HMC-MAC : un protocole MAC hybride et multi-canal pour les réseaux de capteurs sans fil

Rana Diab

Numéro d'ordre : D. U : 2580
E D S P I C : 700

UNIVERSITE BLAISE PASCAL

ECOLE DOCTORALE SCIENCES

POUR L'INGENIEUR DE CLERMONT-FERRAND

Laboratoire d'Informatique, de Modélisation et d'Optimisation des systèmes

THÉSE

Présentée par

Rana Diab

pour obtenir le grade de

DOCTEUR D'UNIVERSITÉ

Discipline : Informatique

HMC-MAC : un protocole MAC hybride et multi-canal pour les réseaux de capteurs sans fil

Soutenue publiquement le 15 Juin 2015 devant le jury :

Rapporteurs :	Mme Pascale Minet	Chercheur HDR à L'INRIA
	M. Congduc Pham	Professeur à l'université de Pau
Examinateurs :	M. Alain QUILLOT	Professeur à l'université Blaise Pascal
	M. Jean-François Perelgritz	Chef de projet - Airbus Group Innovations
	M. Nadir Hakem	Chercheur à L'UQAT
Directeurs :	M. Michel Misson	Professeur à l'université d'Auvergne
	M. Gérard Chalhoub	MCF à l'université d'Auvergne

À ma mère…

À mon père…

À mon mari…

« Quand tout semble être contre vous, souvenez-vous que l'avion décolle face au vent, et non avec lui. »
Henry Ford

Remerciements

En préambule à ce travail, je souhaiterais adresser mes remerciements les plus sincères à toutes les personnes qui m'ont aidée, de près ou du loin, dans la réalisation de ce travail.

Je tiens à exprimer tout d'abord ma profonde reconnaissance à mon directeur de thèse M. Michel Misson, Professeur à l'Université d'Auvergne, ainsi que mon encadrant M. Gérard Chalhoub, Maître de Conférences à l'Université d'Auvergne, qui ont toujours su me guider avec un grand professionnalisme. Leurs expertises, leurs rigueurs scientifiques et leurs conseils pertinents m'ont permis d'avancer dans la bonne direction. Je les remercie également pour leurs disponibilités tout au long de cette période. Je n'oublie pas les encouragements continuels de Michel et l'intelligence de Gérard.

Je souhaite exprimer mes remerciements les plus respectueux aux membres du Jury qui ont accepté d'évaluer mon travail de thèse. Je voudrais remercier Mme Pascale Minet, Chercheur HDR à L'INRIA, et M. Congduc Pham, Professeur à l'Université de Pau, de m'avoir fait l'honneur d'accepter de juger ce mémoire. Je tiens également à remercier M. Nadir Hakem, Chercheur à L'UQAT, et M. Jean-François Perelgritz, Chef de projet - Airbus Group Innovations, d'avoir accepté d'examiner ce travail. Je n'oublie pas de remercier M. Alain Quillot, Professeur à l'Université Blaise Pascal, d'avoir présidé le jury de cette thèse.

Je remercie chaleureusement les membres de l'équipe Réseaux et Protocoles pour leur sympathie, leurs encouragements et leurs conseils. Je pense en particulier à M. Alexandre Guitton et M. Philippe Llamas pour leurs aides et les discussions utiles que j'ai eues avec eux. Je souhaite remercier Madame Marie-Francoise Servajean, Maître de Conférences à l'Université d'Auvergne, qui a eu la gentillesse de relire et commenter ce manuscrit. J'exprime toute ma sympathie aux collègues du bureau C6, plus particulièrement Thérèse, Honoré, Malick, Déthié et Xavier pour les moments précieux et pour l'ambiance très favorable qui régnait dans notre bureau. Je n'oublierai jamais le fameux poster pour ma soutenance effectué par les soins de Thérèse, merci ma chère.

J'adresse toute mon affection à mes chers parents qui ont su rester patients malgré mon éloignement. Leur confiance, leur tendresse et leur prière me guident tous les jours. Merci pour avoir fait de moi ce que je suis aujourd'hui. Il est impossible de trouver des mots pour exprimer à quel point je les aime.

J'adresse un merci très particulier à mes sœurs, mon frère, mes grands parents, mes chères tantes et oncles ainsi que mes cousines et cousins pour leur encouragement et leur soutien.

Un merci plein d'amour à mon mari qui a été à mes côtés durant les moments les plus difficiles de cette thèse. Son soutien quotidien indéfectible, ses encouragements et son aide étaient durant cette période, malgré la distance, ma source de motivation. Je t'aime infiniment.

Mes derniers remerciements vont à tous mes amis libanais à Clermont-Ferrand, avec qui j'ai partagé des moments inoubliables.

Table des matières

Table de figures

Liste des tableaux

Chapitre 1 Introduction

De nos jours, les réseaux de capteurs sans fil sont devenus incontournables grâce à leur facilité de déploiement et à leur autonomie énergétique. Ces réseaux possèdent une large gamme d'applications parmi lesquelles nous pouvons citer : les applications environnementales pour la surveillance de la météorologie ou la détection des risques naturels comme les activités sismiques, les applications militaires pour surveiller les mouvements d'ennemis ou les intrusions, les applications industrielles pour la surveillance de chaînes de production, les applications médicales pour la surveillance de patients, ou encore les applications écologiques pour la surveillance de polluants.

Les réseaux de capteurs sans fil (RCSF) représentent une technologie émergente qui suscite un intérêt croissant dans les communautés scientifiques et industrielles. Un RCSF est constitué d'un ensemble de nœuds capteurs déployés dans une zone d'intérêt afin de surveiller les conditions physiques et environnementales. Chaque capteur est composé d'une ou de plusieurs unités de capture, d'une unité de traitement et d'un module de transmission sans fil. Il est généralement alimenté par une petite pile de petite taille donc de faible capacité de stockage d'énergie d'où la nécessité d'une consommation efficace de l'énergie. Les réseaux de capteurs sans fil permettent de collecter les données, de les traiter, et de les transmettre vers un ou plusieurs points de collecte appelés puits. La figure 1.1 représente une architecture typique d'un réseau de capteurs sans fil.

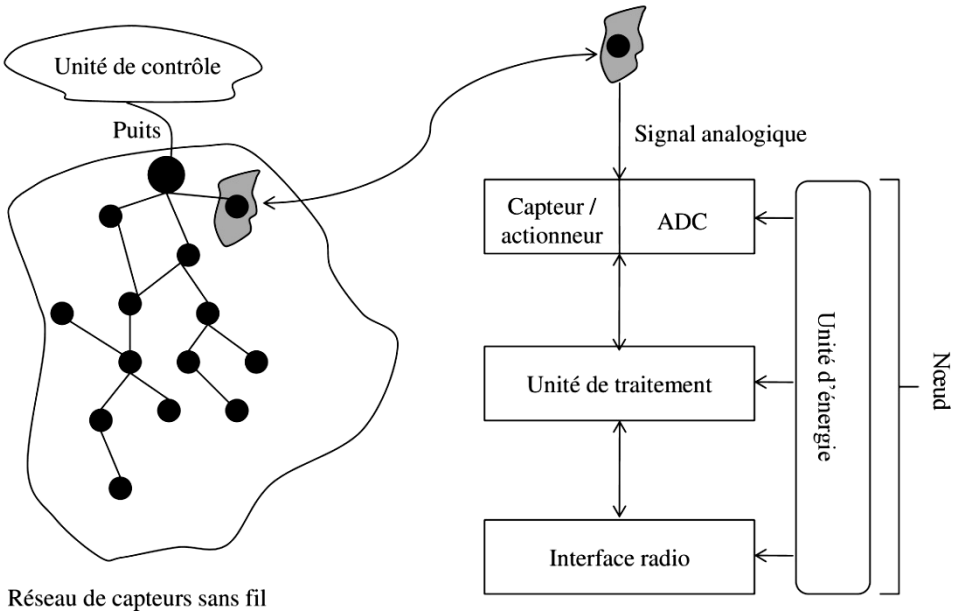


Figure 1.1 Un exemple typique d'un réseau de capteurs sans fil.

Les conditions de déploiement des RCSF nous obligent à concevoir des protocoles et des nœuds économes en énergie. Pour cela, la plupart des protocoles MAC existants proposés pour les réseaux de capteurs sans fil utilisent un seul canal pour la transmission des données. Cependant, la contrainte du passage à l'échelle dans des applications telle que la surveillance environnementale, qui nécessite le déploiement d'un nombre important de nœuds, augmente la charge du réseau. Cette charge est encore plus significative lorsque le réseau fonctionne à haut débit en raison d'une génération de trafic en rafale qui peut être induit par la génération d'alarmes, ou lors de l'exécution simultanée de plusieurs applications. Dans ces conditions, l'utilisation d'un canal unique augmente la probabilité de collision, limite le débit du réseau, et le rend vulnérable aux interférences.

Pour faire face à ces problèmes, les chercheurs ont recours à l'utilisation du multi-canal au niveau des protocoles MAC afin d'améliorer l'utilisation de la bande passante. En effet, la couche physique de la norme IEEE 802.15.4 permet l'utilisation de 16 canaux orthogonaux. L'exploitation de plusieurs canaux permet aux nœuds de transmettre en parallèle des données sur des canaux différents, ce qui augmente le débit et réduit les collisions. Dans le cadre de cette thèse, nous nous intéressons à l'utilisation du multi-canal au niveau de la couche MAC afin de répondre aux exigences des applications de collecte de données à haut débit dans une topologie multi-sauts.

Le choix de la technique d'accès au medium est un facteur essentiel dans les protocoles MAC multi-canaux. Certains protocoles proposés dans la littérature utilisent la technique TDMA. Cette technique offre des performances déterministes et évite les collisions. Cependant, elle nécessite une stricte synchronisation entre les nœuds et une stricte coordination entre les émetteurs et les récepteurs afin d'être actifs au même instant sur le même canal. Ceci entraine l'échange d'un nombre important de messages de contrôle. De plus, ce type de protocoles s'adapte difficilement aux changements de topologies et aux évolutions de réseaux. Or, les réseaux de capteurs sans fil sont dynamiques en raison de la mobilité de certains nœuds, de leur défaillance et/ou du changement de l'état de canal. Contrairement aux protocoles utilisant la technique TDMA, les protocoles utilisant la technique CSMA/CA sont plus flexibles et plus adaptés aux évolutions des réseaux. En revanche, ce type de protocole souffre des effets des collisions.

L'objectif de cette thèse est de proposer un protocole MAC multi-canal adapté aux réseaux de capteurs sans fil qui permet d'augmenter le débit et de minimiser les effets des interférences et des collisions.

Ce travail a été réalisé au sein de l'équipe Réseaux et Protocoles du laboratoire LIMOS-CNRS (Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes) de l'université Blaise Pascal, à Clermont Ferrand, sous la direction de Michel Misson et l'encadrement de Gérard Chalhoub. Ce manuscrit est organisé de la manière suivante.

Le chapitre 1 est la présente introduction dans laquelle nous définissons les réseaux de capteurs sans fil, leurs applications et la problématique du sujet.

Le deuxième chapitre est consacré à une étude bibliographique sur les principales façons d'utiliser des canaux multiples pour gérer l'accès au medium. Nous étudions tout d'abord la norme 802.15.4 qui représente la norme la plus utilisée pour les réseaux de capteurs sans fil. Ensuite, nous abordons les protocoles MAC multi-canaux les plus cités dans la littérature qui sont dédiés aux réseaux de capteurs sans fil. Enfin, nous étudions les solutions normalisées qui se basent sur la norme IEEE 802.15.4.

Dans le troisième chapitre, nous proposons un protocole MAC multi-canal qui vise à améliorer les performances du réseau en combinant les trois techniques TDMA, CSMA et FDMA. Ce protocole adopte une nouvelle méthode d'allocation de canal qui permet aux nœuds de choisir le canal le plus convenable dans ses voisinages jusqu'à 3-sauts. Nous découpons le temps en cycles globaux, chaque cycle étant divisé en plusieurs périodes afin d'améliorer l'utilisation du médium.

Ensuite, nous présentons dans le quatrième chapitre une évaluation des performances par simulation de notre proposition et nous la comparons à d'autres méthodes utilisées dans la littérature. Les résultats obtenus montrent que notre proposition améliore significativement les performances du réseau dans les différents scenarios étudiés.

Dans le cinquième chapitre, nous concluons ce manuscrit et nous accompagnons ce travail de nouvelles perspectives.

Chapitre 2 Etat de l'art

Dans ce chapitre, nous dressons un état de l'art sur les principales façons d'utiliser plusieurs canaux pour réaliser le partage du medium dans le domaine des Réseaux de Capteurs Sans Fil (RCSF). Ce chapitre est composé de trois parties. Dans la première, nous détaillons la norme IEEE 802.15.4 qui répond convenablement aux besoins des réseaux de capteurs sans fils. Dans la seconde partie nous présentons une étude bibliographique sur les protocoles MAC multi-canaux spécialement conçus pour les réseaux de capteurs sans fil. Dans la troisième partie de ce chapitre, nous présentons une étude des principaux standards industriels pour les réseaux de capteurs sans fil qui s'appuient sur le standard IEEE 802.15.4.

1. Le standard IEEE 802.15.4

Dans cette partie, nous allons détailler les différentes caractéristiques et fonctionnalités du standard IEEE 802.15.4 [1] [2]. Il est le premier standard dans le monde des réseaux de capteurs sans fil qui répond convenablement à ses besoins. Le standard IEEE 802.15.4 a été initialement conçu pour les réseaux personnels sans fils à bas débit (*Low Rate Wireless Personal Area Networks*) dont il définit la couche physique et la couche MAC. Il a été développé pour permettre des communications à faible consommation énergétique, ce qui induit une faible portée et un faible débit. Il s'adresse à des domaines applicatifs dont font partie les domaines industriels, agricoles et médicaux. Les couches de ce standard ont été aussi « adoptées » par plusieurs normes telles que ZigBee [3], *WirelessHART* [4] et ISA100.11a [5].

1.1. Architecture du réseau

Le standard IEEE 802.15.4 définit deux types d'entités qui peuvent participer à un réseau selon leurs capacités de fonctionnement : le RFD *(Reduced Function Device)* et le FFD *(Full Function Device)*. Le FFD possède toutes les fonctionnalités disponibles prévues par la norme tandis que le RFD ne possède qu'une partie seulement de ces fonctionnalités.

Le FFD peut prendre un des trois rôles dévolus à un nœud dans un réseau de capteurs sans fil de type WPAN *(Wireless Personal Area Network)* : Coordinateur du réseau personnel (ou CPAN pour coordinateur du PAN), Coordinateur ou Feuille. En revanche le RFD

n'implémente qu'une partie des fonctionnalités du FFD qui lui permet de tenir uniquement le rôle de feuille, ceci par raisons d'économie d'énergie et de facilité d'intégration. Un FFD peut communiquer avec un RFD ou un autre FFD, en revanche un RFD peut uniquement communiquer avec un FFD.

Un RFD comme un FFD est destiné à supporter un ou quelques capteurs/actionneurs associés à des volumes de données échangés en harmonie avec les débits autorisés par la norme. Il ne peut être associé qu'à un seul FFD à un moment donné.

Selon les besoins des applications, la norme IEEE 802.15.4 supporte deux types de topologies: la topologie en étoile et la topologie en point-à-point *(peer-to-peer)*. Ces deux types sont illustrés dans la figure 2.1.

Dans la topologie en étoile les communications sont établies uniquement entre les stations (RFD ou FFD) et le cœur de l'étoile occupé par le CPAN : le contrôleur unique et « central » de ce réseau. Dans cette norme, le CPAN est le coordinateur créateur du réseau, il est le seul nœud capable de gérer comment se construit le réseau en contrôlant la façon d'associer toute nouvelle station à la partie du réseau existant autour du CPAN. Les entités ne peuvent communiquer entre elles qu'en passant par l'intermédiaire du CPAN. La taille de ce type de topologie est limitée par la portée des liens radio des nœuds. Les topologies en étoile sont utilisables pour de nombreuses applications telles que la domotique, la gestion de périphériques, les jeux et le monitoring des processus tels qu'une chaine de production par exemple.

Dans la topologie point-à-point, un coordinateur de type CPAN existe aussi, il est le dépositaire de l'identité du réseau. Ce qui diffère pour cette topologie est qu'elle permet des communications directes entre nœuds sans obligatoirement passer par le CPAN. Les coordinateurs ont la capacité de communiquer entre eux quand ils sont à portée radio, seules les feuilles sont obligées de communiquer au coordinateur auquel elles sont rattachées. Une autre différence vient du fait qu'une hiérarchie peut être exploitée entre coordinateurs : les coordinateurs descendants du coordinateur de PAN peuvent gérer les associations d'autres entités dans le réseau. Ces capacités permettent la formation de réseaux plus complexes et plus étendus comme les réseaux partiellement maillés connus sous l'appellation *Mesh*. Ce type de topologie peut être utilisé pour les applications variées telles que le contrôle et la surveillance industrielle, les applications de traçabilité et de sécurité.

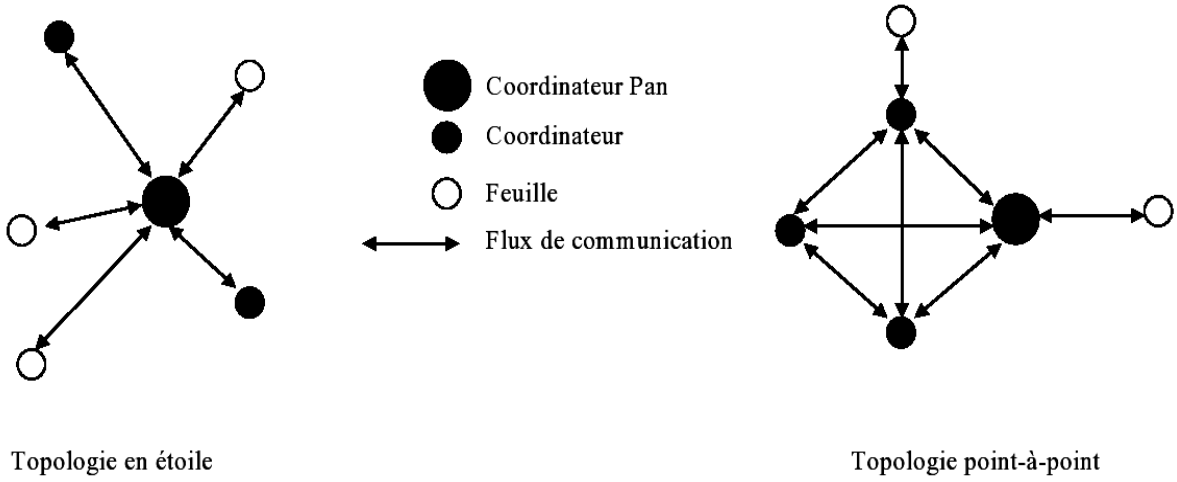


Figure 2.1 Exemple de topologie en étoile et point-à-point.

1.2. La couche physique de la norme IEEE 802.15.4

La couche physique de la norme IEEE 802.15.4 a pour rôle de gérer la transmission et la réception des données à travers un canal de transmission de type radio. Elle est opérationnelle dans trois bandes de fréquences de type ISM (Industrielle, Scientifique et Médicale). Cette couche dispose d'un total de 27 canaux sans recouvrement répartis en :

- 16 canaux dans la plage de fréquences de 2.4 GHz à 2.4835 GHz avec un débit de données de 250 Kbps,
- 10 canaux dans la plage de fréquence de 902 à 928 MHz avec un débit de données de 40 Kbps,
- Un seul canal dans la plage de fréquence de 868 à 870 MHz avec un débit de données de 20 Kbps.

La bande de 2,4 GHz est une des bandes ISM exploitables partout dans le monde, les autres sont des bandes dont l'usage a déjà été affecté dans certains pays. La bande [902-928] MHz par exemple n'est pas libre en France alors qu'elle l'est en Amérique du Nord, en Australie et dans d'autres pays. La bande 868-870 MHz est opérationnelle en Europe. La bande 2,4 GHz permet les plus hauts débits de données radio au détriment de la portée du lien radio. Elle possède le meilleur potentiel pour les applications des réseaux de capteurs sans fils de grande envergure en terme de nombre de stations. Le protocole utilisé pour échanger les données sur le canal radio peut exploiter l'existence d'un nombre important (16) de canaux « non recouvrants » via le concept multi-canal pour réduire les effets des interférences générées par sa propre activité ou des interférences provoquées par l'existence d'activités de réseaux radio qui partagent les mêmes parties du spectre (le WiFi et d'une façon plus générale toutes les solutions basées sur la norme IEEE 802.11 par exemple).

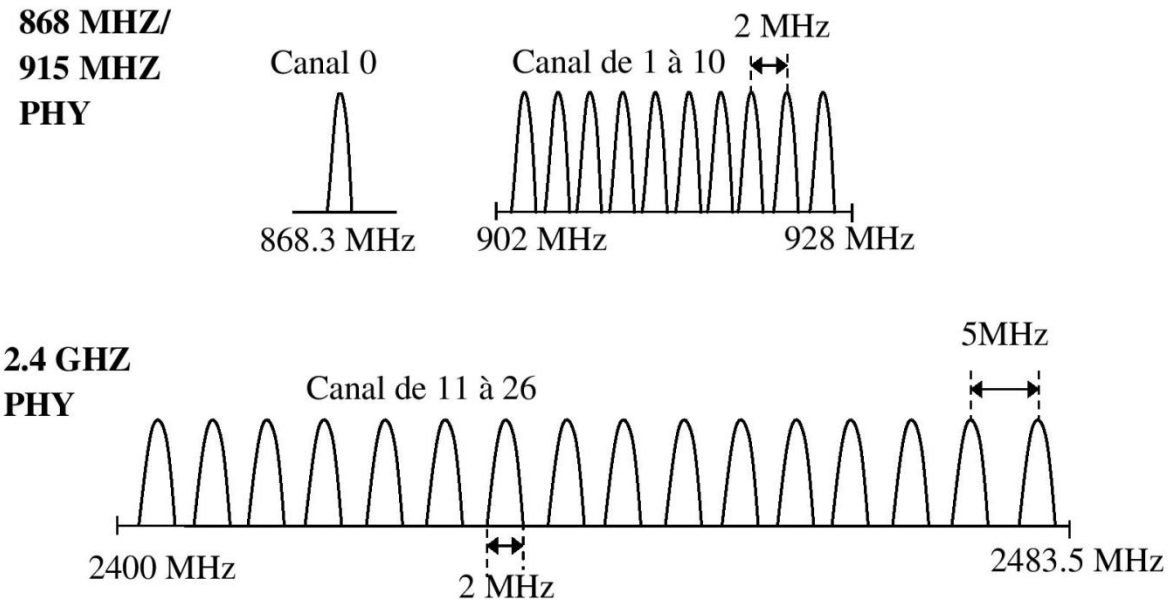


Figure 2.2 Les bandes de fréquences de la couche physique de la norme IEEE 802.15.4 [6].

La couche physique de la norme 802.15.4 offre les fonctions suivantes :

Activation et désactivation du module radio : il existe trois états de fonctionnement différents pour un module radio sous le contrôle de la couche MAC : un état d'émission, un état de réception et un état de veille. Un module radio a besoin au maximum de 192 µs pour passer d'un état de transmission à un état de réception.

Indication de la qualité du signal radio : LQI *(Link Quality Indication)* est une indication de la qualité de lien suite à la réception d'une trame sur ce lien. La façon de donner une valeur à cet indicateur est dépendante et à la discrétion des fabricants de composants. Pour certains composants, cet indicateur est un indice de corrélation moyenné sur les huit premiers symboles reçus de chaque trame [7]. Cet indicateur est essentiel pour les protocoles des couches réseaux et leurs applications.

Test d'occupation du medium ou CCA : le CCA *(Clear Channel Assessment)* est le test effectué pour détecter si le canal est libre en réalisant une détection de porteuse. Le canal est considéré occupé quand la puissance détectée est supérieure à un certain seuil. Le CCA est nécessaire pour respecter le principe de base (la détection de porteuse est assimilée à un canal occupé) de tous les algorithmes de type CSMA/XX.

La sélection de canal : la couche physique offre plusieurs canaux de transmission selon la bande utilisée, elle est capable de faire fonctionner son module radio sur un canal précisé par les couches supérieures et de commuter d'un canal à un autre. Chaque module radio de la couche physique ne peut recevoir ou transmettre que sur un seul canal à un instant donné.

1.3. La couche MAC de la norme IEEE 802.15.4

La couche MAC comme l'indique son nom gère le contrôle d'accès au medium. Elle permet la transmission des trames par l'utilisation du canal physique, la synchronisation, l'exploration de l'environnement radio (ou *scan*), les associations entre les entités et la création du réseau.

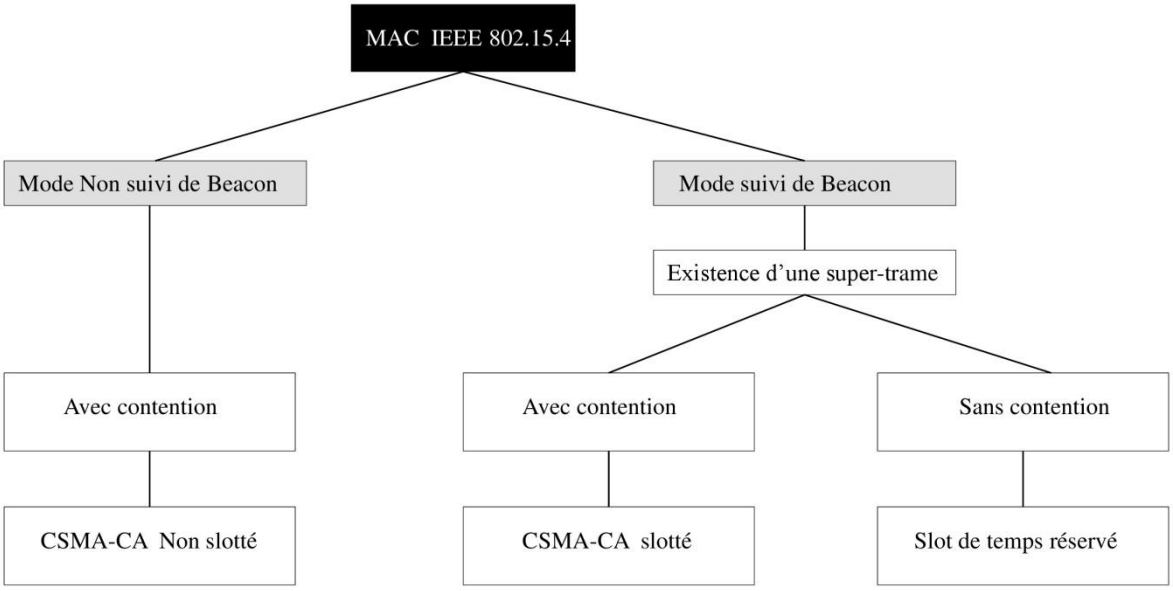


Figure 2.3 Les modes de fonctionnement de la couche MAC de 802.15.4.

Dans la norme IEEE 802.15.4, la couche MAC utilise deux modes de fonctionnement pour accéder au medium : le mode suivi de *beacon* et le mode non suivi de *beacon*. Dans le mode suivi de *beacon*, le coordinateur du réseau envoie périodiquement une trame balise usuellement appelée *beacon* pour synchroniser l'activité des nœuds associés au réseau. Dans ce mode, les nœuds utilisent une structure temporelle et périodique appelée super-trame ou en anglais *superframe*. Dans ce mode, l'algorithme CSMA/CA synchronisé usuellement appelé slotté est utilisé pour gérer l'accès au medium qui sera détaillé dans la suite à partir de la figure 2.4. Dans le mode non suivi de *beacon*, les *beacons* ne sont utilisés que pour la découverte de voisinage et les associations des nœuds. Dans ce mode, la synchronisation et la structure de super-trame ne sont plus exploitées et c'est l'algorithme CSMA/CA non slotté qui est utilisé pour gérer l'accès au medium. Dans ce qui suit, nous détaillerons la structure de la super-trame qui est utilisée dans le mode suivi de *beacon*.

Structure de la super-trame :

Dans le mode suivi de *beacon*, une super-trame est bornée par la transmission du *beacon* suivant. Elle est composée d'une période active suivie d'une éventuelle période inactive. Les coordinateurs peuvent communiquer uniquement dans la période active de la super-trame et entrer en mode sommeil pendant la période inactive permettant ainsi d'économiser de l'énergie. La partie active de chaque super-trame est divisée en 16 intervalles (slots) de temps égaux. Elle est elle-même composée de trois parties : le *beacon*, la CAP *(Contention Access Period)* qui est une période de contention et la CFP *(Contention Free Period)* qui est une période sans contention. Le *beacon* occupe le premier slot. La CAP commence immédiatement après le *beacon* et occupe x des 16 slots de temps suivants. Durant cette partie, les nœuds utilisent le mécanisme CSMA/CA slotté pour accéder au medium pour toutes les trames sauf les acquittements et les *beacons*. Le nœud qui veut transmettre des données durant cette période doit être capable de terminer la transaction (y compris la réception d'un ACK) et d'attendre un IFS (*InterFrame Space*) avant la fin de la CAP. Dans le cas où ce n'est pas possible, la transmission sera reportée à la CAP de la super-trame suivante.

La CFP suit immédiatement la CAP et occupe la fin de la partie active de la super-trame. La CFP est une partie optionnelle de la super-trame. Cette partie est constituée de slots de temps garantis appelé GTS *(Guaranteed Time Slot)* où les nœuds peuvent accéder au medium sans compétition. L'attribution de GTS est gérée par le coordinateur du réseau. Cette attribution se fait à la demande des nœuds par des requêtes envoyées durant la CAP. Lorsqu'une entité transmet dans la CFP, elle doit s'assurer que ses émissions sont terminées avant la fin de ses GTS. La super-trame se répète périodiquement afin que les nœuds du réseau synchronisent leur activité de transmission.

La structure de la super-trame est définie par les deux paramètres BI *(Beacon interval)* et SD *(Superframe Duration)*. BI détermine l'intervalle de temps qui sépare deux *beacons* et représente la durée totale de la super-trame. SD représente la durée de la partie active de la super-trame. Ces deux paramètres sont déterminés en fonction des valeurs de BO *(Beacon Order)* et de SO *(Superframe Order)*, qui sont des informations envoyées par le *beacon*. BI et SD sont calculés à partir des formules suivantes :

$$\begin{cases} BI = aBaseSuperframeDuration * 2^{BO} \ avec \ 0 \leq BO \leq 14 \\ SD = aBaseSuperframeDuration * 2^{SO} \ avec \ 0 \leq SO \leq BO \end{cases}$$

aBaseSuperframeDuration est une durée attribuée par la couche MAC, et est égale par défaut à 15.36 ms. Lorsque SO et BO sont égaux, nous sommes face à une absence d'une période d'inactivité.

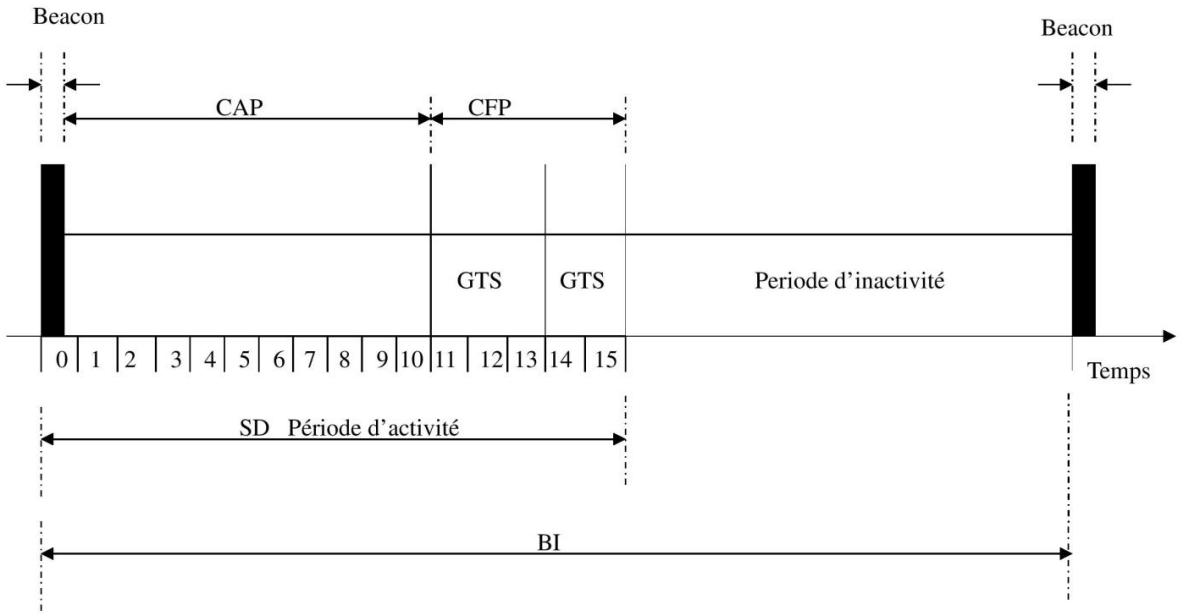


Figure 2.4 La structure d'une super-trame IEEE 802.15.4.

Méthodes d'accès au Medium :

La couche MAC de la norme IEEE 802.15.4 est principalement basée sur la contention. Elle utilise l'algorithme CSMA/CA pour accéder au canal. L'algorithme CSMA/CA est basé sur un tirage de temps aléatoire appelé période de *backoff* qui vise à désynchroniser les instants d'envoi des trames par des nœuds différents. Cette période de *backoff* est un temps multiple d'une unité de temps utilisée dans CSMA/CA qui est égale à *aUnitBackoffPeriod* symboles (qui vaut 20 temps-symboles). Selon le mode de fonctionnement du réseau, deux versions de CSMA/CA sont proposées : le CSMA/CA slotté et le CSMA/CA non slotté.

CSMA/CA slotté :
Le CSMA/CA slotté dépend essentiellement de trois paramètres pour planifier l'accès au medium : NB, BE et CW.

NB *(Number of Backoffs)* : compte le nombre de fois que l'algorithme CSMA/CA applique le tirage de *backoff* pour essayer de transmettre une trame en instance d'émission. Cette valeur est initialisée à 0 avant chaque nouvelle transmission.

BE *(Backoff exponent)* : représente l'exposant du *backoff*. Il permet de calculer le nombre de périodes de *backoff* qu'un nœud doit attendre avant d'évaluer de nouveau l'état du canal. Ce nombre est choisi aléatoirement par le nœud dans l'intervalle $[0, 2^{BE-1}]$.

CW *(Contention Window)* : est la taille de la fenêtre de contention. Il représente une durée définie par un nombre de périodes de *backoff* à la fin de laquelle le canal doit être détecté libre pour que la transmission commence.

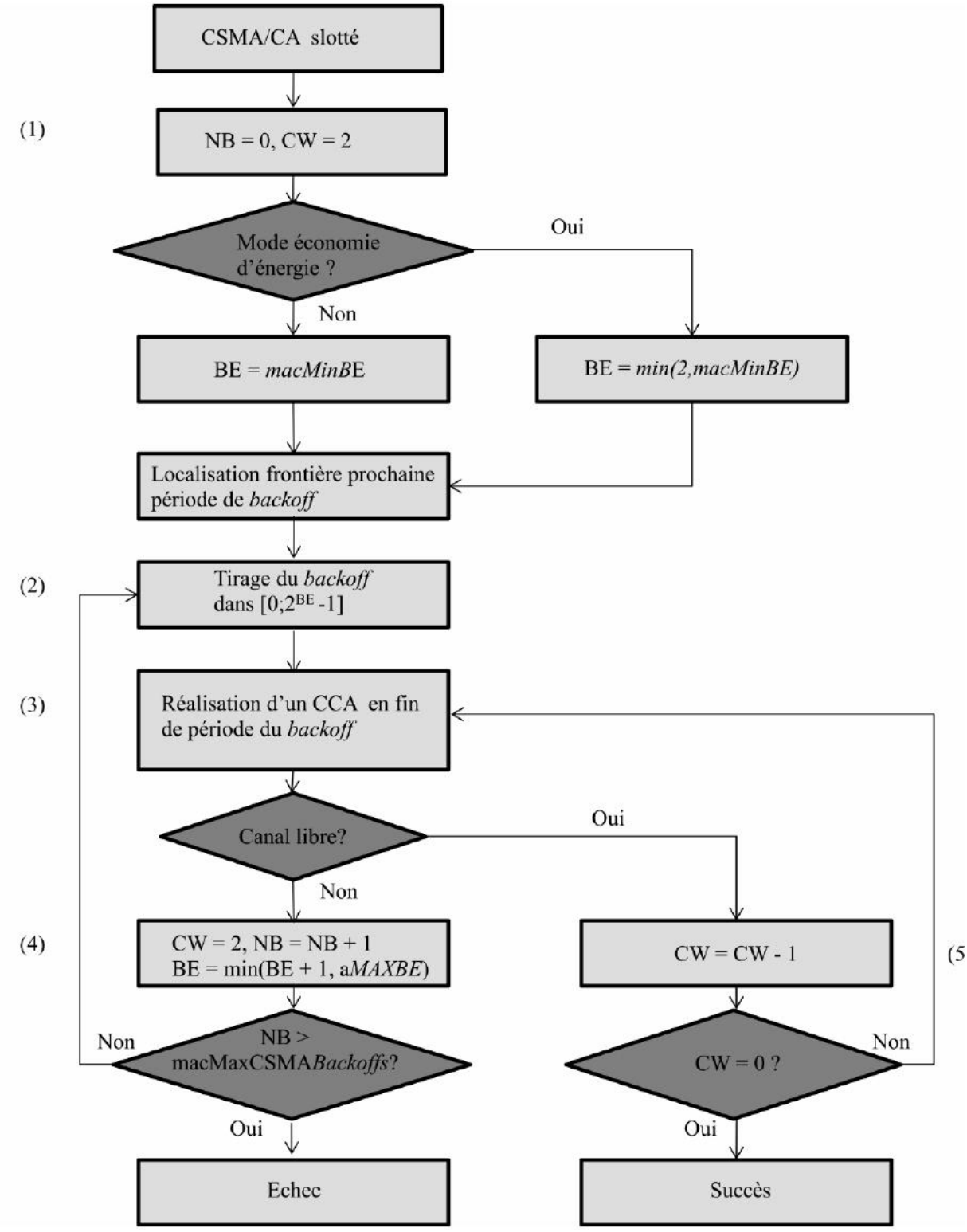


Figure 2.5 L'algorithme CSMA/CA slotté de la norme IEEE 802.15.4.

La figure 2.5 représente l'algorithme CSMA/CA slotté de la norme 802.15.4. Ce dernier peut se résumer en cinq étapes :

Etape 1 : dans cette étape la couche MAC initialise les paramètres de l'algorithme : en premier lieu le NB et le CW sont mis à 0 et à 2 respectivement, puis la valeur du Mode BLE est testée : si la valeur est égale à 1, le BE est initialisé à la valeur minimale entre 2 et *macMinBE*. Dans le cas contraire le BE est initialisé à *macMinBE*. A la fin de cette étape, la couche MAC se synchronise sur une frontière d'une période de *backoff*.

Etape 2 : dans cette étape, le nœud effectue un tirage aléatoire du nombre de périodes de *backoff* dans l'intervalle $[0, 2^{BE} - 1]$ pour désynchroniser les transmissions. Il doit attendre ce délai avant l'évaluation du canal. Si le nombre de période de *backoff* tiré est supérieur aux nombre de périodes de *backoff* restant dans la CAP, l'algorithme de CSMA/CA consomme les périodes de *backoff* jusqu'à la fin de la CAP courante et reporte le nombre restant à la CAP de la prochaine super-trame. Une fois le nombre de période de *backoff* expiré, la couche MAC doit s'assurer que les étapes restantes de l'algorithme (CW, test du canal, transmission de la trame et réception de son acquittement) peuvent être effectuées avant la fin de la CAP. Si c'est le cas, la couche MAC passe à l'étape 3, sinon la couche MAC doit attendre le début de la CAP de la prochaine super-trame pour tester le canal en effectuant un tirage de *backoff* supplémentaire pour éviter les collisions résultant de ce mécanisme de report.

Etape 3 : un test est effectué dans cette étape pour savoir si le canal est libre. La couche MAC sollicite la couche physique pour effectuer un test du canal (CCA), si le canal est détecté occupé l'algorithme passe à l'étape 4. Si le canal est détecté libre l'étape 5 est exécutée.

Etape 4 : la couche MAC réinitialise la variable CW à 2 et incrémente les deux variables NB et BE en s'assurant que BE ne dépasse pas la constante *aMaxBE* (qui est fixé à 5). Si NB est inférieur ou égal à *macMAXCSMABackoffs* (fixé à 4 par défaut) l'algorithme doit retourner à l'étape 2. Sinon l'algorithme se termine avec échec d'accès au canal.

Etape 5 : si le canal est détecté libre, la couche MAC décrémente la variable CW et lorsqu'elle atteint 0 le message est transmis, autrement l'algorithme passe à l'étape 3.

CSMA/CA non slotté :
Cet algorithme est utilisé dans le mode non suivi de *Beacon* où il n'y a pas de synchronisation entre les nœuds. Dans ce mécanisme lorsque le medium est détecté libre, l'émission se fait directement. Il y a une absence de synchronisation des envois sur les débuts des périodes de *backoff*. La figure 2.6 illustre les étapes de l'algorithme.

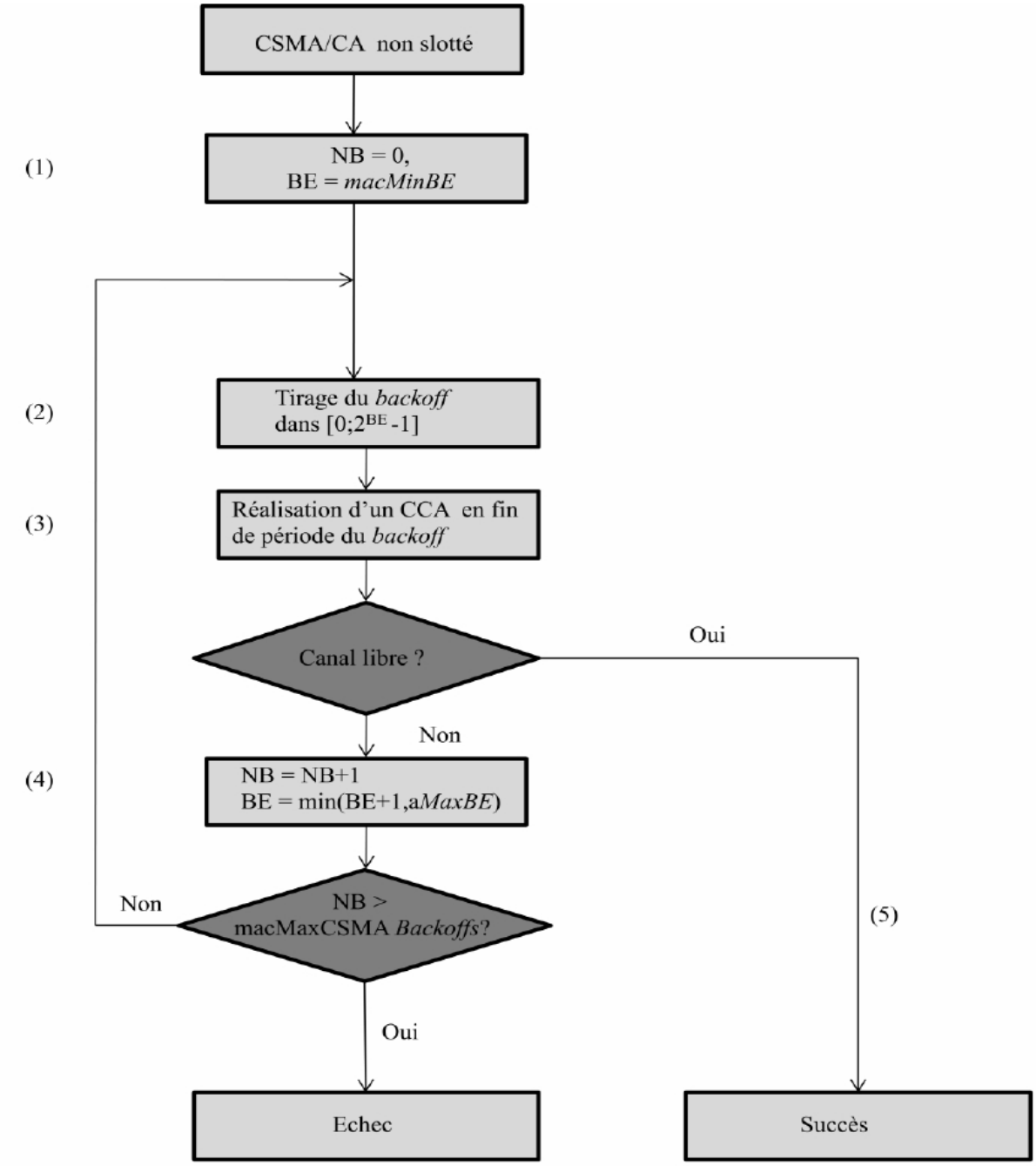


Figure 2.6 L'algorithme CSMA/CA non slotté de la norme IEEE 802.15.4.

