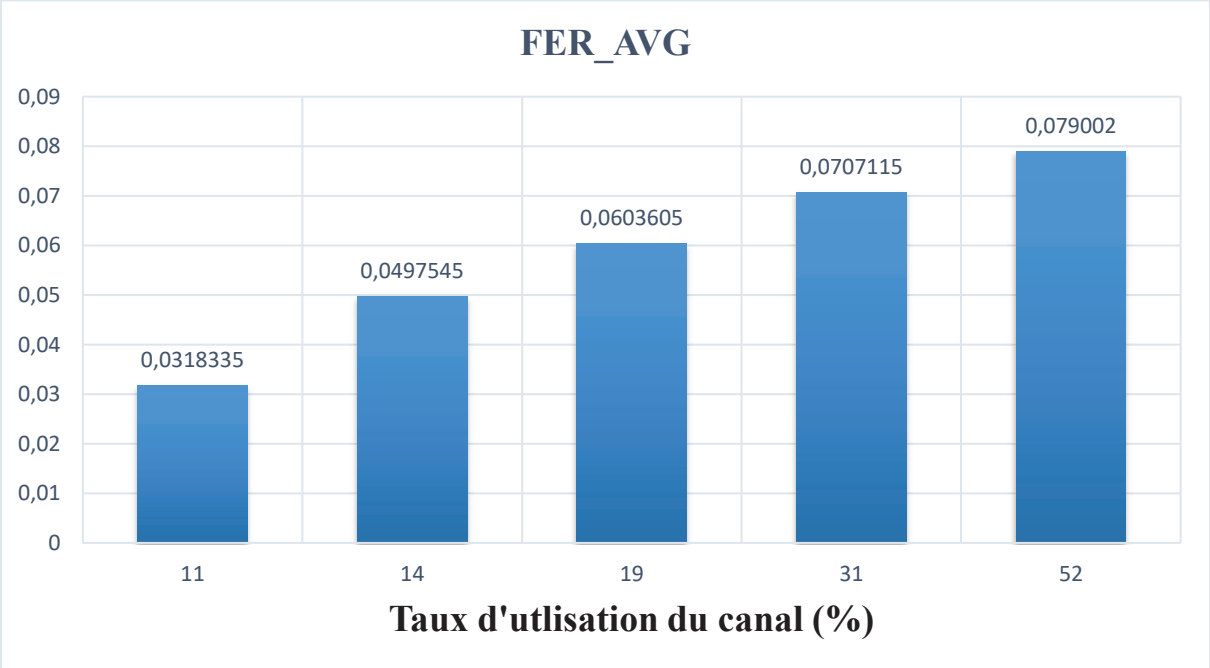

Figure 84 : Taux d'erreur trame moyen en utilisant 1 canal, 8 Tx et 2 Rx

En gardant la même disposition des nœuds (Tx et Rx), et en utilisant un second canal, nous avons procédé à un test avec 2 canaux ; on remarque clairement que le *FER* croit progressivement avec l'augmentation de la charge, mais ce dernier est bien amélioré par rapport au cas avec un seul canal. Cette amélioration du *FER* est bien observée au niveau des graphiques des deux récepteurs Rx1 et

Rx2, mais l'influence du nombre de voisins est encore observée. On voit sur le graphique de Rx1 (Figure 85) qui a moins de voisins qui portent vers lui, que le FER est inférieur quel que soit le taux d'utilisation du canal par rapport à celui observé sur le graphique de Rx2 (Figure 86) qui reçoit de plus de voisins que Rx1. Par conséquent, il y aura plus de collisions par comparaison à Rx1.

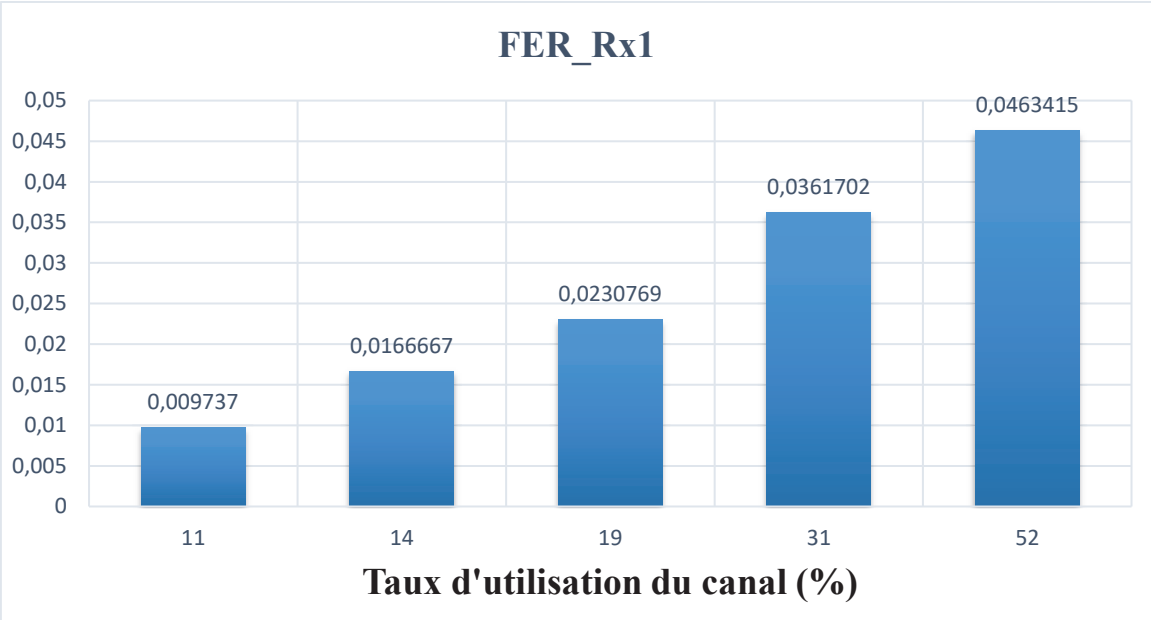

Figure 85 : Taux d'erreur trame en utilisant 2 canaux, 8 Tx et 2 Rx (RX1 : 78)

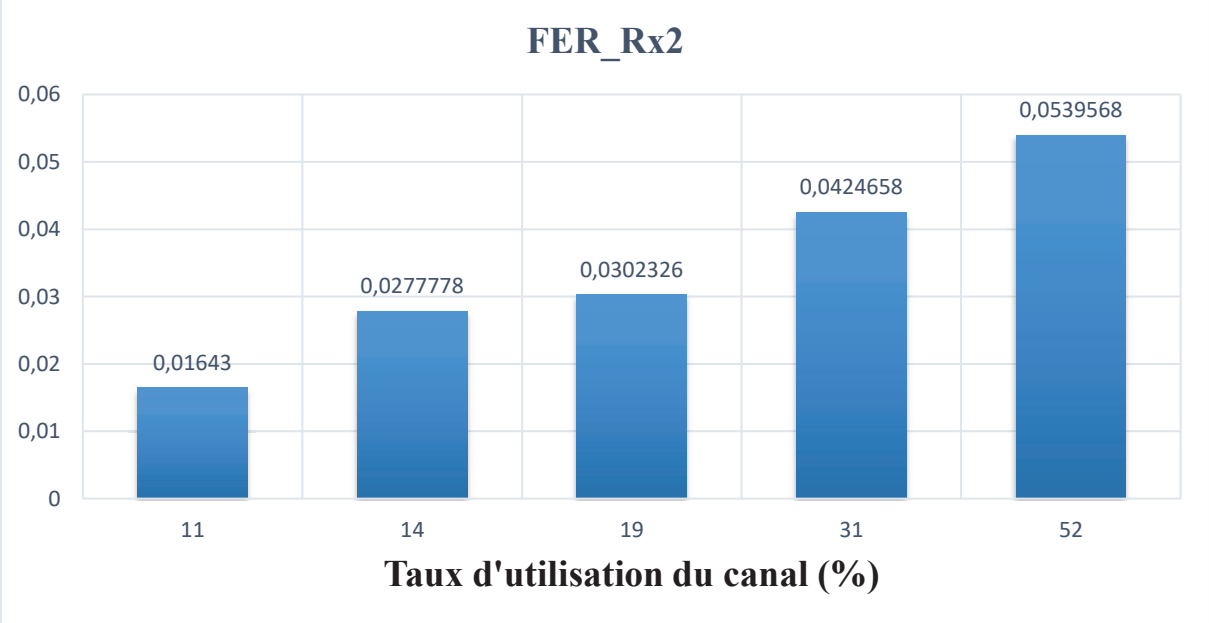

Figure 86 : Taux d'erreur trame en utilisant 2 canaux 8 Tx et 2 Rx (RX2 : 65)

De la même façon, nous comparons le FER moyen, lorsqu'on utilise 2 canaux et 2 Rx (Figure 87) avec le cas où il y a seulement 1 Rx (Figure 80). On observe aussi clairement qu'on a encore un FER moyen plus réduit lorsqu'on utilise 2 Rx par rapport au cas avec 1 Rx.

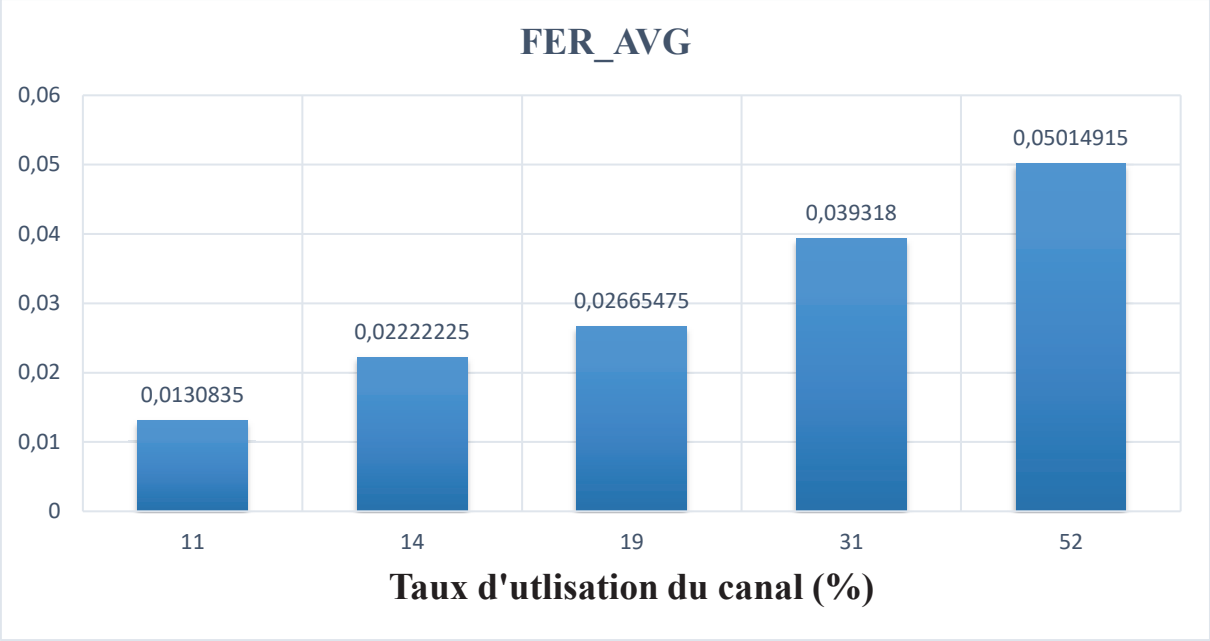

Figure 87 : Taux d'erreur trame moyen en utilisant 2 canaux, 8 Tx et 2 Rx

Nous ajoutons alors un canal de plus pour passer à 3 canaux (Figure 88 et Figure 89). On remarque de nouveau que le FER décroît sur toutes les valeurs du taux d'utilisation du canal de façon très significative par rapport aux autres cas analysés précédemment (Figure 86). L'ajout d'un canal réduit le *FER*, mais on peut toujours observer que le nombre de voisinage a une influence sur le *FER*. Sur le graphique de la figure 89 où le récepteur Rx2 a plus des voisins que Rx1, les valeurs du *FER* sont plus importantes quel que soit le taux d'utilisation du canal par rapport à celles observées au niveau du récepteur Rx1.