Etape 1 : cette étape correspond à l'initialisation des variables, les variables NB et BE sont initialisées respectivement à 0 et à *macMinBE* qui est égal à 3 par défaut.

Etape 2 : durant cette étape, le nœud qui veut transmettre une trame tire au sort un nombre aléatoire de périodes de *backoff* dans l'intervalle $[0, 2^{BE} - 1]$. Il doit attendre ce délai avant l'évaluation du canal.

Etape 3 : lorsque le temps aléatoire tiré expire, le nœud effectue un test du canal et vérifie si le canal est libre. Une fois le canal détecté libre l'algorithme passe à l'étape 5. Quand le canal est détecté occupé, c'est l'étape 4 qui est exécutée.

Etape 4 : cette étape est exécutée lorsque le canal est détecté occupé. Les deux variables NB et BE sont incrémentées de 1 à condition que BE ne dépasse pas *aMaxBE* qui vaut 5. Ensuite la valeur de NB est testée. Si NB est inférieur à *macMAXCSMABackoffs* (fixer à 4 par défaut) l'algorithme retourne à l'étape 2, sinon l'algorithme termine avec échec d'accès au canal.

Etape 5 : cette étape est exécutée lorsque le canal est détecté libre. Dans ce cas la couche MAC peut accéder au canal avec succès.

2. Le multi-canal dans les RCSF

Dans cette partie, Nous allons aborder les protocoles MAC multi-canaux spécialement conçus pour les réseaux de capteurs sans fil. L'utilisation des canaux multiples aide à surmonter les interférences ainsi qu'à améliorer l'ensemble des performances du réseau. Plusieurs protocoles ont été déjà proposés dans la littérature afin d'allouer des canaux et des slots de temps. Ces protocoles ont été classifiés de différentes façons : en fonction de la périodicité de changement de canal, du fonctionnement centralisé ou distribué, synchronisé ou asynchrone, etc. Dans cette partie, nous présentons un aperçu sur le principe des protocoles MAC multi-canaux, les différentes techniques utilisées dans les réseaux de capteurs sans fil qui sont réalisables à partir des ressources offertes par la couche physique du standard IEEE 802.15.4. Nous allons détailler certains des protocoles les plus connus et nous discuterons de leurs avantages et de leurs points de faiblesse.

2.1. Principe des protocoles MAC multi-canaux

Le challenge est de distribuer des ressources dans un réseau, ces ressources sont appelées des canaux. Un canal est centré sur une fréquence particulière généralement allouée pour un certain temps. Nous disposons d'un ensemble de canaux selon la couche physique utilisée. L'allocation de canaux se fera en fonction d'un sous-ensemble de l'ensemble précédant (l'ensemble de canaux donnés gérés par la couche physique), ceci en tenant compte des canaux réservés pour d'autres usages, des canaux exposés aux interférences et des canaux interdits pour des raisons diverses. Cette allocation peut être effectuée pour chaque lien, pour chaque émetteur, ou pour chaque récepteur.

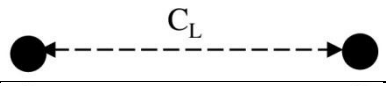
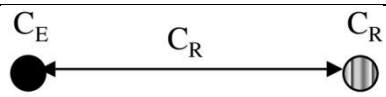
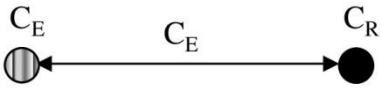
Pour chaque lien	
Pour chaque récepteur	
Pour chaque émetteur	

Tableau 2.1 Les différents types d'allocation des canaux.

La portée d'un lien radio, qui est la distance maximale entre les deux nœuds aux extrémités d'un lien opérationnel, peut être largement inférieure à l'étendue de la topologie, ce qui oblige la mise en place d'une topologie multi-saut afin de relayer les messages envoyés par les nœuds capteurs jusqu'à la destination. Dans une topologie multi-saut, chaque paquet doit être transmis plusieurs fois pour arriver à la destination finale. La surcharge du canal radio induit des interférences et provoque un risque de collision dû au problème du terminal caché [8] que nous détaillerons dans le chapitre 3. Ce phénomène entraîne des pertes de trames qui conduisent à une dégradation des performances et une utilisation inefficace de la bande passante.

Tous les dispositifs conformes à la norme IEEE 802.15.4 [1] disposent d'une puce radio permettant de changer de canal de communication. De nombreux chercheurs ont étudié l'utilisation de multiples canaux afin d'atténuer les conséquences des interférences. Dans le cas où plusieurs canaux sont utilisés, plusieurs nœuds peuvent transmettre des trames à travers des canaux différents simultanément ce qui augmente le débit, offre une protection robuste contre les interférences et augmente la capacité du réseau. La figure 2.7 montre un exemple d'utilisation d'un seul canal et comment les collisions se produisent au niveau des récepteurs lorsque plusieurs nœuds transmettent simultanément. En revanche, la figure 2.8 montre comment l'utilisation de plusieurs canaux permet des transmissions simultanées sans collision pour le même exemple.

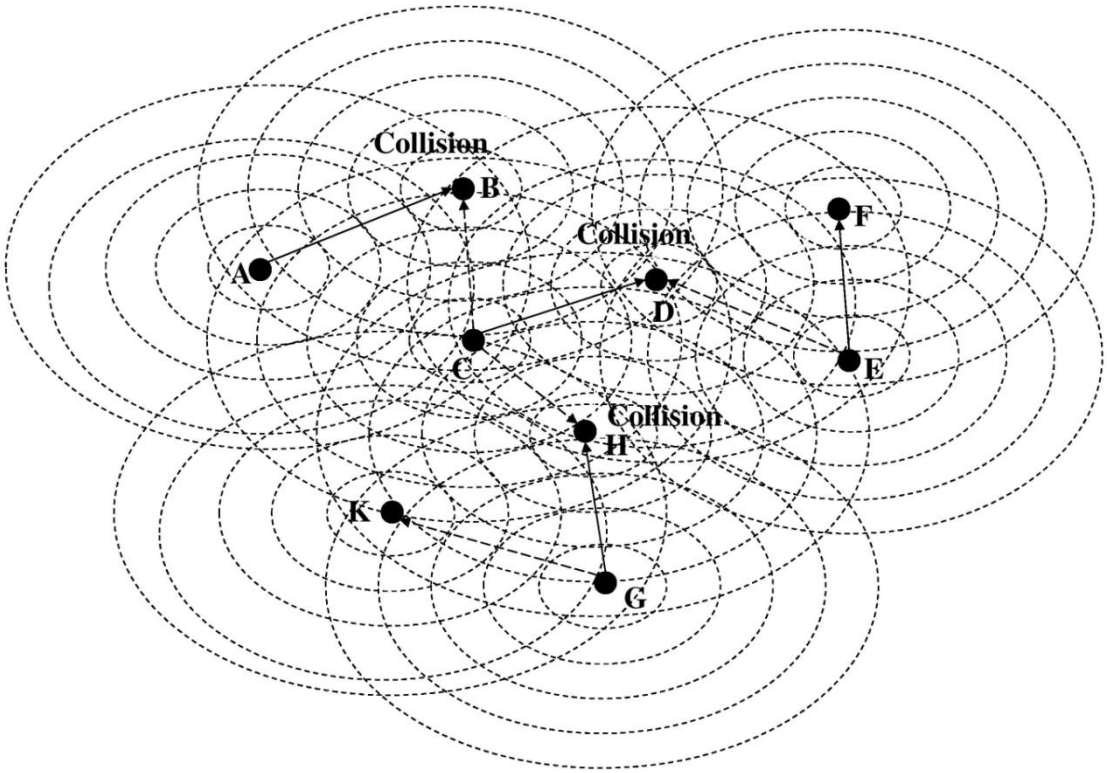


Figure 2.7 Scénario de collision dans le cas d'un canal unique.

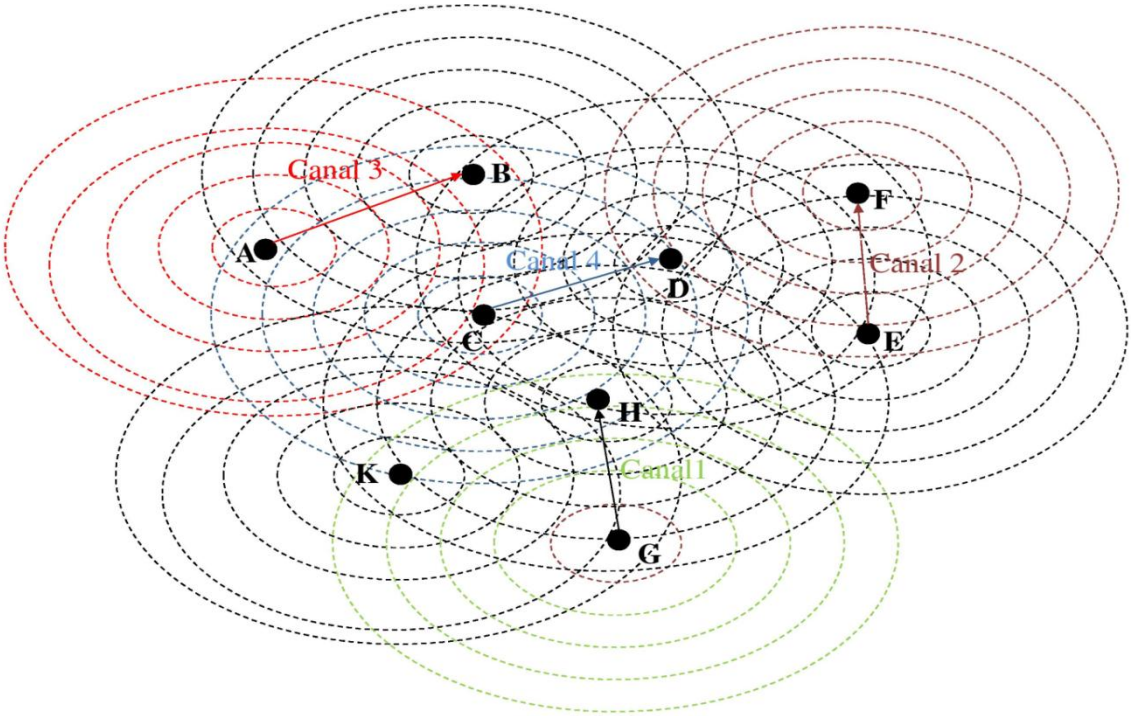


Figure 2.8 Scénario de collision dans le cas de multi-canal.

Dans les deux figures précédentes la portée d'un nœud est matérialisée par le cercle de plus grand rayon dont le centre est le nœud.

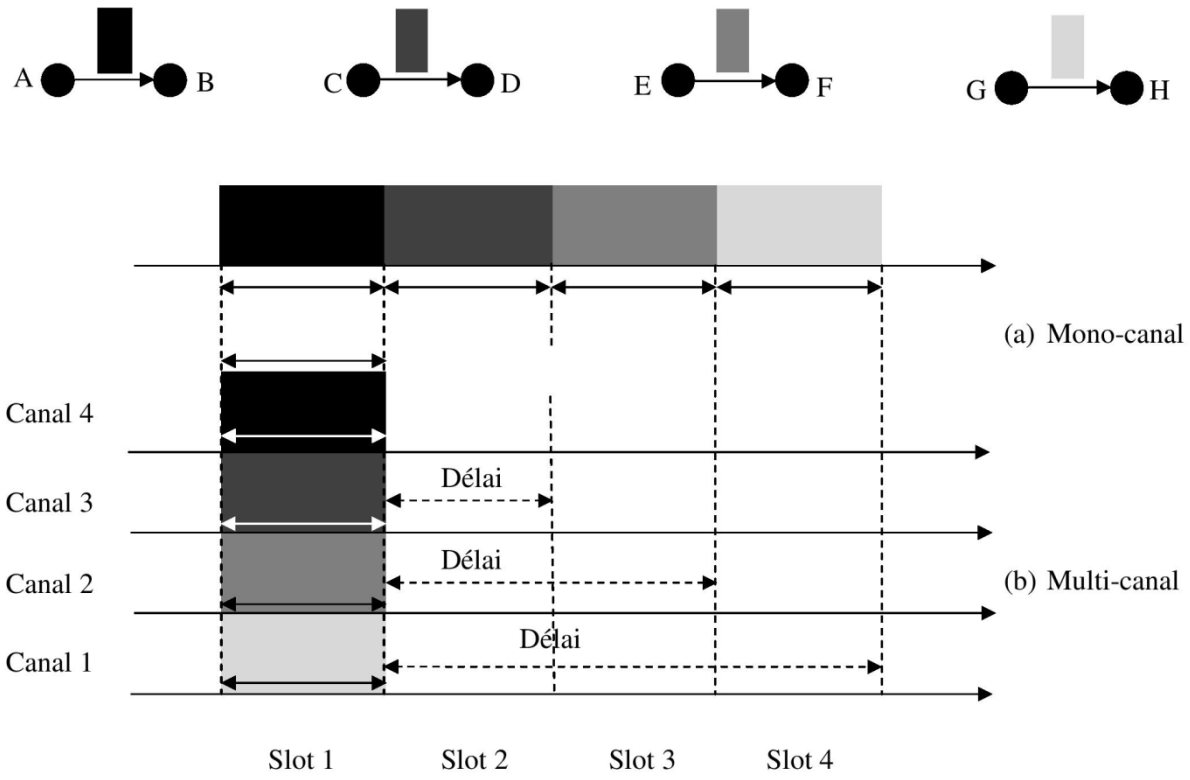


Figure 2.9 Comparaison entre l'utilisation du monocanal et du multi-canal.

Les figures 2.9 (a) et (b) représentent une comparaison entre l'utilisation du monocanal et du multi-canal pour les mêmes exemples représentés respectivement sur les figures 2.7 et 2.8. Dans le cas de l'utilisation d'un seul canal, les quatre transmissions sont effectuées d'une façon séquentielle car une seule transmission peut se faire à la fois afin de limiter les collisions provenant des émissions simultanées de plusieurs nœuds. Alors que dans le cas de l'utilisation de plusieurs canaux où les quatre canaux (1, 2, 3 et 4) sont disponibles, les quatre transmissions peuvent se produire en parallèle sans collision, ce qui améliore la bande passante et réduit les délais.

2.2. Les principales classifications

Les protocoles MAC multi-canaux ont été classés dans la littérature en fonction de plusieurs critères. Ainsi, ils peuvent être classés en fonction de la périodicité de l'attribution des canaux [9] : les protocoles avec attribution fixe de canaux, les protocoles avec attribution semi-dynamique de canaux, et les protocoles avec attribution dynamique de canaux. Chacun de ces protocoles attribue les canaux de manière centralisée ou distribuée. De plus, les protocoles peuvent être également classés en fonction de l'accès au medium : les protocoles basés sur l'ordonnancement temporel ou les protocoles basés sur la contention [10].

Dans les protocoles avec une attribution fixe de canaux, les nœuds utilisent le même canal pour une longue durée. L'idée principale de cette approche est de regrouper les nœuds dans différents *clusters* en attribuant un canal différent à chaque *cluster*. Ceci permet d'éviter les interférences entre les *clusters*, mais ne résout pas le problème des interférences à l'intérieur de chaque *cluster*. Les protocoles d'attribution fixe de canaux ne permettent pas la diffusion des informations, c'est une restriction qui affecte de nombreux protocoles et applications de routage dans les réseaux de capteurs sans fils pour lesquels la diffusion est nécessaire. Dans l'attribution semi-dynamique de canaux, les nœuds s'allouent des canaux fixes, mais ils décident de commuter d'un canal à un autre afin de communiquer avec d'autres nœuds. En revanche, l'affectation semi-dynamique des canaux a besoin d'une coordination entre l'émetteur et le récepteur afin d'être sur le même canal en même temps. Dans l'attribution dynamique de canaux, les nœuds peuvent commuter de canal avant chaque transmission. Une synchronisation précise est nécessaire. Il est également nécessaire que l'émetteur et le récepteur soient actifs en même temps sur le même canal. Cette gestion de la coordination entre les nœuds génère une surcharge du réseau. Dans la pratique, cette surcharge et le délai de commutation de canal ne peuvent pas être ignorés à cause du changement fréquent de canal.

Dans les protocoles centralisés l'allocation de slots et de canaux pour l'ensemble du réseau est gérée par un nœud central responsable de l'ordonnancement de toutes les transmissions. Tandis que, dans les protocoles distribués, chaque nœud choisit son propre slot et canal de communication sans avoir recours à un nœud central.

L'exploitation de plusieurs canaux s'appuie sur une méthode d'accès qui peut être aussi la base d'une classification. Certains protocoles essaient d'ordonnancer toutes les transmissions dans le réseau, ce sont des protocoles MAC sans contention qui sont basés sur une approche TDMA *(Time Division Multiple Access)* [11]. Dans ce type de protocole il n'y a pas de collision, mais ces protocoles souffrent de la surcharge de messages de contrôle. D'autres protocoles adoptent les transmissions asynchrones, ce sont des protocoles MAC basés sur la contention, ces protocoles utilisent CSMA/CA [11] pour accéder au canal. Ce type de protocole est plus flexible et ne nécessite pas de messages de contrôle, mais il souffre de la perte de données due aux collisions.

2.3. Les protocoles MAC multi-canaux dans les RCSF

Dans cette partie, nous allons faire une synthèse des protocoles MAC multi-canaux les plus connus qui ont été proposés pour les réseaux de capteurs sans fil et qui sont applicables à la couche physique de la norme IEEE 802.15.4. Pour chaque protocole étudié, nous décrirons les principaux avantages et nous indiquerons à quelle classe de protocoles il appartient en fonction de la méthode d'attribution de canaux et de l'algorithme d'accès au médium. Nous présenterons également un tableau comparatif de synthèse qui résume les critères essentiels (tableau 2.2) de ces protocoles. Nous avons choisi de classifier les protocoles en deux grandes catégories : les protocoles centralisés et les protocoles distribués.

2.3.1. Les protocoles centralisés

Dans les protocoles centralisés, une entité centrale effectue toutes les allocations de slots et de canaux dans le réseau. Ces protocoles exigent une connaissance de l'ensemble du réseau mais provoquent souvent une surcharge du réseau à cause du trafic de contrôle produit par l'entité centrale. Ils sont utilisés dans le cas des topologies statiques ou de taille limitée.

2.3.1.1. Multi-Channel MAC protocol (MCMAC)

Dans [12], les auteurs proposent MCMAC, un protocole MAC multi-canal centralisé qui est basé sur la méthode TDMA. Le réseau est divisé en *clusters*. Chaque *cluster* possède un chef du *cluster* appelé *Cluster Head* (CH) qui a pour rôle d'attribuer un canal à tous ses descendants qui veulent communiquer. Les nœuds d'un même cluster sont synchronisés. Un *cluster* ne contient pas plus de 64 nœuds. MCMAC suppose aussi qu'il y a plusieurs puits dans le réseau. Le temps est divisé en cycles. Chaque cycle est constitué d'une période d'activité et d'une période d'inactivité dédiée à l'économie d'énergie des nœuds. La période d'activité se compose de quatre phases : la phase de synchronisation des *beacons (synchronous beacon)*, la phase de demande de transmission *(transmission request)*, la phase d'ordonnancement des canaux *(channel schedule)*, et la phase de transmission de données *(data transmit)*. Les CH négocient pour réserver un temps de contact pour les communications entre les CH dans la période d'inactivité sur le canal de contrôle, qui est un canal commun réservé pour la diffusion (messages de contrôle par exemple). Chaque CH envoie un *beacon* synchronisé au début et à la fin de la période d'activité en utilisant le canal de contrôle. Cela permettra aux nœuds d'ajuster leurs horloges de réveil. La phase de demande de transmission est divisée en plusieurs slots de temps. Le nombre de slots de temps est égal au nombre de nœuds dans un cluster donné. Les slots de temps sont attribués conformément à l'identifiant des nœuds de telle sorte que le premier slot de temps est attribué au nœud qui a le plus petit identifiant. Durant la phase de demande de transmission, les CH passent en mode

écoute des requêtes de ses descendants qui demandent l'allocation d'un canal pour l'envoi de leurs paquets. Durant la phase d'ordonnancement de canaux, tous les nœuds du cluster doivent écouter le canal de contrôle sur lequel le CH diffuse un paquet contenant la liste des canaux attribués à chaque nœud. Les nœuds ayant demandé un canal ainsi que les nœuds destinataires recevront le paquet « attribution de canal *(channel assignment)* » et pourront ainsi commuter leurs canaux afin de transmettre/recevoir leurs données sans collision.

MCMAC est simulé en utilisant OMNeT++. L'efficacité énergétique est réalisée par la mise en veille de tous les nœuds des clusters voisins du cluster actif. Ce mécanisme réduit la concurrence et les interférences entre les clusters, mais augmente la latence dans le réseau. Dans MCMAC, avant chaque transmission de trame, une demande d'attribution de canal doit être envoyée au coordinateur ce qui conduit à une augmentation d'*overhead* et du coût énergétique. Le protocole ne spécifie pas comment les CH négocient entre eux l'attribution des slots de temps et des canaux, ni quand les CH échangent des données.

2.3.1.2. Hybrid MAC Protocol (HyMAC)

Dans [13] les auteurs proposent HyMAC comme un protocole hybride qui combine les deux techniques d'ordonnancement temporel et fréquentiel. Ce protocole a été conçu pour des applications de collecte de données[1] dans les réseaux de capteurs sans fil. La période de communication dans HyMAC est un cycle TDMA de longueur fixe. Cette période est composée d'un certain nombre de structures temporelles chacune d'elles étant découpée en plusieurs slots de temps. Un nombre fixe de slots de temps consécutifs dans chaque cycle à partir de son début forment les slots de temps ordonnancés tandis que les slots de temps restants de ce cycle sont des slots de contention. Dans HyMAC, la station de base[2] est responsable de l'allocation de canaux et des intervalles de temps en se basant sur l'algorithme BFS *(Breadth First Search)*. Pour ce faire, elle collecte les messages *Hello* des nœuds voisins pour construire le graphe de la connectivité des nœuds du réseau. Ainsi, chaque nœud est capable d'échanger des données avec son père en utilisant le canal et l'intervalle de temps attribué précédemment par la station de base.

Dans HyMAC, les nœuds éteignent leur module radio quand ils ne sont pas utilisés pour réduire le coût énergétique. Ce protocole a comme avantage d'éviter les collisions entre les nœuds, d'offrir un débit élevé et un délai de bout en bout borné. En revanche, ce protocole n'explique pas certains processus comme la synchronisation des slots des différents nœuds, la façon dont les nœuds s'associent au réseau et la façon dont les collisions sont évitées.

Les auteurs comparent HyMAC au protocole MMSN en termes de nombre de conflits potentiels. Les résultats montrent que HyMAC ne génère pas d'interférences, même avec 2 canaux disponibles uniquement pour les configurations choisies, tandis que MMSN souffre

[1] La collecte de données est un processus de collecte des informations provenant de l'ensemble des nœuds du réseau vers une entité spéciale appelée puits.
[2] Notons que les fonctionnalités du puits sont souvent celles d'une « station de base ».

d'interférences pour ces mêmes configurations. Les auteurs comparent également HyMAC à RT-Link [14]. RT-Link est un protocole qui supporte la voix en *streaming* en temps réel sur les réseaux de capteurs. Les deux protocoles ont été implémentés sur la plateforme FireFly qui utilise le module radio CC2420 [15]. Les résultats de cette implémentation montrent que HyMAC améliore les performances en termes de slots de temps utilisés.

Dans [16], les auteurs proposent EE-MAC *(Energy Efficient hybrid MAC)* qui est une amélioration de HyMAC. Cette amélioration introduit l'utilisation d'une technique de priorité adaptative pour l'accès au medium. Cette priorité est attribuée aux nœuds selon leurs rôles en ajustant la taille initiale de la fenêtre de contention. La priorité dépend des informations recueillies par les nœuds. Ces informations incluent la quantité de données à transmettre, la distance au puits et la charge de batterie restante. Ce protocole diminue le risque de collision car seulement les nœuds du même niveau de priorité se mettent en compétition et non pas tous les nœuds du réseau. En revanche, EE-MAC souffre d'une surcharge liée au mécanisme d'attribution de la priorité aux nœuds qui est faite dynamiquement durant la période de contention.

Les auteurs ont comparé EE-MAC au protocole HyMAC en termes d'efficacité énergétique, de taux de livraison de paquets et de délai de bout-en-bout. Les simulations sont effectuées en faisant varier le nombre de paquets, le nombre de nœuds et la taille de paquets. Les résultats obtenus à l'aide du simulateur NS2 ont montré que EE-MAC apporte une amélioration significative par rapport à HyMAC.

2.3.1.3. Tree-Based Multi-Channel Protocol (TMCP)

Dans [17], les auteurs décrivent un protocole centralisé *« Tree-based Multi-Channel Protocol* (TMCP) *»* pour les applications de collecte de données. Ce protocole est basé sur la méthode d'accès CSMA/CA. Dans TMCP, le réseau est divisé en plusieurs sous-arbres ayant la station de base comme racine commune. A chaque sous-arbre est attribué un canal différent afin de réduire les collisions dans le réseau comme le montre la figure 2.10. TMCP peut fonctionner avec un petit nombre de canaux, et sans la nécessité d'une forte synchronisation. L'algorithme BFS *(Breadth First Search)* est utilisé afin de construire l'arbre durant la phase d'initialisation. Après la construction de l'arbre, la phase d'attribution des canaux commence. Ceci est fait en commençant par la racine du réseau, en n'autorisant qu'un seul père pour chaque nœud tout en s'assurant que chaque nœud est assigné à un sous-arbre dans lequel il crée le moins d'interférences. A chaque sous-arbre est attribué un canal différent afin de minimiser les interférences entre les sous-arbres. La station de base possède plusieurs interfaces afin de pouvoir gérer en parallèle les échanges avec ses fils.

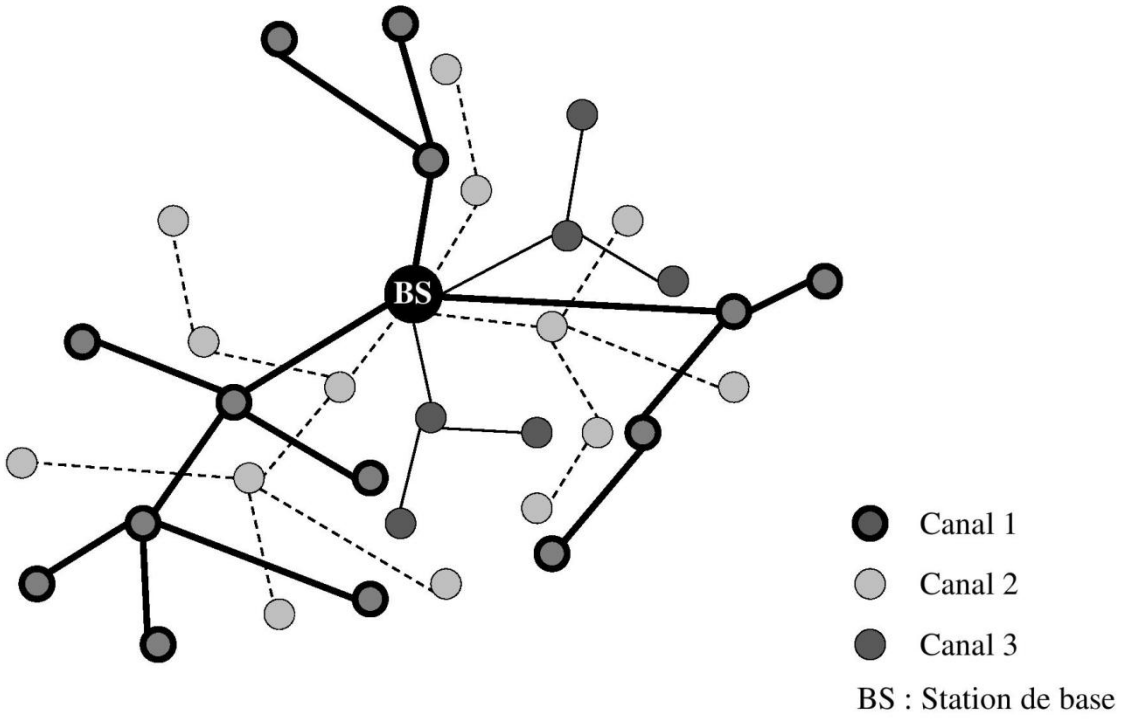


Figure 2.10 La topologie de réseau TMCP.

Ce protocole utilise un algorithme glouton qui diminue les interférences radio. TMCP possède trois types de phases, la détection de canal afin de trouver les canaux orthogonaux disponibles, l'attribution de canal pour diviser l'ensemble du réseau en sous-arbres et attribuer un canal différent pour chaque sous-arbre, et les échanges de données dans le but de gérer la collecte des données à travers chaque sous-arbre.

TMCP est testé en utilisant les modules MicaZ et simulé à l'aide de GloMoSim afin d'évaluer ses performances. Les résultats montrent que TMCP améliore le débit tout en gardant un taux élevé de livraison de paquets et une faible latence. D'autre part, TMCP bloque les communications directes entre les nœuds appartenant à des sous-arbres différents. De plus, les communications à l'intérieur des sous-arbres sont basées sur la contention où l'algorithme CSMA/CA est utilisé ce qui génère beaucoup de perte de trames sous forte charge.

2.3.1.4. Time Synchronized Mesh Protocol (TSMP)

TSMP [18] est un protocole MAC qui est basé sur un ordonnancement temporel. TSMP utilise la technique de saut de canal *(channel hopping)*. Il utilise un coordinateur central qui récupère les listes de voisins de tous les nœuds du réseau afin d'allouer des slots de temps et des canaux de communication pour chaque nœud. TSMP permet l'ordonnancement en utilisant tous les canaux disponibles afin de résister aux interférences et à l'évanouissement

par trajets multiples. TSMP est conçu pour améliorer la fiabilité des transferts de données en combinant trois types de diversité : de temps, de canal et de route.

D'après l'évaluation des mesures en monde réel de TSMP, les auteurs ont démontré [19] les avantages de l'utilisation de sauts de canal en le comparant à la solution d'un canal unique.

D'abord, les auteurs ont comparé l'utilisation d'un canal unique à l'utilisation du saut arbitraire de canal *(blind channel hopping)*. Ce dernier consiste à choisir un canal parmi tous les canaux disponibles de la norme IEEE 802.15.4. Les résultats obtenus ont montré que l'utilisation du saut arbitraire de canal améliore la connectivité, réduit de 56% le nombre moyen de transmissions nécessaires pour relayer un message d'un nœud à un autre et diminue de 38% le nombre de fois où un nœud a changé son parent de routage (la cible de son prochain saut).

Ensuite, l'utilisation de saut arbitraire de canal a été comparée à celle d'une liste blanche *(white listing).* L'utilisation de la liste blanche consiste à choisir les canaux de meilleure qualité pour chaque lien. Les résultats montrent que l'utilisation d'une liste blanche améliore les performances par rapport à celles du saut arbitraire.

Le mécanisme de synchronisation multi-saut utilisé dans TSMP n'est pas détaillé et les auteurs n'expliquent pas comment le message de synchronisation se propage. TSMP souffre également du manque d'évolutivité en raison de la segmentation rigide de TDMA. L'introduction d'un nouveau nœud dans le réseau prendrait un temps important avant qu'il ne soit en mesure de commencer à envoyer des données. Le nouveau nœud doit deviner sur quel canal il faut écouter. De plus, l'absence d'un canal de contrôle entraine des difficultés pour que les nœuds sachent sur quels canaux les voisins[3] travaillent.

2.3.1.5. Multichannel Optimized DElay time Slot Assignment (MODESA)

Dans [20], les auteurs proposent MODESA qui est un protocole basé sur l'ordonnancement temporel (TDMA). MODESA vise à optimiser l'attribution de slots et de canaux pour la collecte de données dans un réseau multi-canal afin de réduire le cycle de collecte TDMA, ceci en prenant en compte le nombre de canaux disponibles. Le but de cet algorithme est de déterminer le nombre minimum de slots de temps qui permet d'assurer aux nœuds du réseau un accès au medium proportionnel à leur trafic.

MODESA s'exécute sur l'ensemble ordonnancé de nœuds en fonction de leurs priorités. C'est-à-dire, le nœud ayant la plus haute priorité est le premier nœud ordonnancé. Les nœuds ont une priorité dynamique qui est égale au produit du nombre de paquets qu'ils ont dans leurs files d'attente à l'itération actuelle et du nombre de paquets que le père du nœud doit recevoir dans un cycle. Ceci favorise les nœuds ayant un grand nombre de données à envoyer au nœud

[3] Nœuds qui sont à portée.

père à condition que ce dernier ait un nombre élevé de données à recevoir dans un cycle. Dans le cas ou plusieurs nœuds possèdent la même priorité, MODESA choisit le nœud ayant le plus petit identifiant. Ensuite, ce nœud utilise le premier canal disponible.

La performance de ce protocole est évaluée par des simulations qui montrent que MODESA a besoin d'une petite taille de file d'attente et réduit le nombre de commutations radio. Les auteurs n'abordent pas la façon de maintenir la synchronisation dans un réseau multi-sauts, et n'expliquent pas comment le réseau est créé. MODESA souffre de la surcharge de messages de contrôle qui sont échangés afin d'ordonnancer toutes les transmissions, surtout quand les répétitions doivent être prises en compte en cas de perte de trames de données.

2.3.1.6. Mulichannel Access for Sensor Networks (MASN)

Dans [21] les auteurs proposent MASN, un protocole multi-canal qui utilise la méthode d'accès CSMA/CA non slotté. Il est conçu pour les réseaux hiérarchiques où le nombre de nœuds est inférieur à 50. Dans ce protocole, les auteurs s'intéressent aux applications de collecte de données utilisant un routage hiérarchique.

Les concepts et le vocabulaire exploités dans cette contribution s'inspirent de ceux de ZigBee Alliance (que nous détaillerons dans la partie 3 de ce chapitre) plutôt que de ceux de la norme IEEE 802.15.4. Dans MASN, l'attribution des différents canaux est centralisée et effectuée par un nœud particulier appelé le coordinateur du réseau (*ZC, ZigBee Coordinator* le CPAN). Il utilise une fonction conçue pour calculer le décalage de canal entre deux fils routeurs *(ZR, ZigBee Router)* successifs inspirée du processus d'attribution d'adresse hiérarchique utilisé par ZigBee. Cette fonction est appelée *CHskip(d)*.

MASN fonctionne sur les 16 canaux de la bande 2.4 GHz du standard IEEE 802.15.4, un canal de contrôle (le canal 11) et 15 canaux utilisés pour la transmission de données (de 12 à 26). La figure 2.11 montre un exemple d'attribution de canal dans MASN. Les canaux sont attribués en fonction de *CHskip*. Considérons l'exemple des fils du ZC, il a attribué le canal 12 (qui est le premier canal disponible) à son premier fils et le canal 20 à son deuxième fils où 20 = 12 + *CHskip(0)*.

Le nombre de canaux utilisés pour l'envoi et pour la réception de données dépend du rôle du nœud dans le réseau. Comme indiqué sur la figure 2.11, pour un ZC, le nombre de canaux de réception *(nCHR, number of reception channels)* est compris entre 1 et 15 tandis que le nombre de canaux de transmission *(nCHS, number of transmission channels)* est égal à 0. Chaque ZR utilise un seul canal pour la transmission et un seul canal pour la réception. En revanche, un ZED (*ZigBee End Device*, une feuille) possède seulement un canal pour la transmission et n'a aucun canal de réception.

L'avantage de ce protocole est sa simplicité et sa facilité d'intégrer les ressources de la norme IEEE 802.15.4 avec de légères modifications de la couche MAC. Les performances de MASN sont évaluées par des simulations en utilisant le simulateur NS-2. Les auteurs comparent

MASN aux protocoles utilisant le monocanal, l'allocation aléatoire et l'attribution de canal par sous-arbre. Les résultats montrent que MASN permet d'améliorer le débit global par un facteur compris entre 2 et 5 par rapport à l'utilisation d'un canal unique, et d'un facteur moyen de 2 par rapport à une allocation aléatoire de canal ou à une distribution simple de sous-arbre. D'autre part, les auteurs ont utilisé une topologie unique pour les simulations qui pourrait être la topologie la plus avantageuse pour MASN.

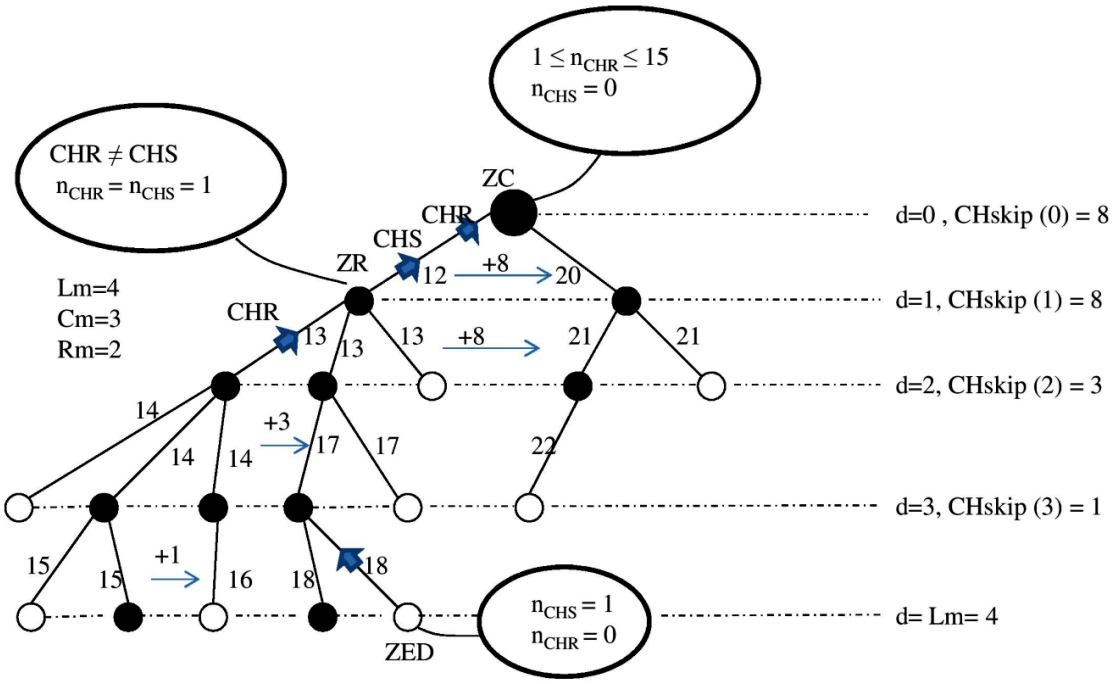


Figure 2.11 Exemple d'attribution de canal dans MASN.

2.3.2. Les protocoles distribués

Les protocoles distribués ne dépendent pas d'un nœud central. Les nœuds ont besoin seulement des connaissances locales de leurs voisins pour choisir leurs slots de temps et leurs canaux de communication. Ces protocoles sont plus adaptés aux changements de la topologie du réseau et en conséquence sont plus appropriés pour les réseaux de capteurs de grande échelle. Dans ce qui suit, nous vous présenterons les principaux protocoles MAC multi-canaux proposés pour les RCSF.

2.3.2.1. Multi-channel Clustered LMAC Protocol

Dans [22], les auteurs proposent un protocole MAC multi-canal basé sur le protocole monocanal économe en énergie LMAC [23]. Ce protocole utilise une méthode d'attribution de canal semi-dynamique. La méthode d'allocation se compose de deux phases. Durant la première phase les nœuds sélectionnent les intervalles de temps selon les règles du protocole LMAC. Ceci empêche les nœuds de choisir le même intervalle de temps utilisé dans leur voisinage à deux sauts. La sélection du canal s'effectue lorsque tous les intervalles de temps sur le canal de base sont utilisés par les voisins à deux sauts. Dans la deuxième phase, le nœud qui n'a pas trouvé un slot libre sur le canal de base scanne les différents canaux pour trouver un nœud pont (un nœud intermédiaire avec lequel il communique afin d'échanger les informations avec les autres nœuds du réseau). Ceci permet au nœud de découvrir tous les slots de temps occupés sur les différents canaux dans son voisinage à deux sauts.

La performance du protocole proposé est évaluée par l'analyse théorique et par des simulations en utilisant OMNeT++. Les résultats montrent que l'utilisation de plusieurs canaux augmente le nombre des nœuds actifs (les nœuds ayant un slot de temps) et réduit les collisions par rapport à LMAC. Cependant ce protocole présente quelques inconvénients. En effet, si deux nœuds voisins utilisent des canaux différents, ils doivent communiquer à travers le nœud pont, ce qui augmente la latence et la consommation d'énergie. Durant la deuxième phase, les diffusions ne peuvent pas être effectuées, et de plus, le fait qu'un nœud puisse être adressé par deux voisins sur deux canaux différents (car les canaux sont attribués aux émetteurs) dans le même intervalle de temps entraîne un gaspillage de la bande passante.

2.3.2.2. Time-Frequency MAC protocol (TFMAC)

Dans [24] les auteurs proposent un protocole MAC hybride nommé TFMAC. C'est un protocole MAC synchronisé qui est basé sur la méthode d'accès TDMA couplé à un mécanisme rapide de sauts de canal *(quick channel hopping mechanism)*. Le temps est divisé en trames (structures temporelles) fixes, chaque trame est composée d'une période de contention et d'une période sans contention. Dans la période de contention, les nœuds surveillent le canal de contrôle sur lequel ils échangent des informations avec leur voisinage. Dans la période sans contention, les nœuds échangent leurs messages de données. Chaque période sans contention est divisée en N slots de temps. Ces slots de temps sont attribués aux nœuds en se basant sur les messages de contrôle échangés au début de chaque trame durant la période de contention.

Le protocole TFMAC comprend deux aspects : l'attribution de canaux et l'accès au médium. TFMAC utilise un simple schéma d'attribution de canaux avec k canaux disponibles. Chaque nœud est capable de sélectionner son canal de réception et de le diffuser à ses voisins afin que chaque nœud puisse savoir le canal à utiliser pour transmettre des données à chacun de ses voisins.