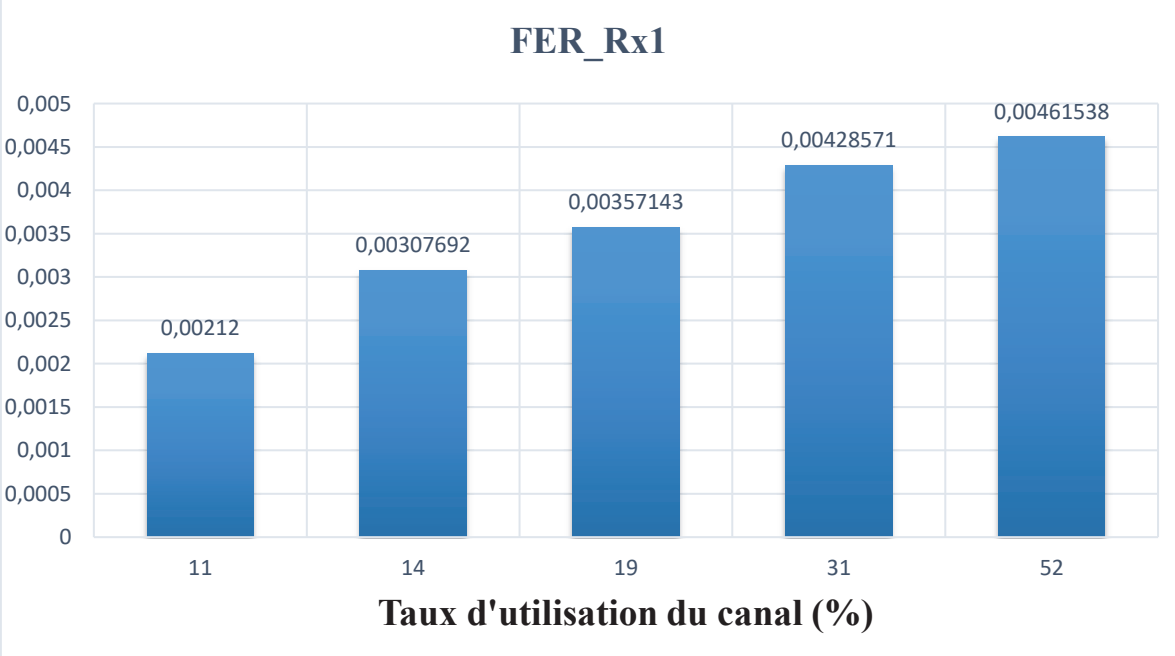

Figure 88 : Taux d'erreur trame en utilisant 3 canaux, 8 Tx et 2 Rx (RX1 : 78)

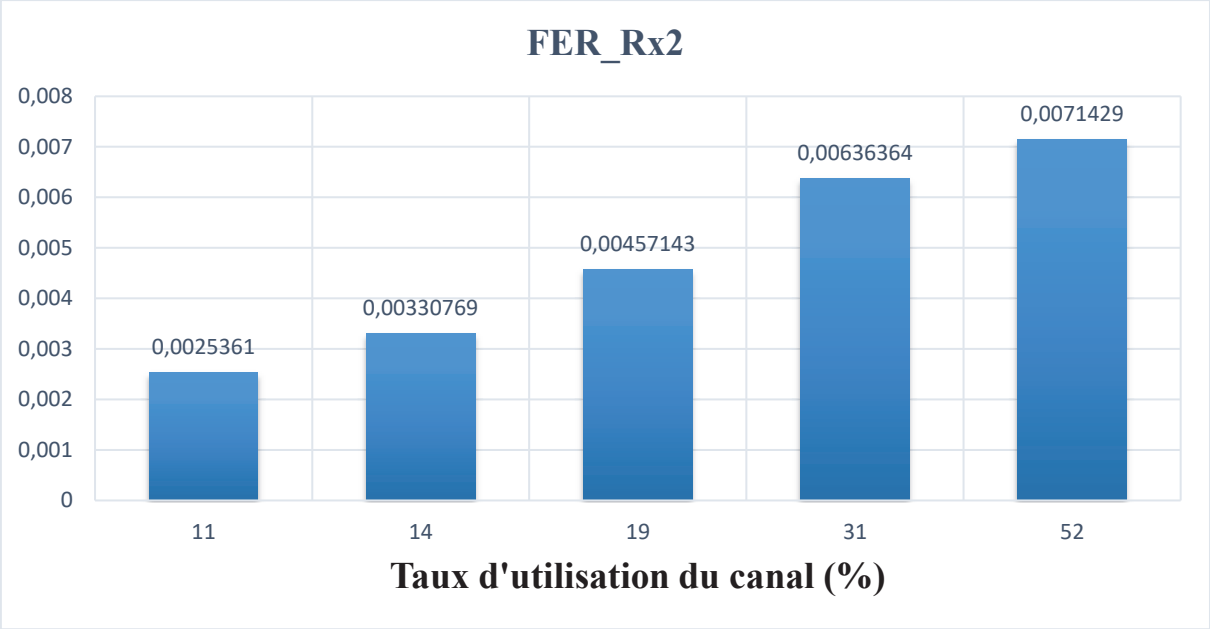

Figure 89 : Taux d'erreur trame en utilisant 3 canaux 8 Tx et 2 Rx (RX2 : 65)

Plus on augmente le nombre des canaux, meilleure est la valeur moyenne du FER. Le graphique de la figure 90 représente le FER moyen en utilisant 2 Rx et 3 canaux, par comparaison à celui où on utilise seulement 1 Rx et 3 canaux (Figure 34). Ce graphique fournit une valeur significative du FER moyen, même avec la valeur la plus importante du taux d'utilisation du canal.

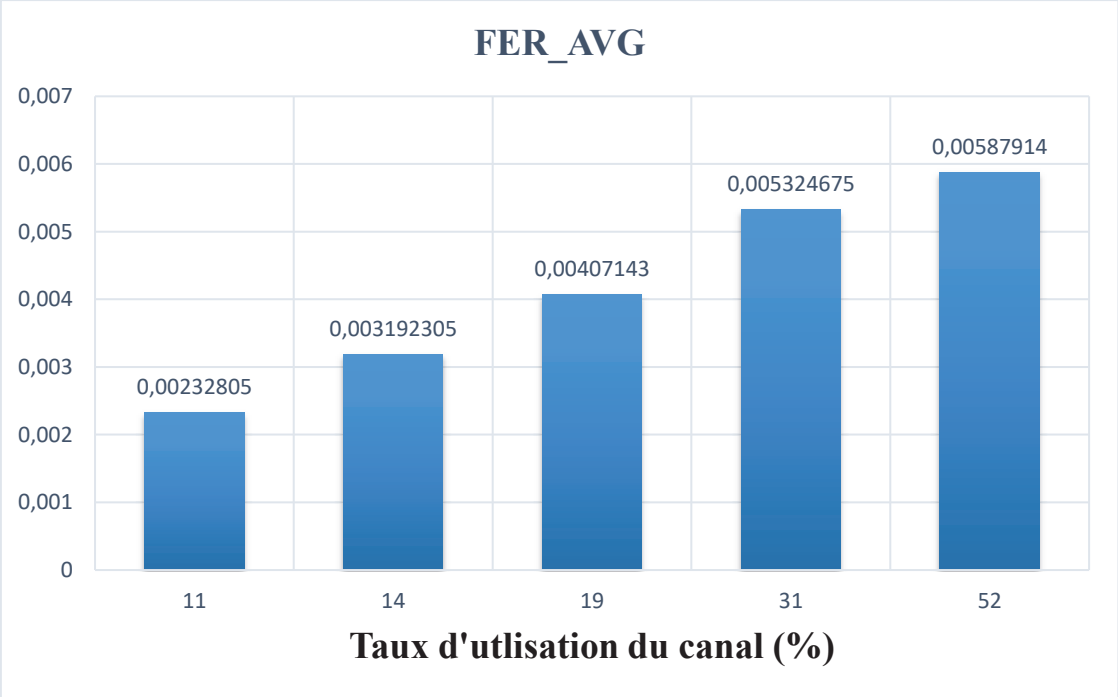

Figure 90 : Taux d'erreur trame moyen en utilisant 3 canaux, 8 Tx et 2 Rx

Le testbed Ophelia nous offre plus des nœuds par rapport aux tests précédents, réalisés seulement avec 4 nœuds. Il est également très pratique car il nous permet de "flasher" à distance les nœuds, c'est un atout pour la souplesse et la rapidité des tests. De plus, la disposition des nœuds nous offre une

approche évidente d'une topologie réelle d'un réseau sans fil, sur laquelle on a pu identifier certaines contraintes qu'on peut souvent rencontrer dans ces types des réseaux. Nous pouvons par exemple citer entre autres le testbed avec un seul Rx qui ne reçoit que des Tx qui se trouvent dans sa portée. Par contre les Tx hors de portée tentent souvent de joindre le Rx sans succès. Ce sont des cas fréquents qui surviennent dans les réseaux locaux sans fil. On a observé également que le FER moyen avec un Rx unique est plus important, puisque tous les nœuds du réseau sollicitent le même Rx, par conséquent les collisions sont très nombreuses. Ceci fait que, lorsqu'on ajoute un second Rx dans le réseau, le FER moyen régresse avec une valeur significative. C'est aussi une raison pour montrer la nécessité d'une transmission multi-saut dans les réseaux sans fil. Enfin, nous avons vu que avec le second Rx, les nœuds qui n'atteignent pas la destination (cas d'un seul Rx), portent au moins vers une autre destination un peu plus proche. On a toujours observé une meilleure valeur du FER à chaque fois qu'on a un canal supplémentaire. L'approche MAC multi-canal est donc bien mise en valeur.

3.10 Conclusion

Ce chapitre présente le protocole MAC multi-canal sans RDV multi-saut de façon plus détaillé. Nous avons présenté les différents aspects et procédures qui ont conduit à l'implémentation du protocole. Nous avons posé et exprimé nos métriques d'étude de performance, qui sont ensuite illustrer par des diagrammes des séquences. Des automates qui expliquent le processus par étape pour l'implémentation d'un protocole d'accès au médium MAC mono-canal et multi-canal de l'émetteur Tx et du récepteur Rx sont également mis en place.

Différentes topologies des testbeds composées de 4 nœuds sont configurées afin d'évaluer les différentes métriques d'étude de performance exprimées, ce qui a abouti aux comparaisons des plusieurs résultats et par conséquent à l'amélioration du protocole d'accès multi-canal sans RDV précédemment proposé. Les résultats des certains testbed sont aussi comparé par rapport à un modèle analytique dans un contexte multi-canal mono-saut et multi-saut, tout en admettant que ce dernier exclut certains paramètres qui influencent significativement les données réelles des testbed.

Enfin d'autres types des topologies avec plus des nœuds que les précédentes sont mise en place avec le projet Ophelia, et les résultats de testbed montrent que la méthode d'accès multi-canal proposée est plus fiable lorsque le réseau est composé de plusieurs nœuds, est moins fiable dans le cas contraire.

Conclusion et perspectives

Notre étude bibliographique des protocoles d'accès multi-canal existants, nous a permis d'identifier les lacunes des uns et les avantages des autres, mais aussi de soulever des questions pertinentes sur lesquelles se sont focalisés nos principales pistes pour la contribution de cette thèse. Via des modèles analytiques ou des simulateurs, certains travaux ont démontré les avantages de l'accès multi-canal par rapport au mono-canal en termes de débit et de réduction du délai de transmission, mais aussi des inconvénients, dont certains sont classiques et liées aux méthodes d'accès MAC mono-canal, alors que d'autres par contre sont inhérentes aux MAC multi-canal.

Au cours de ces études des protocoles d'accès multi-canal, nous avons remarqué que les méthodes d'accès multi-canal actuellement proposées, traitent le plus souvent de l'accès multi-canal mono-saut. Le concept multi-canal multi-saut a été abordé avec tant de contraintes que leur implémentation ne serait pas facile. Pour une transmission multi-saut, les méthodes d'accès multi-canal proposées ont souvent recourt à plusieurs interfaces radios pour contrôler l'utilisation des canaux. Elles nécessitent aussi parfois plusieurs échangent de trames de contrôles, ou utilisent également des diffusions permanentes des *beacons* (balises) pour gérer les différents problèmes qui se produisent pendant les transmissions. Par contre, ces échanges des trames de contrôles et diffusions permanentes des beacons peuvent considérablement pénaliser les performances des réseaux en présence d'une forte charge de trafics. Nous constatons aussi que la réservation des canaux par les méthodes que nous avons étudiés, s'effectue en majorité après un rendez-vous sur un canal prédéfini à l'avance (par exemple en utilisant le concept du canal de contrôle dédié ou du *Split phase*). Les protocoles d'accès multi-canal ont très souvent abordé le contexte multi-saut de façon théorique, face auxquelles nous posons la question qui est de savoir s'il serait facile de les implémenter afin de tester leurs performances réelles en multi-saut. Une autre piste est de s'affranchir des négociations de prises de RDV qui posent des problèmes en contexte multi-saut, car il est nécessaire d'indiquer aux nœuds éloignés à n sauts les RDV pris en local. On s'est donc orienté alors vers des MAC multi-saut multi-canal sans RDV, donc aléatoires, de type Aloha ou CSMA multi-canal. La plupart des méthodes d'accès multi-canal qui ont été proposées traitent principalement uniquement du cas des réseaux mono-saut, et risquent d'être inefficaces pour les réseaux distribués multi-saut de grande taille, puisqu'il faut propager le RDV pris en local aux voisins au-delà d'un saut.

Il est donc primordial pour nous de proposer une méthode d'accès multi-canal adaptée à une topologie multi-saut, qui passe à l'échelle. Cette étude nous a amenée à prototyper, premièrement, une méthode d'accès sans fil MAC mono-canal basée sur *Slotted-Alloha*. Par la suite, nous avons étendu notre étude pour prototyper l'accès multi-canal multi-saut sans RDV qui a été améliorée par une stratégie de rémanence sur le dernier canal de succès.

Nous avons fixé trois paramètres essentiels pour évaluer nos études des performances : le nombre de trames reçues, le nombre des trames perdues et le taux d'erreur trame (FER) indiquant les collisions principalement. Nous prenons comme référence le numéro de séquence de chaque trame reçue pour déduire ce FER.

Nous avons commencé nos évaluations de performances en comparant deux protocoles, qui sont l'Aloha slottée mono-canal et l'Aloha slottée multi-canal. Un testbed a été mis en place, constitué de cinq nœuds (un émetteur de beacons, deux émetteurs et deux récepteurs), avec certains cas de topologies qu'on peut généralement rencontrer dans les réseaux ad hoc. La première topologie nous a permis d'observer l'impact du nombre de voisinage sur le récepteur, car nous avons placé chaque récepteur est à une portée d'un seul émetteur (le voisin), alors que les deux émetteurs sont à portée l'un de l'autre. En mono-canal, le testbed nous offre un FER quasiment nul au niveau de chacun de deux récepteurs. Dans la seconde topologie, on a placé un récepteur à portée de deux émetteurs, nous

avons observé au niveau de ce récepteur un FER qui croit avec l'augmentation de la charge réseau (en générant plus de trafic en utilisant un backoff croissant).