Le schéma d'accès au medium offre pour chaque nœud un ordonnancement pour l'accès sans collision en temps et en canal. Les nœuds sont capables de sélectionner un ensemble de *k* slots de temps (un pour chaque canal) pour l'envoi de leurs paquets de données aux voisins.

Après l'échange de canaux attribués, les nœuds créent les *timetables* (un échéancier) de telle sorte que les messages émis par les nœuds soient sans collision en temps et en canal.

Les auteurs ont comparé TFMAC (avec *k* canaux disponibles) au protocole TDMA (avec un seul canal). La simulation à l'aide d'un outil en C++ montre que TFMAC améliore le délai moyen et le débit maximum du réseau. Cependant TFMAC souffre d'une forte surcharge due à la sélection distribuée de slots/canaux dans le voisinage à deux sauts, mais cet aspect n'a pas été évalué par les auteurs.

2.3.2.3. A Practical Multi-Channel MAC Protocol

Dans [25], les auteurs ont proposé un protocole MAC qui découpe le réseau en clusters. Ce protocole utilise une approche dynamique pour l'allocation des canaux. Afin d'attribuer des canaux différents pour chaque cluster les auteurs utilisent des approches heuristiques et un *feedback* (retour d'information) obtenu d'une manière asynchrone et distribuée. L'utilisation de nouveaux canaux est effectuée seulement lorsque c'est nécessaire. Les auteurs prennent en compte la consommation réelle d'énergie de la commutation de canal.

Un nœud commute son canal dans deux cas. Le premier cas se produit lors de la surcharge du canal utilisé, celle-ci est détectée en fonction du taux de perte de paquets. La décision de commutation de canal dépend de l'impact des collisions et du rôle du nœud dans le réseau. En effet, les nœuds qui se comportent comme des puits ont la priorité de changer leurs canaux en premier pour initier la création d'un nouveau cluster. Ainsi, les voisins qui sont en surcharge de trafic destiné à un nœud commutent sur le même canal afin de s'associer au cluster. Cette phase est appelée expansion du canal *(channel expansion).* Le deuxième cas a lieu quand un canal n'est plus surchargé, les nœuds qui utilisent ce canal invitent ceux du suivant dans la liste des canaux pour partager le canal utilisé par ces nœuds (forme d'agrégation). Cette phase est appelée le rétrécissement du canal *(channel shrinking).*

Ce protocole est évalué en utilisant les modules MicaZ. Les résultats ont montré que le protocole proposé évite la congestion du réseau et apporte une augmentation de débit de 50 % dans les réseaux denses par rapport à l'utilisation d'un canal unique. D'autre part, les nœuds diffusent périodiquement des messages d'état *(status messages)* à leurs voisins, ce qui peut surcharger le réseau. De plus, le choix du canal est basé sur une liste fixe de canaux et ne prend pas en compte la possibilité d'une réutilisation spatiale pertinente.

2.3.2.4. An Energy-Efficient Multi Channel MAC Protocol (Y-MAC)

Y-MAC [26] est un protocole MAC multi-canal basé sur l'ordonnancement temporel. Les auteurs ont adopté une méthode distribuée basée sur les algorithmes proposés dans LMAC [23] et MMSN [27]. Dans ce protocole, les slots de temps sont attribués aux récepteurs d'une façon dynamique. Y-MAC utilise une méthode d'accès hybride où le temps est divisé en cycles. Chaque cycle est constitué d'une période de diffusion et d'une période d'unicast (pour la transmission des messages de données) comme le montre la figure 2.12. Au début de la période de diffusion, les nœuds se réveillent afin d'échanger des messages de contrôle. Les nœuds échangent aussi le temps restant de la trame actuelle afin de maintenir la synchronisation. Y-MAC utilise un vecteur d'allocation de slots qui permet aux nouveaux nœuds de rejoindre le réseau et de choisir un slot de temps d'une manière distribuée. Ce vecteur est utilisé pour stocker des informations sur les slots de temps occupés par les voisins directs et il est diffusé aux voisins à un saut.

Y-MAC réduit la latence en offrant la possibilité aux émetteurs potentiels qui ont des trames en attente destinées à un même récepteur de commuter sur le canal du récepteur et de se mettre en compétition à nouveau.

Y-MAC a été implémenté en utilisant le système d'exploitation RTOS sur les modules TmoteSky. Les auteurs ont comparé Y-MAC à LPL [28] et Crankshaft [29] dans des environnements d'un seul saut avec une fréquence d'envoi d'un paquet chaque 10 secondes. Les résultats montrent que Y-MAC et Crankshaft ont sensiblement la même latence et le même *duty cycle* (comme un indicateur d'efficacité énergétique) et qu'ils sont tous les deux plus performants que LPL. Dans des conditions de trafic plus élevé (de 1 paquet par seconde), Y-MAC obtient un meilleur taux de réception par rapport aux deux autres protocoles. Les auteurs ont également comparé Y-MAC à Crankshaft dans des environnements multi-sauts, les résultats montrent que le *duty cycle* de Y-MAC est légèrement supérieur à celui du Crankshaft. De plus, lorsque la charge de trafic augmente, la latence moyenne par saut de Crankshaft augmente plus rapidement que celle de Y-MAC. Notons aussi que Y-MAC est plus performant que Crankshaft en terme de taux de réception de données.

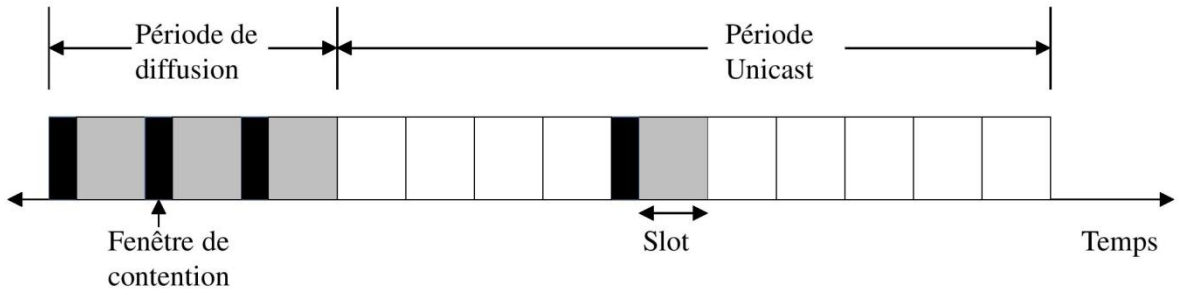


Figure 2.12 Découpage temporel de Y_MAC.

2.3.2.5. Multi-Frequency Media Access Control protocol (MMSN)

MMSN [27] est un protocole qui combine les deux mécanismes FDMA et CSMA. Chaque nœud du réseau doit choisir un canal qui lui servira à la réception des données, les nœuds dans ce protocole sont synchronisés et utilisent un slot de temps divisé en deux parties : la première partie pour la diffusion et la deuxième pour les communications *unicast*. Au début de chaque time slot, les nœuds se mettent en compétition pour accéder au medium afin de diffuser leurs trafics de contrôle en utilisant le canal de diffusion commun.

MMSN propose, évalue et analyse différentes approches pour l'allocation de canaux: attribution des canaux exclusifs, *even-selection*, *eavesdropping*, et *implicit-consensus*.

Selon l'approche exclusive, le nombre de canaux doit être supérieur ou égal au nombre de nœuds dans le voisinage à 2-sauts. Les nœuds échangent leur identifiant (ID) dans leur voisinage à 2-sauts. Les nœuds choisissent des canaux dans l'ordre croissant de leurs identifiants. Si un nœud possède le plus petit ID dans son voisinage à 2-sauts, il choisit le plus petit canal parmi ceux qui sont disponibles, puis il envoie le canal choisi à ses voisins à 2-sauts. Cette stratégie garantit aux nœuds l'attribution de canaux différents dans leur voisinage à 2-sauts, mais elle provoque une surcharge de communication élevée en raison des échanges d'ID et de canaux avec leurs voisins à 2-sauts en utilisant le CSMA/CA.

L'approche *even-selection* est plus adaptée aux réseaux de capteurs sans fils denses où le nombre de canaux est plus petit que le nombre de nœuds dans un voisinage à 2-sauts. Cette stratégie est similaire à la méthode exclusive, mais si tous les canaux ont déjà été attribués dans un voisinage donné alors qu'il reste encore des nœuds n'ayant pas des canaux assignés, ces nœuds doivent choisir d'une façon aléatoire l'un des canaux les moins utilisés.

Selon la méthode *eavesdropping*, chaque nœud prend un *backoff* aléatoire durant lequel il laisse ses voisins à un saut choisir un canal. Lorsque son *backoff* expire, il choisit aléatoirement un des canaux les moins utilisés dans son voisinage à un saut et ensuite il le diffuse. Cette stratégie a moins de surcharge par rapport aux deux premières, mais les nœuds collectent seulement des informations des canaux dans leur voisinage à un saut durant un intervalle de temps aléatoire limité ce qui provoque des conflits, ainsi que des allocations de canaux non optimums qui provoquent des interférences.

Selon l'approche *implicite-consensus*, les nœuds échangent leur ID dans leur voisinage à 2-sauts comme dans les deux premières stratégies. Cette stratégie repose sur un générateur pseudo-aléatoire. Pour chaque canal, chaque nœud calcule un nombre aléatoire pour lui-même et un nombre aléatoire pour chaque nœud de son voisinage à 2-sauts en utilisant le même générateur de nombre pseudo-aléatoire. Un nœud gagne le canal actuel si son numéro aléatoire est le plus élevé parmi tous les autres numéros aléatoires. Cette stratégie garantit moins d'*overhead* mais suppose qu'un grand nombre de canaux est disponible par rapport à la densité.

2.3.2.6. Traffic-Aware Channel Assignment mechanism

Dans [30], les auteurs proposent un mécanisme distribué d'allocation de canaux basé sur la connaissance du trafic. Chaque nœud a un poids de trafic en fonction de son débit de données. Les nœuds collectent les ID et les poids de trafic de tous les voisins à 2-sauts. Ensuite, ils choisissent le canal dans l'ordre décroissant de leurs poids de trafic. C'est à dire que si un nœud possède le plus grand poids de trafic dans son voisinage à 2-sauts, il choisit le canal ayant le moins de charge parmi les canaux disponibles. Si deux nœuds ont le même poids de trafic, le nœud qui a le plus petit ID choisit son canal en premier. Quand un nœud choisit un canal, il l'envoie à ses voisins à 2-sauts.

Ce mécanisme est caractérisé par une forte surcharge de communication en raison des échanges d'ID et de poids du trafic entre les voisins à 2-sauts. De plus, les auteurs n'ont pas pris en compte les fluctuations de la charge du trafic, ce protocole suppose que les débits sont constants et connus à l'avance.

2.3.2.7. Multi-Channel Lightweight MAC Protocol (MC-LMAC)

MC-LMAC [31] repose sur le protocole LMAC[23]. Le but de MC-LMAC est de maximiser le débit du réseau de capteurs en coordonnant les transmissions sur plusieurs canaux. Ce protocole garantit que le même slot/canal n'est pas utilisé simultanément par des voisins à 2-sauts. Chaque slot est composé de deux parties. La première partie est consacrée à la signalisation, durant laquelle tous les nœuds commutent leur module radio sur le canal de contrôle. La deuxième partie du slot est utilisée pour transférer les données. Chaque nœud ayant des données à transmettre au nœud destinataire doit le rejoindre sur son canal de communication. Lorsqu'un nœud n'a pas de données à envoyer, il doit commuter son canal sur le canal de contrôle pour voir si un nœud veut communiquer avec lui. Chaque nœud gère un vecteur de slots occupés pour chaque canal avec une longueur égale au nombre de slots de temps. Chaque bit dans ce vecteur est mis à 1 si le slot de temps à la même position est occupé. Ensuite chaque nœud envoie ce vecteur à ses voisins à un saut.

La figure 2.13 montre un exemple de 7 nœuds, 2 canaux et 5 slots de temps. Tous les nœuds ont choisi un slot de temps et un canal à l'exception du nœud central. Le nœud central effectue l'opération « OR » sur les vecteurs des slots occupés recueillis de ses voisins directs et donc détecte qu'il peut utiliser le slot numéro 5 sur le canal 2.

La performance de MC-LMAC est évaluée par simulation en utilisant le simulateur GloMoSim. Il est montré qu'en augmentant le nombre de canaux, un meilleur débit est atteint et dans des conditions particulières il semble se rapprocher du débit maximal théorique. D'autre part, MC-LMAC souffre de la surcharge des messages de contrôle qui sont échangés afin de découvrir les canaux et les slots de temps libres dans le voisinage à 2-sauts et de la

surcharge de transmission car chaque trame de données doit être précédée d'une trame de contrôle. Le problème s'accentue avec l'augmentation de la densité du réseau.

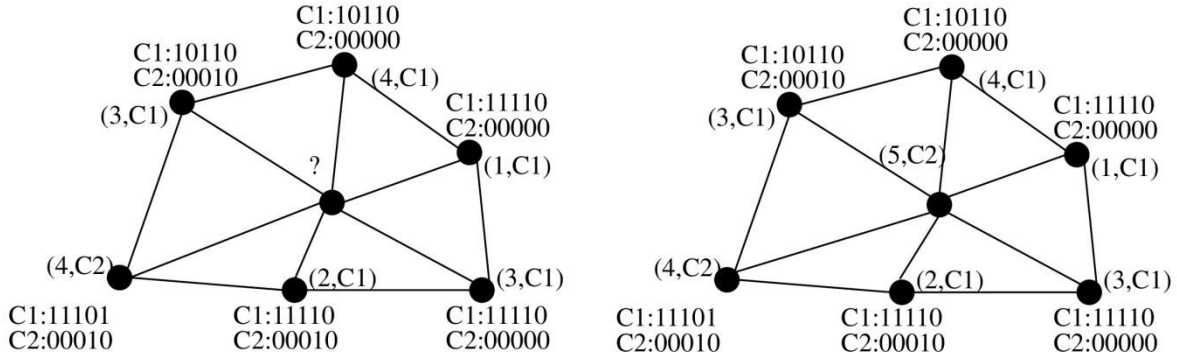


Figure 2.13 L'allocation de slot/canal basée sur des vecteurs d'occupation.

2.3.2.8. Low Overhead MAC Protocol Much MAC (MuChMAC)

Dans [32], les auteurs proposent le protocole MuChMAC. Il s'agit d'un protocole MAC à faible surcharge qui combine les techniques MAC TDMA et asynchrone. Ce protocole utilise une approche dynamique d'attribution de canal. Selon MuChMAC, le temps est divisé en slots. Au début de chaque slot de temps, les nœuds commutent leur canal. Chaque nœud est capable de choisir d'une manière indépendante la séquence de commutations de son canal de réception. Ceci est fait en utilisant un générateur pseudo aléatoire qui prend comme entrée l'ID du nœud et le numéro du slot en cours d'utilisation.

Un slot de diffusion est inséré tous les *n* slots comme le montre la figure 2.14. Ces slots de diffusion suivent également une séquence de saut de canal pseudo-aléatoire, mais durant ces slots le canal choisi est le même pour tous les nœuds. Un émetteur est capable de calculer le canal du récepteur en utilisant le même générateur pseudo aléatoire. Ensuite, il commute son module radio sur le canal du récepteur et commence à envoyer des petits messages de préambule. Lorsque le récepteur se réveille, il entend un préambule et il informe l'émetteur qu'il est réveillé. Dans le but de supporter les réseaux denses, les auteurs proposent de segmenter les slots en sous-slots, ceci permet de diminuer le risque que plusieurs nœuds choisissent la même combinaison « canal, sous-slot » pendant un slot donné.

MuChMAC est testé en utilisant le module Sentilla JCreate afin d'évaluer ses performances. Les auteurs comparent MuChMAC à X-MAC [33] (protocole MAC à *duty cycle* asynchrone). Les résultats montrent que MuChMAC est plus économe en énergie et améliore la fiabilité. D'autre part, l'attribution des canaux est basée sur un mécanisme aléatoire qui ne prend pas en compte les canaux utilisés dans les voisinages des nœuds, ce qui provoque des collisions importantes.

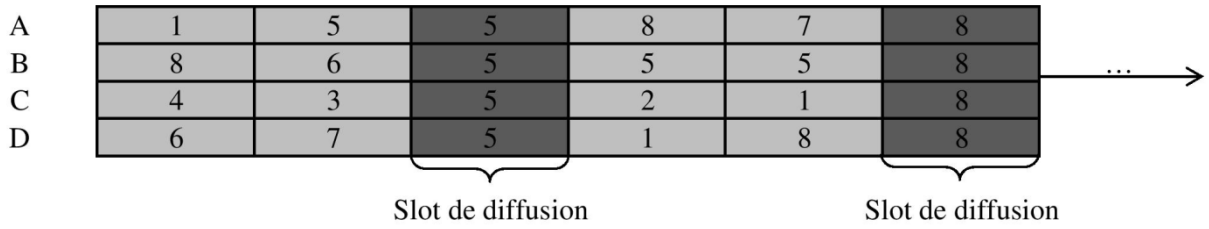


Figure 2.14 Rendez-vous parallèles avec des slots de diffusion.

2.3.3. Synthèse :

Différentes propositions ont été faites afin de gérer l'allocation des canaux dans le réseau. La plupart des protocoles gèrent la découverte de voisinage jusqu'à 2-sauts dans leur algorithme d'allocation de slots de temps et de canaux. Ceci n'est pas suffisant pour éviter les collisions et les interférences quand les acquittements de couche MAC sont utilisés. D'autres protocoles attribuent les canaux d'une manière aléatoire ou en évitant uniquement les canaux utilisés par les voisins directs, ceci génère de fréquentes collisions qui dégradent les performances du réseau. D'autres solutions proposent de fragmenter le réseau en plusieurs groupes et d'attribuer à chaque groupe un canal différent. Ces protocoles évitent les interférences entre les groupes mais ne résolvent pas le problème des collisions à l'intérieur des groupes.

Le tableau 2.2 récapitule les caractéristiques des différents protocoles MAC multi-canaux discutés dans l'état de l'art.

Protocole	Comparaison des protocoles multi-canaux MAC pour les réseaux de capteurs								
	Année de publication	**Méthode d'attribution des canaux**	**Synchronisation**	**Accès au medium**	**Diffusion supportée**	**Canal utilisé pour le transfert de données**	**Allocation de canal**	**Méthode d'évaluation**	**Objectif**
MCMAC	2006	Centralisée	Requise	Basé sur l'ordonnancement temporel (TDMA)	Intérieur des clusters	Canal nommé *(appointed channel)*	Dynamique	Simulations (OMNeT++)	L'efficacité énergétique
Multi-Channel Clustered LMAC	2006	Distribuée	Requise	Basé sur l'ordonnancement temporel (TDMA),LMAC	Oui	Emetteur	Semi-dynamique	Simulations (OMNeT++)	Augmenter les transmissions parallèles dans les réseaux denses de capteur sans fils
TFMAC	2007	Distribuée	Requise	Basé sur l'ordonnancement temporel (TDMA)	Oui	Récepteur	Semi-Dynamique	Simulations (C++)	Augmenter le débit du réseau
HyMAC	2007	Centralisée	Requise	Basé sur l'ordonnancement temporel (TDMA)	Pas d'info.	Emetteur	Semi-Dynamique	Implémentation FireFly +Simulations	Offrir un débit élevé et un délai de bout en bout borné
TMCP	2008	Centralisée	Non requise	Basé sur la contention (CSMA)	Intérieur des branches	Sous-arbre	Fixe	Simulations (GloMosim)	Pour les applications de collecte de données
TSMP	2008	Centralisée	Requise	Basé sur l'ordonnancement temporel (TDMA)	Oui	Canal nommé *(appointed channel)*	Dynamique	Développé dans un réseau géré	Améliorer la fiabilité de transfert de données en combinant le temps, la fréquence et la diversité de routage
Practical	2008	Distribuée	Non Requise	No info	Entre les clusters	Récepteur	Dynamique	Implémentation Micaz +simulations	Utilisation efficace du multi-canal
Y-MAC	2008	Distribuée	Requise	Hybride	Oui	Récepteur	Dynamique	système d'exploitation RTOS sur les motes TmoteSky	Réduire la latence

Protocole	Comparaison des protocoles multi-canal MAC pour les réseaux de capteurs								
	Année de pubilication	**Méthode d'attribution des canaux**	**Synchronisation**	**Accès au medium**	**Diffusion supportée**	**Canal utilisé pour le transfert de données**	**Allocation de canal**	**Méthode d'évaluation**	**Objectif**
MMSN	2010	Distribuée	Requise	Hybride	Oui	Récepteur	Semi-Dynamique	Simulations (GloMosim)	Augmenter les transmissions parallèles
Traffic Aware	2010	Distribuée	Requise	Hybride	Oui	Récepteur	Semi-Dynamique	Simulations	Considérer les volumes de trafic
MC-LMAC	2010	Distribuée	Requise	Basé sur l'ordonnancement temporel (TDMA)	Oui	Emetteur	Semi-Dynamique	Simulation (GloMosim)	Maximiser le débit du réseau
Much MAC	2010	Distribuée	Requise	Basé sur l'ordonnancement temporel (TDMA), X-MAC	Oui	Récepteur	Dynamique	Le mote Sentilla JCreate	Améliorer la bande passante
MODESA	2012	Centralisée	Requise	Basé sur l'ordonnancement temporel (TDMA)	Pas claire	Emetteur	Dynamique	Simulations	Déterminer le Nombre minimum de slots afin de fournir un débit élevé
EE-MAC	2013	Centralisée	Requise	Hybride	Oui	Emetteur	Semi-dynamique	NS2	Améliorer l'efficacité énergétique, le délai et le taux de livraison de paquets
MASN	2013	Centralisée	Non requise	Basé sur la contention (CSMA)	Pas claire	Récepteur	Semi-dynamique	NS2	Améliorer la bande passante

Tableau 2.2 Synthèse des protocoles MAC multi-canaux pour les réseaux de capteurs.

3. Présentation synthétique des solutions basées sur la norme IEEE 802.15.4

Dans cette partie, nous allons présenter une synthèse des standards sans fils qui se basent sur la norme IEEE 802.15.4.

3.1. La couche réseau de ZigBee

Les spécifications ZigBee [3] sont définies et développées par la ZigBee Alliance [34] qui est une organisation internationale travaillant sur la création des normes de réseaux sans fils. Le standard ZigBee a gagné un intérêt croissant dans les réseaux sans fil pour les applications à faible débit, à faible coût, à courte portée et à faible consommation énergétique. Cet intérêt a entrainé l'utilisation de ce standard par un grand nombre des nouvelles applications comme la domotique [35] [36] (éclairage, chauffage et système d'alarme), la surveillance de la santé [37] (suivi de patients) et la surveillance environnementale.

La pile protocolaire ZigBee est composée de quatre couches principales : la couche physique (PHY), la couche d'accès au medium (MAC), la couche réseau (NWK) et la couche application (APL). Le standard ZigBee définit les couches supérieures (la couche NWK et la couche APL) en se basant sur le standard IEEE 802.15.4 pour les couches PHY et MAC. Chaque couche offre un ensemble de services spécifiques pour la couche supérieure. Les différentes couches communiquent à travers des interfaces.

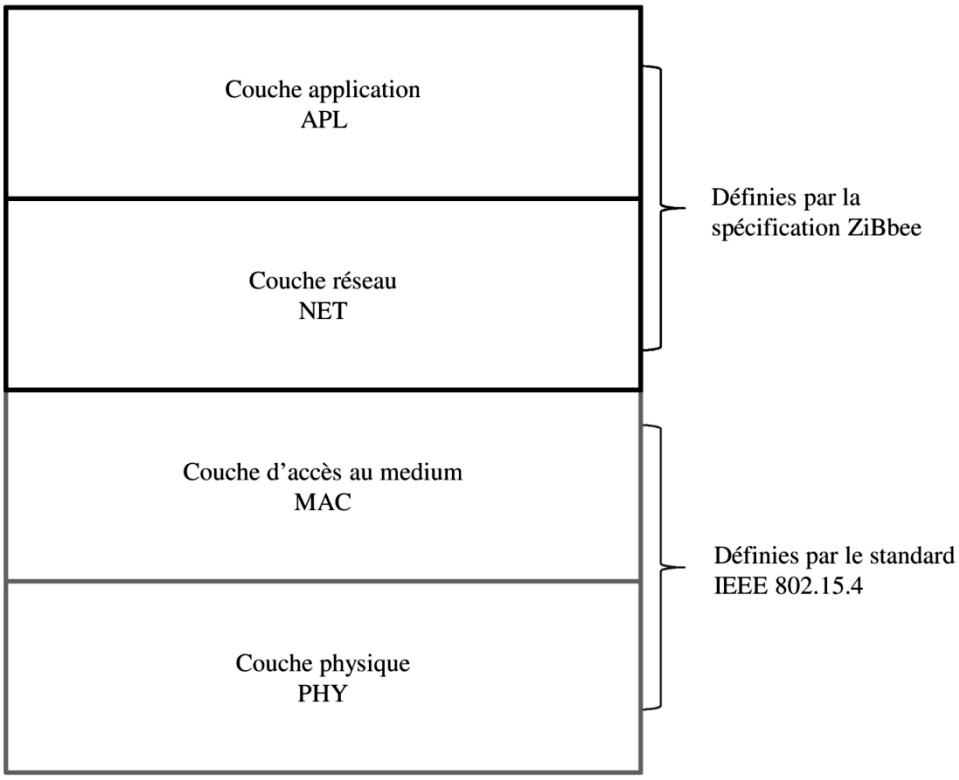


Figure 2.15 La pile protocolaire définie par les spécifications de la ZigBee Alliance.

La couche réseau du standard ZigBee est responsable de la gestion de la table de routage, la découverte du voisinage à plusieurs sauts, la création de la topologie, l'allocation des adresses logiques, le choix du prochain saut et le routage multi-sauts.

ZigBee identifie trois types de nœuds : le Coordinateur ZigBee *(ZC, ZigBee Coordinator)*, le Routeur ZigBee *(ZR, ZigBee Router)* et le nœud terminal ZigBee *(ZED, ZigBee End Device)* qui correspondent respectivement au coordinateur du PAN, au coordinateur et au nœud terminal ou à la feuille de la norme IEEE 802.15.4.

La couche réseau supporte trois types de topologies : la topologie en étoile, la topologie maillée et la topologie en arbre. La figure 2.16 montre des exemples de ces topologies.

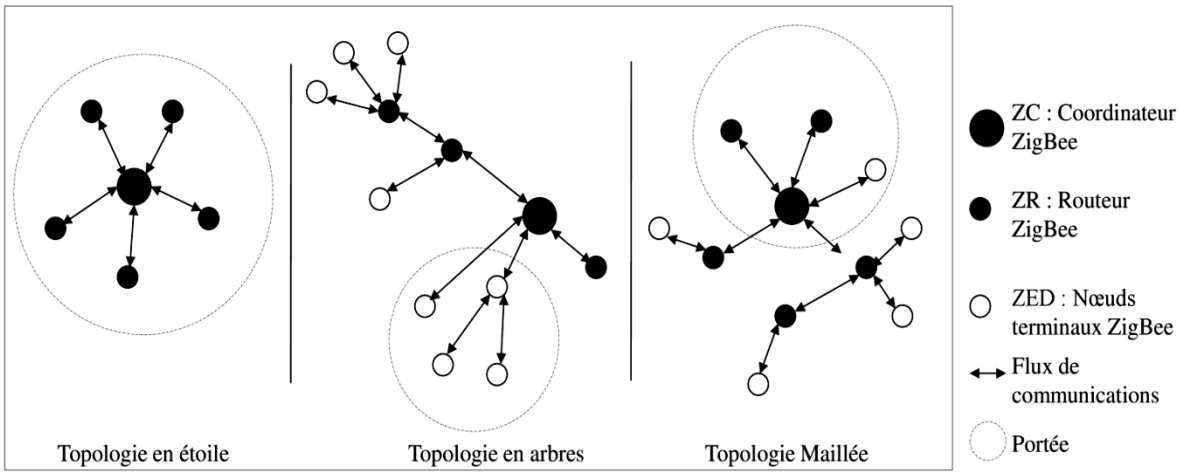


Figure 2.16 Types de topologies supportés par le standard ZigBee.

La topologie en étoile correspond à celle définie par la norme IEEE 802.15.4. La topologie maillée et la topologie en arbre sont plus complexes et plus étendues. Dans ces topologies, le ZC est l'initiateur du réseau. Ces topologies sont composées de ZR et de ZED. Un nœud peut rejoindre le réseau en s'associant au ZC ou à un ZR. Dans une topologie maillée, les routes sont créées et sont maintenues à l'aide d'un ensemble de mécanismes basés sur le protocole de routage AODV *(Ad hoc On-demand Distance Vector)* [38]. Dans ce type de topologie, il n'existe pas d'hiérarchie entre les nœuds. En revanche, la topologie en arbre est un cas particulier d'une topologie maillée où il existe un chemin unique de routage entre n'importe quelle paire de nœud grâce au mécanisme d'attribution d'adresses logiques qui permet de calculer les routes en se basant uniquement sur les adresses hiérarchiques.

3.1.1. Adressage

Chaque entité conforme à la norme IEEE 802.15.4 possède une adresse unique MAC IEEE nommée adresse longue. Cette adresse est composée de 8 octets, elle est donnée lors de la fabrication de la carte réseau. Dans le but de réduire la taille des champs d'adressage dans les trames échangées, la couche réseau utilise des adresses courtes qui sont limitées à 2 octets et assure la correspondance entre ses adresses courtes et les adresses longues.

Lorsqu'un nœud s'associe au réseau, le coordinateur de réseau lui attribue une adresse courte. L'attribution d'adresses se fait par deux méthodes différentes : l'attribution des adresses hiérarchiques et l'attribution des adresses aléatoires. Dans ce qui suit nous nous intéressons uniquement à l'attribution des adresses hiérarchiques.

Attribution des adresses hiérarchiques : dans une topologie en arbre, l'allocation d'adresses courtes est effectuée d'une manière distribuée en se basant sur un mécanisme hiérarchique. Ce mécanisme offre pour chaque coordinateur une plage d'adresses limitée et unique qui lui permet de distribuer des adresses à ses nœuds fils (à sa descendance).

Cette allocation est basée sur les trois paramètres qui caractérisent l'arbre associé au réseau ainsi que sur la profondeur du nœud dans cet arbre. Ces trois paramètres sont connus par le ZC et transmis dans le *beacon*. Ces trois paramètres sont :

- C_m : le nombre maximal de fils associés à un routeur ZigBee,
- R_m : le nombre maximal de fils routeurs associés à un routeur ZigBee,
- L_m : la profondeur maximale de la topologie.

La profondeur du nœud, notée d *(depth)*, dans l'arbre représente le nombre de sauts à effectuer pour atteindre le coordinateur ZC. Le ZC est à la profondeur 0 tandis que les fils du ZC sont à la profondeur 1.

Chaque parent routeur utilise les valeurs d, R_m, C_m et L_m pour définir la plage d'adresses de ses descendants selon la formule *Cskip(d)*. Il distribue cette plage à ses descendants en fonction de sa profondeur dans le réseau. Pour une profondeur donnée d, *Cskip(d)* est calculé selon la formule suivante :

$$Cskip(d) = \begin{cases} 1 + C_m * (L_m - d - 1) & si\ R_m = 1 \\ \dfrac{1 + C_m - R_m - C_m * R_m^{L_m - d - 1}}{1 - R_m} & sinon \end{cases}$$

Un nœud qui a une valeur de *Cskip(d)* égale à 0 n'est pas capable d'associer des fils et est considéré comme un nœud terminal. En revanche, un nœud qui a une valeur de *Cskip(d)* supérieure à 0 est capable d'associer des fils et de leur distribuer des adresses courtes.

L'adresse du $n^{ième}$ fils routeur A_{ZR} est calculée selon la formule suivante :

$$A_{ZR} = A_p + 1 + nb_{ZR} * Cskip(d)$$

L'adresse du $n^{ième}$ fils ZED est calculée selon la formule suivante :

$$A_{ZED} = A_p + R_m + Cskip(d) + nb_{ZED}$$

Nous désignons par A_p l'adresse courte du nœud parent de profondeur d qui possède nb_{ZR} fils routeurs et nb_{ZED} fils ZED.

3.1.2. Routage

Le routage est un mécanisme d'acheminement de paquets d'une source vers une destination finale. Dans un réseau ZigBee, le mécanisme utilisé dépend de la topologie du réseau. Dans ce qui suit, nous nous intéressons uniquement au routage hiérarchique. Ce mécanisme est basé sur le schéma d'adressage hiérarchique qui a été définit par Motorola [39]. Le routage hiérarchique est caractérisé par les liens père-fils qui sont construits en fonction de la topologie où chaque nœud dans le réseau possède un seul père. Ce type de routage ne nécessite pas d'échanges de données supplémentaires pour l'acheminement des paquets, chaque nœud est capable de calculer le prochain saut en fonction de l'adresse de destination. Le nœud terminal achemine directement les paquets à son père. Un nœud routeur applique l'algorithme ci-dessous.

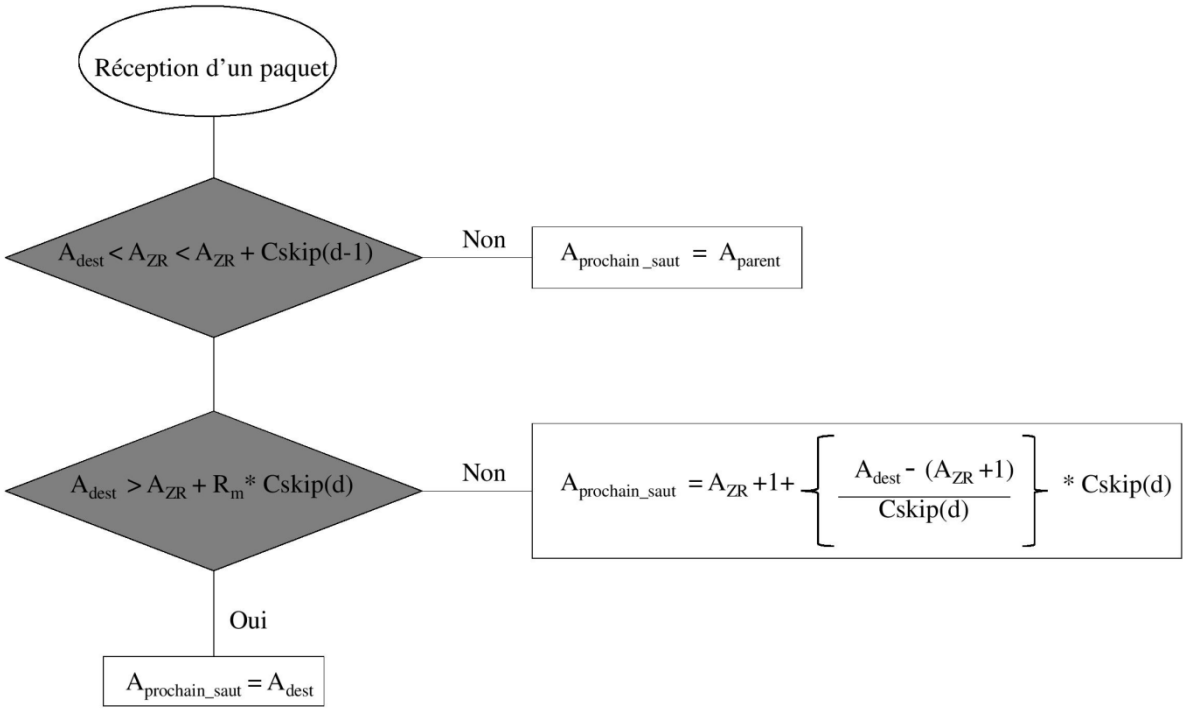


Figure 2.17 L'algorithme de routage hiérarchique.

Dans cet algorithme nous désignons par *A* l'adresse courte du nœud routeur *ZR* qui possède une profondeur d, par A_p l'adresse courte de son père et par A_{dest} l'adresse de destination finale. Lorsqu'un nœud routeur reçoit un paquet de données, il vérifie tout d'abord s'il est la destination finale, si c'est le cas il envoie le paquet à la couche supérieure. Si le nœud routeur n'est pas le destinataire final, il doit calculer l'adresse du prochain saut. Pour ce faire, ce nœud vérifie si l'adresse de destinataire est comprise entre A_{ZR} et $A_{ZR} + Cskip(d-1)$, si c'est le cas la destination finale est un de ses descendants. Ensuite, il vérifie si l'adresse de la destination finale est supérieure à $A_{ZR} + R_m + Cskip(d)$, deux cas peuvent se présenter :

- Si oui, le nœud destinataire est l'un de ses nœuds terminaux, alors le paquet est acheminé au nœud approprié.
- Sinon il calcule l'adresse de son nœud fils, vers lequel il doit acheminer le paquet, en appliquant la formule suivante : $A_{prochain\_saut} = A_{ZR} + 1 + \left\{\frac{A_{dest} - (A_{ZR}+1)}{Cskip(d)}\right\} * Cskip(d)$.

Dans le cas où l'adresse du nœud destinataire n'est pas dans la plage d'adresses du nœud routeur, ce dernier achemine le paquet à son père.

Notons que cette technique de routage sera exploitée dans notre contribution.

3.2. *WirelessHART (Wireless Highway Addressable Remote Transducer protocol)*

Le standard *WirelessHART* [4] est un standard international qui a été développé par la fondation de communication HART et approuvé en juin 2007. Ce standard offre une communication sans fils centralisée dans un réseau maillé. Il est conçu pour les applications de contrôle et de surveillance, pour les applications industrielles et pour les applications d'automatisation de procédés.

3.2.1. Composants du réseau *WirelessHART*

Un réseau *WirelessHART* est composé de différents appareils comme illustré sur la figure 2.18. Ces appareils sont :

- Les appareils de terrain *WirelessHART (WFD, WirelessHART Field Device)* : ils peuvent s'agir d'appareils avec *WirelessHART* intégré ou d'appareils HART existants ayant un adaptateur *WirelessHART* greffé. Ces appareils sont reliés au processus ou au

matériel industriel. Ils peuvent mesurer les variables du processus et retransmettre les paquets reçus aux autres appareils.

- Les passerelles *WirelessHART* : elles permettent la communication entre les appareils de terrain et les applications hôtes. Chaque réseau *WirelessHART* nécessite une passerelle et un point d'accès. La passerelle supporte un ou plusieurs points d'accès.

- Le gestionnaire du réseau : il gère la configuration du réseau, la planification de la communication entre les appareils, la table de routage et la surveillance de la santé du réseau. Un réseau *WirelessHART* peut inclure plusieurs gestionnaires de réseau pour des raisons de redondance, mais un seul gestionnaire peut être actif à la fois. Le gestionnaire du réseau peut être intégré à la passerelle, à l'application hôte ou au SNCC0 (Systeme Numérique de Contrôle-Commande). Ce système est utilisé pour contrôler automatiquement un processus.

- Le gestionnaire de sécurité : il a pour rôle de gérer et de distribuer les clés de cryptage de sécurité. Ce gestionnaire contrôle les appareils qui sont autorisés à rejoindre le réseau.

- Les adaptateurs : ils permettent l'intégration des appareils de terrain HART dans le réseau *WirelessHART*.

- Les ordinateurs de poche (ou tablettes) : ils assurent l'accès aux appareils de terrain adjacents.

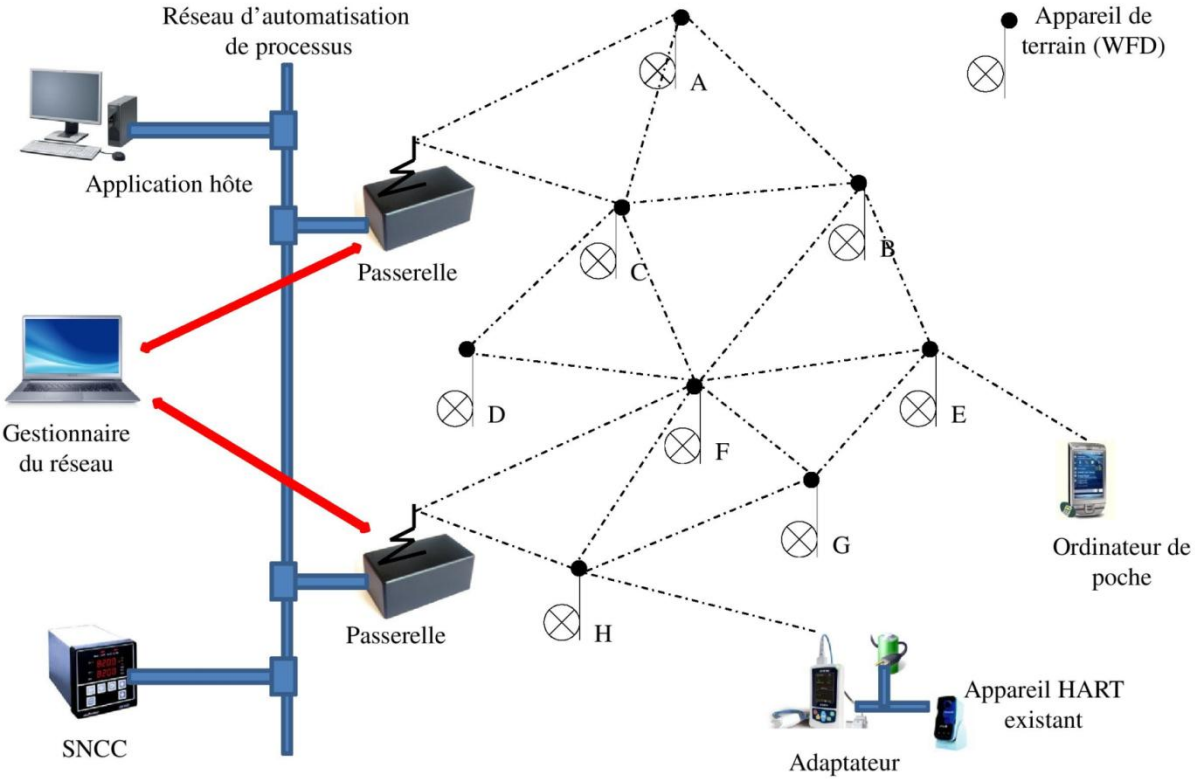


Figure 2.18 Présentation d'un réseau *WirelessHART* [40].

3.2.2. La pile protocolaire de *WirelessHART* :

Dans *WirelessHART*, la pile protocolaire définit la majorité des couches du modèle OSI [41], ces couches sont les suivantes : la couche physique, liaison de données, réseau, transport et application. Dans cette partie, nous nous intéressons uniquement à la couche physique, la couche liaison de données et la couche réseau.

La couche physique de *WirelessHART* :

La couche physique de *WirelessHART* est basée sur la couche physique de la norme IEEE 802.15.4 et fonctionne sur les 15 canaux (11-25) définis par cette norme dans la bande 2.4 GHz avec un débit de données de 250 Kbps. Notons que le canal 26 n'est pas utilisé dans *WirelessHART* puisqu'il n'est pas autorisé dans certains pays.

La couche liaison de données de *WirelessHART* :

La couche liaison de données de *WirelessHART (Data Link Layer DLL)* est basée sur la couche MAC de la norme IEEE 802.15.4. Cette couche améliore le déterminisme de la couche MAC en utilisant la technique TDMA avec un saut de canal de communication

synchronisé, ceci permet d'assurer des communications planifiées sans collisions entre les appareils HART. Les intervalles de temps ont une durée de 10 ms, ce qui est suffisant pour la transmission ou la réception d'un message de données avec accusé de réception. Dans le cas des messages de diffusion l'acquittement n'est pas nécessaire et plusieurs récepteurs peuvent se voir affecter le même slot de temps. Une super-trame est définie pour gérer les intervalles de temps. Elle est composée d'une séquence d'intervalles de temps consécutifs. Cette super-trame se répète périodiquement avec un taux fixe pendant toute la durée de vie du réseau. D'autres super-trames peuvent être ajoutées comme des super-trames additionnelles pour supporter des trafics supplémentaires. Notons qu'il y a au moins une super-trame utilisée dans le réseau. Dans une super-trame, chaque slot de temps peut être dédié à un nœud seul ou partagé par différents nœuds. Pour les slots partagés, en cas de collision, les appareils sources utilisent le mécanisme de back-off aléatoire pour procéder à de nouvelles tentatives. En effet, à chaque fois qu'une tentative d'accès au medium échoue, la taille de la fenêtre d'attente s'élargit. Ceci est similaire à d'autres protocoles basés sur la contention.