En ajoutant un deuxième canal (car multi-canal) dans la seconde topologie, on a constaté un FER plus important que celui précédemment observé en mono-canal, qui provient non seulement des collisions lorsque les deux émetteurs sélectionnent le même canal et la même destination, mais également à cause du manquement du canal de réception (le récepteur n'est pas en écoute sur le canal sélectionné). Ceci montre que l'accès multi-canal n'est pas bien adapté au réseau de petite taille.

La solution que nous avons proposée pour réduire les erreurs liées au manquement de canal de réception est d'adopter une stratégie de rémanence sur le dernier canal de succès, c'est un paramètre très important pour nos analyses et études de performance. Lorsque nous avons mis en place une stratégie de rémanence, nous avons constaté qu'il y a une amélioration significative du FER par rapport aux précédents résultats.

Un modèle analytique a été développé dans un contexte multi-canal mono-saut et multi-saut en utilisant les processus de Poisson qui semblent bien adaptés pour modéliser les méthodes d'accès aléatoire. Notons que le modèle exclut beaucoup de paramètres essentiels pour l'étude de performances des réseaux ad hoc sans fil, par exemple les perturbations sur les canaux, le manquement de canal de réception, l'asymétrie des liens… Par contre, notre modèle prend en considération la charge et le nombre des canaux. Ce modèle analytique nous a servi de référence pour voir le comportement des protocoles d'accès aléatoire multi-canal de façon générale. Ainsi, un modèle de 4 nœuds et 2 canaux du testbed, montre que le débit est toujours meilleur en multi-canal (2 canaux) qu'en mono-canal, mais ce dernier s'affaiblit avec l'augmentation de la charge dans le deux cas mono-canal et multi-canal. Avec le même modèle en multi-saut, le débit se dégrade par rapport au contexte mono-saut qui sera bien évidemment impacté par le nombre de voisins du nœud sur chaque saut à effectuer par les trames afin d'atteindre leur destination.

Nous avons conclu nos testbeds en utilisant la plateforme développée dans le projet Ophelia, qui nous offre un nombre important de nœuds avec des topologies réseaux semblables à celles des réseaux ad hoc sans fil, où nous avons rencontré les problématiques et contraintes liés à ces types des réseaux. Nous avons remarqué que, en utilisant un récepteur (Rx) et plusieurs émetteurs (Tx) dont certains tentent de joindre le Rx sans succès, car limités par la portée de leur puissance du signal, le recours aux transmissions multi-saut est nécessaire. Pour cette raison, nous avons mis en place un second Rx. Utilisant les deux cas de figure (1 Rx et 2 Rx) et avons effectué également des tests en mono-canal et en multi-canal. En tenant compte de ces deux cas, le multi-canal offre un FER meilleur que le mono-canal. De plus, on a toujours des bons résultats lorsqu'on dispose de 2 Rx par rapport à 1 Rx.

Avec le testbed Ophelia, nous avons rencontré des contraintes réseaux sans fil réelles, comme par exemple le problème lié aux signaux, à la radio, le problème d'asymétrie…, lesquels ne sont pas considérés par la simulation réseau, ou encore par la modélisation.

Les travaux réalisés dans cette thèse offrent de nombreuses pistes de recherches, et rejoignent en grande partie notre positionnement face aux méthodes de développement que nous employons. Les performances des méthodes d'accès multi-canal proposées par la communauté scientifique, sont, dans la plupart des cas, testées à travers des modèles analytiques ou à travers des simulateurs. Nous proposons de valider ces simulations et ces modèles par des prototypages et testbeds réels, afin de tirer le meilleur de leurs études de performance pour savoir si elles sont bien adaptées aux transmissions multi-saut.

Pour tester encore d'avantage les performances de notre méthode d'accès multi-canal aléatoire multi-saut, nous suggérons premièrement dans les travaux futurs de réaliser des testbeds avec un grand

nombre des nœuds, typiquement une centaine. Il serait également utile d'implémenter une méthode de synchronisation très souple qui permettra aux nœuds de détecter indépendamment le début et la fin du slot de temps plutôt que d'attendre qu'un nœud synchronisateur diffuse un beacon pour signaler le slot de temps. En effet, il y a un risque que le nœud synchronisateur cesse de fonctionner, ce qui risque par conséquence de bloquer le réseau.

Un autre paramètre très important pour étendre l'étude de performance est le délai de rémanence, que nous pensons évaluer et adapter en fonction de la taille du réseau en nombre des nœuds, et en fonction du nombre de canaux disponibles. En d'autres termes, l'évaluation future des performances va se baser sur la taille du réseau, sur la période de rémanence, et sur le nombre de canaux. Cette évaluation permettra ainsi d'identifier à quelle taille du réseau correspondra le nombre des canaux et la période de rémanence de façon optimale.

Nous envisageons aussi de réfléchir à d'autres méthodes que la stratégie de rémanence qui permettent d'améliorer les méthodes d'accès multi-canal aléatoires afin de fiabiliser la probabilité pour que l'émetteur et le récepteur commutent leurs interfaces sur le même canal de transmission. Ces méthodes devront être comparées à la stratégie de rémanence.

Nous suggérons aussi dans les travaux futurs de prototyper d'autres méthodes d'accès multi-canal aléatoire, par exemple le CSMA, même si cette dernière ne semble pas offrir beaucoup d'intérêt pour le multi-canal en raison du délai de l'écoute du canal avant tout émission. Il serait intéressant également de prototyper des méthodes d'accès multi-canal à Rendez-Vous. La comparaison de ces méthodes avec notre méthode d'accès multi-canal aléatoire sans RDV utilisant une stratégie de rémanence, permettra de voir sur quels paramètres il faut agir, en quoi elle est meilleure et quels sont les paramètres pénalisants.

Il serait également intéressant d'implémenter et de comparer la méthode d'accès multi-canal sans RDV avec d'autres méthodes d'accès multi-canal aléatoires et d'autres méthodes d'accès multi-canal avec RDV en utilisant les simulateurs réseaux comme par exemple OMNeT++. Ceci permettra également de comparer leurs comportements en simulations par rapport à leurs prototypages réels.

Nous recommandons également des testbeds de la méthode d'accès multi-canal sans RDV avec d'autres types des nœuds que les nœuds WiNo, par exemple ceux utilisant la technologie LoRaWAN ou UWB.

Enfin, nous encourageons des futurs travaux sur l'accès multi-canal en s'orientant particulièrement vers des méthodes sans RDV, favorable aux multi-sauts. Il serait utile, par exemple, de retravailler la méthode d'accès multi-canal 802.15.4e, afin d'éliminer son RDV caché, diffusé par le coordinateur du PAN, et ainsi revoir la conception de ce protocole.