Pour chaque slot de temps, une transaction est identifiée par le vecteur suivant : {trame id, indice, type, adresse source, adresse de destination, offset de canal} où la « trame id » identifie la super-trame concernée. L'indice identifie le numéro du slot dans la super-trame. Le type indique si c'est un slot d'émission, de réception ou de veille. Le canal offset indique le canal logique qui doit être utilisé dans la transaction.

WirelessHART évite l'utilisation de certains canaux en introduisant l'idée d'une liste noire *(black listing)* de canaux. Les canaux qui sont affectés par des interférences pourraient être mis en liste noire. De cette façon, le gestionnaire du réseau peut désactiver totalement l'utilisation des canaux de cette liste noire. Pour supporter le saut de canal, chaque appareil maintient une table de canaux actifs. Ainsi, la table peut avoir moins de 15 canaux si une liste noire de canaux est gérée. Pour un slot de temps et un *offset* de canal donnés, le numéro de canal est calculé selon la formule suivant : $Numéro\ de\ canal = \frac{offset\ de\ canal + ASN}{nombre\ de\ canaux}$

Ce numéro de canal est utilisé comme indice dans la liste des canaux actifs pour obtenir le numéro de canal à utiliser. L'ASN [4] *(Absolute Slot Number)* est initialisé à zéro, il s'incrémente ensuite de 1 après chaque slot de temps et est partagé par tous les nœuds du réseau. Puisque l'ASN augmente, le même *offset* de canal peut produire différents canaux dans différents slots, ceci assure une diversité de canaux et améliore la fiabilité.

Couche réseau de *WirelessHART* :

La couche réseau assure plusieurs fonctionnalités dont les plus importantes sont le routage et la sécurité. *WirelessHART* utilise deux types de routage : le routage en utilisant les graphes et le routage à la source. Dans le routage qui utilise les graphes, le gestionnaire du réseau détermine les différents chemins qui forment les graphes et attribue un identifiant à chaque graphe. Ces chemins sont enregistrés dans la table de routage de chaque nœud et serviront pour l'acheminement des données dans le graphe. Notons qu'un nœud dispose de plusieurs

[4] L'ASN est le nombre total (modulo N) de slots de temps écoulés depuis l'initialisation du réseau.

chemins pour envoyer ses données, cette redondance augmente la fiabilité et diminue la latence. Le routage à la source définit une route unique entre la source et la destination. La liste des nœuds formant le chemin entre la source et la destinataire est contenue dans l'entête des paquets. De plus, la couche réseaux de *WirelessHART* offre les transmissions en *unicast*, *multicast* et *broadcast*. La sécurité de la couche réseau garantit l'intégrité et la confidentialité des données de bout-en-bout à travers le réseau sans fil.

Un autre standard qui se base sur la norme IEEE 802.15.4 et qui traite le multi-canal est ISA 100.11a.

3.3. ISA100.11a

Le standard ISA100.11a [5][42] est un standard international qui a été développé par ISA *(International Society of Automation)* [43] et approuvé en 2009. Ce standard est conçu pour les applications d'automatisation des procédés. ISA100.11a supporte trois types de topologies: topologie en étoile, topologie maillée et topologie étoile-maillée.

3.3.1. Composants du réseau ISA100.11a

La figure 2.19 montre une topologie typique d'un réseau ISA100.11a. Un réseau ISA100.11a se compose de plusieurs appareils :

- Les appareils de terrain sans fils *(WFD, Wireless Field Device)* : ces appareils sont connectés au processus, ils se divisent en deux types : ceux qui supportent le routage et ceux qui ne le supportent pas.

- Le routeur de *backbone* : ce routeur a pour rôle d'acheminer des paquets entre les WFD, ou entre un WFD et la passerelle.

- La passerelle : elle agit comme une interface entre le réseau de terrain et le réseau d'usine.

- Le gestionnaire du système : il est l'administrateur de l'ensemble du réseau. Il surveille le réseau et il est responsable de la gestion du système, la gestion des appareils, la gestion des tâches exécutées dans le réseau et la configuration des communications.

- Le gestionnaire de sécurité : il est responsable de la fourniture des services de sécurité basés sur les politiques de sécurité définies. Il effectue également la gestion des clés de sécurité et il garantit le fonctionnement sécurisé du système.

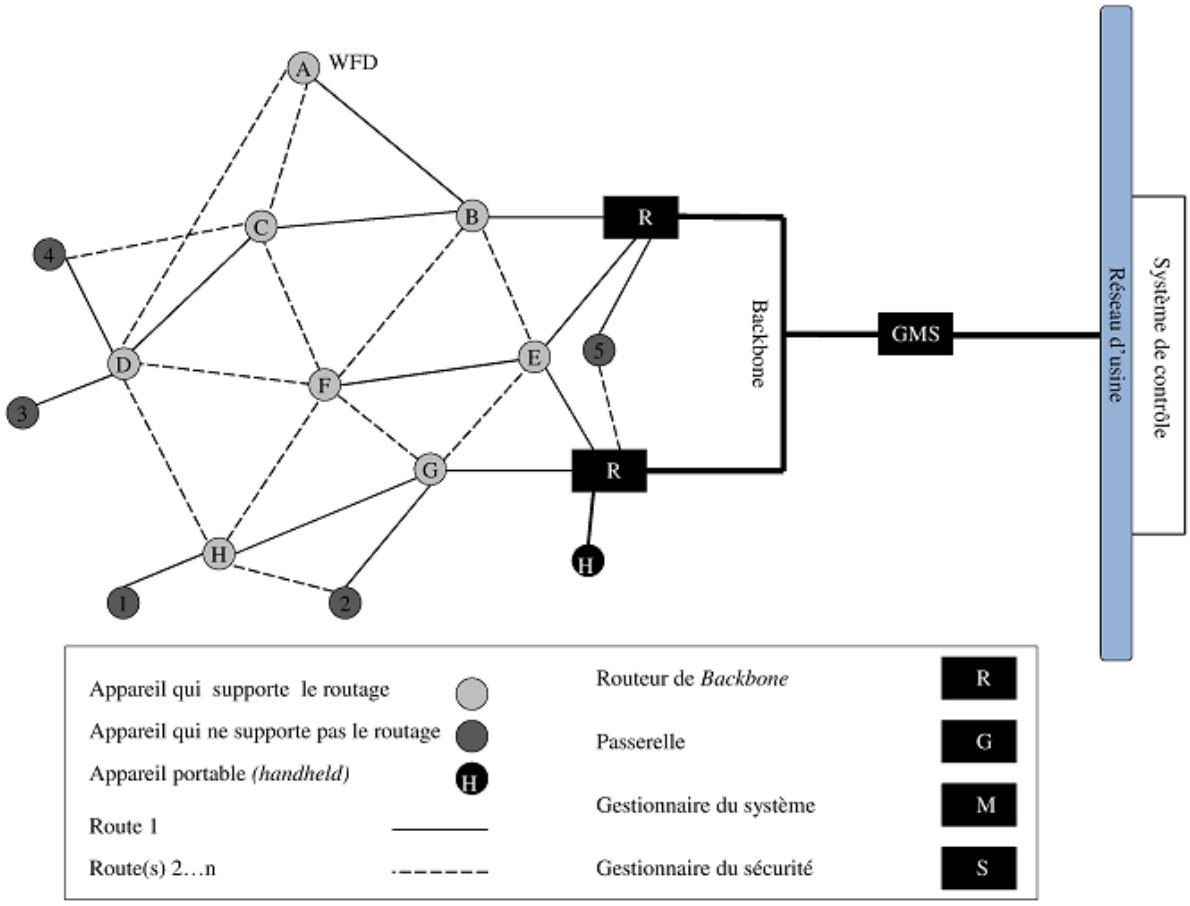


Figure 2.19 Un réseau ISA100.11a [44].

3.3.2. La pile protocolaire de ISA100.11a :

Dans cette partie, nous allons présenter la pile protocolaire définie par ISA100.11.a. Ce standard définit les couches : physique, liaison de données, réseau, transport et application. Nous décrivons ici uniquement la couche physique, la couche liaison de données et la couche réseau.

La couche physique de ISA100.11a :

D'une manière similaire à *WirelessHART*, ISA100.11a se base sur la couche physique de la norme IEEE 802.15.4 et fonctionne sur l'ensemble des 16 canaux définis dans la bande 2.4GHz. Ceci augmente la robustesse contre les interférences.

La couche liaison de données de ISA100.11a :

ISA100.11a adopte la couche MAC de la norme IEEE 802.15.4. Ce standard utilise un ordonnancement temporel (TDMA) avec des super-trames similaires à celles définies dans le standard *WirelessHART* mais avec des slots de temps de taille variable et configurable. La principale différence avec le standard *WirelessHART* est que ISA100.11a effectue le routage (à l'intérieur du réseau ou le *forwarding* des trames) dans la couche liaison de données. Ce standard supporte le routage en utilisant les graphes et le routage à la source.

Le standard ISA100.11a propose trois méthodes de saut de canal :

Channel-hopping slots : c'est une méthode déterministe basée sur TDMA dont chaque intervalle de temps utilise un nouveau canal de communication. Comme montré sur la figure 2.20, le motif TDMA se répète plusieurs fois.

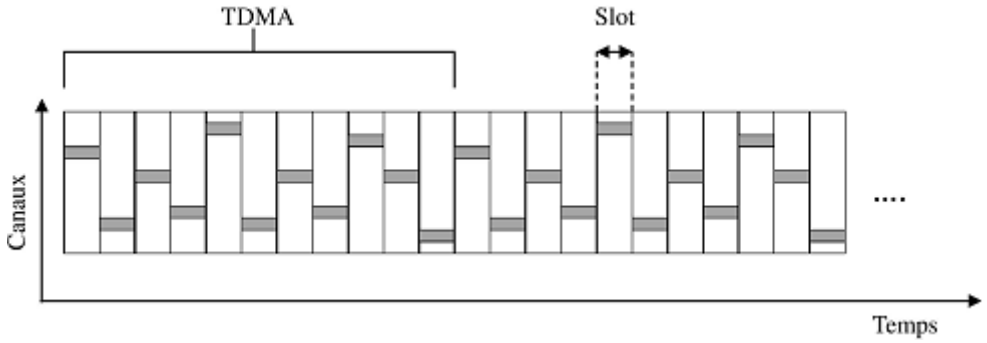


Figure 2.20 Exemple de sauts de canal slotté.

Slow hopping : dans cette méthode, CSMA/CA est utilisé. Ce mécanisme regroupe plusieurs intervalles de temps en utilisant le même canal avant d'être changé (le saut de canal est lent). Chaque groupe de slots de temps est partagé entre plusieurs appareils.

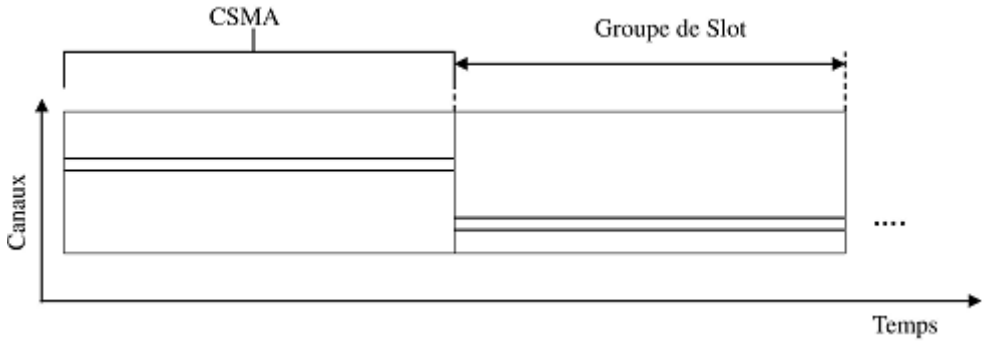


Figure 2.21 Exemple de sauts de canal lent.

Hybride hopping : c'est une combinaison des deux méthodes citées ci-dessus, c'est-à-dire des slots de TDMA suivis par des slots de CSMA. Les sauts de canal sont indépendants les uns des autres.

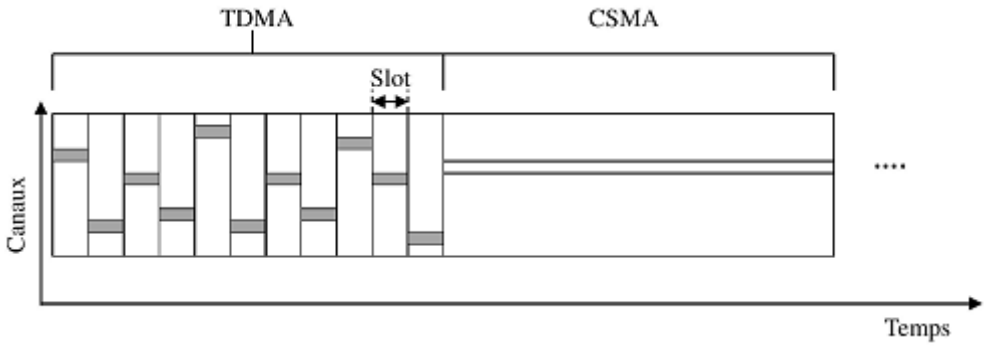


Figure 2.22 Exemple de sauts de canal hybride.

La couche réseau de ISA100.11a :

Dans ISA100, la couche réseau du modèle OSI traditionnel est remplacée par 6LoWPAN *(IPv6 Low power Wireless Personal Area Networks)* [45]. 6LoWPAN permet l'utilisation des paquets IPv6 dans un réseau 802.15.4. Dans ISA100, la couche réseau est responsable de la fragmentation, du réassemblage des paquets et de la compression de header car les paquets IPv6 ont une grande taille par rapport à la taille des trames de la norme IEEE 802.15.4. Cette couche gère aussi le routage entre les réseaux (routage de maille à maille).

3.4. Le standard 802.15.4e

IEEE 802.15.4e [46] améliore la couche MAC de la norme IEEE 802.15.4 existant sans nécessiter un changement de matériel. Le but de cette amélioration est d'ajouter des fonctionnalités à la couche MAC afin de mieux s'adapter aux besoins des communications industrielles.

La norme IEEE 802.15.4e comporte les options suivantes :

- DSME *(Deterministic and Synchronous Multi-channel Extension)* : c'est une extension multi-canal de la super-trame utilisée dans la norme IEEE 802.15.4.

- TSCH *(Time Slotted Channel Hopping)* : ce mode se base sur une technique de saut de canal dans un réseau multi-saut.
- LLDN *(Low Latency Deterministic Network)* : cette fonctionnalité est dédiée à la topologie de réseau en étoile, dans laquelle il existe des capteurs et des actionneurs pour observer et contrôler un système.

Dans cette partie, nous décrivons un aperçu de ces options.

3.4.1. *Deterministic and Synchronous Multi-channel Extension*

Le standard 802.15.4 offre sept slots de temps garantis (GTS), et ces GTS sont limités par l'utilisation d'un seul canal. Le DSME augmente le nombre de slots GTS et également le nombre de canaux utilisés. En effet, le mode DSME adopte une structure multi-super-trame qui est une extension de la super-trame de 802.15.4. La multi-super-trame est définie par le coordinateur du réseau et comprend plusieurs super-trames. Elle commence par la transmission des *beacons* suivis par une période de contention (CAP) puis une autre sans contention (CFP). La figure 2.23 montre la structure de la multi-super-trame avec l'utilisation d'un seul canal dans la partie (a) mais exploitant la diversité de canaux dans la partie (b). La multi-super-trame est définie par BO *(Beacon Order)*, SO *(Superframe Order)* et MO *(Multiframe Order)*. La durée de la multi-super-trame est calculée selon la formule suivante:

$$MD = aBaseSuperframeDuration * 2^{MO} \quad avec \ 0 \leq SO \leq MO \leq 14$$

Le nombre de super-trames dans une multi-super-trame est égale à 2^{MO-BO}.

Pour surmonter la vulnérabilité des communications sans fils aux interférences, IEEE 802.15.4e offre deux mécanismes de diversité de canal : l'adaptation de canal et le saut de canal.

- Le mécanisme d'adaptation de canal : lorsque la qualité du signal reçu devient inférieure à une valeur de seuil donnée, ce mécanisme change le canal utilisé. Dans le cas contraire, la communication se poursuit sur le même canal.
- Le mécanisme de saut de canal : dans chaque slot de temps, l'entité commute son canal sur un canal différent selon une liste prédéfinie des canaux.

Selon la norme 802.15.4e, un nœud doit utiliser le canal du nœud récepteur afin de transmettre une trame de données. Si le récepteur reçoit la trame, il doit envoyer un acquittement (ACK) au nœud émetteur sur le même canal.

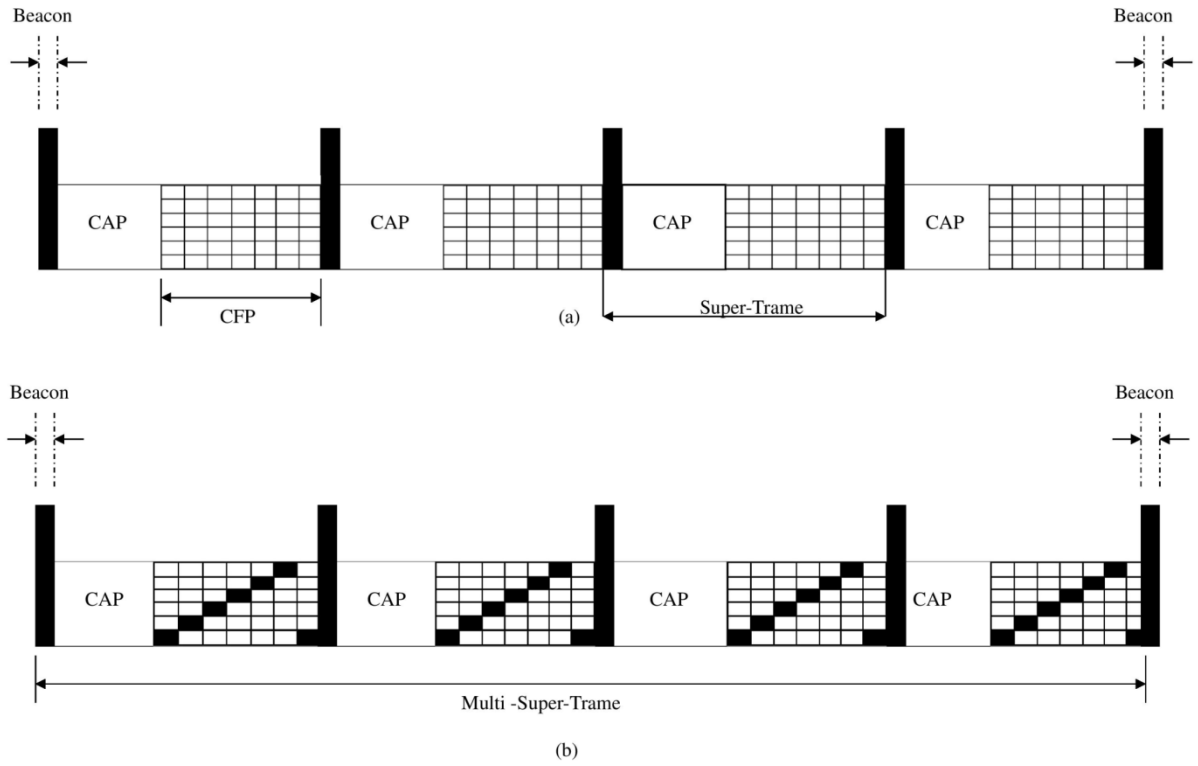


Figure 2.23 Exemple de la structure de multi-super-trame dans le mode DSME.

3.4.2. *Time Slotted Channel Hopping*

Dans le but d'offrir une robustesse dans les réseaux sans fils, le mode TSCH utilise la communication synchronisée et le saut de canal. Dans le mode TSCH, la super-trame est remplacée par une slot-trame *(slotframe)*. Une slot-trame est un groupe de slots qui se répète dans le temps**.** La figure 2.24 illustre la structure d'une slot-trame composée de trois slots de temps. Dans un slot de temps, une paire de nœuds peut échanger une trame et un éventuel acquittement. Le saut de canal permet l'utilisation d'un slot de temps par plusieurs liens simultanément. Ceci augmente la capacité du réseau.

Un lien en mode TSCH peut être représenté par un slot de temps et un *offset* de canal. Lien = (Slot de temps, *offset* de canal). Le canal utilisé par un lien est donné selon la formule suivante :

$$Canal = F[(ASN + offset\ de\ canal)\ \%\ nombre\ des\ canaux\ disponibles]$$

Où *F* est une fonction qui utilise la table des canaux disponibles dans le réseau.

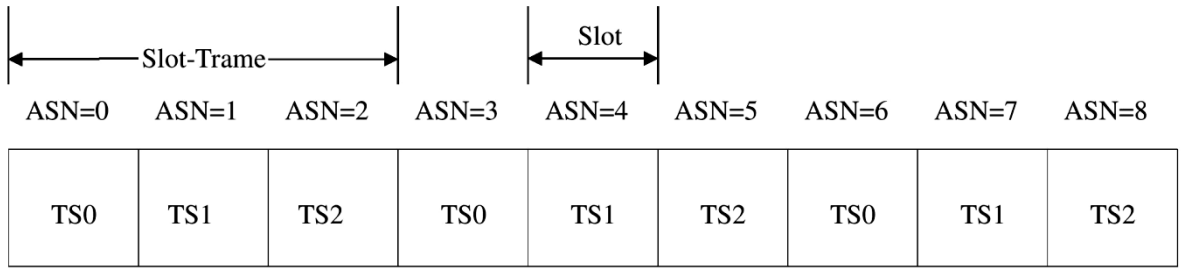


Figure 2.24 La structure de slot-trame.

3.4.3. *Low Latency Deterministic network*

Le LLDN améliore le mode suivi de *beacon* pour mieux servir les applications d'automatisation industrielle. Ces applications nécessitent une grande fiabilité et une faible latence (inférieure à 10ms). Dans ce mode, le réseau est composé de capteurs et d'actionneurs organisés dans une topologie en étoile autour d'un coordinateur unique appelée passerelle. La communication entre les capteurs et la passerelle est unidirectionnelle ascendante alors que la communication entre les actionneurs et la passerelle est bidirectionnelle. Afin de réduire la durée de transmission, un nouveau format de trame est introduit avec une réduction d'*overhead*. Comme le montre la figure 2.25, une super-trame LLDN est composée d'un intervalle de temps de *beacon*, de deux intervalles de temps pour la gestion, d'un nombre de slots de temps de taille égale pour les capteurs, d'un groupe d'accusés de réception (GACK)

et d'un nombre de slots de temps de taille égale pour les actionneurs. Ces deux intervalles de temps cités ci-dessus sont dédiés pour la gestion de communications, ascendantes et descendantes. Le groupe d'accusé de réception (GACK) contient un bitmap qui contrôle les transmissions des capteurs en fonction de leur ordre de transmission. Ce GACK est utilisé également pour favoriser les retransmissions (les transmissions qui ont échoué) dans les intervalles de temps des communications ascendantes.

Dans le réseau de LLDN, il existe deux types d'intervalle de temps : les intervalles de temps dédiés et les intervalles de temps partagés par plusieurs nœuds. La direction de la communication entre les actionneurs et la passerelle est indiquée dans le *beacon*. Notons bien que si la direction est descendante, la passerelle transmet ses paquets sans l'utilisation du CSMA/CA slotté.

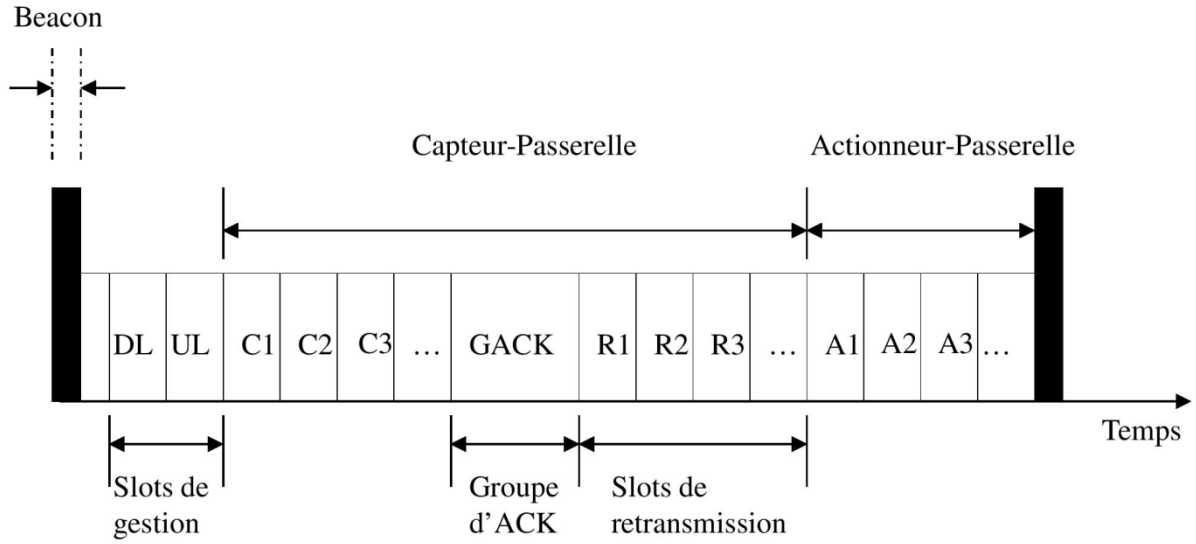


Figure 2.25 La structure de la super-trame LLDN de la norme IEEE 802.15.4e.

4. Conclusion

Dans ce chapitre, nous avons étudié les principes et les caractéristiques de l'utilisation du multi-canal sur lesquels nous allons nous baser dans la suite de cette thèse. Dans la première partie, nous avons détaillé la norme IEEE 802.15.4, ensuite dans la deuxième partie nous avons décrit différents protocoles MAC multi-canaux conçus pour les réseaux de capteurs sans fils et qui sont compatibles avec la norme IEEE 802.15.4. Dans la troisième partie nous avons abordé différents standards qui se basent sur la norme IEEE 802.15.4.

D'après l'étude des différents protocoles et standards existants dans la littérature, l'utilisation de plusieurs canaux améliore les performances globales du réseau, en évitant les collisions et en permettant des transmissions simultanées.

Dans le chapitre suivant, nous allons proposer un protocole MAC multi-canal conçu pour les réseaux de capteurs sans fils basé sur la norme IEEE 802.15.4/ZigBee et qui vise à améliorer les performances globales du réseau.

Chapitre 3 Contribution : Utilisation du multi-canal dans les réseaux de capteurs sans fil.

Après avoir présenté le standard IEEE 802.15.4 et analysé les principaux protocoles MAC multi-canaux proposés dans la littérature, nous allons présenter dans un premier temps une étude des conséquences de la connaissance du voisinage sur les interférences et les collisions. Cette étude est faite dans une topologie multi-saut pour nous servir à faire face aux problèmes des interférences et des collisions. Nous détaillons ensuite les motivations qui nous ont permis de fixer l'objectif de notre proposition. Enfin, nous présentons notre proposition qui répond aux problèmes de partage des canaux entre les nœuds dans les protocoles MAC multi-saut.

Dans notre proposition, nous nous intéressons à une topologie multi-saut dans laquelle la collecte de données est le type d'application visée. La collecte de données ou le *converge-cast* est un processus de collecte des informations provenant de l'ensemble des nœuds d'un réseau vers une entité spéciale appelée puits.

Un réseau multi-saut dense peut entrainer une augmentation de congestion et du nombre de répétitions afin d'acheminer les paquets vers le puits. En effet ce sera le cas si dans un réseau à haut débit, un flot d'alarmes est généré par le processus contrôlé, ou si, par l'exécution de plusieurs applications, il y a accumulation du trafic pour chacune d'entre elles. Si le protocole proposé utilise l'ordonnancement temporel (TDMA) pour envoyer ces paquets, une difficulté d'ordonnancement de communication va être rencontrée. Ceci entraine une augmentation de l'*overhead* impliquant une dégradation des performances du réseau. Si le protocole proposé utilise la technique CSMA/CA pour envoyer ces paquets, l'augmentation de la densité entraine une augmentation de la concurrence entre les nœuds impliquant une augmentation du risque de collision et du nombre de retransmissions. Ceci induit aussi une dégradation des performances. Un des critères essentiels qui caractérisent les protocoles MAC multi-canaux est le choix de la technique à utiliser pour gérer les communications entre les nœuds et allouer des canaux aux différents nœuds. Cela permet de déterminer la planification de transmissions entre les nœuds sur différents canaux.

1. Les transmissions improductives dans un RCSF :

Les collisions représentent une des principales causes de la dégradation des performances dans les réseaux de capteurs sans fil. Dans ce qui suit nous expliquons les conséquences de l'utilisation du même canal dans les voisinages jusqu'à 3-sauts. Ensuite nous abordons les principales sources de transmissions improductives : les pertes dues au problème du nœud sourd, collisions dues au problème du terminal caché et celles dues au problème des interférences externes.

1.1. Etude des conséquences de la connaissance du voisinage sur l'estimation des interférences et des collisions dues à l'activité des nœuds

La procédure d'attribution des canaux aux différents nœuds est un processus essentiel dans les protocoles MAC multi-canaux. L'attribution des canaux consiste à les distribuer aux nœuds associés au réseau afin d'optimiser localement des communications parallèles. Dans cette partie nous allons aborder l'impact de la réutilisation du même canal dans les voisinages jusqu'à 3-sauts (plus précisément, les voisinages à 1-saut, 2-sauts et 3-sauts). Dans ce qui suit, nous définissons 3 types de collisions : Données-Ack quand il s'agit d'une collision entre une trame de données et une trame d'acquittement, Données-Données quand il s'agit d'une collision entre deux trames de données, Ack-Ack quand il s'agit d'une collision entre deux trames d'acquittement. Notons que le canal utilisé pour la transmission des données est celui du nœud récepteur.

1.1.1. Impact de la réutilisation du même canal de réception dans un voisinage à 1-saut.

Dans cette partie, nous allons étudier l'effet de la réutilisation du même canal dans un voisinage à 1-saut. Pour ce faire, nous considérons systématiquement deux nœuds voisins à 1-saut exploitant le même canal de réception C0. Nous faisons ensuite varier le nombre de sauts séparant les émetteurs concurrents. Ainsi, nous commençons par des scenarios dans lesquels les émetteurs de paquets de données sont des voisins à 1-saut, puis à 2-sauts et enfin à 3-sauts. Dans chaque scénario nous considérons 2 récepteurs et 2 émetteurs.

Emetteurs de paquets de données voisins à 1-saut

Nous considérons que les émetteurs B et C sont des voisins à 1-saut. B envoie des données à A et C envoie des données à D. Nous supposons encore comme indiqué sur la figure 3.1 que A et D sont des récepteurs, exploitant simultanément le canal C0. Afin de compter les risques de collisions qui peuvent avoir lieu, nous prenons le scénario où tous les nœuds sont des voisins à 1-saut comme illustré sur la figure 3.1.

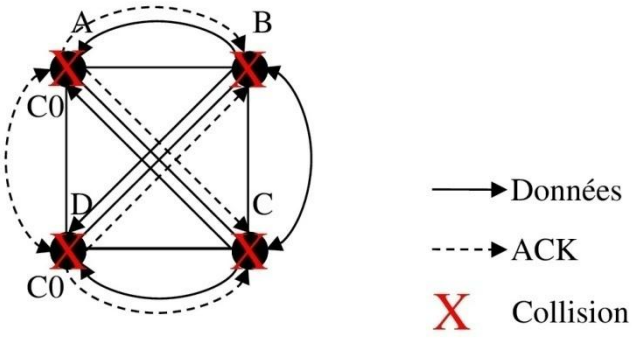


Figure 3.1 Huit risques de collision peuvent être identifiés lorsque les émetteurs de paquets de données sont des voisins à 1-saut et les récepteurs sont des voisins à 1-saut.

Les huit risques de collision sont les suivants :

1. Une collision Données-Ack peut se produire au niveau du nœud A quand le nœud B transmet une trame de données au nœud A pendant que le nœud D transmet un Ack au nœud C.

2. Une collision Données-Données peut se produire au niveau du nœud A lorsque les deux nœuds B et C transmettent en même temps des trames de données destinées aux nœuds A et D respectivement.

3. Une collision Données-Ack peut se produire au niveau du nœud B quand le nœud C transmet une trame de données au nœud D pendant que le nœud A transmet un Ack au nœud B.

4. Une collision Ack-Ack peut se produire au niveau du nœud B lorsque les deux nœuds A et D transmettent des Ack en même temps destinés aux nœuds B et C respectivement.

5. Une collision Données-Ack peut se produire au niveau du nœud C quand le nœud B transmet une trame de données au nœud A pendant que le nœud D transmet un Ack au nœud C.

6. Une collision Ack-Ack peut se produire au niveau du nœud C lorsque les deux nœuds A et D transmettent des Ack en même temps destinés aux nœuds B et C respectivement.

7. Une collision Données-Ack peut se produire au niveau du nœud D lorsque le nœud C transmet une trame de données au nœud D pendant que le nœud A transmet un Ack au nœud B.

8. Une collision Données-Données peut se produire au niveau du nœud D lorsque les deux nœuds B et C transmettent en même temps des trames de données destinées aux nœuds A et D respectivement.

Certaines de ces situations présentent une symétrie par rapport à l'axe horizontal de la figure (2 et 8 par exemple).

Emetteurs de paquets de données voisins à 2-sauts

Dans ce scenario, nous prenons en compte des émetteurs de paquets de données (B et C) voisins à 2-sauts. Ainsi, les trames de données émises, par le nœud B au nœud A et par le nœud C au nœud D, n'affectent plus les nœuds C et B respectivement. Par conséquence, le troisième et le cinquième risque de collisions (cités ci-dessus) ne peuvent plus avoir lieu. De ce fait, six types de collisions potentielles peuvent être identifiés lorsque que B et C sont des voisins à 2-sauts et A et D sont récepteurs, exploitant simultanément le canal C0.

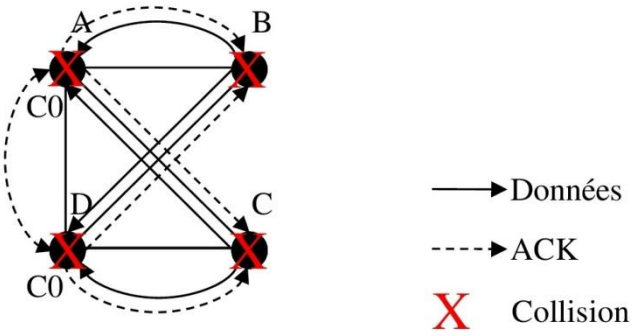


Figure 3.2 Six risques de collisions peuvent être identifiés lorsque les émetteurs de paquets de données sont des voisins à 2-sauts et les récepteurs sont des voisins à 1-saut.

Emetteurs de paquets de données voisins à 3-sauts

Dans ce scenario, nous considérons que les émetteurs de paquets de données (B et C) sont des voisins à 3-sauts. Comme le montre la figure 3.3, B est un voisin à 1-saut de A, à 2-sauts de D et à 3-sauts de C. Ainsi, B et D ne sont plus des voisins à 1-saut de même que A et C. De ce fait, Il nous reste seulement les deux risques de collision 1 et 7 énumérés ci-dessus.

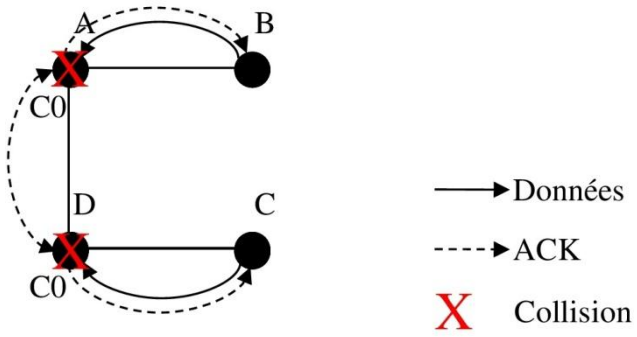


Figure 3.3 Deux risques de collision peuvent être identifiés lorsque les émetteurs de paquets de données sont des voisins à 3-sauts et les récepteurs sont des voisins à 1-saut.

1.1.2. Impact de la réutilisation du même canal de réception dans un voisinage à 2-sauts.

Nous considérons dans cette partie que le même canal C0 est utilisé dans un voisinage à 2-sauts. Nous allons aussi analyser 3 cas de figure avec des émetteurs voisins à 1-saut, 2-sauts et 3-sauts.

Emetteurs de paquets de données voisins à 1-saut

Nous prenons le scénario où B et D sont des voisins à 1-saut, C et A sont voisins à 2-sauts. Nous supposons, comme indiqué sur la figure 3.4, que A et C sont récepteurs, qui partagent simultanément le canal C0. Ce scenario nous permet de compter les risques de collisions qui peuvent avoir lieu lorsque les émetteurs de paquets de données sont des voisins à 1-saut et les récepteurs sont des voisins à 2-sauts. Ainsi, six risques de collisions peuvent avoir lieu comme le montre la figure 3.4. Ces risques sont les suivants :

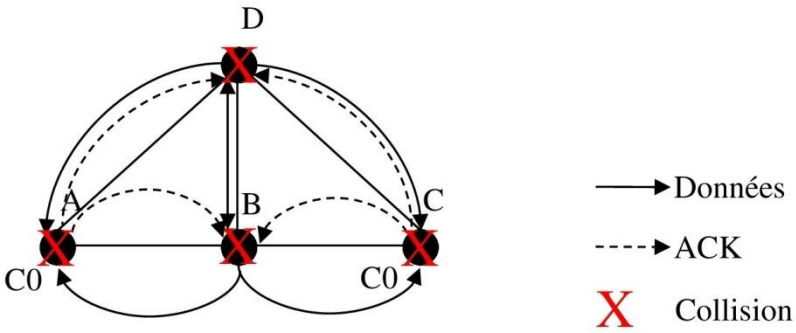


Figure 3.4 Six risques de collisions peuvent être identifiés lorsque les émetteurs de paquets de données sont des voisins à 1-saut et les récepteurs sont des voisins à 2-sauts.

1. Une collision Données-Données peut se produire au niveau du nœud A lorsque les deux nœuds B et D transmettent des trames de données en même temps destinées aux nœuds C et A respectivement.

2. Une collision Données-Ack peut se produire au niveau du nœud B quand le nœud D transmet une trame de données au nœud A pendant que le nœud C transmet un Ack au nœud B.

3. Une collision Ack-Ack peut se produire au niveau du nœud B lorsque les deux nœuds A et C transmettent des Ack en même temps destinés aux nœuds D et B respectivement.

4. Une collision Données-Données peut se produire au niveau du nœud C lorsque les deux nœuds B et D transmettent des trames de données en même temps destinées aux nœuds C et A respectivement.

5. Une collision Ack-Ack peut se produire au niveau du nœud D lorsque les deux nœuds A et C transmettent des Ack en même temps destinés aux nœuds D et B respectivement.

6. Une collision Données-Ack peut se produire au niveau du nœud D quand le nœud B transmet une trame de données au nœud C pendant que le nœud A transmet un Ack au nœud D.

Certaines de ces situations présentent une symétrie par rapport à l'axe vertical de la figure.

Emetteurs de paquets de données voisins à 2-sauts

Dans ce scenario, nous considérons que les émetteurs de paquets de données (B et D) sont des voisins à 2-sauts (voir figure 3.5). Ainsi, les trames de données émises par le nœud D au nœud A et par le nœud B au nœud C n'affectent plus les nœuds B et D respectivement. Par conséquence, le deuxième et le sixième risque de collision cités précédemment ne sont plus pris en compte. De ce fait, quatre risques de collisions peuvent être identifiés dans ce type de scenario.

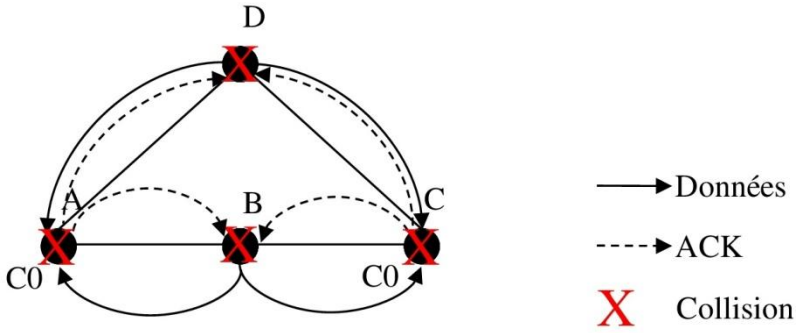


Figure 3.5 Quatre risques de collisions peuvent être identifiés lorsque les émetteurs de paquets de données sont des voisins à 2-sauts et les récepteurs sont des voisins à 2-sauts.

Emetteurs de paquets de données voisins à 3-sauts

Nous considérons que les émetteurs de paquets de données (E et D) sont des voisins à 3-sauts. Figure 3.6 montre que E est un voisin commun à 1-saut de A et B, à 2-sauts de C et à 3-sauts de D. Dans ce cas de figure, il n'existe aucun risque de collisions. E et A, étant tous les 2 à 2-sauts de C, n'affectent pas les réceptions au niveau de C, et inversement, les émissions de C n'atteignent pas les nœuds A et E. D, étant au moins à 2-sauts de A et à 3-sauts de E, n'affectent pas les réceptions au niveau de A, et inversement D n'est pas perturbé ni par A ni par E.

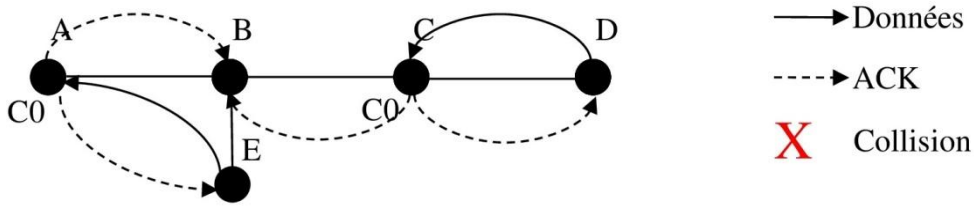


Figure 3.6 Aucun risque de collision lorsque les émetteurs de paquets de données sont des voisins à 3-sauts et les récepteurs sont des voisins à 2-sauts.

1.1.3. Impact de la réutilisation du même canal de réception dans un voisinage à 3-sauts.

Nous considérons dans cette partie que le même canal est utilisé par 2 voisins à 3-sauts.

Emetteurs de paquets de données voisins à 1-saut

A est un voisin à 1-saut de B, à 2-sauts de C et à 3-sauts de D. Ce sont maintenant les nœuds A et D voisins à 3-sauts qui partagent le même canal C0. Ainsi, deux risques de collision Données-Ack peuvent avoir lieu lorsque les nœuds émetteurs sont des voisins à 1-saut.