Actuellement, l'implémentation du protocole d'accès au médium TDMA (Time Division Multiple Access) inspire beaucoup de travaux dans ce contexte lié au multi-canal Un de ses inconvénients majeurs entraîne des doutes pour son efficacité en multi-canal. Mais, l'utilisation d'une synchronisation fine répartie de type SiSP entre les nœuds dans le réseau pourrait être une piste de solution à tester.

Bibliographie

1. A. A. K. Jeng, R. H. Jan, C. Y. Li, C. Chen (2011). Release-time-based multi-channel MAC protocol for wireless mesh networks. Computer Networks, 55(9), 2176-2195.
2. Hissein, Mahamat Habib Senoussi, Adrien Van Den Bossche, and Thierry Val (2014). Survey on multi-channel access methods for wireless LANs. *International Journal of Engineering Research and Technology* 3.12: 673-680.
3. J. Crichigno, M. Y. Wu, W. Shu (2008). Protocols and architectures for channel assignment in wireless mesh networks. Ad Hoc Networks, 6(7), 1051-1077.
4. J. Mo, H. S. So, J. Walrand (2008). Comparison of multichannel MAC protocols. IEEE Transactions on Mobile Computing, 7(1), 50-65.
5. P. Bahl, R. Chandra, J. Dunagan (2004). SSCH: slotted seeded channel hopping for capacity improvement in IEEE 802.11 ad-hoc wireless networks. In Proceedings of the 10th annual international conference on Mobile computing and networking, pp. 216-230.R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
6. W. So, J. Walrand, J. Mo (2007). McMAC: a parallel rendezvous multi-channel MAC protocol. In Wireless Communications and Networking Conference, pp. 334-339.
7. A. El Fatni, G. Juanole (2012). Split Phase Multi-channel MAC Protocols-Formal Specification and Analysis. In Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), IEEE, pp. 485-488.
8. J. So, N. H. Vaidya (2004). Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, ACM, pp. 222-233.
9. J. So, N. H. Vaidya (2004). Multi-channel mac for ad hoc networks: handling multi-channel hidden terminals using a single transceiver. In Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, ACM, pp. 222-233.
10. M. Wang, L. Ci, P. Zhan, Y. Xu (2008). Multi-channel MAC protocols in wireless ad hoc and sensor networks. In Computing, Communication, Control, and Management. CCCM'08, 1(2) , pp. 562-566.
11. A. El Fatni (2013). Modélisation, analyse et conception de protocoles MAC multi-canaux dans les réseaux sans fil, rapport de thèse, UT2J.
12. A. Nasipuri, J. Zhuang, S. R. Das (1999). A multichannel CSMA MAC protocol for multihop wireless networks. In Wireless Communications and Networking Conference, IEEE, pp. 1402-1406.
13. A. Nasipuri, J. Zhuang, S. R. Das (1999). A multichannel CSMA MAC protocol for multihop wireless networks. In Wireless Communications and Networking Conference, IEEE, pp. 1402-1406.
14. N. Jain, S. R. Das, A. Nasipuri (2001). A multichannel CSMA MAC protocol with receiver-based channel selection for multihop wireless networks. In Proceedings of the 10th International Conference on Computer Communications and Networks, IEEE, pp. 432-439.
15. J. Chen, Y. D. Chen (2004). AMNP: Ad hoc multichannel negotiation protocol for multihop mobile wireless networks. IEEE International Conference on Communications, IEEE, 1(6), pp. 3607-3612.
16. D.M. Shila, T. Anjali, Y, Cheng (2010). A Cooperative Multi-Channel MAC Protocol for Wireless Networks. Global Telecommunications Conference (GLOBECOM 2010), IEEE, pp.1-5.
17. W. T. Chen, J. C. Liu, C. C. Chang (2007). An Efficient Flow Control and Medium Access in Multihop Ad Hoc Networks with Multi-Channels. In Vehicular Technology Conference, IEEE, pp.56-60.
18. I. Chlamtac, M. Conti, J. J. N. Liu (2003). Mobile ad hoc networking: imperatives and challenges. Ad Hoc Networks, 1(1), pp.13-64.

19. K. Kredo II, P. Mohapatra (2007). Medium access control in wireless sensor networks. Computer Networks, 51(4), pp. 961-994.
20. I. F. Akyildiz, X. Wang, W. Wang (2005). Wireless mesh networks: a survey. Computer networks, 47(4), pp. 445-487.
21. Roberts, L. G. (1975). ALOHA packet system with and without slots and capture. ACM SIGCOMM Computer Communication Review, 5(2), pp.28-42.
22. N. Abramson (1977). The throughput of packet broadcasting channels. IEEE Transactions on Communications, 25(1), pp.117-128.
23. T. B. Reddy, I. Karthigeyan, B. S. Manoj, C. Murthy (2006). Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions. Ad Hoc Networks, 4(1), pp. 83-124.
24. S. Choi, J. Del Prado, N. Sai Shankar, S. Mangold (2003). IEEE 802.11e contention-based channel access performance evaluation. In ICC'03, IEEE, 2(1), pp.1151-1156.
25. A. L. Ruscelli, G. Cecchetti, A. Alifano, G. Lipari (2012). Enhancement of QoS support of HCCA schedulers using EDCA function in IEEE 802.11e networks. Ad Hoc Networks, 10(2), pp.147-161.
26. T. B. Reddy, I. Karthigeyan, B. S. Manoj, C. Murthy (2006). Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions. Ad Hoc Networks, 4(1), pp. 83-124.
27. P. Kyasanur, N. H. Vaidya (2006). Routing and link-layer protocols for multi-channel multi-interface ad hoc wireless networks. ACM SIGMOBILE Mobile Computing and Communications Review, 10(1), pp. 31-43.
28. A. Adya, P. Bahl, J. Padhye, A. Wolman, L. Zhou (2004). A multi-radio unification protocol for IEEE 802.11 wireless networks. In Proceedings of the First International Conference on Broadband Networks, IEEE, pp. 344-354.
29. E. Shim, S. Baek, J. Kim, D. Kim (2008). Multi-channel multi-interface MAC protocol in wireless ad hoc networks. In ICC'08, IEEE, pp. 2448-2453.
30. C. Toham, F. Jan (2006). Multi-interfaces and Multi-channels Multi-hop Ad hoc Networks: Overview and Challenges. In IEEE International Conference on Mobile Adhoc and Sensor Systems, IEEE, pp. 696-701.
31. Y. C. Cho, S. J. Yoon, Y. B. Ko (2011). Modifying the IEEE 802.11 MAC Protocol for Multi-hop Reservation in MIMC Tactical Ad Hoc Networks. In IEEE Workshops of International Conference on Advanced Information Networking and Applications, IEEE, pp. 178-183.
32. P. Kyasanur, N. H. Vaidya (2005). Routing and Interface Assignment in Multi-Channel Multi-Interface Wireless Networks. In Proceedings IEEE Wireless Communications and Networking Conference, IEEE, 1(4), pp. 2051-2056.
33. R. Draves, J. Padhye, B. Zill (2004). Routing in multi-radio, multi-hop wireless mesh networks. In Proceedings of the 10th annual international conference on Mobile computing and networking, ACM, pp. 114-128.
34. A. Raniwala, K. Gopalan, T. C. Chiueh (2004). Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. In ACM SIGMOBILE Mobile Computing and Communications Review, ACM, 8(2), pp. 50-65.
35. Z. Tang, J. Garcia-Luna-Aceves (1998). Hop Reservation Multiple Access (HRMA) for Multichannel Packet Radio Networks. In Proceedings of the IEEE IC3N'98, Seventh International Conference on Computer Communications and Networks, pp. 388-395.
36. A. Tzamaloukas, J. Garcia-Luna-Aceves (2000). Channel-hopping multiple access with packet trains for ad hoc networks. In Proceedings IEEE Mobile Multimedia Communications, pp, 23-26.
37. A. Tzamaloukas, J. J. Garcia-Luna-Aceves (2000). Channel-hopping multiple access. In ICC, 1(1), pp. 415–419.