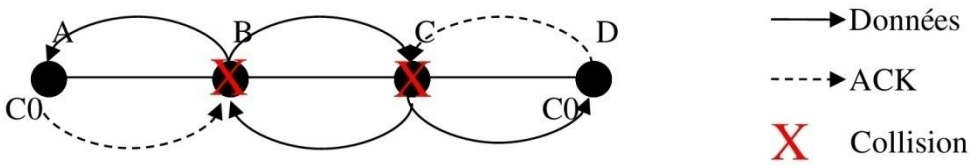


Figure 3.7 Deux risques de collision lorsque les émetteurs de paquets de données sont des voisins à 1-saut et les récepteurs sont des voisins à 3-sauts.

1. Une collision Données-Ack peut se produire au niveau du nœud B lorsque le nœud C transmet une trame de données au nœud D pendant que le nœud A transmet un Ack au nœud B.

2. Une collision Données-Ack peut se produire au niveau du nœud C lorsque le nœud B transmet une trame de données au nœud A pendant que le nœud D transmet un Ack au nœud C.

Emetteurs de paquets de données voisins à 2-sauts

Nous considérons que les émetteurs de paquets de données (E et C) sont des voisins à 2-sauts. Ainsi, les paquets peuvent être transmis en parallèles sans risque de collision. En effet, les réceptions de données et d'Ack ne sont pas affectées par les transmissions qui peuvent avoir lieu.

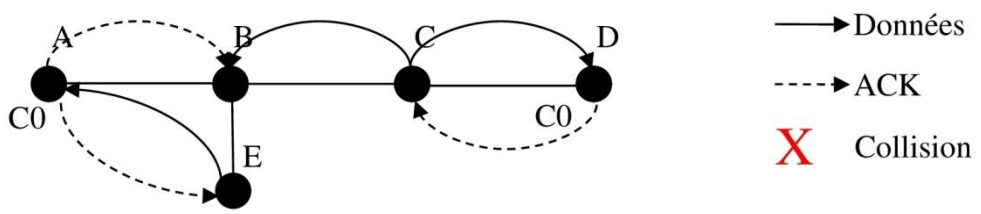


Figure 3.8 Aucun risque de collision lorsque les émetteurs de paquets de données sont des voisins à 2-sauts et les récepteurs sont des voisins à 3-sauts

Emetteurs de paquets de données voisins à 3-sauts

Dans ce scenario, nous considérons que les émetteurs (E et C) sont des voisins à 3-sauts. Comme nous pouvons le voir sur la figure 3.9, il n'existe aucun risque de collision du fait de l'éloignement des émetteurs et des récepteurs.

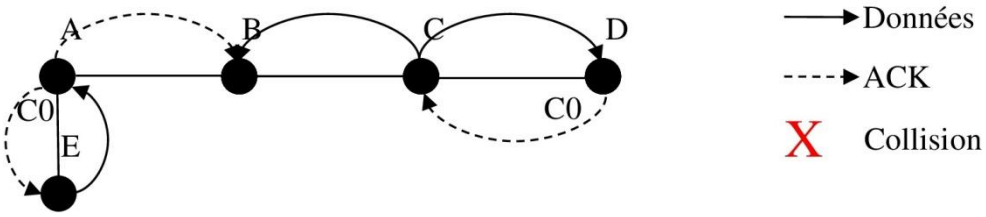


Figure 3.8 Aucun risque de collision lorsque les émetteurs de paquets de données sont des voisins à 3-sauts et les récepteurs sont des voisins à 3-sauts.

1.1.4. Synthèse

D'après cette étude, nous pouvons conclure que les collisions générées par la réutilisation du même canal par des voisins à 1-saut, 2-sauts et 3-sauts ne sont pas similaires. Ces trois cas montrent que le risque de collision lorsque le même canal est utilisé dans le voisinage à 1-saut est plus important que celui à 2-sauts. Similairement, le risque de collision lorsque le même canal est utilisé dans le voisinage à 2-sauts est plus important que celui à 3-sauts. Ce point sera développé dans le chapitre 4 et notre méthode d'allocation de canaux intégrera cette conclusion dans le choix des canaux.

Le tableau 3.1 récapitule les collisions générées par la réutilisation du même canal par des voisins à 1-saut, 2-sauts et 3-sauts en faisant varier le nombre de sauts séparant les émetteurs de paquets de données.

	Même canal C0 utilisé par des voisins à 1-saut	Même canal C0 utilisé par des voisins à 2-sauts	Même canal C0 utilisé par des voisins à 3-sauts
Emetteurs voisins à 1-saut	8	6	2
Emetteurs voisins à 2-sauts	6	4	Aucun risque de collision
Emetteurs voisins à 3-sauts	2	Aucun risque de collision	Aucun risque de collision
Totale	16	10	2

Tableau 3.1 Impact de la réutilisation du même canal dans le voisinage à 1-saut, 2-sauts et 3-sauts.

1.2. Nœud sourd

Le problème de nœud sourd se produit lorsqu'un nœud essaie de communiquer sur un canal donné avec un autre nœud qui est occupé par un échange de données sur un canal différent. Considérons le scenario de la figure 3.9 basé sur l'existence de 3 nœuds : A, B et C. Le nœud B envoie une trame de données au nœud C sur le canal C2 pendant que le nœud A essaie de communiquer avec le nœud B sur le canal C1. Dans ce cas, plusieurs retransmissions de la trame provenant du nœud A surchargent inutilement le canal jusqu'à ce que le nœud A épuise son crédit de répétitions ou bien jusqu'à ce que le nœud B soit sur le bon canal.

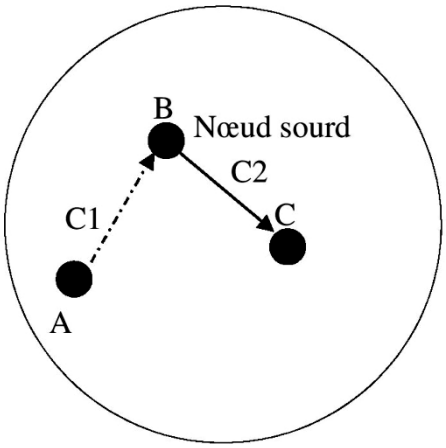


Figure 3.9 Exemple d'un nœud sourd.

1.3. Terminal caché

Ce problème se produit lorsque plusieurs nœuds qui ne sont pas à portée les uns des autres transmettent des trames de données vers le même récepteur. Considérons le scenario de la figure 3.10. Le nœud C n'est pas à portée du nœud A et donc ne peut pas détecter l'activité de A. Lorsque le nœud C teste le canal afin de transmettre une trame au nœud B, il détecte que le medium est libre pendant que A est en train de communiquer avec B. Par conséquence, une collision se produit au niveau du nœud B.

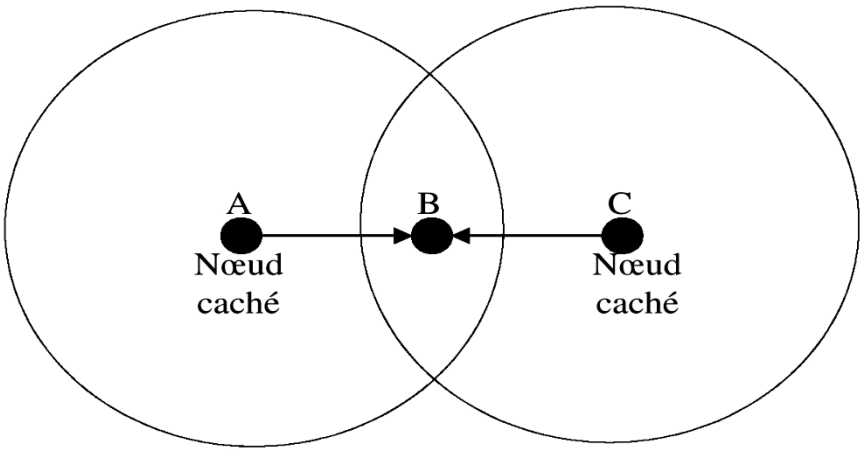


Figure 3.10 Exemple de terminal caché.

1.4. Terminal caché multi-canal :

Dans le cas du réseau multi-canal le problème du terminal caché prend une dimension supplémentaire. Ce problème est lié à l'utilisation du mécanisme RTS/CTS pour la réservation des canaux. Il se produit lorsque les paquets de contrôle (RTS/CTS) envoyés sur le canal de contrôle[5] ne peuvent pas être reçus par des nœuds voisins communiquant sur des canaux différents. Considérons le scenario de la figure 3.11 pour lequel le RTS est utilisé pour porter la requête (réservation de C1). Dans ce scenario, nous avons 4 nœuds nommés respectivement A, B, C et D. Nous possédons également 3 canaux. Les nœuds A et B échangent des messages de contrôle (RTS/CTS) sur le canal C0 afin de sélectionner un canal pour l'échange de leurs données. Supposons que C1 est le canal sélectionné. En même temps, les nœuds C et D sont occupés par un échange de données sur le canal C2, ce qui empêche C de recevoir la trame CTS envoyé par B. Ensuite C veut échanger des données avec D. Supposons qu'ils choisissent le canal C1. Ainsi C1 est utilisé simultanément par les échanges des données entre A et B et entre C et D, ce qui provoque un risque de collisions au niveau du nœud B.

[5] Un canal de contrôle est un canal réservé pour l'échange des messages de contrôle.

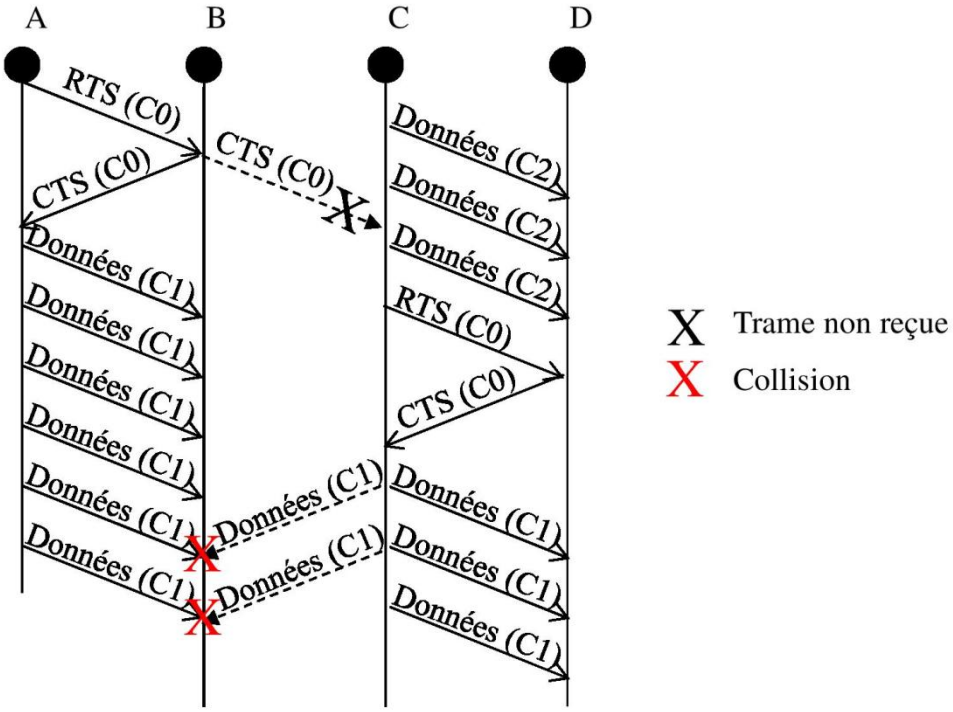


Figure 3.11 Exemple de terminal caché multi-canal.

1.5. Interférences externes :

Les réseaux de capteur sans fils utilisent la bande 2.4 GHz qui est partagée avec d'autres technologies et systèmes comme le WI-FI (IEEE 802.11), Bluetooth (IEEE 802.15.1) et les dispositifs exploitant les micro-ondes notamment car la liste est courte. Cette coexistence est souvent une source d'interférences qui provoque une dégradation des performances. L'utilisation d'une liste noire[6] dans les protocoles multi-canaux permet de réduire ce type d'interférences, appelé interférences entre les réseaux ou interférences externes, en évitant d'utiliser les canaux perturbés pour mieux résister aux interférences des technologies co-existantes localement. La figure 3.12 montre un exemple de canaux utilisés par IEEE 802.11 qui utilisent des parties de la bande ISM 2,4 GHz aussi occupées par certains canaux définis pour la couche physique de la norme IEEE 802.15.4. Par exemple, le canal 1 de wifi recouvre complètement les canaux 11, 12, 13 et 14 de IEEE 802.15.4.

[6] Liste noire : liste contenant les canaux perturbés par les interférences externes.

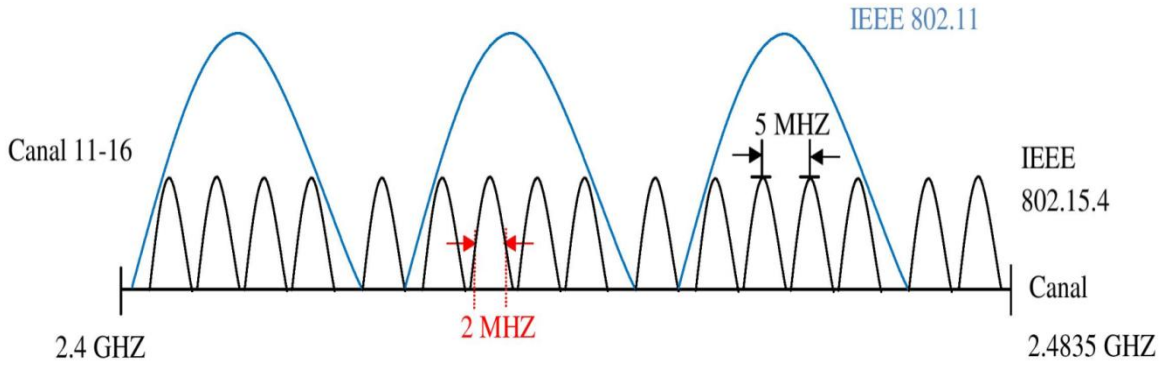


Figure 3.12 Exemple de recouvrement de canaux utilisés par IEEE 802.11 avec ceux définis par l'IEEE 802.15.4.

1.6. Conclusion :

L'augmentation du nombre de collisions provoque une augmentation du nombre de retransmissions et de paquets perdus ce qui dégrade les performances du réseau. Pour cela dans notre protocole, nous proposons de minimiser le plus possible les effets des interférences et des collisions.

2. Problématiques et motivations

Dans le chapitre 2, nous avons présenté différents protocoles MAC multi-canaux proposés dans la littérature qui ont pour but d'augmenter le nombre de transmissions locales en parallèle sur différents canaux.

Dans le but de transmettre les messages de données, l'accès au medium est réalisé selon la technique TDMA ou CSMA. Les protocoles MAC multi-canaux qui sont basés sur TDMA exigent une stricte synchronisation entre les nœuds et une stricte coordination de la commutation du canal entre les émetteurs et les récepteurs afin d'être sur le même canal au même moment. Ceci nécessite aussi d'importants échanges de messages de contrôle entre les nœuds afin d'établir une allocation adéquate de slot/canal. Cette allocation est souvent basée sur la découverte de voisinage et sur le taux de transmission de données de chaque nœud.

Ce type de protocoles permet des transmissions parallèles de données sans collision et limite l'*overhearing*. Cependant, la nécessité d'une stricte synchronisation entre les nœuds limite l'utilisation de ce type de protocoles et le rend rigide, non-évolutif et non-adapté aux

changements de topologies. En effet, à chaque fois qu'un nouveau nœud va s'associer au réseau ou qu'un nœud est désassocié ou lors du changement du volume de données à transmettre par chaque nœud, le processus d'attribution de slots de temps et de canaux va être réactivé dans le but de prendre en compte ces nouveaux changements.

En revanche, les protocoles qui utilisent la technique CSMA/CA sont plus flexibles et plus adaptés aux changements du réseau. Cependant, l'utilisation de cette technique provoque plusieurs problèmes qui ne sont pas résolus dans la littérature. Ces problèmes sont les suivants :

- la surdité des nœuds dans la plupart des protocoles utilisant la technique CSMA/CA, ce qui induit un grand nombre de transmissions infructueuses et de retransmissions de paquets.
- l'absence de gestion des collisions entre les nœuds associés au même père augmente considérablement la probabilité de collisions et de retransmissions.
- l'augmentation de la densité des nœuds et des paquets générés dans un réseau multi-saut augmente la contention et ainsi le risque de collisions. En conséquence, une réduction de l'effet de cette densité doit être effectuée.

3. HMC-MAC *(Hybrid Multi-Channel MAC Protocol)*

Dans cette partie, nous proposons HMC-MAC, un protocole MAC multi-canal hybride répondant aux exigences des réseaux de capteurs sans fil à haut débit. Ce protocole vise à minimiser les effets des interférences et des collisions et à augmenter le débit de réseau. HMC-MAC découpe le temps en cycles globaux, chaque cycle étant divisé en plusieurs périodes afin d'améliorer l'utilisation du médium.

HMC-MAC propage des messages de contrôle, appelé *beacons*, en multi-saut afin de maintenir la synchronisation entre les nœuds. Ces messages contiennent des informations dont les nœuds ont besoin pour connaitre les intervalles d'envoi, d'écoute et de mise en veille. Nous proposons d'allouer un canal de contrôle qui est utilisé pour que les nœuds diffusent leurs *beacons*, ainsi il suffira aux nouveaux nœuds qui souhaitent rejoindre le réseau de simplement écouter sur ce canal de contrôle. Afin d'éliminer les collisions des messages de *beacon*, HMC-MAC utilise la technique TDMA pour l'envoi de ces derniers [47].

Afin de maintenir un réseau évolutif et d'éviter les difficultés d'ordonnancement de communication qui entraine une surcharge de messages de contrôle lors de la planification des communications entre les nœuds, HMC-MAC exploite la flexibilité offerte par la méthode d'accès CSMA/CA. Plusieurs techniques sont mises en place dans HMC-MAC afin de réduire les risques de collisions induits par l'utilisation de CSMA/CA.

Dans le but d'améliorer les performances globales du réseau, HMC-MAC exploite des transmissions parallèles sur plusieurs canaux. Nous utilisons l'ensemble des canaux autorisés par la norme IEEE 802.15.4 pour construire un réseau qui fonctionne sur plusieurs canaux.

Notre objectif est de réduire au maximum les problèmes conduisant à des transmissions improductives déjà expliqués dans la partie 3.1, pour ce faire HMC-MAC utilise une nouvelle méthode d'allocation de canaux qui exploite la réutilisation spatiale des canaux et qui permet aux nœuds de choisir le canal le plus convenable dans leurs voisinages jusqu'à 3-sauts.

Comme déjà indiqué, dans cette thèse nous nous intéressons aux applications de type *converge-cast* dans une topologie multi-saut (voir figure 3.13) dans laquelle les nœuds du réseau génèrent un volume conséquent de données à destination d'un puits. Dans le but d'augmenter le débit de réseau et de diminuer le risque de saturation de ce réseau, nous nous basons dans notre domaine d'étude sur une topologie caractérisée par la présence de puits multi-interface.

Pour contourner le problème de nœud sourd et pour réduire les effets de l'augmentation de la densité de réseau comme expliqué dans la partie 3.1, HMC-MAC fragmente le réseau en deux groupes en définissant les intervalles d'émission et d'écoute de chaque groupe.

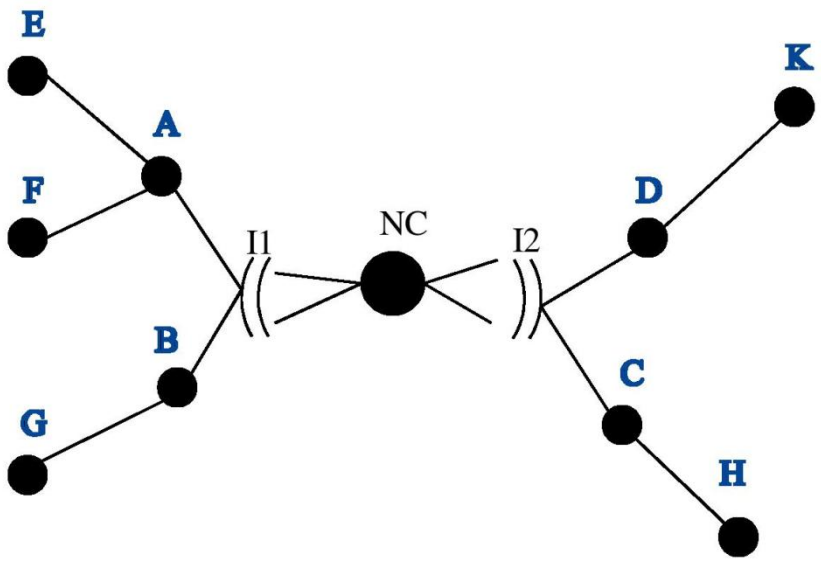


Figure 3.13 Exemple d'une topologie multi-saut avec un puits multi-interface.

Dans ce qui suit, nous allons expliquer les phases de démarrage du réseau avant que les nœuds commencent à échanger des messages de données.

Dans HMC-MAC, la configuration du réseau est réalisée en deux phases : la phase de synchronisation et d'initialisation du réseau, et la phase d'attribution des canaux. Le but de la séparation en deux phases différentes est de laisser se stabiliser la connaissance de la

topologie pour permettre aux nœuds de choisir leurs canaux. La stabilité de la connaissance d'une topologie est la stabilité des tables de voisinages de tous les nœuds du réseau. Nous considérons la stabilité du voisinage d'un nœud donné lorsque ce nœud considère les mêmes voisins à 1-saut, 2-sauts et 3-sauts pour *n* cycles globaux consécutifs (n dépend de la durée de la phase de déploiement, dans notre cas nous l'avons à 5).

Durant la phase de synchronisation et d'initialisation du réseau comme le montre la figure 3.14 (cette figure se réfère à la précédente), chaque cycle global est composé de deux intervalles [T0, T1] et [T1, T2]. Durant [T0, T1], les nœuds propagent les *beacons* pour que chaque nœud puisse construire sa table de voisinage à 1-saut, 2-sauts et 3-sauts. L'intervalle [T1, T2] est dédié aux associations des nouveaux nœuds qui souhaitent joindre le réseau.

Comme nous utilisions une topologie arborescente où tous les messages vont être transmis vers le puits, chaque coordinateur envoie un message de stabilité à son parent lorsqu'il détecte une stabilité dans son voisinage jusqu'à 3-sauts pour *n* cycles globaux consécutifs, et lorsqu'il a reçu des messages de stabilité de tous ses fils. Cette phase se termine lorsque le coordinateur du réseau, appelé *Network Coordinator* (NC), reçoit des messages de stabilité de tous ses fils. Ensuite le NC lance la phase suivante. Notons que ce message de stabilité est représenté par un bit dans le *beacon*, ce bit est égal à 1 lorsqu'un nœud atteint sa stabilité et la stabilité de tous ses fils, dans le cas contraire ce bit est égal à 0. Avec cette représentation des messages de stabilité, HMC-MAC n'a pas besoin de surcharger le réseau avec des messages de contrôle.

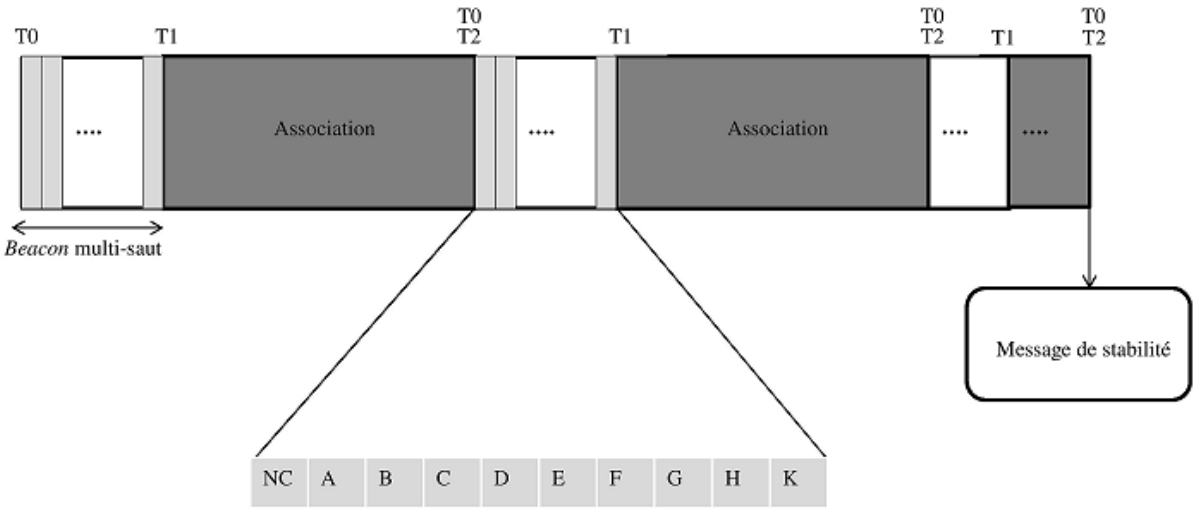


Figure 3.14 Phase de synchronisation et d'initialisation du réseau.

Nous allons maintenant donner un descriptif détaillé des différents mécanismes proposés dans ces deux phases.

3.1. Synchronisation et initialisation du réseau

Dans ce qui suit nous détaillons la phase de synchronisation et d'initialisation du réseau afin de faire la découverte des voisinages.

3.1.1. Création du réseau et choix du nœud père

Lorsqu'un nouveau nœud est activé et souhaite rejoindre le réseau, il commence par la scrutation de son environnement radio à la recherche de *beacons*. Si aucun *beacon* n'est détecté, le nœud considère qu'il est le premier nœud actif dans le réseau et donc responsable de la création d'un nouveau réseau. Ce nœud est nommé le coordinateur du réseau (NC), il possède l'adresse courte 0 et il est de profondeur 0. Ensuite le NC commence à diffuser un *beacon* périodiquement. Les nœuds qui se trouvent à portée du NC reçoivent ce *beacon* et envoient une demande d'association au NC afin de rejoindre le réseau. Une fois cette demande acceptée, le nouveau nœud commence à propager le *beacon* périodiquement afin d'annoncer sa présence pour que les autres nœuds puissent rejoindre le réseau. Par conséquence une topologie arborescente multi-saut est créée entre les différents nœuds associés au réseau en formant des liens père-fils.

Dans HMC-MAC, le nouveau nœud reçoit au moins un *beacon* de ces nœuds voisins s'il est à la portée du réseau. Le *beacon* transporte les informations sur le C_m, le R_m, le L_m (cf. chapitre 2 de ce document), l'adresse de la source, sa profondeur et son nombre de fils. Ce qui permet au nouveau nœud de construire sa table de voisinage en se basant sur les informations contenues dans les *beacons*.

Avec comme objectif de choisir son nœud père (le nœud avec lequel il souhaite s'associer), un nouveau nœud vérifie d'abord si le RSSI des nœuds voisins dépasse le seuil pour garantir des liaisons robustes entre nœuds associés. Ensuite, il cherche le nœud qui a la plus petite profondeur parmi ceux qui ont la capacité d'accepter des associations. Un nœud a la capacité d'accepter une association si son nombre de fils est inférieur à C_m. Si plusieurs nœuds possèdent la plus petite profondeur parmi ses voisins qui sont capables d'accepter une association, le nouveau nœud choisit le nœud qui a moins de fils afin de repartir la topologie. Si plusieurs nœuds possèdent la petite profondeur et le plus petit nombre de fils, il choisit aléatoirement un de ces nœuds. Notons que HMC-MAC limite le nombre de fils par coordinateur en initialisant une petite valeur de C_m afin de minimiser le nombre de collision entre les fils de même nœud père.

3.1.2. Association des nœuds

Dans certains protocoles proposés dans la littérature, lorsqu'un nœud souhaite rejoindre le réseau, il doit scanner tous les canaux disponibles dans le réseau afin de collecter tous les *beacons* transmis dans son voisinage. Ceci provoque d'une part une forte consommation énergétique et d'une autre part un délai non négligeable. Dans HMC-MAC, afin d'éviter ces problèmes par hypothèse tous les nœuds utilisent un même canal de contrôle qui peut être différent pour chaque réseau.

Durant la phase d'association, lorsqu'un nœud choisit son nœud père, il lui envoie une requête d'association pour rejoindre le réseau. Cette requête est envoyée sur le canal de contrôle dans le cas où le réseau est dans la phase de configuration. En revanche, si le réseau est dans la phase d'échange des données, le nœud souhaitant rejoindre le réseau utilise le canal alloué au nœud qu'il souhaite avoir comme nœud père et lui envoie une requête d'association.

Une fois l'association acceptée, le nœud père génère une adresse courte pour le nouveau nœud. Ensuite, il lui envoie une réponse d'association *(Association Response)* dans laquelle il inclut des paramètres nécessaires à la synchronisation.

Après chaque nouvelle association, le nœud père remonte un message d'indication au NC dans le but de l'informer de la présence de ce nouveau nœud dans le réseau. Ce changement de topologie a un impact sur la façon de propager le *beacon* dans le réseau. Le *beacon* va servir de vecteur pour informer les nœuds de ce changement. Ceci peut être réalisé de la façon suivante. Ce message d'indication inclut l'adresse du nouveau nœud, ce qui permet au NC de modifier l'ordre de propagation de *beacon*. Ensuite le NC inclut cette modification (voir champs renseignés dans la partie 3.3.2.5) dans le *beacon* durant *m* cycles consécutifs. Cela permet aux nœuds de modifier l'ordre de propagation.

Dans le cas où le coordinateur prévu comme père n'accepte pas l'association, le nouveau nœud essaie de trouver un autre coordinateur avec lequel il peut s'associer afin de rejoindre le réseau.

Un nœud est considéré comme désassocié lorsqu'il ne reçoit pas de *beacon* pour *m* cycles consécutifs (m dépend de l'évolutivité de la topologie, dans notre cas nous l'avons fixé à 3). Dans ce cas, il essaie de trouver à nouveau un coordinateur père afin de rejoindre le réseau en se mettant à l'écoute du canal de contrôle.

3.1.3. Découpage temporel

Dans HMC-MAC, le temps est divisé en cycles globaux. Chaque cycle est divisé en périodes durant lesquelles les communications sont organisées de manière spécifique. Ces périodes sont les suivantes :

- Une période de propagation du *beacon* [T0; T1] qui est une période de synchronisation durant laquelle un *beacon* multi-saut se propage entre les nœuds du réseau.

- Une période d'échange de données [T1; T2] qui est elle-même découpée en plusieurs intervalles de temps. Durant cette période, les nœuds du réseau transmettent leurs données.

- Une période d'inactivité [T2; T0] où tous les nœuds du réseau peuvent économiser de l'énergie en se mettant en mode sommeil.

En fonction des besoins de l'application, la durée de ces intervalles peut être ajustée pour supporter le trafic généré. Notons que, plus la période [T1 ; T2] est longue, plus le débit de transit de paquets à travers le réseau est élevé. Notons aussi que, plus la période d'inactivité est longue plus les nœuds économisent de l'énergie. Les différentes périodes sont représentées dans la figure 3.15.

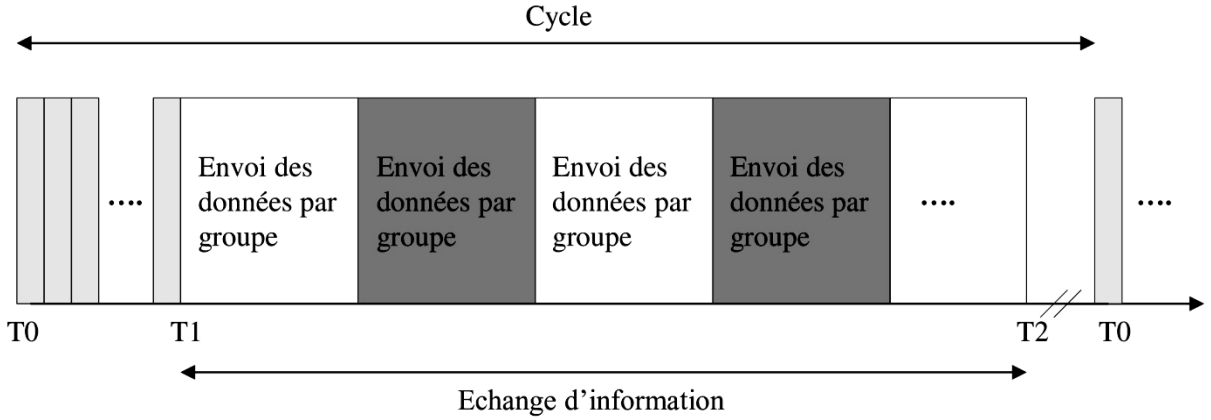


Figure 3.15 Les différentes périodes du cycle global.

3.1.4. Propagation de *beacon*

Pendant [T0; T1], NC diffuse un *beacon* de synchronisation contenant les informations permettant aux nœuds de synchroniser leur activité selon les différents intervalles du cycle. Afin d'atteindre l'ensemble des nœuds du réseau qui ne sont pas forcément à portée du NC, les coordinateurs du réseau propagent à tour de rôle (en cascade) le *beacon* diffusé par le NC. Cette propagation est basée sur le protocole MaCARI [47].

Chaque nœud du réseau dispose d'un slot de temps unique pour la propagation du *beacon*. Ce *beacon* contient l'ordre dans lequel les nœuds le propagent dans le réseau. L'ordre de propagation de *beacon* est défini par une liste des adresses des nœuds du réseau, c'est-à-dire si l'adresse d'un nœud est au quatrième rang dans la liste qui définit l'ordre de propagation, cela signifie qu'il est le quatrième nœud autorisé à envoyer son *beacon* pendant [T0; T1]. Cet ordre de propagation garantit une propagation sans collision des *beacons*. Le *beacon* est propagé en utilisant un canal de contrôle commun à tous les nœuds. De cette façon, les nouveaux nœuds souhaitant rejoindre le réseau se mettent à l'écoute uniquement sur le canal de contrôle.

3.1.5. Allocation des adresses courtes

La topologie est organisée en plusieurs profondeurs selon la topologie arborescente utilisée dans ZigBee [34]. L'allocation des adresses courtes permet de donner une adresse unique à chaque nœud dans le réseau. Le coordinateur de réseau possède toujours l'adresse 0. Les adresses courtes sont calculées selon les règles du protocole ZigBee en utilisant les paramètres R_m, L_m et C_m décrites dans la partie 3.1.1 du chapitre 2. Ces adresses courtes permettent aux nœuds d'appliquer le protocole de routage hiérarchique de ZigBee. Afin d'attribuer des adresses courtes aux interfaces de puits, nous supposons que ces interfaces sont comme des nœuds situés à la profondeur 1. Ainsi, nous supposons que les nœuds situés normalement à une profondeur P du puits auront une profondeur égale à P+1 (avec P supérieur ou égale à 1).

Reprenons notre architecture générique donnée figure 3.13 et considérons le jeu de paramètres suivant $L_m = 3$, $C_m = 2$ et $R_m = 2$, le tableau 3.1 donne les valeurs de $Cskip$ pour chaque profondeur. La figure 3.16 représente un exemple d'allocation des adresses hiérarchiques. La première interface qui est supposée être à la profondeur 1 a attribué l'adresse 2 à son premier fils A et l'adresse 5 à son deuxième fils B où $5 = 2 + Cskip(1)$.

Profondeur	$Cskip$
0	7
1 (interfaces du NC)	3
2	1
3	0

Tableau 3.2 Valeurs de $\boldsymbol{Cskip}$ pour l'exemple (3, 2, 2).

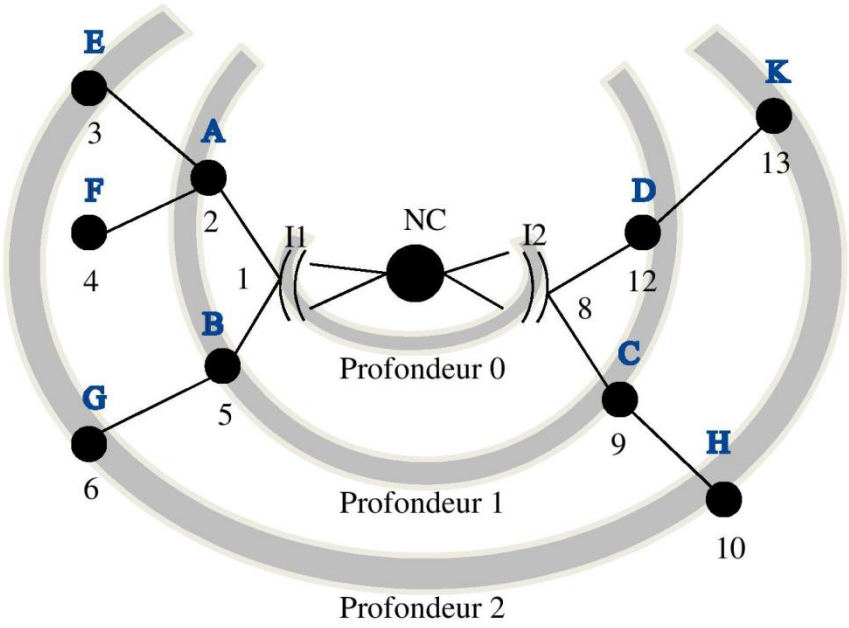


Figure 3.16 Exemple d'attribution des adresses hiérarchiques.

3.1.6. Découverte de voisinage

Afin de minimiser l'énergie consommée suite à de longues durées en écoute *(Idle Listening)* comme expliqué précédemment, nous utilisons un canal unique sur lequel nous nous basons pour notre méthode de découverte de voisinage.

Dans le but de planifier les communications entre les nœuds et pour savoir quels nœuds ont tendance à interférer et quels autres sont autorisés à transmettre des trames de données simultanément, HMC-MAC réalise une découverte de voisinage.

Afin d'éviter de surcharger le réseau avec de longs messages de contrôle pour échanger les informations de voisinage entre les nœuds, nous avons choisi de nous servir de *bitmaps* pour représenter les voisins. Pour ce faire, nous prenons comme base, l'ordre de propagation expliqué dans la section 3.4.1.3, ceci permet à chaque nœud de construire et de gérer un *bitmap* qui représente toutes les adresses des nœuds du réseau. Chaque place (indice) du *bitmap* correspond à l'adresse du nœud ayant le même rand (indice) dans la liste qui correspond à l'ordre de propagation.

D'après la figure 3.8, le même canal ne peut pas être utilisé dans un voisinage à 3-sauts afin d'éviter les interférences et les collisions lorsque les accusés de réception sont utilisés [48]. Pour cela, la découverte de voisinages jusqu'à 3-sauts est nécessaire.

Comme dit précédemment, pendant [T0, T1], chaque nœud à son tour propage un *beacon.* Ce qui permet aux nœuds voisins de construire leur table de voisinage à 1-saut en utilisant l'adresse courte du nœud émetteur. Quand un nœud reçoit un *beacon* d'un autre nœud, il le considère comme un voisin et il met à 1 le bit correspondant à l'adresse de ce nœud voisin dans sa table de voisinage à 1-saut.

Par exemple, si l'ordre de propagation est le suivant {NC, A, B, C, D, E, F, H, K} selon la topologie de la figure 3.16, la table de voisinage à 1-saut du nœud F est représentée par le *bitmap* ci-dessous. Ce qui signifie que le nœud A est un voisin à F (F entend A). De cette façon, chaque nœud est capable de représenter ses voisins car la liste de diffusion du *beacon* contient tous les nœuds du réseau.

NC	A	B	C	D	E	F	G	H	K
0	1	0	0	0	0	0	0	0	0

Tableau 3.3 *Bitmap* de voisins à 1-saut du nœud F.

Afin que les nœuds puissent construire les *bitmaps* de voisins à 2-sauts, chaque nœud inclut le « *bitmap* de son voisinage à 1-saut » dans le *beacon*. A chaque fois qu'un nœud reçoit un *beacon*, il vérifie dans ce dernier chaque bit du « *bitmap* de voisinage à 1-saut » :

- Dans le cas où ce bit est égal à 1, il met à 1 le bit correspondant au même indice dans son « *bitmap* de voisinage à 2-sauts ».

- Dans le cas où ce bit est égal à 0, il ne fait aucune modification.

Ensuite, il met à 0 le bit correspondant à son indice et les bits qui correspondent aux indices de ses voisins à 1-saut dans son « *bitmap* de voisinage à 2-sauts ».

Ceci permet à chaque nœud de construire le « *Bitmap* de ses voisins à 2-sauts » lorsqu'il reçoit tous les *bitmaps* de ses voisins via la diffusion des *beacons*.

Par exemple, la liste de voisinage à 2-sauts du nœud F est représentée par le *bitmap* ci-dessous. Ce qui signifie que NC et E sont des voisins à 2-sauts de F.

NC	A	B	C	D	E	F	G	H	K
1	0	0	0	0	1	0	0	0	0

Tableau 3.4 *Bitmap* de voisins à 2-sauts du nœud F.

D'une façon similaire à la construction des listes de voisinage à 2-sauts, les listes de voisinage à 3-sauts peuvent être construites. Chaque nœud inclut le « *bitmap* de ses voisins à 2-sauts » dans le *beacon*. Ainsi, à chaque fois qu'un nœud reçoit un *beacon*, il contrôle chaque bit du « *bitmap* de voisinage à 2-sauts » du voisin émetteur. Lorsqu'un 1 est détecté, il met à 1 le bit correspondant au même indice dans son « *bitmap* de voisinage à 3-sauts ». Ensuite, il met à 0 le bit correspondant à son indice et les bits qui correspondent aux indices de ses voisins à 2-sauts dans son « *bitmap* de voisinage à 3-sauts ». Ceci permet à chaque nœud de construire la liste de ses voisins à 3-sauts.

Par exemple, la liste de voisinage à 3-sauts du nœud F est représentée par le *bitmap* ci-dessous. Ce qui signifie que B, C et D sont des voisins à 3-sauts de F.

NC	A	B	C	D	E	F	G	H	K
0	0	1	1	1	0	0	0	0	0

Tableau 3.5 *Bitmap* de voisins à 3-sauts du nœud F.

En utilisant les *beacons* et l'ordre de propagation nous venons de montrer qu'il est possible de construire les listes de voisinage à 1-saut, 2-sauts et 3-sauts. Cette façon de faire nous évite d'utiliser des messages de contrôle supplémentaires comme les messages HELLO qui

surchargent le réseau (ici le *beacon* est vu comme un HELLO et il contient des informations de voisinage).

Notons que seuls les nouveaux nœuds qui rejoignent le réseau sont inclus comme une demande de mise à jour de l'ordre de propagation, ceci pour éviter d'envoyer la liste complète des nœuds dans le *beacon* et par conséquent de surcharger le *beacon*.

Dés que tous les nœuds ont détecté la stabilité de ses voisinages, la phase d'attribution des canaux peut avoir lieu.

3.2. Phase d'attribution des canaux

Dans cette partie, nous abordons les différentes étapes que les nœuds doivent respecter pour sélectionner leurs canaux de réception. Dans un premier temps, nous expliquons les hypothèses de travail suivantes :

- Nous supposons que chaque nœud est équipé d'une seule interface radio et le puits possède plusieurs interfaces radio afin d'augmenter le taux de réception des trames de données sur le point de collecte.

- Nous considérons que tout le trafic est destiné au NC, c'est la raison pour laquelle les échanges peuvent être séquencés en fonction de la profondeur des nœuds dans la topologie. Ceci permet d'allouer des canaux en tenant compte de la direction du trafic.

- Pour contourner le problème des interférences externes provoquées par la cohabitation avec d'autres technologies ou autres réseaux qui utilisent les mêmes bandes spectrales, nous disposons dans notre protocole de *n* canaux de communication sans interférence avec un maximum de 16 canaux comme c'est le cas dans la norme IEEE 802.15.4. Ceci permet de minimiser l'effet des interférences externes et d'offrir des meilleures performances au réseau.