38. S. L. WU, C. Y. LIN, Y. C. TSENG, all (2000). A new multi-channel MAC protocol with on-demand channel assignment for multi-hop mobile ad hoc networks. In Proceedings of Parallel Architectures, Algorithms and Networks, IEEE, pp. 232-237.
39. S. L. Wu, Y. C. Tseng, C. Y. Lin, J. P. Sheu (2002). A multi-channel MAC protocol with power control for multi-hop mobile ad hoc networks. In The Computer Journal, 45(1), pp. 101-110.
40. Y. J. CHOI, S. PARK, S. BAHK (2006). Multichannel random access in OFDMA wireless networks. IEEE Journal on Selected Areas in Communications, 2006, 24(3), pp. 603-613.
41. B. Mawlawi (2015). Random access for dense networks: Design and Analysis of Multiband CSMA/CA, rapport de thèse, INSA Lyon.
42. IEEE Standard for Local and metropolitan area networks--Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer," in IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011) , vol., no., pp.1-225, April 16 2012.
43. D. De Guglielmo, A. Seghetti, G. Anastasi, M. Conti (2014). A performance analysis of the network formation process in IEEE 802.15. 4e TSCH wireless sensor/actuator networks. In Computers and Communication (ISCC), IEEE, pp. 1-6.
44. C. F. Shih, A. E. Xhafa, J. Zhou (2015). Practical frequency hopping sequence design for interference avoidance in 802.15. 4e TSCH networks. In IEEE International Conference on Communications, IEEE, pp. 6494-6499.
45. R. Soua, P. Minet, E. Livolant (2015). Wave: a Distributed Scheduling Algorithm for Convergecast in IEEE 802.15. 4e Networks, rapport de thèse, Inria.
46. I. Guvenc, S. Gezici, Z. Sahinoglu, U. C. Kozat (2011). Reliable communications for short-range wireless systems. Cambridge University Press.
47. N. Accettura, M. R. Palattella, M. Dohler, L. A. Grieco, G. Boggia (2012). Standardized power-efficient & internet-enabled communication stack for capillary M2M networks. In Wireless Communications and Networking Conference Workshops (WCNCW), IEEE, pp. 226-231.
48. G. Alderisi, G. Patti, O. Mirabella, L. L. Bello (2015). Simulative assessments of the IEEE 802.15.4e DSME and TSCH in realistic process automation scenarios. In IEEE 13th International Conference on Industrial Informatics (INDIN),IEEE, pp. 948-955.
49. S. C. Mukhopadhyay, N. K. Suryadevara (2014). Internet of Things: Challenges and Opportunities, Springer International Publishing, pp. 1-17.
50. K. Pister, L. Doherty (2008). TSMP: Time synchronized mesh protocol. In IASTED Distributed Sensor Networks, pp. 391-398.
51. J. Kacprzyk (2012). Advances in Intelligent Systems and Computing. Springer.
52. M. R. Palattella, N. Accettura, L. A. Grieco, G. Boggia, M. Dohler, T. Engel (2013). On optimal scheduling in duty-cycled industrial IoT applications using IEEE802.15.4e TSCH. Sensors Journal, IEEE, 13(10), pp. 3655-3666.
53. T. Chang, T. Watteyne, K. Pister, Q. Wang (2015). Adaptive synchronization in multi-hop TSCH networks. Computer Networks, 76(1), pp. 165-176.
54. C. Toham, F. Jan (2006). Multi-interfaces and Multi-channels Multi-hop Ad hoc Networks: Overview and Challenges. In IEEE International Conference on Mobile Ad Hoc and Sensor Systems, IEEE, pp. 696-701.
55. E. S. Shim, S. Baek, J. Kim, D. Kim (2008). Multi-channel multi-interface MAC protocol in wireless ad hoc networks. In IEEE International Conference on Communications, IEEE, pp. 2448-2453.
56. Y. C. Cho, S. J. Yoon, Y. B. Ko (2011). Modifying the IEEE 802.11 MAC Protocol for Multi-hop Reservation in MIMC Tactical Ad Hoc Networks. In IEEE Workshops of International Conference on Advanced Information Networking and Applications (WAINA), IEEE, pp. 178-183.

57. H. Kim, Q. Gu, M. Yu, W. Zang, P. Liu (2010). A simulation framework for performance analysis of multi-interface and multi-channel wireless networks in INET/OMNET++. In Proceedings of the 2010 Spring Simulation Multiconference Society for Computer Simulation International, Springer, pp. 101.
58. W. YUE, Y. MATSUMOTO (2007). Performance analysis of multi-channel and multi-traffic on wireless communication networks. Springer Science & Business Media, Springer.
59. B. H. Walke, S. Mangold, L. Berlemann (2007). In IEEE 802 wireless systems: protocols, multi-hop mesh/relaying, performance and spectrum coexistence. John Wiley & Sons.
60. R. ROM, M. SIDI (2012). Multiple access protocols: performance and analysis. Springer Science & Business Media, Springer.
61. A. BRAND, H. AGHVAMI (2002). Multiple access protocols for mobile communications: GPRS, UMTS and beyond. John Wiley & Sons.
62. A. Van Den Bossche, T. Val, R. Dalcé (2011). SISP: A lightweight synchronization protocol for Wireless Sensor Networks. ETFA2011,IEEE, pp. 1-4.
63. J. Elson, L. Girod, D. Estrin (2002). Fine-grained network time synchronization using reference broadcasts. ACM SIGOPS Operating Systems Review, 36(SI), pp. 147-163.
64. H. Tall, G. Chalhoub, M. Misson (2015). Implementation of IEEE 802.15.4 unslotted CSMA/CA protocol on Contiki OS, In PEMWN.
65. A. Van Den Bossche, R. Dalce, T. Val (2016). OpenWiNo: an open hardware and software framework for fast-prototyping in the IoT. In International Conference on Telecommunications (ICT), IEEE, pp. 1-6.
66. A. Van den Bossche, T. Val (2013). WiNo: une plateforme d'émulation et de prototypage rapide pour l'ingénierie des protocoles en réseaux de capteurs sans fil. In 9emes Journees francophones Mobilite et Ubiquite (UBIMOB 2013), pp-1.
67. P. W. de Graaf, J. S. Lehnert (1998). Performance comparison of a slotted ALOHA DS/SSMA network and a multichannel narrow-band slotted ALOHA network. IEEE transactions on communications, IEEE, 46(4), 544-552.
68. D. Shen, V. O. Li (2002). Stabilized multi-channel ALOHA for wireless OFDM networks. In Global Telecommunications Conference, IEEE, 1(1), pp. 701-705.
69. Hissein, Mahamat Habib Senoussi, Adrien Van Den Bossche, and Thierry Val (2017). Prototyping of a new Multi-channel Multi-hop Access Method without RendezVous. International Journal of Computer Science and Information Security 15.1 68.
70. Van Den Bossche, A. (2007). Proposition d'une nouvelle méthode d'accès déterministe pour un réseau personnel sans fil à fortes contraintes temporelles(Doctoral dissertation, Université Toulouse le Mirail-Toulouse II).
71. Shih-Lin Wu, Chih-Yu Lin, Yu-Chee Tseng, and Jang-Ping Sheu (2000). A Dynamic Multi-Channel MAC for Ad-Hoc LAN. In Proc. International Symposium on Parallel Architectures, Algorithms and Networks page 232, Dallas/Richardson, Texas, USA.
72. ZHANG, Jingbin, ZHOU, Gang, HUANG, Chengdu, and al (2007). TMMAC: An energy efficient multi-channel mac protocol for ad hoc networks. In International Conference on. IEEE, 2007. p. 3554-3561.