- Nous considérons l'existence d'un canal unique de contrôle sur lequel le réseau est crée. Il suffira aux nouveaux nœuds qui souhaitent rejoindre le réseau de simplement écouter sur ce canal de contrôle.

3.2.1. Segmentation du réseau

Un grand nombre d'applications proposés pour les réseaux de capteurs sans fil nécessitent des débits de plus en plus importants (pour un réseau de capteurs) et une forte densité des nœuds due au besoin de déploiement d'un nombre important de nœuds dans certains domaines. Cette densité provoque des congestions et une forte contention entre les nœuds du réseau, ce qui conduit à l'apparition d'un grand nombre de collisions.

Pour diminuer l'effet des congestions, les auteurs dans [25] [17] ont proposé de faire une segmentation du réseau sur plusieurs canaux. Ainsi les réseaux ont été fragmentés en plusieurs groupes et à chaque groupe a été alloué un canal différent. Dans notre protocole, nous exploitons cette idée en faisant une segmentation du réseau en deux groupes.

D'autre part, pour éviter le problème de nœud sourd, les nœuds émetteurs doivent savoir quand les récepteurs seront prêts à recevoir des trames de données sur le canal choisi sans pour cela échanger des préambules comme dans [32].

Comme dans notre considération tout le trafic est destiné au NC, dans le but de maximiser le taux de réception au puits, nous proposons de fragmenter le réseau de sorte que deux groupes alimentent le puits en trames de données d'une façon continue (Il sera préjudiciable que les interfaces du puits soient alternativement émettrice puis réceptrice dans la mesure où elles n'ont rien à émettre dans cette phase). De cette façon, les interfaces de puits seront donc toujours réceptrices pour accueillir le flux venant des deux groupes qui ont été créés.

Pour effectuer une partition en groupes de nœuds travaillant alternativement, nous n'allons pas utiliser des messages de contrôle mais déduire l'appartenance à un groupe à partir d'une adresse logique. Afin d'alimenter le puits d'une façon continue, nous allons garder une partie des nœuds de profondeur 1 en mode émission. Pour ce faire, nous allons diviser le réseau en branches comme le montre la figure 3.17. Chaque branche comprend un fils d'une interface avec tous ses descendants et possède un indice en fonction de l'ordre d'association du fils de l'interface. D'autre part, pour séquencer les échanges de trames de données dans la topologie, nous utilisons la profondeur des nœuds. L'idée est de diviser les nœuds en deux groupes en fonction de leur profondeur et de leur indice de branche en s'arrangeant que les nœuds du premier groupe soient des émetteurs pendant que ceux du deuxième groupe soient des récepteurs et vice-versa.

Dans ce qui suit, nous allons expliquer comment chaque nœud est capable de savoir à quelle branche il appartient afin de construire les deux groupes des nœuds.

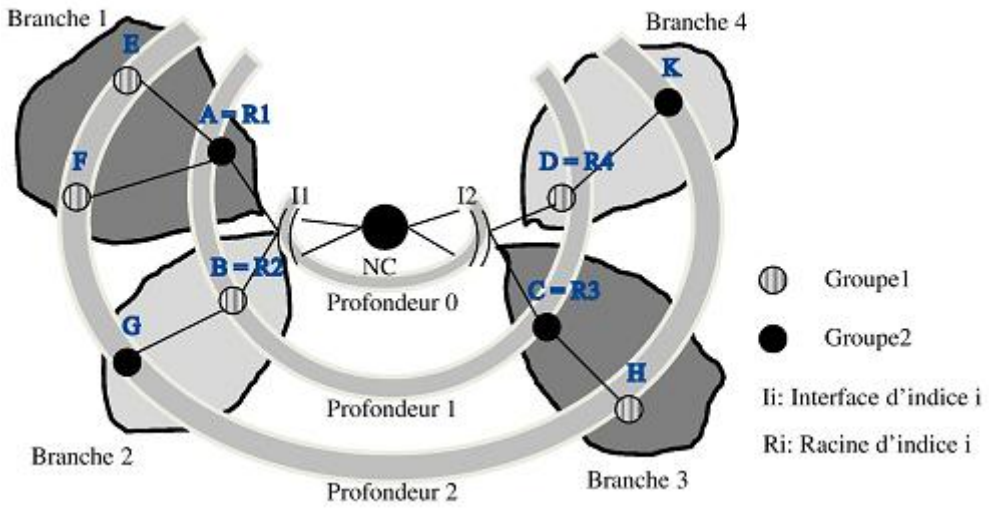


Figure 3.17 Segmentation du réseau en deux groupes.

Détection de la branche du nœud

Nous avons proposé d'identifier l'ascendant à partir de la plage d'adresses à laquelle le nœud appartient en utilisant les règles des adresses hiérarchiques du protocole ZigBee déjà expliqué dans la partie 3.1.1 du chapitre 2. Afin que chaque nœud puisse connaître la branche à laquelle il appartient, il doit en premier lieu calculer l'adresse de l'interface du NC dont il est un descendant. Ce qui revient à identifier l'interface avec la plus grande adresse parmi celles qui sont plus petites que sa propre adresse.

En second lieu, le nœud doit déduire la branche à laquelle il appartient. Ces branches sont routées vers les fils des interfaces du puits qui sont appelés racine comme illustré dans la figure 3.17 (soit Ri la racine d'indice i). Ce qui revient à identifier la racine avec la plus grande adresse parmi celles qui sont plus petites que sa propre adresse. L'algorithme 1 décrit le processus de détection de la branche d'un nouveau nœud.

Prenons l'exemple du nœud H de la figure 3.16, ce nœud possède l'adresse courte 10. Pour que le nœud H détecte à quelle branche il appartient, d'abord il doit calculer $Cskip(d = 0)$ et $Cskip(d = 1)$.

$$Cskip(d = 0) = \frac{1 + 2 - 2 - 2 * 2^{3-1-0}}{1 - 2} = 7$$

$$Cskip(d = 1) = \frac{1 + 2 - 2 - 2 * 2^{3-1-1}}{1 - 2} = 3$$

Ensuite le nœud H doit savoir l'interface de laquelle il est un descendant. Pour cela :

Il initialise l'adresse de l'interface à 1.

Il vérifie si son adresse est entre l'adresse de l'interface et l'adresse de l'interface + $Cskip(d = 0)$, c'est-à-dire 1 < 10 < 1 + 7 ➔ faux.

L'adresse de l'interface = adresse de l'interface + $Cskip(d = 0)$ = 1 + 7 = 8.

Il vérifie si son adresse est entre 8 et 15, 8 < 10 < 15 ➔ vrai.

Alors le nœud H est un descendant de l'interface qui possède l'adresse courte 8.

Puis le nœud H doit savoir à quelle branche il appartient. Pour cela,

il initialise l'adresse de la racine à 8 + 1 = 9 et l'indice de la branche à 1.

Il vérifie si son adresse est entre l'adresse de la racine et l'adresse de la racine + $Cskip(d = 1)$, c'est-à-dire si 9 < 10 < 12 ➔ vrai.

Par conséquence, le nœud H appartient à la branche d'indice 1 et il est un descendant de la racine 9.

Algorithme 1 Algorithme de calcul de l'indice de la branche à laquelle un nœud appartient

Prérequis L_m: $profondeur\ maximale\ de\ l'arbre$,
R_m: $nombre\ de\ fils\ coordinateur\ maximum$,
C_m: $nombre\ de\ fils\ maximum$,
adr_noeud : $l'adresse\ courte\ du\ noeud\ avec\ adr\_noeud > 0$

$interface\_detectée = faux$
Calculer:

$$Cskip\ (prof = 0) = \frac{1 + C_m - R_m - C_m * R_m^{L_m - 1 - 0}}{1 - R_m}$$

$$Cskip\ (prof = 1) = \frac{1 + C_m - R_m - C_m * R_m^{L_m - 1 - 1}}{1 - R_m}$$

$adr\_interface = 1$
répéter
si $adr\_interface < adr\_noeud < adr\_interface + Cskip(prof = 0)$
$interface\_detectée = vrai$
sinon
$adr\_interface = adr\_interface + Cskip(prof = 0)$

fin si
jusque $interface\_detectée == vrai$

$branche\_detectée = faux$
$adr\_racine = adr\_interface + 1$
$indice\_branche = 1$
répéter
si $adr\_racine \leq adr\_noeud < adr\_racine + Cskip(prof = 1)$
$branche\_detectée = vrai$
sinon
$adr\_racine = adr\_racine + Cskip(prof = 1)$
$indice\_branche = indice\_branche + 1$
fin si
jusque $branche\_detectée == vrai$
retourner $indice\_branche$

Formation de groupes et échange de données

Afin de permettre les transmissions de trafic de données vers le puits durant tous les intervalles, nous expliquons comment les deux groupes sont formés. Pour identifier facilement les groupes il suffit de considérer que :

Chaque nœud qui a une profondeur paire et qui appartient à une branche d'indice impaire, et chaque nœud qui a une profondeur impaire et qui appartient à une branche d'indice paire font partie du groupe 1.

Chaque nœud qui a une profondeur impaire et qui appartient à une branche d'indice impaire, et chaque nœud qui a une profondeur paire et qui appartient à une branche d'indice paire font partie du groupe 2.

Pour que les interfaces du puits soient toujours réceptrices, elles devront faire partie des deux groupes, alors qu'elles ne sont jamais émettrices. Comme nous pouvons le voir dans la figure 3.17 le groupe 1 inclut les nœuds E, F, B, D, H. Tous les autres nœuds font partie du groupe 2.

Comme le montre la figure 3.18, lorsque les nœuds du groupe 1 sont en mode transmission, les nœuds du groupe 2 sont en mode réception et vice-versa. Ceci maximise les transmissions en parallèle dans le réseau et garantit que les interfaces du NC puissent avoir des nœuds fils en mode transmission.

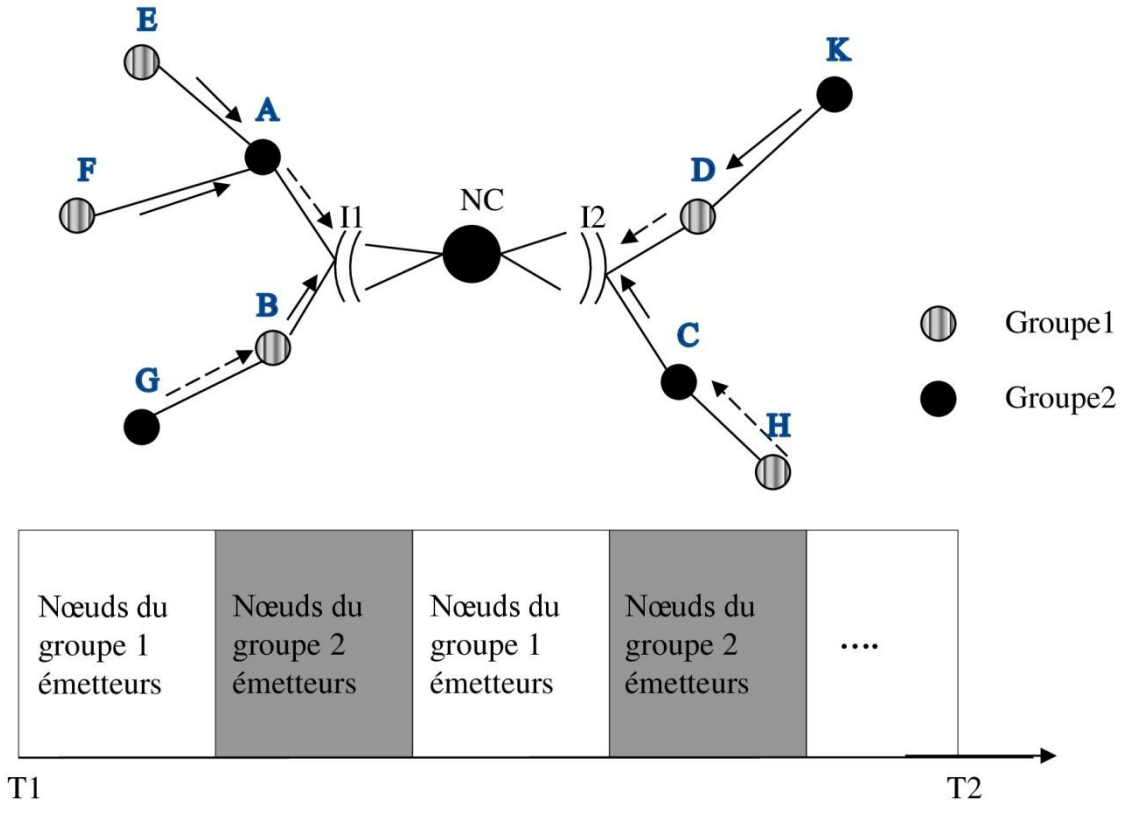


Figure 3.18 Activité des nœuds dans le réseau fragmenté.

3.2.2. Echange de *bitmap* pair et impair

Afin d'ordonnancer les communications entre les nœuds sur les différents canaux disponibles, chaque nœud a besoin de connaitre les canaux utilisés dans ses voisinages à 1-saut, 2-sauts et 3-sauts de son groupe. Selon HMC-MAC, chaque nœud échange des informations concernant l'utilisation des canaux. Ceci permet aux nœuds de choisir des canaux en évitant de générer des interférences et des collisions.

La liste des canaux disponibles est un paramètre initial connu de tous les nœuds du réseau, cette liste va servir de base pour le partage de la connaissance de l'occupation locale de ces canaux.

Afin d'éviter de surcharger le réseau avec de longs messages de contrôle pour échanger les informations d'utilisation des canaux, nous utilisons des *bitmaps* pour représenter l'occupation des canaux. Ces *bitmaps* sont appelés *bitmaps* de canaux. La taille de chaque *bitmap* est égale au nombre des canaux disponibles dans le réseau (tableau 3.5). Chaque bit dans les *bitmaps* indique l'occupation du canal équivalent dans le voisinage du groupe concerné. Plus précisément, dans un *bitmap* de canaux à m-saut, le bit 0 signifie que le canal qui correspond à ce bit est disponible dans son voisinage à m-saut ($1 \leq m \leq 3$), en revanche, le bit 1 signifie que ce canal est occupé dans ce voisinage. Initialement, ces *bitmaps* sont remplis avec des 0 ce qui signifie que tous les canaux disponibles[7] sont libres.

Lorsqu'un nœud choisit son canal, il diffuse son choix dans le *beacon*, ce qui permet à ses voisins de construire leurs *bitmaps* des canaux utilisés à 1-saut. Chaque nœud construit un bitmap de n bits (n égal au nombre des canaux disponibles) pour chaque groupe de nœuds. Plus précisément lorsqu'un nœud reçoit un *beacon* de son voisin, il récupère l'adresse courte et la profondeur de la source du *beacon* afin de vérifier à quel groupe il appartient, il récupère le canal choisi par ce nœud voisin et met à jour son *bitmap* de canaux à 1-saut du groupe auquel le nœud appartient en mettant à 1 le bit qui correspond au canal choisi.

Afin que chaque nœud construise un *bitmap* de canaux utilisés par ses voisins à 2-sauts pour chacun des deux groupes, il a besoin de l'union des *bitmaps* de canaux utilisés à 1-saut de ses voisins. Pour ce faire, chaque nœud inclut dans le *beacon* son bitmap des canaux à 1-saut pour chaque groupe. Ceci permet à un nœud qui reçoit ces *bitmaps* de construire son *bitmap* de canaux à 2-sauts de n-bits pour chaque groupe en exécutant l'opération « OR » sur tous les *bitmaps* à 1-saut reçus du groupe concerné pour cumuler tous les canaux utilisés au moins une fois dans son voisinage.

D'une façon similaire, chaque nœud inclut son bitmap de canaux à 2-sauts de chaque groupe dans le *beacon* dans le but de construire une liste de canaux utilisés dans le voisinage à 3-sauts pour chaque groupe de nœuds.

[7] La liste des canaux disponibles représente un sous-ensemble de canaux utilisables parmi les 16 canaux de la couche physique 802.15.4.

A noter que dans notre protocole, un nœud reçoit des *beacons* de nœuds voisins qui n'appartiennent pas forcément à son groupe. Dans le but de propager les *bitmaps* des canaux à tous les nœuds du réseau, chaque nœud doit inclure dans le *beacon* les *bitmaps* des canaux utilisés par chacun des deux groupes.

3.2.3. Priorité du choix du canal

Nous allons détailler la procédure permettant à chaque nœud d'avoir connaissance de son tour pour choisir un canal. Les interfaces du puits posent les problèmes les plus critiques en cas d'utilisation du même canal par plusieurs interfaces. Ceci peut dégrader de façon significative les performances du réseau. Pour éviter ce problème, HMC-MAC commence par l'attribution d'un canal différent à chaque interface du puits. Ensuite, le nœud le plus prioritaire choisit à son tour le canal le plus convenable.

Dans notre proposition, nous considérons que les priorités sont attribuées selon les adresses courtes du réseau. Le nœud ayant la plus petite adresse courte dans son voisinage à 3-sauts qui n'a pas encore de canal alloué, a la plus haute priorité. Le nœud qui a la plus haute priorité parmi ces voisins jusqu'à 3-sauts choisit son canal en premier.

Afin de connaitre le nœud prioritaire, chaque nœud doit avoir un *bitmap* de voisinage jusqu'à 3-sauts et un *bitmap* annonçant les nœuds ayant déjà leur canal. Le *bitmap* de voisinage jusqu'à 3-sauts est obtenu en exécutant une opération « OR » sur les bitmaps de voisins à 1-saut, 2-sauts et 3-sauts. Le *bitmap* annonçant les nœuds ayant déjà leur canal représente tous les nœuds du réseau et il est également utilisé pour annoncer aux autres nœuds du réseau les nœuds qui ont déjà terminé leur processus d'attribution de canal.

Ceci permet à chaque nœud d'obtenir le « *bitmap* d'allocation de canaux du voisinage » en exécutant l'opération suivante. Notons que ces bitmaps sont inclus dans le tableau 3.5.

$$bitmap\ d'allocation\ de\ canaux\ du\ voisinage = bitmap\ de\ voisinage\ jusqu'à\ 3-sauts\ \&\&\ !\ bitmap\ annonçant\ les\ nœuds\ ayant\ déjà\ leur\ canal \quad (3.1)$$

Ce *bitmap* est utilisé par chaque nœud dans le but de savoir si c'est son tour en contrôlant les bits qui le constituent. Dans ce *bitmap*, le bit 0 signifie un des deux cas : le nœud qui correspond à ce bit a déjà choisi un canal ou il ne fait pas partie du voisinage jusqu'à 3-sauts du nœud concerné. Le bit 1 signifie que le nœud qui correspond à ce bit est parmi les voisins jusqu'à 3-sauts du nœud concerné et il n'a pas choisi un canal. En conséquence, le nœud qui possède l'adresse courte la plus petite correspondant à un bit 1 est le nœud prioritaire pour le choix d'un canal.

Lorsqu'un nœud choisit son canal, il le diffuse dans le *beacon*, ce qui permet aux autres nœuds de modifier le « *bitmap* annonçant les nœuds ayant déjà leur canal ». Ce *bitmap* est également diffusé dans le *beacon* pour que les autres nœuds du réseau sachent quand sera leur tour pour

choisir un canal. Notons que la phase d'allocation des canaux se termine lorsque le « *bitmap* d'allocation de canaux » ne comporte que des 1. Ceci indique que tous les nœuds du réseau ont choisi leur canal.

Algorithme 2 Algorithme d'élection du nœud ayant à faire le choix du canal

Présréquis $ordre\_de\_synchro : ordre\ de\ synchronisation,$
$adr\_noeud : l'adresse\ courte\ du\ noeud,$
$bitmap\ de\ voisinage\ à\ n - saut : bitmap\ indiquant\ si\ le\ i^{ème}\ coordinateur\ de\ l'ordre\_de\_synchro\ est\ un\ voisin\ à\ n - saut\ de\ ce\ nœud,$
$bitmap\ annonçant\ les\ nœuds\ ayant\ des\ canaux : bitmap\ indiquant\ si\ le\ i^{ème}\ coordinateur\ a\ choisi\ un\ canal.$

pour $chaque\ adr\_courte\ dans\ l'ordre\_de\_synchro$ **faire**
$bitmap\ de\ voisinge\ jusqu'à\ 3 - sauts = bitmap\ de\ voisinage\ à\ 1 - saut\ ||$
$bitmap\ de\ voisinage\ à\ 2 - sauts\ ||\ bitmap\ de\ voisinage\ à\ 3 - sauts$
fin pour
pour $chaque\ adr\_courte\ dans\ l'ordre\_de\_synchro$ **faire**
$bitmap\ d'allocation\ de\ canaux\ des\ voisinages =$
$bitmap\ de\ voisinage\ jusqu'à\ 3 - sauts\ \&\&\ !\ bitmap\ annonçant\ les\ nœuds\ ayant\ déjà\ leur\ canal$
fin pour
pour $chaque\ adr\_courte\ dans\ l'ordre\_de\_synchro$ **faire**
    si $le\ bitmap\ d'allocation\ de\ canaux\ des\ voisinages\ est\ vrai$ **alors**
        si $adr\_courte < adr\_courte\_minimal$ **alors**
            $adr\_courte\_minimal = adr\_courte$
        fin si
    fin si
fin pour
si $adr\_noeud = adr\_courte\_minimal$
retourner $vrai$
sinon retourner faux

3.2.4. Algorithme proposé pour l'attribution des canaux aux nœuds

Dans les parties précédentes, nous avons expliqué la façon dont le réseau est divisé et comment un nœud connait son groupe afin de savoir l'intervalle de temps durant lequel il est autorisé à envoyer ses données. Ensuite, nous avons abordé le mécanisme d'échange des *bitmaps* concernant les canaux utilisés dans chaque groupe entre les nœuds du réseau. Cet échange permet à chaque nœud du réseau de choisir son canal d'une façon distribuée sans passer par le NC. Le choix du canal se base sur les informations transmises dans les *bitmaps* des canaux utilisés dans les voisinages à 1-saut, 2-sauts et 3-sauts. Enfin nous avons détaillé la méthode avec laquelle chaque nœud est capable de connaitre son tour pour choisir un canal.

Dans notre protocole, nous nous intéressons au rapport (nombre de nœuds /nombre de canaux disponibles) afin d'optimiser la distribution des canaux. En effet, dans la plupart des cas le nombre de nœuds dans le voisinage à 3-sauts de chaque groupe est supérieur au nombre de canaux disponibles. Ceci signifie que les canaux disponibles ne sont pas en nombre suffisant pour permettre à tous les nœuds de communiquer dans leur intervalle de temps sans risque d'interférences. Nous proposons dans notre processus d'attribution de canal d'exploiter la réutilisation spatiale.

Comme dit précédemment, notre but est de diminuer les interférences autant que possible même dans les cas où le même canal est utilisé par des nœuds interférents. D'après l'étude des conséquences de la réutilisation du même canal dans le voisinage jusqu'à 3-sauts, nous avons prouvé que le risque de collisions lors de la réutilisation du même canal dans un voisinage à 3-sauts est moins important que celui à 2-sauts. Similairement, le risque de collisions lorsque le même canal est utilisé dans le voisinage à 2-sauts est moins important que celui à 1-saut. Pour cela nous proposons que chaque nœud à son tour applique l'algorithme expliqué ci-dessous pour choisir le canal le plus convenable dans son voisinage jusqu'à 3-sauts.

Dans un premier temps, le nœud le plus prioritaire localement essaie de trouver un canal libre dans son voisinage jusqu'à 3-sauts de son groupe. S'il n'en trouve pas, il essaie de trouver un canal libre dans son voisinage jusqu'à 2-sauts de son groupe. Dans le cas où il n'y a aucun canal libre dans son voisinage jusqu'à 2-sauts, le nœud vérifie s'il y a un canal libre parmi ses voisins à 1-saut de son groupe. Lorsque plusieurs canaux sont libres, le nœud en choisit aléatoirement un. Enfin, si le nœud ne trouve pas un canal libre, il choisit aléatoirement un canal parmi la liste des canaux les moins utilisés dans son voisinage à 1-saut qui font partie de son groupe.

Ces opérations se déroulent pendant la phase de configuration du réseau et si des changements se produisent dans la topologie. La phase de configuration se termine lorsque tous les nœuds ont choisi leur canal. La phase d'échange de données peut ensuite commencer.

Notons que durant la phase d'échange de données, lorsqu'un nœud est désassocié ou un nouveau nœud rejoint le réseau, il exécute les mêmes phases annoncées ci-dessus avec la seule différence qu'il doit utiliser le canal exploité par le nœud père pressenti afin de demander l'association.

Algorithme 3 Algorithme de choix du canal

Présréquis $liste\_des\_canaux\_libres\ jusqu'à\ 1-saut$,
$liste\_des\_canaux\_libres\ jusqu'\ à\ 2-sauts$,
$liste\_des\_canaux\_libres\ jusqu'à\ 3-sauts$,
$liste\_des\_canaux\_les\_moins\_utilisés\ à\ 1-saut\ dans\ le\ groupe\ auquel\ le\ noeud\ appartient$.

si *le noeud n est le noeud prioritaire dans le réseau alors*
  si $liste\_des\_canaux\_libres\ jusqu'\ à\ 3-sauts\ \neq \emptyset\ alors$
    $indice\_canal\ = rand(liste\_des\_canaux\_libres\ jusqu'\ à\ 3-sauts)$
  sinon si $liste\_des\_canaux\_libres\ jusqu'\ à\ 2-sauts\ \neq \emptyset\ alors$
      $indice\_canal\ = rand(liste\_des\_canaux\_libres\ jusqu'\ à\ 2-sauts)$
      sinon si $liste\_des\_canaux\_libres\ à\ 1-saut\ \neq \emptyset\ alors$
        $indice\_canal\ = rand(liste\_des\_canaux\_libres\ à\ 1-saut)$
          sinon
            $indice\_canal\ = rand(liste\_des\_canaux\ les\ moins\ utilisés\ à\ 1-saut)$
          fin si
      fin si
  fin si
fin si

3.2.5. Liste des *bitmaps* générés par nœud

Le tableau 3.5 récapitule les *bitmaps* générés par chaque nœud en précisant la taille de chaque *bitmap* et spécifie s'ils sont envoyés dans le *beacon*.

Liste des *bitmaps* qu'un nœud doit gérer	Bitmaps transmis dans le *beacon*	Taille (en bit)
Bitmap de voisins à 1-saut	Oui	Nombre de nœuds
Bitmap de voisins à 2-sauts	Oui	Nombre de nœuds
Bitmap de voisins à 3-sauts	Non	Nombre de nœuds
Bitmaps des canaux à 1-saut groupe 1	Oui	Nombre de canaux disponibles
Bitmaps des canaux à 1-saut groupe 2	Oui	Nombre de canaux disponibles
Bitmaps des canaux à 2-sauts groupe 1	Oui	Nombre de canaux disponibles
Bitmaps des canaux à 2-sauts groupe 2	Oui	Nombre de canaux disponibles
Bitmaps des canaux à 3-sauts groupe 1	Non	Nombre de canaux disponibles
Bitmaps des canaux à 3-sauts groupe 2	Non	Nombre de canaux disponibles
Bitmaps des canaux jusqu'à 3-sauts groupe 1	Non	Nombre de canaux disponibles
Bitmaps des canaux jusqu'à 3-sauts groupe 2	Non	Nombre de canaux disponibles
Bitmaps des canaux jusqu'à 2-sauts groupe 1	Non	Nombre de canaux disponibles
Bitmaps des canaux jusqu' à 2-sauts groupe 2	Non	Nombre de canaux disponibles
Bitmap de voisinage jusqu'à 3-sauts	Non	Nombre de nœuds
Bitmap annonçant les nœuds ayant déjà leur canal	Oui	Nombre de nœuds

Tableau 3.6 La liste des *bitmaps* générés par chaque nœud jusqu'à x-sauts n'est que facilité d'implémentation fonctionnellement ce n'est pas un besoin absolu.

3.2.6. Structure et taille de *beacon*

La trame de *beacon* est propagée dans le réseau afin d'échanger des informations permettant l'initialisation et la synchronisation du réseau et l'attribution des canaux. Comme dit précédemment HMC-MAC se base sur la couche physique de la norme IEEE 802.15.4. En conséquence, la taille de la trame de *beacon* est limitée à 127 octets. Le tableau 3.6 représente la structure du *beacon*.

	Information transmise dans le *beacon*	Taille
En-tête	Contrôle de la trame	2 octet
	Numéro de séquence	1 octet
	Adresse de la source	2 octet
Données	Nombre de nœuds	1 octet
	Durée des activités des nœuds	1 octet
	Durée d'inactivité	1 octet
	Canal utilisé	1 octet
	Champs renseignés	6 octets
	Nombre de fils d'un coordinateur	1 octet
	Lm, Rm	2 octet
	Profondeur	1 octet
	Stabilité	1 bit
	Les *bitmaps* transmis	(3 * nombre de nœuds + 4 * nombre de canaux) en bits

Tableau 3.7 Structure du *beacon*

L'en-tête du *beacon* :

- Contrôle de la trame : ce champ indique le type de la trame.
- Numéro de séquence : c'est le numéro de séquence du *beacon* (modulo 256). Ce numéro est géré par le NC.
- Adresse de la source : l'adresse courte du nœud transmettant la trame.

Le champ de données du *beacon* :

- Nombre de nœuds : c'est le nombre des coordinateurs existants dans le réseau (dans notre solution tous les nœuds sont des coordinateurs), ils participent à la propagation du *beacon*.
- Durée des activités des nœuds : c'est la durée de la période d'échange de données des nœuds [T1 ; T2].

- Durée d'inactivité : c'est la durée durant laquelle les nœuds peuvent être en mode sommeil.
- Canal utilisé : c'est le canal choisi par le coordinateur diffusant le *beacon*.
- Champs renseignés : ces champs sont utilisés afin d'inclure les modifications lorsqu'un nouveau nœud s'associe au réseau. En effet, NC inclut dans le *beacon* l'adresse du nouveau nœud ainsi que celles des deux nœuds (celui qui le précède et celui qui le suit) dans la liste qui définit l'ordre de propagation.
- Nombre de fils : c'est le nombre de nœuds associés au coordinateur diffusant le *beacon*.
- Lm, Rm : les paramètres utilisés du standard ZigBee. Notons que dans HMC-MAC, la valeur de Rm est similaire à celle de Cm car il n'existe pas des nœuds feuilles.
- Profondeur : le nombre de sauts à effectuer à partir de l'émetteur de *beacon* pour atteindre le NC.
- Stabilité : ce champ spécifie si le nœud diffusant le *beacon* a atteint sa stabilité.
- Les bitmaps transmis : les *bitmaps* qui sont transmis dans le *beacon* sont définis dans le tableau 3.5.

Nous nous intéressons au calcul de la taille du *beacon* afin de vérifier si HMC-MAC supporte le passage à l'échelle.

- Pour 50 nœuds et 10 canaux, la taille de la trame de *beacon* dans HMC-MAC est égale : (19 * 8) + 1 +3 * nombre de nœuds + 4 * nombre de canaux = 152 + 1 + 3 * 50 + 4 * 10 = 343 bits = 43 octets < 127 octets.
- Pour 100 nœuds et 16 canaux, la taille de la trame de *beacon* dans HMC-MAC est égale : (19 * 8) + 1 +3 * nombre de nœuds + 4 * nombre de canaux = 152 + 1 + 3 * 100 + 4 * 16 = 517 bits = 65 octets < 127 octets.
- Pour 16 canaux, qui est le nombre maximal des canaux disponibles, le nombre de nœuds pour une trame de *beacon* de taille maximale dans HMC-MAC est égal :
    - (19 * 8) + 1 +3 * nombre de nœuds + 4 * nombre de canaux = 127 octets.
    - Nombre de nœuds $= \frac{(127*8)-(4*16)}{(19*8)+1+3}$ bits.
    - Nombre de nœuds = 266.

Par conséquence, HMC-MAC supporte l'extensibilité du réseau et ainsi le passage à l'échelle jusqu' à une limite de 266 nœuds quand les16 canaux sont utilisés.

4. Etude de cas

Afin d'illustrer notre proposition, nous allons présenter un scenario pour décrire d'une façon détaillée le processus exécuté par les nœuds afin que chacun choisisse son canal.

Comme dit précédemment, nous nous basons sur une topologie arborescente avec un seul puits multi-interface, le NC, qui possède 2 interfaces I1 et I2 (voir figure 3.19). Chacune de ces interfaces possède des nœuds fils. Nous nous intéressons aux applications de *converge-cast* pour cela nous supposons que tous les paquets sont transmis à destination du NC. Nous disposons de 5 canaux disponibles {C0, C1, C2, C3, C4} en supposant que les autres canaux ont été utilisés localement par des technologies exploitant la même bande de fréquences. Dans les tableaux 3.7, 3.8 et 3.9 chaque ligne représente uniquement les différents *bitmaps* du nœud dont c'est le tour de faire le choix d'un canal.

Dans un premier temps, Le NC démarre avec les canaux 0 et 1 présélectionnés (réservés) pour ses deux interfaces. Ensuite, l'allocation des canaux pour les différents nœuds du réseau peut s'exécuter.

Dans la deuxième étape, nous commençons par déterminer le nœud qui a la plus petite adresse courte parmi les nœuds dans son voisinage jusqu'à 3-sauts qui n'ont pas encore choisi un canal. Pour ce faire, nous nous servons de l'équation 3.1 et nous vérifions les bits résultant de cette équation comme illustré dans le tableau 3.7. Puisque le nœud A possède la plus petite adresse courte correspondant à un bit 1, il choisit son canal en premier. Comme le nœud A appartient au groupe 2, il exécute l'opération « OR » sur les bitmaps de canaux utilisés dans son voisinage à 1-saut, 2-sauts, et 3-sauts du groupe 2 afin d'obtenir le *bitmap* de canaux utilisés dans ses voisins jusqu'à 3-saut (voir tableau 3.8 et tableau 3.9). Le nœud A doit choisir un canal aléatoire parmi ceux qui sont libres. Supposons qu'il choisisse le canal 2.

Le même processus peut être exécuté de l'étape 3 à l'étape 8. Chaque nœud choisit à son tour un canal libre parmi ses voisinages jusqu'à 3-sauts de son groupe. Ces canaux sélectionnés sont illustrés dans le tableau 3.9.

Dans l'étape 9, comme nous pouvons le voir dans le tableau 3.7, c'est le tour du nœud D. Le nœud D applique l'algorithme 2 et détecte qu'il appartient au groupe 1. Ainsi, il cherche un canal libre parmi ses voisins jusqu'à 3-sauts qui appartiennent à son groupe. Comme nous pouvons le voir dans le tableau 3.9, le bitmap des voisins jusqu'à 3-sauts est rempli par des 1. Ceci indique qu'il n'y a plus de canaux libres dans son voisinage jusqu'à 3-sauts. Alors il essaie de trouver un canal libre dans son voisinage jusqu'à 2-sauts dans son groupe en exécutant l'opération « OR » sur les *bitmaps* de canaux à 1-saut et 2-sauts. D'après le bitmap qui représente les canaux utilisés dans les voisinages jusqu'à 2-sauts, D choisit le canal C2.

Ensuite, c'est le tour du nœud K qui appartient au groupe 1. Pour cela il contrôle les bitmaps de canaux jusqu'à 3-sauts du groupe 1 et détecte que le canal C2 est le canal qui n'a pas été utilisé par son voisinage jusqu' à 3-sauts.

Notons que le nœud K a exploité la réutilisation spatiale en choisissant un canal qui a été utilisé par le nœud G. Les nœuds G et K sont des voisins à 4-sauts et appartiennent au même groupe (groupe 2). Ces deux nœuds peuvent recevoir des données en même temps en utilisant le même canal C3. Notons aussi que H est à 4-sauts de E et ils appartiennent au groupe 1. Ces deux nœuds ont également utilisé le même canal C2.

Ce qui est intéressant à noter aussi à travers ce scenario, est que notre processus permet aux nœuds voisins qui appartiennent à des groupes différents d'utiliser le même canal mais dans des intervalles de temps consécutifs pour réduire la contention. Par exemple le nœud A est un voisin à E mais ils peuvent utiliser le même canal C2 car chacun appartient à un groupe différent. C'est le même cas pour les deux nœuds B et C.

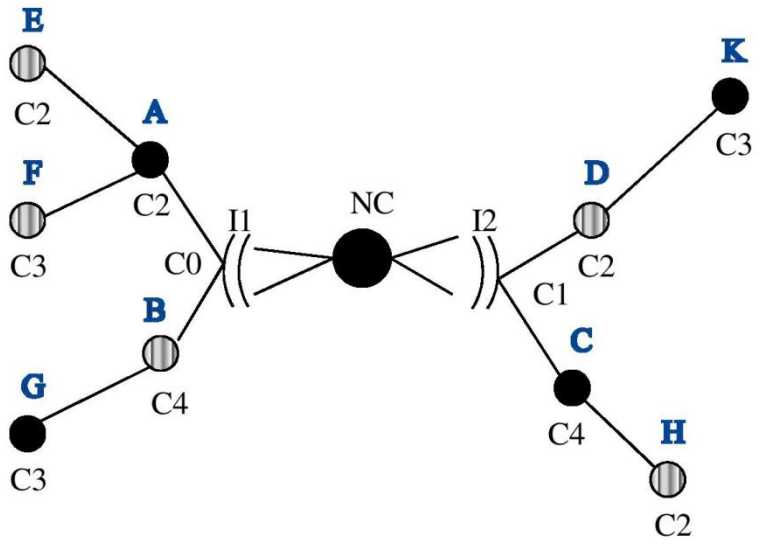


Figure 3.19 Exemple de distribution des canaux dans une topologie multi-saut.

		Bitmap annonçant les nœuds ayant des canaux										Bitmap de voisinage jusqu'à 3-saut										! Bitmap annonçant les nœuds ayant des canaux && Bitmap de voisinage jusqu'à 3-saut										Tour
Adresse courte		0	2	5	9	12	3	4	6	10	13	0	2	5	9	12	3	4	6	10	13	0	2	5	9	12	3	4	6	10	13	
Etape	Nœud	NC	A	B	C	D	E	F	G	H	K	NC	A	B	C	D	E	F	G	H	K	NC	A	B	C	D	E	F	G	H	K	
1	NC	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	NC
2	A	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	A
3	E	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	E
4	F	1	1	0	0	0	1	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0	F
5	B	1	1	0	0	0	1	1	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	0	0	1	1	1	B
6	G	1	1	1	0	0	1	1	0	0	0	1	1	1	1	1	0	0	1	0	0	0	0	0	1	1	0	0	1	0	0	G
7	C	1	1	1	0	0	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	1	1	0	0	0	1	1	C
8	H	1	1	1	1	0	1	1	1	0	0	1	1	1	1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	H
9	D	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1	D
10	K	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	K

Tableau 3.8 La procédure de connaissance du tour pour le choix de canal.

Etape	Nœud	Bitmaps des canaux à 1-saut groupe 1					Bitmaps des canaux à 1-saut groupe 2					Bitmaps des canaux à 2-saut groupe 1					Bitmaps des canaux à 2-saut groupe 2					Bitmaps des canaux à 3-saut groupe 1					Bitmaps des canaux à 3-saut groupe 2				
		C0	C1	C2	C3	C4	C0	C1	C2	C3	C4	C0	C1	C2	C3	C4	C0	C1	C2	C3	C4	C0	C1	C2	C3	C4	C0	C1	C2	C3	C4
1	NC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	A	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	E	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	F	0	0	0	0	0	0	0	1	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5	B	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0
6	G	0	0	0	0	1	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0
7	C	1	1	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	1	0	0	0	0	1	0
8	H	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0
9	D	1	1	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	1	0	0	0	0	1	0
10	K	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	1	0	1

Tableau 3.9 Les bitmaps de canaux utilisés dans les voisinages à 1-saut, 2-sauts et 3-sauts de chaque groupe.

	Nœud	Bitmap des canaux jusqu'à 3-saut groupe 1					Bitmap des canaux jusqu'à 3-saut groupe 2					Bitmap des canaux jusqu'à 2-saut groupe 1					Canal choisi
Etape		C0	C1	C2	C3	C4	C0	C1	C2	C3	C4	C0	C1	C2	C3	C4	
1	NC																C0 et C1
2	A						1	1	0	0	0						C2
3	E	1	1	0	0	0											C2
4	F	1	1	1	0	0											C3
5	B	1	1	1	1	0											C4
6	G						1	1	1	0	0						C3
7	C						1	1	1	1	0						C4
8	H	1	1	0	0	1											C2
9	D	1	1	1	1	1						1	1	0	0	1	C2
10	K						1	1	1	0	1						C3

Tableau 3.10 La procédure de choix du canal. Le contenu du tableau est construit à partir du tableau 3.6. Pour chaque nœud nous présentons uniquement les *bitmaps* utilisés pour le choix du canal selon son groupe

5. Echange de messages de données

L'espace temporel [T1; T2] est divisé en plusieurs parties dont chacune est divisée en deux intervalles de temps. Durant le premier intervalle, les nœuds du groupe 1 sont des émetteurs pendant que les nœuds du groupe 2 sont des récepteurs. Durant l'intervalle suivant, les nœuds échangent leur rôle, les récepteurs deviennent des émetteurs et les émetteurs deviennent des récepteurs. Ainsi, les nœuds reçoivent des trames de données de leurs nœuds fils du niveau inférieur et transmettent ces trames de données à leur nœud parent du niveau supérieur dans la prochaine période en utilisant l'algorithme CSMA/CA. Ceci permet de séquencer les échanges de données dans la topologie.

Dans le mécanisme de CSMA/CA lorsqu'une collision est détectée, le nœud génère un *backoff* afin de retransmettre le paquet. Dans le cas où le *backoff* généré dépasse le temps de communication restant du groupe, le nœud doit se placer sur le début de la prochaine période de son groupe. De plus, lorsqu'un nœud génère une trame à un moment donné où il n'est pas autorisé à envoyer des données, il doit se placer également sur le début de son prochain intervalle de temps où il est autorisé à envoyer des données. La figure 3.20 illustre ces deux problématiques. Cette accumulation de trafic au début de chaque période augmente le risque de collision entre les nœuds associés au même père.

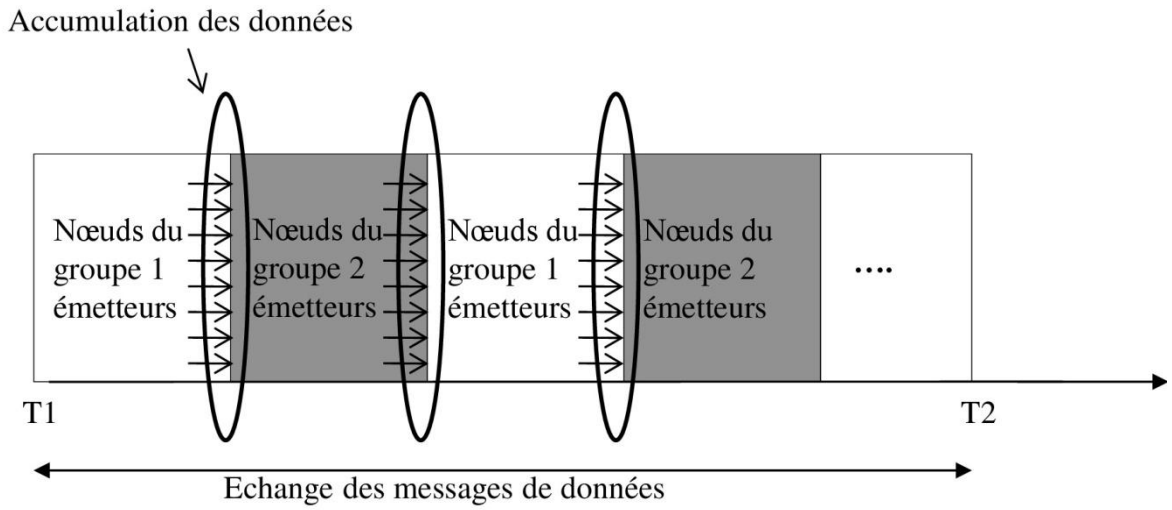


Figure 3.20 Le placement des nœuds au début du prochain intervalle de leur groupe.

Pour minimiser ce risque de collision, nous proposons d'utiliser un délai aléatoire qui permet de différer les activités des nœuds durant leur intervalle de temps autorisé. C'est-à-dire chaque nœud tire un nombre aléatoire qui peut aller de 0 à 90% de la durée de chaque intervalle avant

d'émettre sa première trame. Quand le délai aléatoire expire, ce nœud commence la transmission de ses paquets dans sa file d'attente jusqu'à la fin de l'intervalle de temps en cours en utilisant le CSMA/CA slotté. Ceci évite l'accès simultanés des nœuds associés au même père au début de chaque intervalle de temps et ainsi diminue les collisions générées dans le réseau.

6. Conclusion

Les collisions et les interférences représentent l'une des causes essentielles de la dégradation des performances dans les réseaux de capteurs sans fil. Pour cela, nous avons étudié l'impact de la réutilisation du même canal dans les voisinages jusqu'à 3-sauts.

Dans ce chapitre, nous avons proposé le protocole multi-canal HMC-MAC qui supporte l'existence d'un puits multi-interface dans un réseau multi-saut. Il s'agit d'un protocole distribué qui utilise une approche d'attribution de canaux semi-dynamique. Ce protocole vise à améliorer les performances du réseau en combinant les trois techniques TDMA, CSMA et FDMA.

HMC-MAC permet de réduire les collisions entre les fils du même nœud en proposant d'une part une politique d'association des nœuds au réseau qui vise à repartir le nombre de fils des nœuds de la même profondeur, et d'autre part une désynchronisation du temps d'envoi de données de ces nœuds sur l'intervalle de temps autorisé.

Nous avons proposé de fragmenter le réseau en deux groupes de sorte que ces derniers alimentent le puits de façon continue afin d'améliorer les performances du réseau en terme de débit et de diminuer l'effet de la densité des nœuds. Nous avons aussi proposé une nouvelle méthode d'allocation des canaux qui permet au nœud de choisir leur canal d'une façon distribuée en se basant sur notre étude concernant la réutilisation du même canal.

Dans le chapitre suivant, nous allons étudier les performances de notre protocole en termes de débit, taux de pertes, nombre de collisions, et délai de bout-en-bout en les comparant à d'autres protocoles multi-canaux.

Chapitre 4 Résultats

Après avoir présenté notre contribution dans le chapitre précédent, dans ce chapitre, nous présentons une évaluation des performances par simulation de notre proposition et nous les comparons à celles d'autres méthodes utilisées dans la littérature. Cette évaluation est effectuée en utilisant le simulateur NS-2. Ce chapitre est composé de deux parties. Dans la première, nous présentons les performances de notre proposition en fonction du nombre de conflits potentiels et du taux d'interférence en se basant sur les canaux attribués par les nœuds. Dans la deuxième partie, nous nous intéressons aux conséquences sur la transmission des paquets de HMC-MAC afin d'étudier le comportement de plusieurs métriques.

1. Evaluation de la méthode d'allocation des canaux

Dans le but d'identifier les conséquences d'une allocation imparfaite des canaux, nous évaluons par simulation les performances de notre méthode d'allocation de canaux en termes de nombre de conflits et de taux d'interférence en fonction d'une densité moyenne du réseau. Pour ce faire, nous avons généré des topologies aléatoires dans lesquelles le puits est situé au centre. Afin d'obtenir des densités différentes, nous avons fait varier le nombre de nœuds de 15 à 100 nœuds. Nous faisons croitre par pas de 5 et pour chaque étape la simulation est réalisée N fois (N est indiqué dans les tableaux représentants les paramètres de simulation). Ces topologies sont utilisées en fonction de leur densité explorée de 2 à 11 par pas de 1. Les courbes obtenues représentent les moyennes des densités pour les différentes topologies générées. Dans ces simulations, nous considérons que les 16 canaux de la couche physique de la norme IEEE 802.15.4 sont disponibles.

En se basant sur l'étude des protocoles multi-canaux effectuée dans la partie 2 du chapitre 2, nous avons choisi de comparer les performances de notre méthode d'allocation de canaux à celles des méthodes d'allocation suivantes :

- Méthode HMC-MAC sans division : chaque nœud applique le même mécanisme d'allocation de canaux que HMC-MAC sans segmenter le réseau.
- Méthode aléatoire : chaque nœud choisit aléatoirement un canal parmi les canaux disponibles.

- Méthode 2-sauts : chaque nœud choisit un canal libre dans son voisinage jusqu'à 2-sauts, si tous les canaux sont utilisés, il choisit aléatoirement un canal parmi les canaux disponibles.
- Méthode 3-sauts : chaque nœud choisit un canal libre dans son voisinage jusqu'à 3-sauts, si tous les canaux sont utilisés, il choisit aléatoirement un canal parmi les canaux disponibles.

1.1. Evaluation du nombre de conflits

Nous évaluons par simulation le nombre de conflits potentiels en fonction de la densité du réseau afin de montrer l'intérêt de notre méthode d'allocation de canaux. Pour cela, nous étudions trois scenarios. Dans le premier, nous considérons que tous les nœuds sont des destinataires potentiels, dans ce cas un message peut être acheminé vers n'importe quel nœud du réseau. Dans le deuxième scenario, nous considérons que tout le trafic est orienté vers le puits. Dans le troisième scenario, nous prenons en compte les émissions réalisées alternativement par les nœuds du groupe 1 et les nœuds du groupe 2. Le tableau 4.1 synthétise les paramètres de simulation.

Simulateur utilisé	NS-2
Dimensions du réseau	100*100 m^2
Débit	250kbps
Portée de communication	20 m
Nombre de nœuds	15-100
Nombre des interfaces du puits	1
Nombre des canaux disponibles	16
Nombre de topologies générés	2 topologies aléatoires[8]
Taille de la file d'attente	50 paquets
Paramètre des topologies	Lm = 7, Rm = Cm = 5
Fenêtre d'observation	100 secondes
Nombre de répétitions N	20 répétitions pour chaque méthode
Modèle de propagation	*Two-ray ground*

Tableau 4.1 Paramétres de simulation utilisés pour l'évaluation du nombre de conflits.

[8] Dans ce type de topologies, les nœuds sont déployés aléatoirement dans le réseau.

1.1.1. Premier Scenario : dénombrement des risques de conflits.

Dans ce scenario, nous évaluons le nombre de tous les conflits qui peuvent survenir entre les nœuds du réseau. Pour quantifier ceci, nous avons récapitulé tous les conflits potentiels dans la figure 4.1 en nous basant sur l'étude des conséquences de la réutilisation des canaux dans les voisinages jusqu'à 3-sauts (voir partie 1.1.1 du chapitre 3). La figure 4.1 nous permet de compter tous les conflits potentiels. Le dénombrement est réalisé en considérant dans un premier temps tous les conflits survenant lors de l'émission de A vers un de ses voisins. Puis en appliquant le même raisonnement à tous les autres nœuds du réseau, ainsi les trames reçues par A seront aussi prises en compte. Notons que chaque message de données devrait être acquitté par le récepteur.

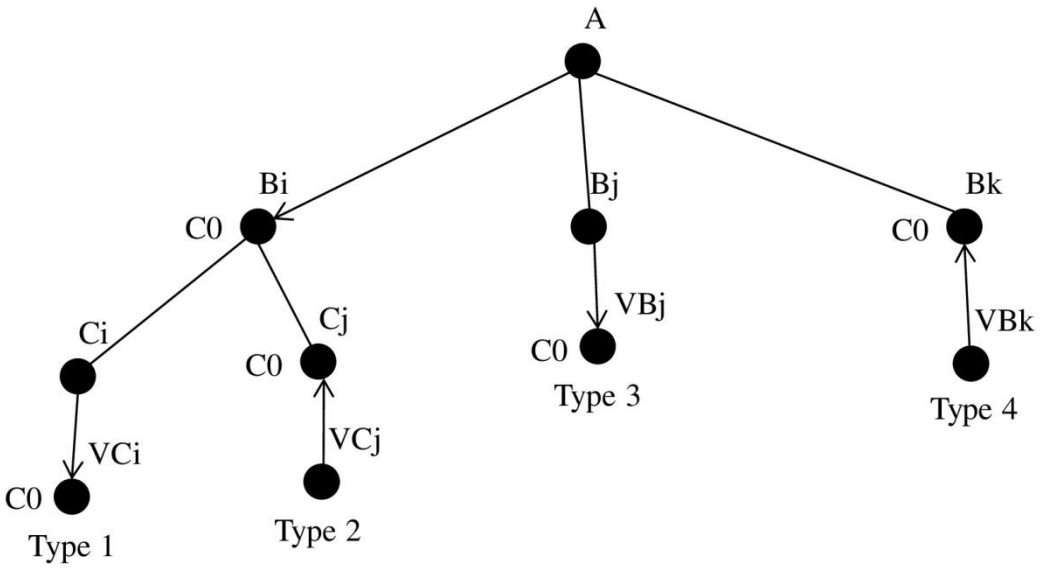


Figure 4.1 Types de conflits potentiels résultant de l'émission d'une trame par A.

Soit Bi, Bj et Bk les nœuds voisins à 1-saut de A, Ci et Cj les nœuds voisins à 2-sauts de A, et VX les nœuds voisins à 1-saut d'un nœud quelconque X. Nous considérons quatre types de conflits comme illustré sur la figure 4.1.

Type 1 : conflit données-données

Il y a un conflit données-données pour chaque couple (Bi, VCi) ayant un voisin commun à 1-saut Ci, si Bi et VCi partagent le même canal.

Type 2 : conflit Données-Ack

Il y a un conflit Données-Ack pour chaque couple (Bi, VCj) ayant un voisin commun à 1-saut Cj, si Bi et Cj utilisent le même canal.

Type 3 : conflit Ack- Données

Il y a un conflit Données-Ack pour chaque couple (Bi, VBj) où Bj est un voisin à un saut de A différent de Bi, si Bi et VBj utilisent le même canal.

Type 4 : conflit Ack-Ack

Il y a un conflit Ack-Ack pour chaque couple (Bk, VBk) où Bk est un voisin à 1-saut de A différent de Bi et VBk est un voisin à 1-saut de Bk (n'étant ni A ni Bi), si Bi et Bk partagent le même canal.

Prenons l'exemple du type 1 (conflit données-données). En effet, une collision Données-Données se produit au niveau du nœud Bi lorsque les deux nœuds A et Ci transmettent des trames de données en même temps destinées aux nœuds Bi et VCi respectivement. Notons que le même raisonnement s'applique pour les autres types.

Afin de calculer tous les conflits potentiels, nous calculons pour chaque nœud du réseau le nombre de conflits potentiels comme nous l'avons précédemment fait pour le nœud A.

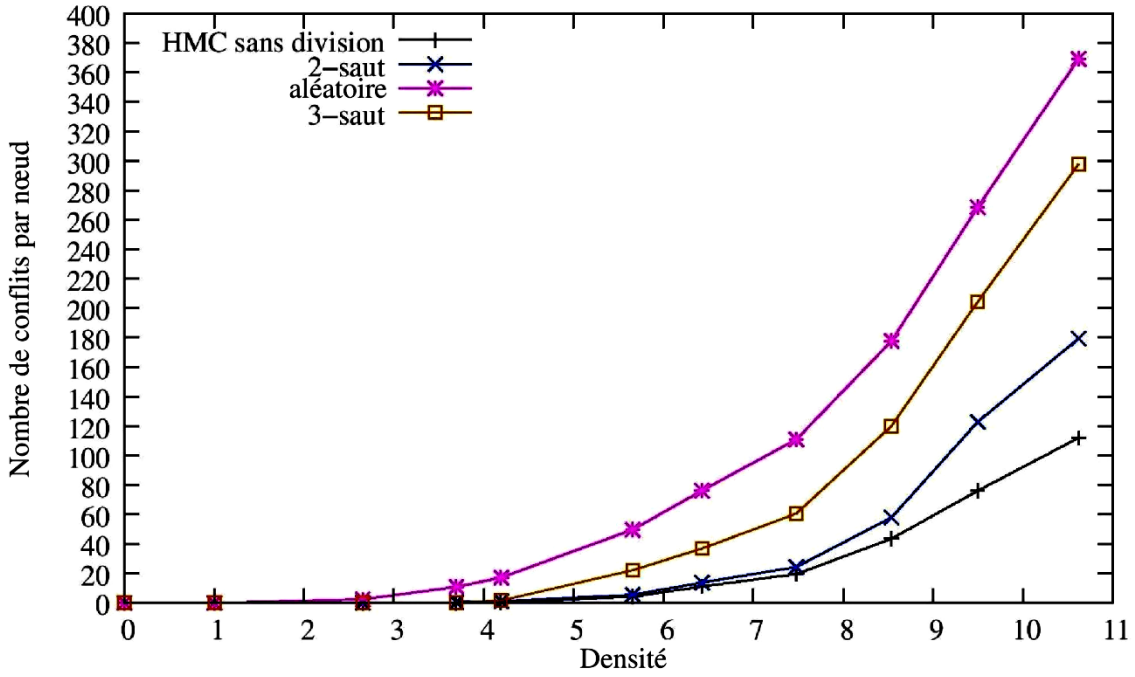


Figure 4.2 Influence de la densité sur le nombre de conflits potentiels.

La figure 4.2 présente le nombre de conflits potentiels par nœud en fonction de la densité du réseau. Nous observons que le nombre de conflits augmente avec la densité. Ceci est attendu car le nombre de nœuds dans une zone donnée partageant le même canal augmente. Nous pouvons constater que HMC-MAC sans division réduit le nombre de conflits par rapport aux autres méthodes. Par exemple, pour des densités comprises entre 5 et 6, HMC-MAC sans division réduit le nombre de conflit d'environ 91%, 81%, et 25% par rapport aux méthodes aléatoire, 3-sauts, 2-sauts respectivement. Pour des densités comprises entre 10 et 11, la réduction du nombre de conflits est d'environ 70%, 62% et 38% respectivement. Cette réduction est due au fait que HMC-MAC sans division prend toujours en considération les canaux utilisés avant de choisir un canal.

Notons que la méthode 3-sauts est plus performante que celle de 2-sauts tant que le nombre des canaux est inférieur au nombre des voisins jusqu'à 3-sauts. En revanche lorsque le nombre de voisinages jusqu'à 3-sauts dépasse le nombre des canaux disponibles, la méthode 2-sauts devient plus performante que celle de 3-sauts. Ce comportement n'est pas très intuitif, il s'explique par le fait qu'un nœud commence à choisir un canal d'une façon aléatoire plus tôt en appliquant la méthode 3-sauts.

Dans un réseau de capteurs, le trafic est souvent destiné à un ou quelques points de collecte. Donc dans notre explication basée sur les trames émises par A, les destinataires ne seront pas tous les voisins de A mais ceux qui le sépare du point de collecte visé. Ceci devrait réduire le nombre de conflits potentiels qui dans le cas que nous venons d'étudier à une croissance quadratique en fonction de la densité.

1.1.2. Deuxième scenario : trafic orienté vers le puits.

Dans ce scenario, nous tenons compte du fait que le trafic est orienté vers un seul point de collecte appelé « le puits ». C'est le « convergecast » c'est-à-dire que les échanges se font du plus loin au plus près de ce nœud de collecte. Ainsi, les trames de données émises par un nœud sont destinées soit au puits lui-même, soit à un nœud plus proche de ce dernier (ayant donc une profondeur inférieure dans un arbre dont le puits est la racine). De ce fait, nous avons des contraintes supplémentaires pour chaque type de conflits. Ces contraintes qui résultent de l'orientation du trafic se traduisent de la façon suivante :

Type 1 : conflit données-données

Ce type de conflit est considéré uniquement si VCi est plus près du puits que Ci et Bi plus près du puits que A.

Type 2 : conflit Données-Ack

Ce type de conflit est considéré uniquement si Cj est plus près du puits que VCj et Bi plus près du puits que A.

Type 3 : conflit Ack- Données

Ce type de conflit est considéré uniquement si VBj est plus près du puits que Bj et Bi plus près du puits que A.

Type 4 : conflit Ack-Ack

Ce type de conflit est considéré uniquement si Bk est plus près du puits que VBk et Bi plus près du puits que A.

Prenons l'exemple du type 2 (conflit Données-Ack). Nous avons considéré dans le scenario précédent qu'une collision ce produit au niveau du nœud Bi lorsque que le nœud A transmet une trame de données au nœud Bi pendant que le nœud Cj transmet un Ack au nœud VCj. Cependant, dans ce scenario nous tenons compte du fait que le trafic est orienté vers le puits. C'est-à-dire cette collision se produit uniquement lorsque les destinataires des trames sont plus proches du puits que les émetteurs. De ce fait, nous ajoutons une contrainte supplémentaire pour ce type de conflit qui consiste à vérifier si A et VCj sont plus près du puits que Bi et Cj respectivement. Notons que pour les autres types, le raisonnement est effectué de la même façon.

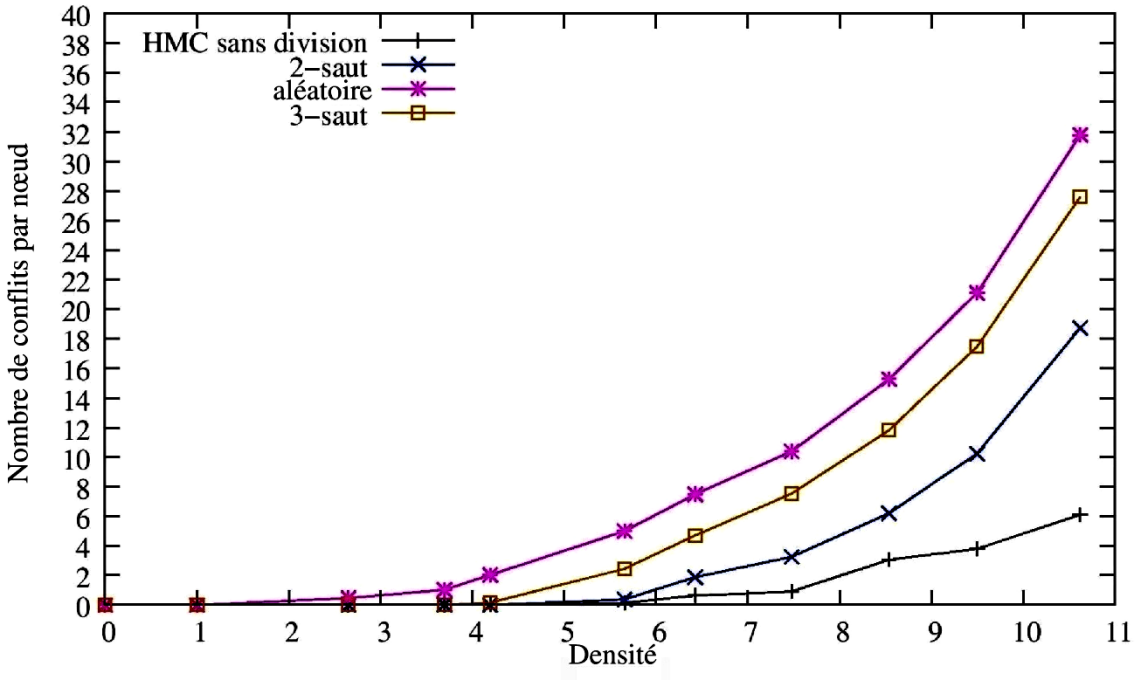


Figure 4.3 Influence de la densité sur le nombre de conflits lorsque le trafic est orienté vers le puits.

La figure 4.3 présente le nombre de conflits potentiels en fonction de la densité du réseau lorsque le trafic est orienté vers le puits. Ces résultats montrent que HMC-MAC est plus performant que les autres méthodes en termes de nombre de conflits. Pour les topologies avec des densité comprises entre 5 et 6, HMC-MAC sans division réduit le nombre de conflit d'environ 97%, 94% et 62% par rapport aux méthodes aléatoire, 3-sauts et 2-sauts respectivement. Pour des densités comprises entre 10 et 11, la réduction du nombre de conflits est d'environ 81%, 78% et 68% respectivement. Il est intéressant de noter qu'en comparant ce scenario avec le premier, nous obtenons beaucoup moins de conflits et ces derniers apparaissent à partir d'une densité plus élevée.

1.1.3. Troisième scenario : émissions alternées.

Afin de montrer l'amélioration apportée par la segmentation du réseau, nous considérons dans ce scenario que les émissions sont produites alternativement par les nœuds du premier groupe et du deuxième groupe en se basant sur la profondeur de chaque nœud. Les contraintes supplémentaires relatives à l'alternance de l'activité des deux groupes sont rajoutées par rapport aux deux premiers scenarios. Ces contraintes sont les suivantes :

Type 1 : conflit données-données

Ce type de conflit est considéré uniquement si A et Ci appartiennent au même groupe.

Type 2 : conflit Données-Ack

Ce type de conflit est considéré uniquement que si VCj et A appartiennent au même groupe.

Type 3 : conflit Ack-Données

Ce type de conflit est considéré uniquement que si Bj et A appartiennent au même groupe.

Type 4 : conflit Ack-Ack

Ce type de conflit est considéré uniquement que si VBk et A appartiennent au même groupe.

Prenons l'exemple du type 3 (conflit Ack-Données). Nous avons considéré dans le scenario précédent qu'une collision ce produit au niveau du nœud A lorsque :

- le nœud Bi transmet un Ack au nœud A pendant que le nœud Bj transmet une trame de données au nœud VBj,
- Bi et VBj sont plus près du puits que A et Bj respectivement.

Cependant, dans ce scenario nous tenons compte du fait que les émissions sont réalisées alternativement par les nœuds des deux groupes. C'est-à-dire cette collision se produit uniquement lorsque les émetteurs de trames appartiennent au même groupe. De ce fait, nous ajoutons une contrainte supplémentaire pour ce type de conflit qui consiste à vérifier si A et Bj appartiennent au même groupe de nœuds.

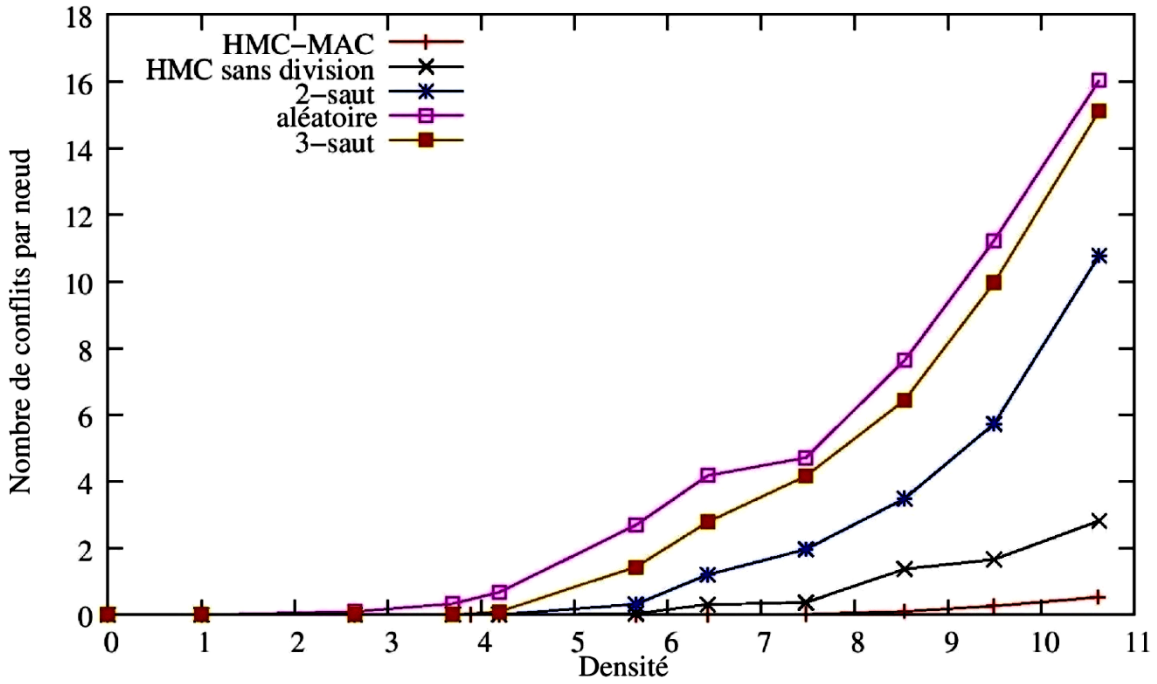


Figure 4.4 Influence de la densité sur le nombre de conflits avec des émissions sont réalisées alternativement par les nœuds du premier et du deuxième groupe.

La figure 4.4 présente le nombre de conflits en fonction de la densité lorsque les émissions sont réalisées alternativement entre les nœuds du groupe 1 et les nœuds du groupe 2 tel que cela a été défini dans le chapitre 3. Nous pouvons constater que la méthode HMC-MAC génère moins de conflits que celle sans division, cela est dû au fait que la segmentation du réseau nous a permis d'améliorer l'utilisation des canaux disponibles. En effet, avec HMC-MAC nous considérons deux groupes de nœuds possédant chacun tous les canaux disponibles, contrairement aux méthodes où les canaux sont partagés par tous les nœuds du réseau. Ceci revient d'une façon virtuelle et par effet de segmentation à doubler le nombre de canaux disponibles. Nous constatons également que HMC-MAC sans division génère moins de conflits que les autres méthodes puisqu'il prend en compte l'utilisation des canaux dans le voisinage des nœuds. En comparant ce scenario avec le deuxième scenario, nous obtenons moins de conflits et ces derniers apparaissent à partir de densités plus élevées (comme c'est le cas pour le deuxième scenario par rapport au premier). Notons que la méthode 2-sauts est plus performante que celle de 3-sauts à partir d'une densité égale à 4 et que la méthode 3-sauts est plus efficace que la méthode aléatoire.

1.2. Evaluation du taux d'interférence

Dans cette partie, la métrique utilisée est le taux d'interférence. En se basant sur l'étude des conséquences d'une allocation imparfaite des canaux effectuée dans la partie 1.1 du chapitre 3, nous constatons que la réutilisation du même canal dans un voisinage à 1-saut provoque 5 conflits tandis qu'à 2-sauts il provoque 2 conflits et un seul conflit lorsque le même canal est réutilisé à 3-sauts. De ce fait, nous allons associer à chaque cas une pondération qui reflète le risque de produire des conflits afin de calculer le taux d'interférence. Soit Ci le nombre des nœuds conflictuels à i-saut du nœud X (c'est-à-dire les nœuds exploitant le même canal que celui du nœud X) et soit N le nombre des nœuds dans le réseau. Les taux d'interférence jusqu'à 3-sauts (TI_3), 2-sauts (TI_2) et 1-saut (TI_1) sont calculés selon les formules suivantes :

$$TI_3 = \sum_{n=0}^{N-1} \frac{16 * C_1 + 10 * C_2 + 2 * C_3}{N}$$

$$TI_2 = \sum_{n=0}^{N-1} \frac{16 * C_1 + 10 * C_2}{N}$$

$$TI_1 = \sum_{n=0}^{N-1} \frac{16 * C_1}{N}$$

Dans ce qui suit, nous allons évaluer par simulation les taux d'interférence jusqu'à 3-sauts, 2-sauts et 1-saut. Les différents paramètres de simulation sont résumés dans le tableau 4.2 et les valeurs moyennes pour chaque densité sont représentées sur les différentes courbes.

Simulateur utilisé	NS-2
Dimensions du réseau	100*100 m^2
Débit	250kbps
Portée de communication	20 m
Nombre de nœuds	10-100
Nombre des interfaces du puits	1
Nombre des canaux disponibles	16
Nombre de topologies générés	5 topologies aléatoires
Taille de la file d'attente	50 paquets
Paramètre des topologies	Lm = 7, Rm = Cm = 5
Fenêtre d'observation	100 secondes
Nombre de répétitions N	10 répétitions pour chaque méthode
Modèle de propagation	*Two-ray ground*

Tableau 4.2 Paramètres de simulation utilisés pour l'évaluation du taux d'interférence.

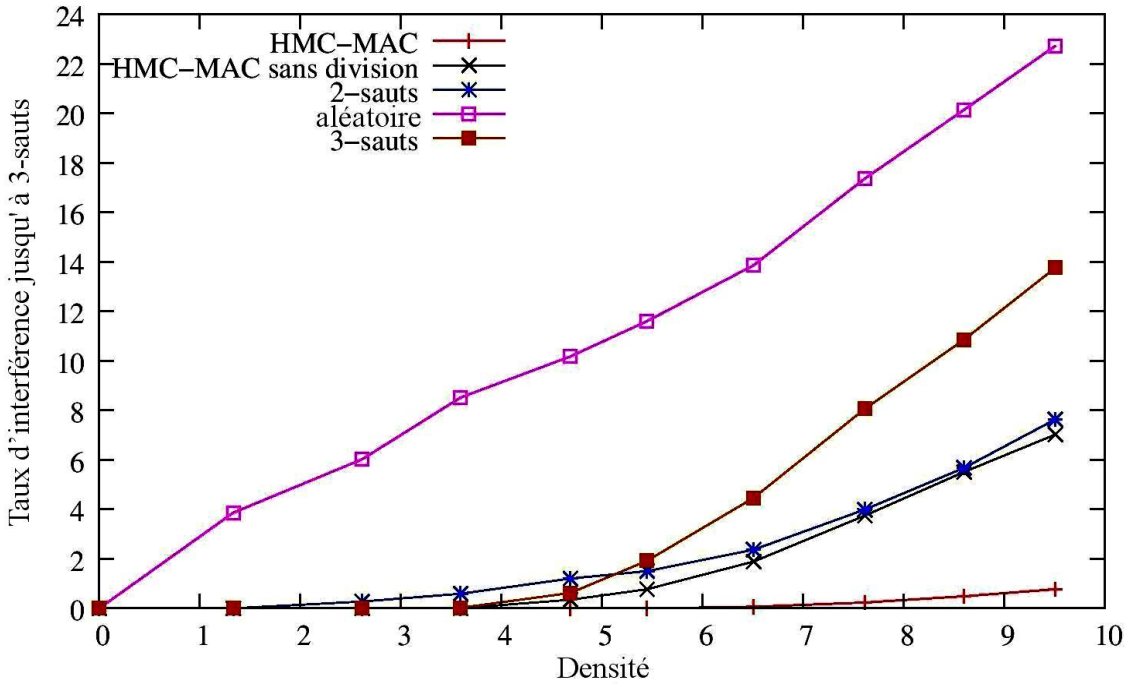


Figure 4.5 Taux d'interférence jusqu'à 3-sauts en fonction de la densité.

La figure 4.5 présente le taux d'interférence jusqu'à 3-sauts par nœud en fonction de la densité du réseau. Nous pouvons constater que le taux d'interférence augmente avec la densité. Ceci est attendu car quand la densité du réseau augmente, le nombre de canaux disponibles devient insuffisant pour servir chaque nœud sans risque d'interférences, un même canal est alors utilisé plusieurs fois dans le même voisinage. Ces résultats montrent l'efficacité de HMC-MAC par rapport aux autres méthodes. Par exemple, pour des densités comprises entre 7 et 8, HMC-MAC réduit le taux d'interférence d'environ 93%, 94%, 97% et 98% par rapport aux méthodes HMC sans division, 2-sauts, 3-sauts et aléatoire respectivement. Ce qui est intéressant à noter à travers ces résultats est la robustesse offerte par HMC-MAC face à l'augmentation de la densité. En effet, la segmentation du réseau permet une double utilisation des canaux, ce qui explique également l'efficacité de HMC-MAC par rapport à HMC-MAC sans division. Nous observons aussi selon cette métrique que HMC-MAC sans division est plus performant que la méthode 2-sauts, la méthode 3-sauts et la méthode aléatoire. Par exemple, pour des densités comprises entre 5 et 6, HMC-MAC réduit le taux d'interférence d'environ 93%, 59% et 48% par rapport aux méthodes aléatoire, 3-sauts et 2-sauts respectivement. Cela prouve l'efficacité de la méthode d'allocation des canaux effectuée par HMC-MAC par rapport aux autres méthodes.

Notons aussi que pour des densités supérieures à 5, la méthode 2-sauts est plus performante que celle de 3-sauts. Comme expliqué précédemment, cela est dû au fait que la méthode 3-sauts commence le choix aléatoire de canaux avant la méthode 2-sauts car le nombre des canaux non alloués dans le voisinage jusqu'à 3-sauts expire plus rapidement que celui dans le voisinage jusqu'à 2-sauts.

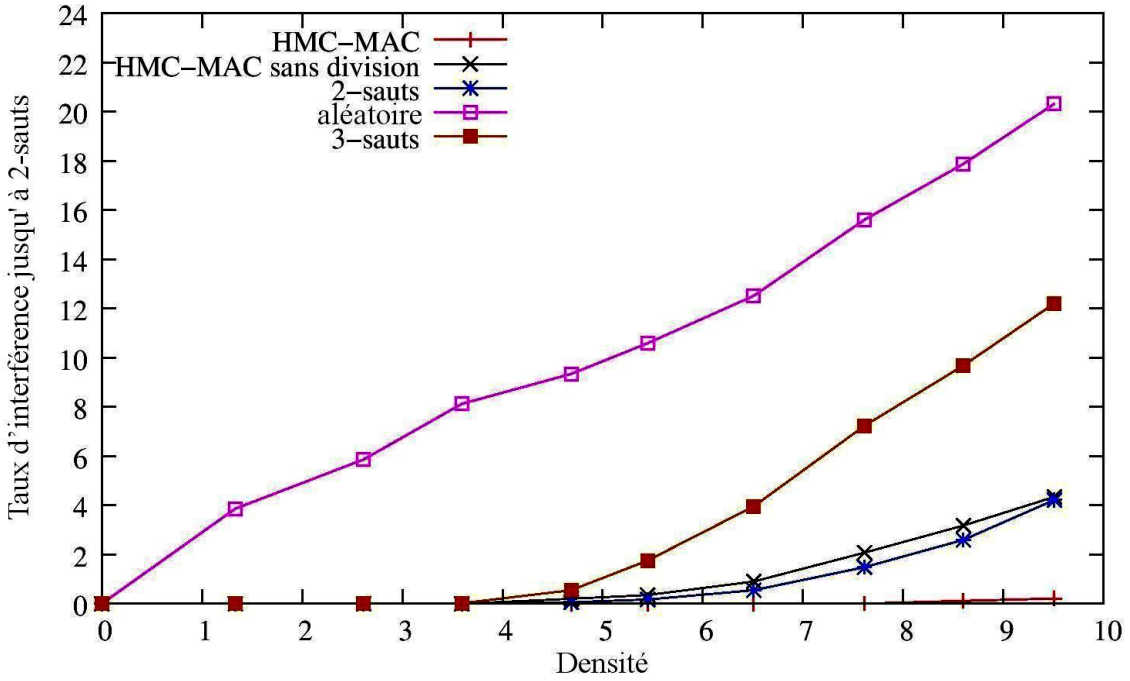


Figure 4.6 Taux d'interference jusqu'à 2-sauts en fonction de la densité.

La figure 4.6 montre le taux d'interférence jusqu'à 2-sauts en fonction de la densité du réseau. Nous observons que le taux d'interférence de HMC-MAC sans division est comparable à celui de la méthode 2-sauts. Nous observons aussi que HMC-MAC ne provoque pas d'interférences malgré l'augmentation de la densité.

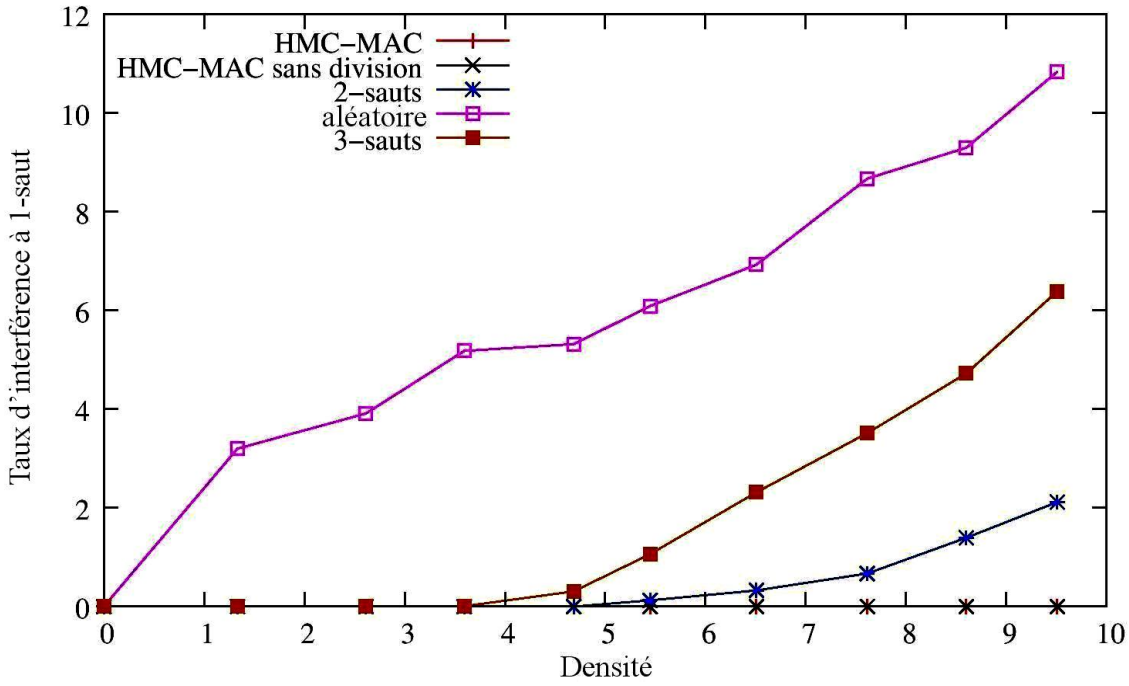


Figure 4.7 Taux d'interférence à 1-saut en fonction de la densité.

La figure 4.7 présente le taux d'interférence à 1-saut par nœud en fonction de la densité du réseau. Les résultats montrent que les méthodes HMC-MAC et HMC-MAC sans division évitent l'utilisation du même canal dans le voisinage à 1-saut d'où leur efficacité par rapport aux autres méthodes.

2. Évaluation des performances de HMC-MAC

Nous nous intéressons dans cette partie à l'évaluation des performances de notre proposition dans des scenarios où les nœuds génèrent des paquets dans le réseau à différentes fréquences. Nous considérons que le trafic est orienté vers le puits. Nous avons fixé la valeur de Rm à 3 afin de limiter le nombre de fils par nœuds, ce qui diminue la compétition entre les nœuds associés au même coordinateur. Pour chacune des topologies, nous avons fait varier le taux de génération de paquets de 1 à 20 paquets par seconde et par nœud. Chaque cas est répété N fois (N est donné sur les tableaux illustrant les paramètres de simulation), la moyenne pour toutes les topologies est représentée sur les différentes courbes. Notons que les résultats sont issus d'une fenêtre d'observation de 20 secondes une fois le régime de croisière atteint.

Nous comparons notre proposition avec les méthodes HMC-MAC sans division, 2-sauts et aléatoire définies dans la partie 1 de ce chapitre et la méthode cluster définie comme suit. Dans la méthode cluster, le réseau est segmenté en cluster dont chacun est formé d'une interface avec ses descendants. Nous attribuons un canal différent à chaque cluster. Il s'agit d'une méthode qui évite les interférences entre les clusters mais ne résout pas le problème de collision à l'intérieur de chaque cluster. Cette méthode est comparable à celle utilisée dans TMCP [15] mais avec un autre processus de construction des clusters. Dans cette méthode, tous les nœuds appartenant au même cluster utilisent le mécanisme CSMA/CA pour accéder au medium. Notons que le nombre des canaux exploités dans la méthode cluster est égal au nombre des interfaces de puits.

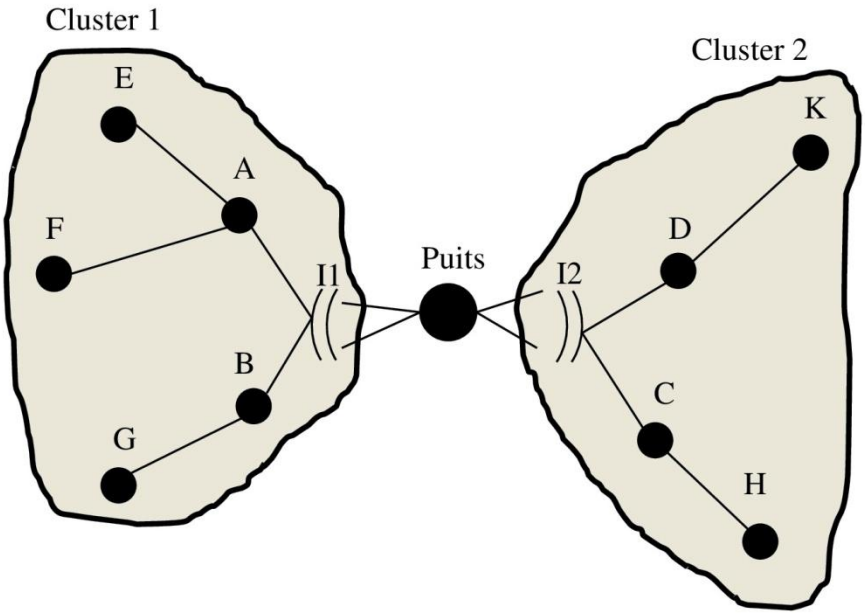


Figure 4.8 Un exemple de la méthode cluster avec un puits equipé de deux interfaces.

Afin de montrer les performances de notre proposition, nous évaluons les métriques suivantes. Notons que nous distinguons le nombre de paquets générés à la source du nombre de paquets émis sur le réseau.

- Débit agrégé : c'est le nombre des paquets reçus par le puits par seconde.
- Taux de réception de paquets : c'est le rapport du nombre de paquets reçus par le puits sur le nombre des paquets envoyés par les sources.
- Taux de débordement de files d'attente : c'est le rapport du nombre de paquets perdus dus aux débordements de files d'attente sur le nombre des paquets générés dans le réseau.
- Nombre de collisions : c'est le nombre de transmissions point-à-point non acquittées par seconde.

- Nombre de paquets perdus dus aux collisions : c'est le nombre de paquets supprimés après quatre tentatives d'envoi sans succès pour accéder au medium comme c'est le cas dans le mécanisme CSMA / CA de la norme IEEE 802.15.4.
- Nombre de paquets reçus par tous les nœuds : ce nombre nous permet d'avoir une vision sur la charge du réseau au niveau de la couche MAC.

Dans HMC-MAC, nous avons considéré que la durée de la période d'échange de données [T1; T2] est de 10 secondes. Cette période est découpée en 40 parties dont chacune est divisée en 2 intervalles de temps. Ainsi, chaque intervalle de temps vaut 125 ms. Notons que les nœuds génèrent les paquets d'une façon désynchronisée. Dans ce qui suit, nous allons étudier le comportement de notre protocole par rapport aux autres dans des scenarios différents.

2.1. Puits mono-interface

Dans ces simulations, nous évaluons les performances de HMC-MAC en considérant que le puits est équipé d'une seule interface. Le tableau 4.3 résume les paramètres de simulation.