Publications personnelles

Revues internationales

Mahamat Habib Senoussi Hissein, **Adrien Van den Bossche**, **Thierry Val**. Survey on Multi-channel access methods for Wireless LANs. Dans : *International Journal of Engineering Research & Technology*, Vol. 3 N. 12, décembre 2014.

Mahamat Habib Senoussi Hissein, **Adrien Van den Bossche**, **Thierry Val**. Prototyping of a new Multi-channel Multi-hop Access Method without RendezVous. Dans : *International Journal of Computer Science and Information Security*, *International Journal of Computer Science and Information Security*, USA, Vol. 15 N. 1, 2017.

Conférences et workshops nationaux

Mahamat Habib Senoussi Hissein, **Adrien Van den Bossche**, **Thierry Val**. Etat de l'art des méthodes d'accès multi-canal pour les réseaux locaux sans fil. Dans : *Journées Nationales des Communications Terrestres (JNCT 2014)*, *Toulouse-Blagnac*, *22/05/2014-23/05/2014*, Éditions Universitaires Européennes, mai 2014.

Mahamat Habib Senoussi Hissein, **Adrien Van den Bossche**, **Thierry Val**. Prototypage d'une nouvelle méthode d'accès multi-canal multi-saut sans RDV. Dans : *Journées Nationales des Communications Terrestres (JNCT 2016)*, *Montbéliard, France*, *01/09/2016-02/09/2016*, Laboratoire FEMTO-ST, septembre 2016.

Résumé

L'émergence de l'Internet des Objets révolutionne les réseaux locaux sans fil et inspirent de nombreuses applications. L'une des problématiques majeures pour les réseaux locaux sans fil est l'accès et le partage du médium radio sans fil. Plusieurs protocoles MAC mono-canal ont été proposés et abordent cette problématique avec des solutions intéressantes. Néanmoins, certains problèmes majeurs liés à l'accès au canal (nœud caché, synchronisation, propagation des RDV…) pour un contexte de transmission multi-saut, persistent encore et font toujours l'objet d'intenses études de la communauté scientifique, surtout lorsqu'il s'agit de réseaux de capteurs sans fil distribués sur des topologies étendues. Certains travaux de recherches ont proposé des protocoles MAC multi-canal, traitant souvent le cas idéal, où tous les nœuds dans le réseau sont à portée les uns des autres. Les émissions et réceptions des trames de données sont généralement précédées de trames des contrôles pour l'établissement de Rendez-vous (RDV) entre les nœuds concernés. Nous constatons que les RDV ne garantissent pas la réservation des canaux de façon déterministe sans conflit entre les nœuds dans le réseau, et peuvent rendre difficile les transmissions en multi-saut. Une solution complexe serait de propager ces RDV vers les voisins du nœud récepteur au-delà de 2 sauts. C'est face à cette complexité de gestion de RDV multi-sauts que s'inscrit notre contribution. Il s'agit pour nous de proposer une méthode d'accès multi-canal aléatoire sans RDV, en topologie multi-saut. Notre solution est implémentée sur un *testbed*réel constitué de nœuds WiNo mono-interface, elle est basée sur la méthode ALOHA slottée améliorée pour notre contexte multi-canal, dont nous évaluons les performances qui sont comparées au cas mono-canal. Un modèle analytique lié au contexte multi-canal sans RDV a été développé également, et comparé aux résultats de notre *testbed*.

Mots clés — Accès au médium multi-canal, Multi-saut sans RDV ; Prototypage ; Évaluation de performances ; WiNo ; MAC ; réseaux de capteurs sans fil.

The emergence of the Internet of Thingsrevolutionizeswireless local area networks and inspiringnumerous applications. One of the main issues for wirelessLANsis the access and sharing of the wireless radio medium. Several single-channel MAC protocols have been proposed and addressthis issue withinteresting solutions. However, some major problemsrelated to the channelaccess (hiddennode, synchronization, propagation of RDV) for a multi-hop transmission context, persist and are still the subject of intensive studies by the scientificcommunity, especiallywhenitcomes to a distributedwirelesssensors networks over extended topologies. Someresearch has proposed multi-channel MAC protocols, oftenaddressingthe ideal case, where all nodes in the network are within range of eachother. The transmissions and receptions of the data frames aregenerallypreceded by controls frames for the establishment of Rendez-Vous (RDV) among the nodesconcerned. Wefindthat the RDVs do not guarantee the channelsreservation in a deterministicwaywithoutconflictamong the nodes in the network, and maymakeitdifficult the multi-hop transmissions. A complex solution wouldbe to propagatetheseRDVs to the neighbors of the receivernodebeyond 2 hops. Facedwiththiscomplexity of multi-hop RDV management thatmakesour contribution. It is important for us to propose arandom multi-channelaccessmethodwithout RDV, in multi-hop topology. Our solution isimplemented on a real testbed made of multi-channel single-interface "WiNo" nodes, of whichweevaluate the performance that are compared to the single-channel case. An analytical model related to the multi-channelcontextwithout RDV wasalsodeveloped, And compared to the results of ourtestbed.

Keywords — Multi-channel medium access; Multi-hop without RDV; Prototyping; Performance evaluation ; WiNo ; MAC; Wireless Sensors Networks.