Simulateur utilisé	NS-2
Dimensions du réseau	100*100 m^2
Débit	250kbps
Portée de communication	20 m
Nombre de nœuds	49
Nombre des interfaces du puits	1
Nombre des canaux disponibles	16
Génération de trafic	Trafic périodique
Nombre de topologies générés	10 topologies aléatoires
Taille de la file d'attente	200
Paramètre des topologies	Lm = 7, Rm = Cm = 3
Taux de génération de paquets	1-20 paquets par seconde et par nœud
Fenêtre d'observation	20 secondes
Nombre de répétitions N	5 répétitions pour tous les protocoles
Modèle de propagation	*Two-ray ground*
Longueur des paquets	50 octets

Tableau 4.3 Paramètres de simulation utilisés pour l'évaluation des performances de HMC-MAC avec un puits mono-interface.

2.1.1. Débit agrégé

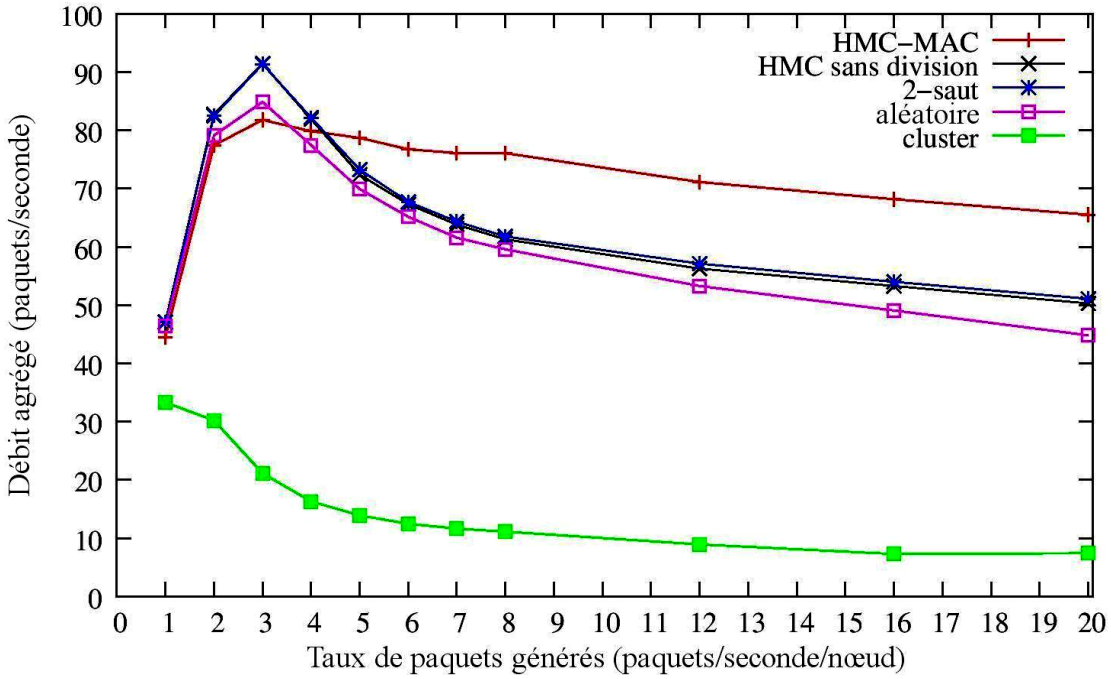


Figure 4.9 Nombre de paquets reçus par seconde par le puits mono-interface en fonction de la charge produite par chacun des noeuds.

La figure 4.9 illustre le débit en fonction de la charge produite[9] par chacun des nœuds. Nous constatons que le débit augmente avec le taux de génération de paquets avant d'atteindre la saturation. Cette saturation est atteinte à partir d'un taux de génération de 3 paquets par seconde et par nœud pour HMC-MAC et à partir de 5 paquets par seconde et par nœud pour les autres protocoles. Nous pouvons constater aussi que HMC-MAC se porte mieux que les autres méthodes sous forte charge. Par exemple, avec un taux de génération de 12 paquets par seconde et par nœud, HMC-MAC augmente le débit d'environ 26.3%, 24,3%, 33.4% et 696% par rapport aux méthodes HMC-MAC sans division, 2-sauts, aléatoire et cluster respectivement. Ces résultats de simulation nous permettent également de constater que toutes les méthodes multi-canal sont plus performantes que la méthode cluster qui utilise un seul canal, ce qui prouve l'intérêt de l'utilisation de plusieurs canaux. Afin d'expliquer les causes de la saturation du débit, nous évaluons dans ce qui suit le taux de débordement de files d'attente.

[9] La charge produite représente le taux de génération de paquets par seconde.

2.1.2. Taux de débordement de files d'attente

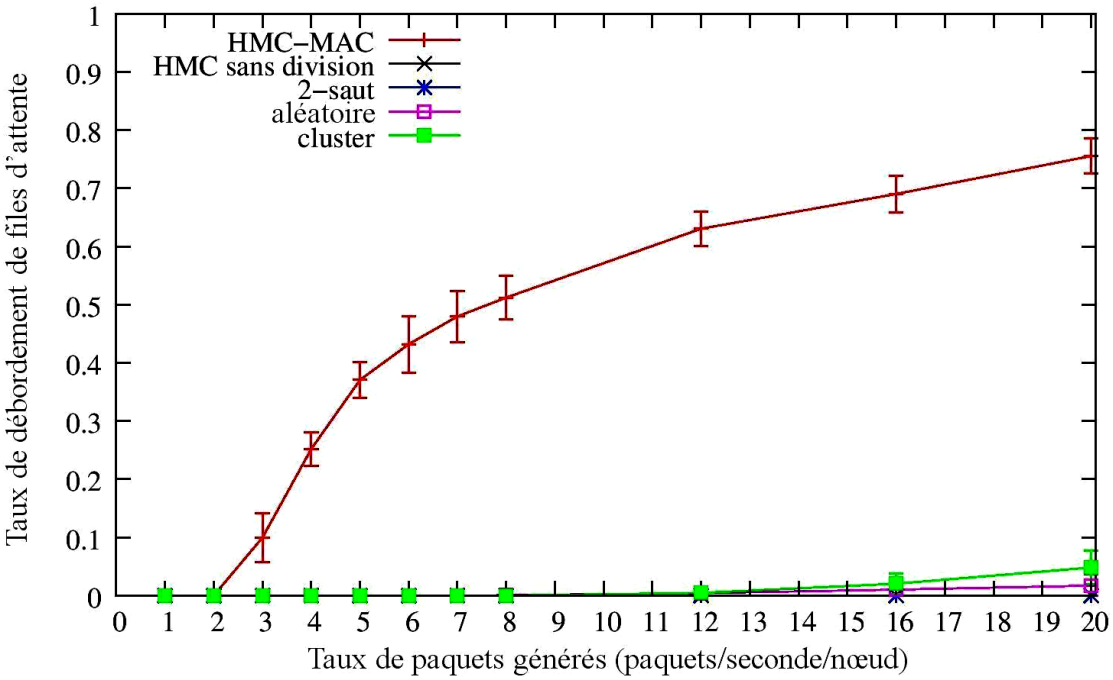


Figure 4.10 Taux de débordement de files d'attente des nœuds du réseau en fonction de la charge produite par chacun des nœuds.

La figure 4.10 montre le taux de paquets perdus à cause du débordement des files d'attente en fonction du taux de génération des paquets par nœud et par seconde. Cette étude est menée avec une taille de files d'attente de 200 paquets dont la longueur de chaque paquet est de 50 octets. Pour HMC-MAC, nous constatons que lorsque le nombre de paquets générés augmente, le pourcentage de pertes de paquets augmente jusqu'à 80%. Cela peut être expliqué par le fait que l'accumulation des paquets dans HMC-MAC provoque le débordement des files d'attente des nœuds autour de puits. En revanche, les autres protocoles ne souffrent pas de problème du débordement des files d'attente. En effet, ces protocoles perdent un nombre important de paquets durant leur acheminement dû essentiellement aux problèmes de collisions. Il est intéressant de constater que pour HMC-MAC les pertes dues aux débordements de files d'attente provoquent la saturation du débit (voir figure 4.9). En effet, les pertes de paquets apparaissent en même temps que le phénomène de saturation est obtenu pour le débit. Pour voir les conséquences de débordement de files d'attente sur les performances du réseau, nous évaluons par la suite le taux de réception de paquets.

2.1.3. Taux de réception

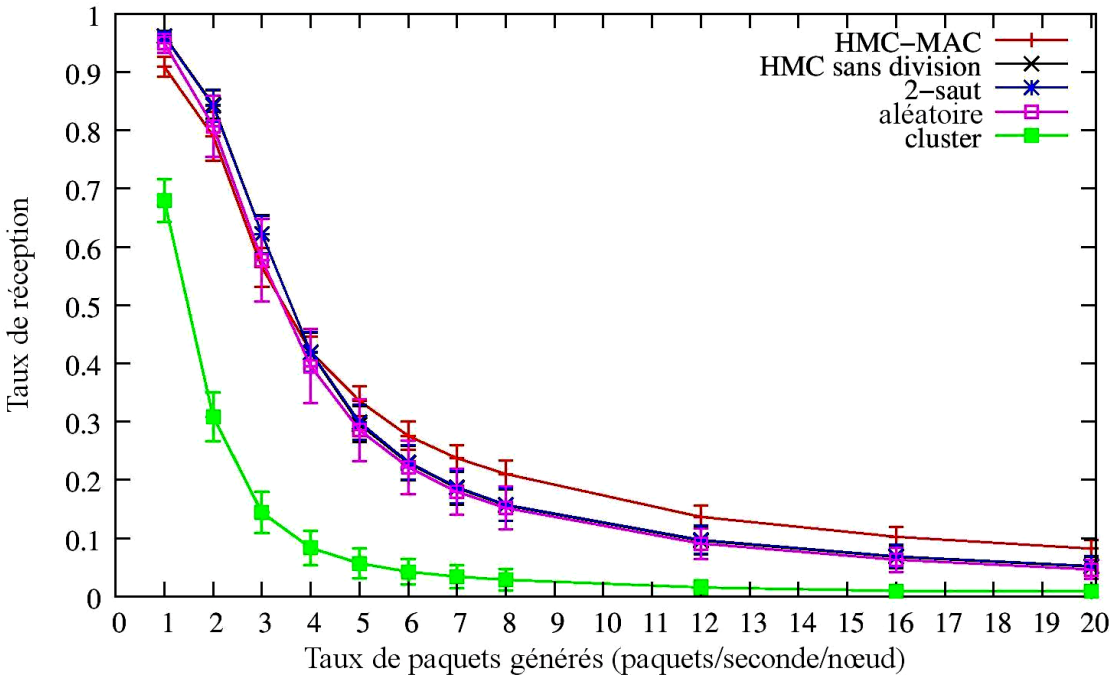


Figure 4.11 Taux de réception du puits mono-interface en fonction de la charge produite par chacun des nœuds.

La figure 4.11 illustre le taux de réception en fonction du taux de génération de paquets par nœud et par seconde. Pour tous les protocoles, nous constatons que le taux de réception diminue significativement avec le taux de génération des paquets. Pour HMC-MAC, cette forte diminution est due au problème des débordements de files d'attente (voir figure 4.10) alors que pour les autres protocoles cette diminution est essentiellement due au problème des collisions et des interférences entre les nœuds du réseau. Notons que malgré le problème des débordements de files d'attente, HMC-MAC donne de meilleures performances sous forte charge que celles des autres protocoles considérés dans ce travail. Dans le but de tester les performances de HMC-MAC au niveau de la couche MAC, nous évaluons par la suite le nombre de paquets reçus par l'ensemble des nœuds.

2.1.4. Nombre de paquets reçus par l'ensemble des nœuds

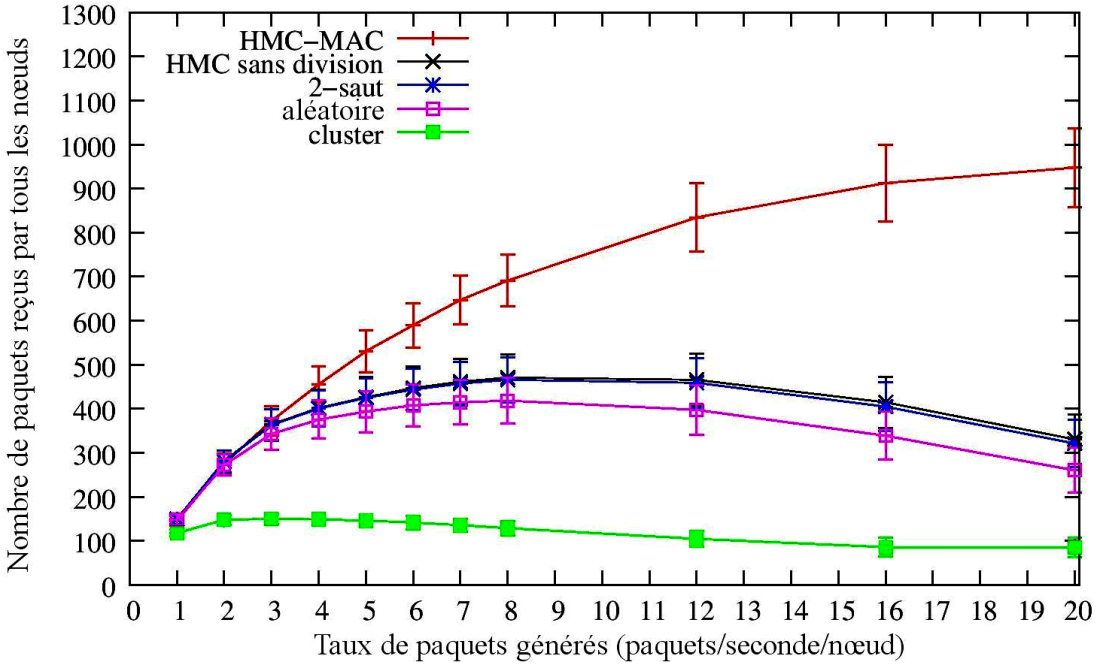


Figure 4.12 Nombre de paquets reçus par l'ensemble des nœuds du réseau par second.

La figure 4.12 illustre le nombre de paquets reçus par tous les nœuds du réseau en fonction de la charge produite par chacun des nœuds. Rappelons que la longueur de chaque paquet est de 50 octets. A faible charge (moins de 3 paquets par seconde), le nombre de paquets reçus par tous les nœuds est similaire pour toutes les méthodes comparées à l'exception de la méthode cluster qui utilise un seul canal. Lorsque le taux de génération de paquets augmente, HMC-MAC améliore significativement le nombre de paquets reçus par tous les nœuds du réseau par rapport aux autres méthodes. Par exemple, avec un taux de génération de 12 paquets par seconde et par nœud, HMC-MAC augmente le nombre de paquets reçus par tous les nœuds d'environ 79%, 81,8%, 109% et 699% par rapport aux méthodes HMC-MAC sans division, 2-sauts, aléatoire et cluster respectivement. Nous pouvons constater que pour les autres protocoles le nombre de paquets reçus par tous les nœuds atteint la saturation à partir de 6 paquets par seconde et par nœud. Cette saturation est provoquée par le fait que le nombre de collisions augmente de façon significative pour les protocoles qui n'appliquent pas la segmentation du réseau, comme nous allons montrer par la suite.

2.1.5. Nombre de collisions

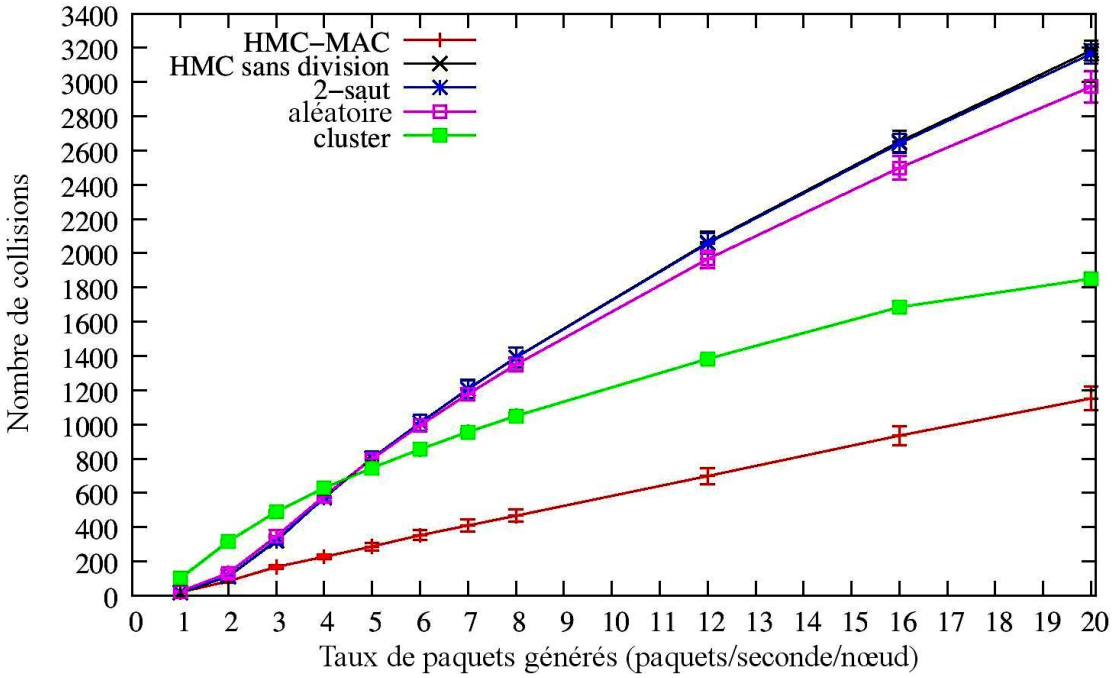


Figure 4.13 Nombre de collisions par seconde dans le réseau en fonction de la charge produite par chacun des nœuds.

La figure 4.13 illustre le nombre de collisions en fonction du taux de génération des paquets par nœud et par seconde. Le nombre de collisions augmente avec le taux de génération de paquets pour tous les protocoles. Cela est dû au fait qu'avec l'augmentation du taux de génération de paquets, la compétition entre les nœuds utilisant le même canal augmente. Ces résultats montrent un nombre important de collisions pour les protocoles qui n'appliquent pas un mécanisme de rendez-vous, ils souffrent donc du problème du nœud sourd (voir partie 1.2 du chapitre 3). Contrairement à ces protocoles, HMC-MAC évite le problème du nœud sourd en définissant pour chaque groupe des intervalles d'émission et de réception, ce qui réduit le nombre de collisions. Prenons l'exemple de taux de génération de 12 paquets par seconde et par nœud, HMC-MAC réduit le nombre de collisions d'environ 66%, 66%, 64% et 49% par rapport aux méthodes HMC-MAC sans division, 2-sauts, aléatoire et cluster respectivement. Rappelons que les collisions provoquent des pertes de paquets qui sont évaluées dans la suite.

2.1.6. Nombre de paquets perdus dus aux collisions

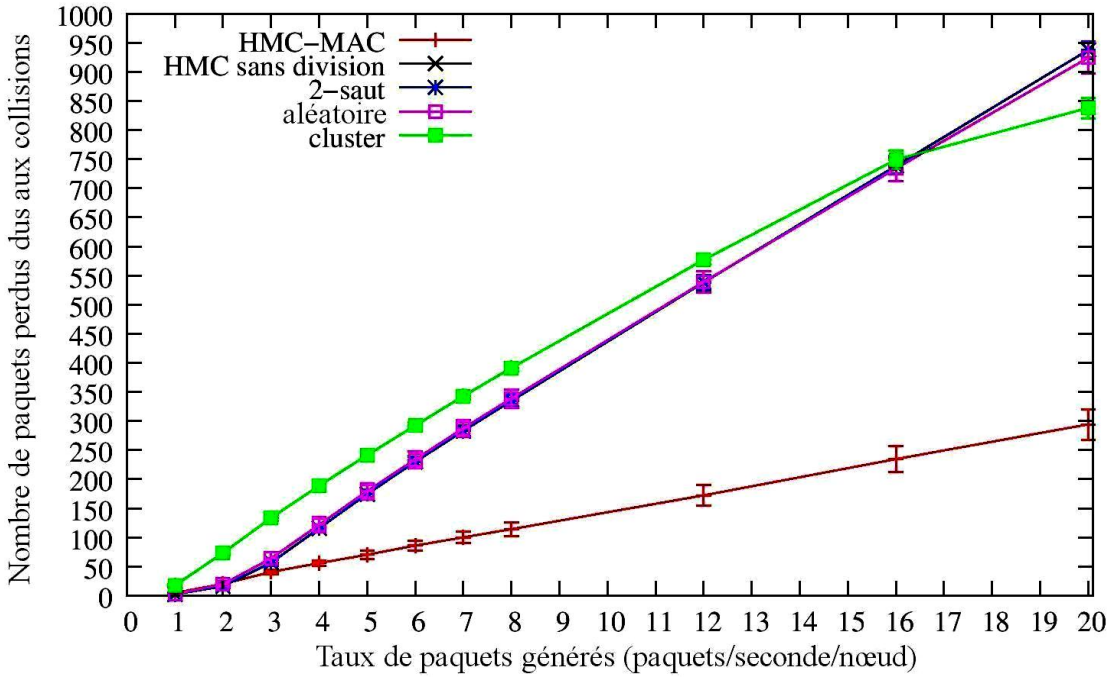


Figure 4.14 Le nombre de paquets perdus par seconde dus aux collisions.

La figure 4.14 montre le nombre de paquets perdus dans le réseau dus aux collisions en fonction du taux de génération de paquets par nœud et par seconde. Comme attendu, le nombre de paquets perdus augmente avec le nombre de paquets générés. Nous constatons que HMC-MAC réduit significativement les pertes de paquets par rapport aux autres protocoles, cela est dû au fait que HMC-MAC génère moins de collisions comme nous l'avons montré dans la figure 4.13. Cette dernière illustre le nombre de collisions en fonction du taux de génération de paquets de HMC-MAC par rapport aux autres méthodes. Prenons l'exemple de taux de génération de 12 paquets par seconde et par nœud, HMC-MAC réduit le nombre de paquets perdus dus aux collisions d'environ 67,9%, 67,9%, 68% et 70% par rapport aux méthodes HMC sans division, 2-sauts, aléatoire et cluster respectivement.

Les différentes métriques évaluées ci-dessus nous mènent à équiper le puits de plusieurs interfaces, ce choix est justifié dans ce qui suit.

2.1.7. Nombre de paquets reçus par les nœuds associés au puits

Afin de mieux comprendre la raison pour laquelle le débit dans HMC-MAC atteint rapidement la saturation, nous allons évaluer le nombre de paquets reçus par les fils du puits.

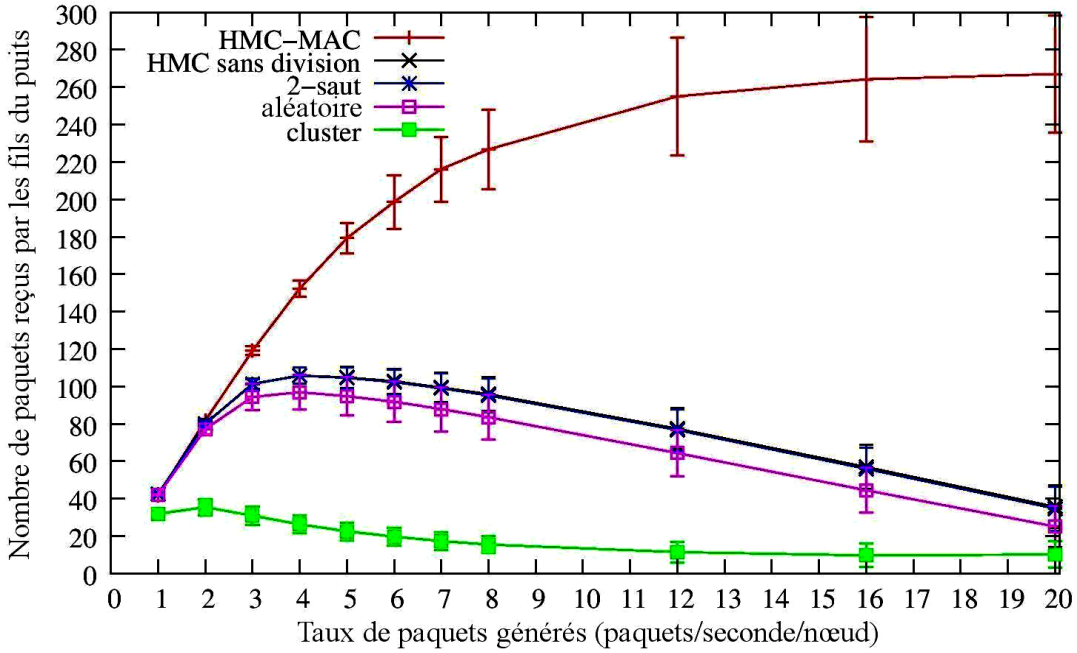


Figure 4.15 Nombre de paquets reçus par les fils du puits.

La figure 4.15 montre le nombre de paquets reçus par les fils du puits en fonction de la charge produite par chacun des nœuds. HMC-MAC augmente considérablement le nombre de paquets reçus par les fils du puits par rapport aux autres méthodes. Nous pouvons constater qu'avec HMC-MAC le nombre de paquets reçus par les fils du puits augment avec le taux de génération de paquets car n'a pas encore atteint son maximum dans la plage de trafic considérée, alors qu'avec les autres méthodes la saturation est atteinte rapidement à partir de 5 paquets par seconde et par nœud (le même point de saturation pour le débit). Nous pouvons conclure grâce à ces résultats que pour HMC-MAC, il y a un effet d'accumulation de paquets dans les files d'attentes des nœuds associés au puits. En effet, le puits mono-interface n'a pas la capacité d'absorber tous les paquets provenant de ses fils sous forte charge. La solution est donc d'augmenter cette capacité en utilisant un puits équipé de plusieurs interfaces.

Dans ce qui suit, nos allons prouver l'intérêt d'utiliser plusieurs interfaces au puits et évaluer l'impact de type de profile de trafic sur les performances des protocoles comparés.

2.2. Evaluation des performances de HMC avec deux types de profils de trafic

Dans ce scenario, nous considérons deux types de profils de trafic: trafic périodique et trafic en rafale. La figure 4.16 illustre le comportement de ces deux profils. Dans le cas du trafic périodique, les nœuds génèrent périodiquement des trames de données de façon similaire à un trafic CBR *(Constant Bit Rate)*. Dans le cas de trafic en rafale, les nœuds génèrent des paquets en doublant le taux de génération de trafic périodique dans la première seconde, puis suspendent cette génération de trafic durant la seconde suivante. Ce qui conduit à une augmentation de la contention entre les nœuds durant la première seconde. Ce type de trafic peut être représenté par une génération d'alarme par exemple. Notons qu'avec ces deux profils de génération de trafic, nous obtenons le même nombre de paquets générés. Dans ce qui suit, nous allons étudier l'impact de ces deux types de profils sur les performances des protocoles comparés.

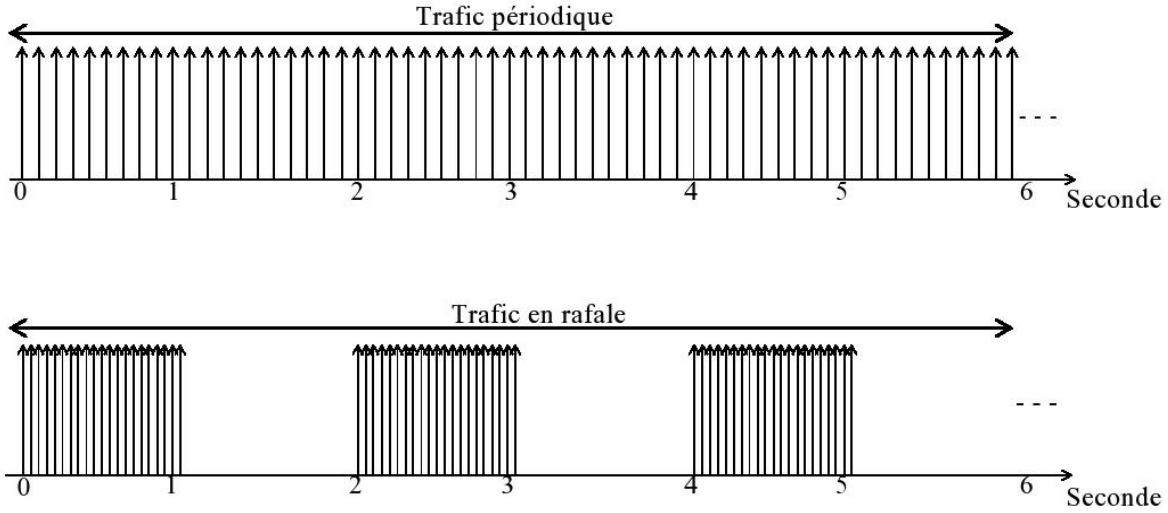


Figure 4.16 Définition des deux types de trafic : périodique et en rafale.

Dans ces simulations, nous avons considéré que le puits possède uniquement 3 interfaces dont chacune possède 2 fils. Notons que 3 interfaces n'est qu'un cas d'étude. Nous avons limité le nombre de fils par interface à 2 afin d'éviter la compétition entre les nœuds associés à chaque interface et par conséquent d'éviter les collisions au niveau des interfaces du puits. Le tableau 4.4 présente les paramètres de simulation.

Simulateur utilisé	NS-2
Dimensions du réseau	100*100 m^2
Débit	250kbps
Portée de communication	20 m
Nombre de nœuds	49
Nombre des interfaces du puits	3
Nombre des canaux disponibles	16
Génération de trafic	Trafic périodique et trafic en rafale
Nombre de topologies générés	10 topologies aléatoires
Taille de la file d'attente	200 paquets
Paramètre des topologies	Lm = 7, Rm = Cm = 3
Taux de génération de paquets	1-20 paquets par seconde par nœud
Fenêtre d'observation	20 secondes
Nombre de répétitions N	5 répétitions pour la méthode aléatoire et 1 répétition pour les autres méthodes en utilisant le premier canal libre.
Modèle de propagation	*Two-ray ground*
Longueur des paquets	50 octets

Tableau 4.4 Paramètres de simulation utilisés pour l'évaluation des performances de HMC-MAC avec deux types de profils de trafic.

2.2.1. Débit agrégé

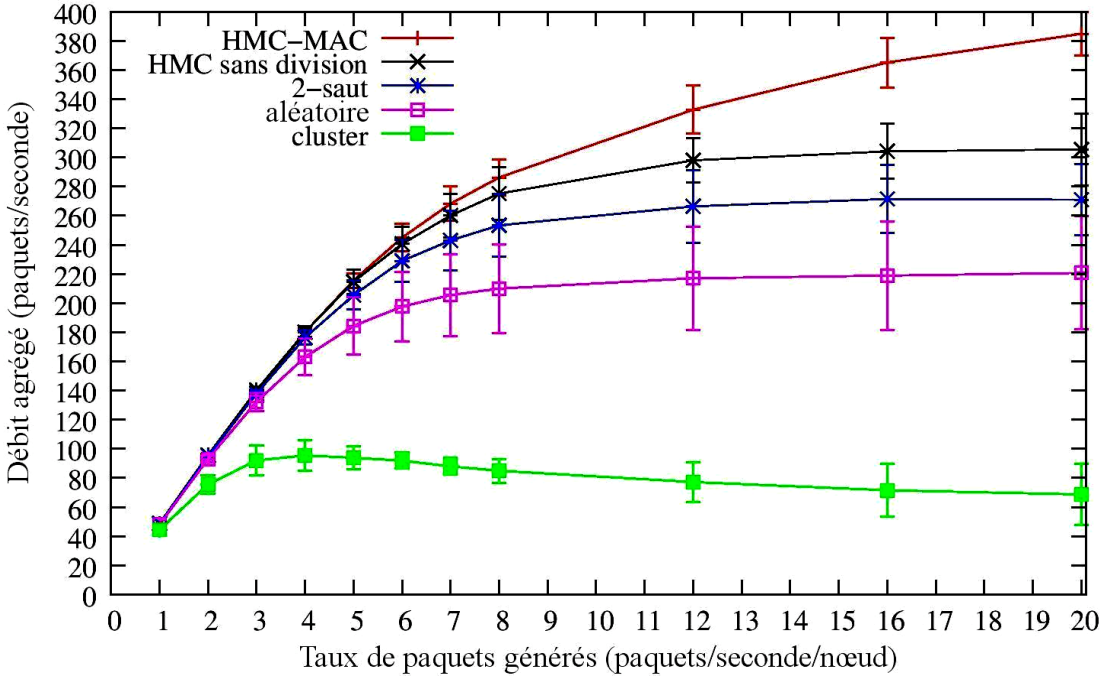


Figure 4.17 Comportement du débit agrégé pour le trafic périodique.

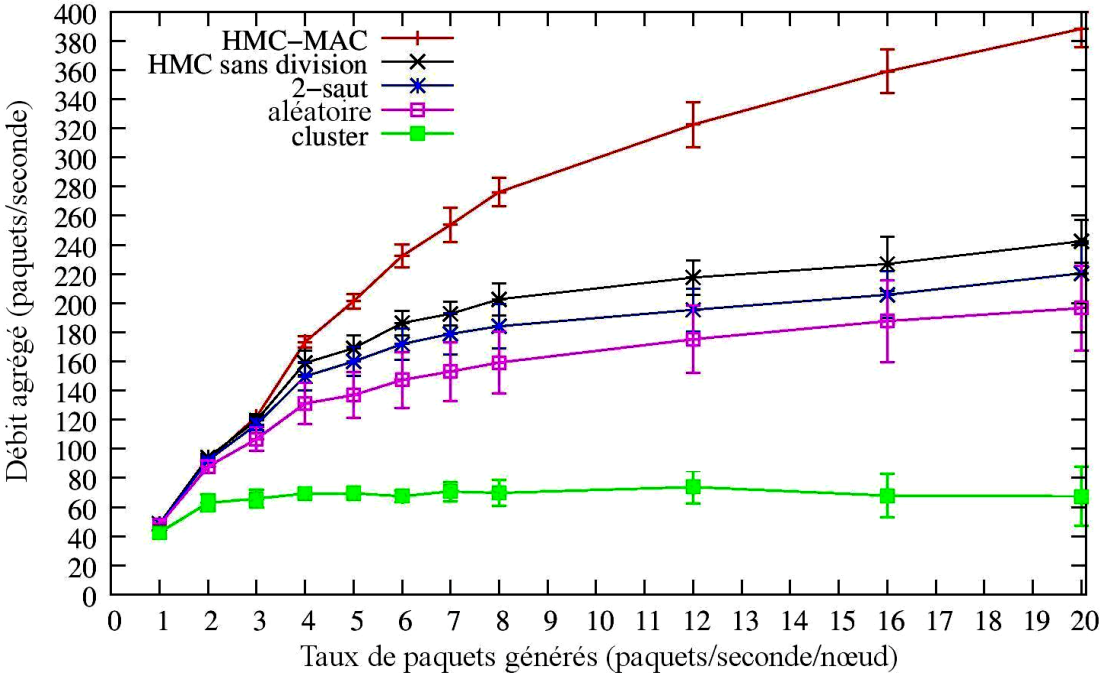


Figure 4.18 Comportement du débit agrégé pour le trafic en rafale.

Le débit agrégé est défini par le nombre de paquets reçus par le puits par seconde. Les figures 4.17 et 4.18 présentent le débit agrégé en fonction du taux de génération de paquets par nœud et par seconde. Pour HMC-MAC, nous constatons que le débit augmente avec le taux de génération de paquets sans atteindre la saturation pour les deux profils de trafic, sur cette plage de trafic explorée. En revanche, pour les autres protocoles, le débit atteint la saturation dans le cas de trafic périodique (voir figure 4.17) et augmente lentement dans le cas de trafic en rafale (figure 4.18) lorsque le taux de génération dépasse 8 paquets par seconde et par nœud. Cela est dû au nombre élevé des collisions et au fait que des paquets sont perdus. Pour la méthode cluster, le débit est toujours inférieure à 100 paquets par seconde et atteint la saturation à partir de 4 paquets par seconde par nœud. Cela s'explique par le fait que la méthode cluster utilise uniquement 3 canaux.

Nous pouvons aussi constater que les performances de HMC-MAC sont proches de celles obtenues avec HMC-MAC sans division sous faible charge, plus précisément, lorsque le taux de génération de paquets est inférieur à 6 et 3 paquets par seconde et par nœud pour le trafic périodique et en rafale respectivement. Cependant, HMC-MAC augmente significativement le débit par rapport aux autres protocoles sous forte charge de trafic. Nous retrouvons ici un argumentaire déjà développé concernant ce bon comportement. HMC-MAC offre 16 canaux disponibles pour chaque groupe des nœuds alors que les autres méthodes offrent 16 canaux pour tous les nœuds du réseau. De plus, la segmentation du réseau en définissant les récepteurs et les émetteurs durant chaque intervalle permet à HMC-MAC d'éviter le problème du nœud sourd d'où son efficacité par rapport aux autres méthodes.

Il est intéressant de remarquer que HMC-MAC améliore plus significativement le débit pour un trafic en rafale par rapport à un trafic périodique. Prenons l'exemple de taux de génération de 12 paquets par seconde et par nœud. Dans le cas d'une génération de trafic périodique, HMC-MAC augmente le débit d'environ 76,8%, 34,7%, 20% et 10,4% par rapport aux méthodes cluster, aléatoire, 2-sauts et HMC-MAC sans division respectivement. Cependant dans le cas d'une génération du trafic en rafale, l'augmentation du débit est d'environ 77%, 45,7%, 39,4% et 32,5% respectivement. Cela est dû au fait que la transmission d'un trafic en rafale crée des vagues de surcharges qui subissent beaucoup de pertes avec les protocoles qui ne segmentent pas le réseau. En revanche, HMC-MAC résiste mieux et subit peu de pertes supplémentaires à celles d'un trafic périodique.

Nous pouvons également noter que pour les deux types de profils de trafic, HMC-MAC sans division se porte mieux que la méthode 2-sauts, cela est dû au fait que la méthode 2-sauts ne prend pas en compte les canaux utilisés dans son voisinage à 3-sauts. Notons aussi que la méthode 2-sauts se porte mieux que la méthode aléatoire, cela est attendu car la méthode aléatoire ne prend pas en considération les canaux utilisés dans le voisinage des nœuds.

2.2.2. Taux de débordement des files d'attente

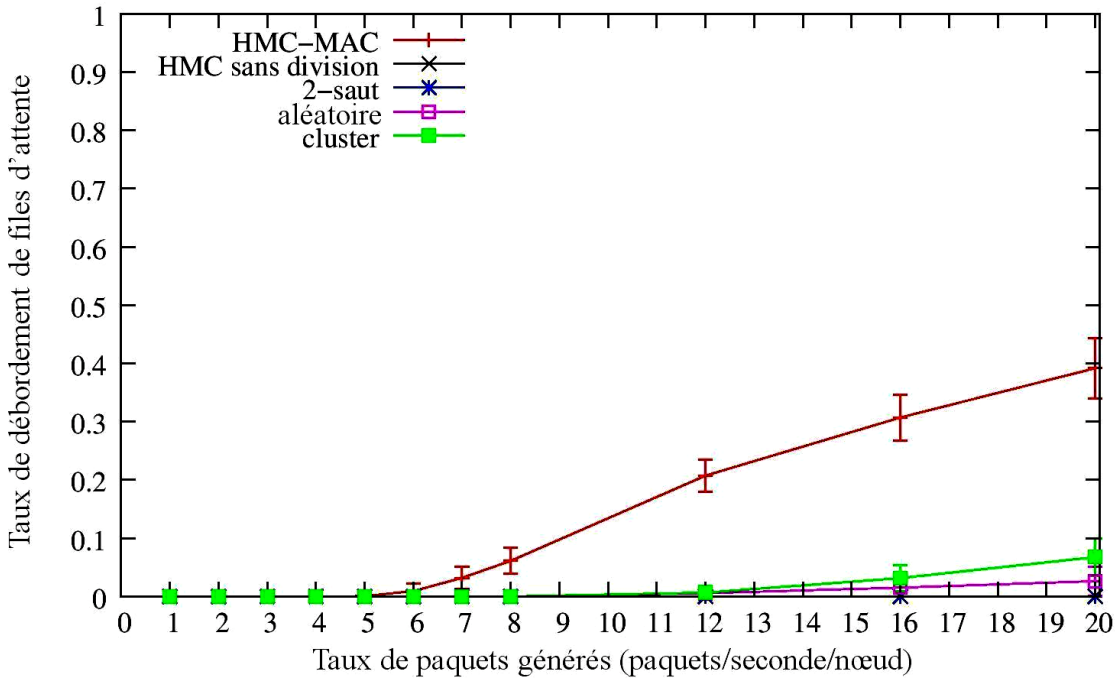


Figure 4.19 Taux de débordement des files d'attente pour le trafic périodique.

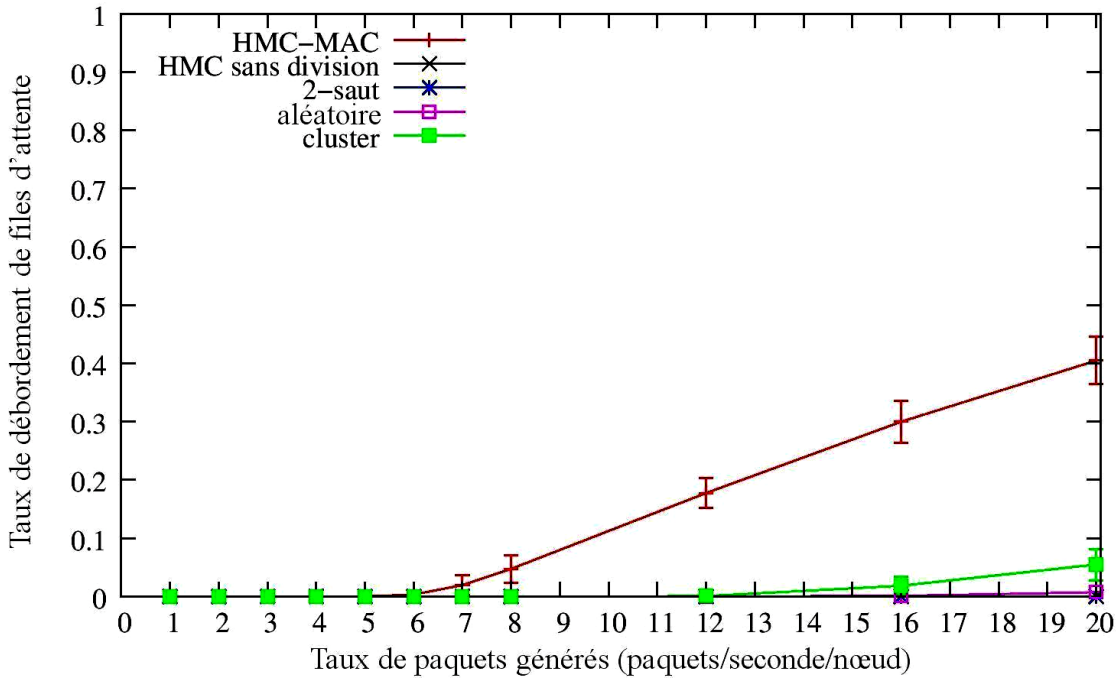


Figure 4.20 Taux de débordement des files d'attente pour le trafic en rafale.

Les figures 4.19 et 4.20 montrent les résultats en termes de paquets perdus dus aux débordements des files d'attente. Nous observons un taux élevé de débordement des files d'attente pour HMC-MAC dans les deux profils de trafic. En effet, HMC-MAC est capable de mieux gérer les pertes de paquets en cours d'acheminement comparés aux autres protocoles. Ce fait conduit à une accumulation de paquets dans les files d'attente des nœuds qui sont proches du puits surtout quand ils ont un grand nombre de descendants. Ces nœuds sont donc incapables de transmettre les paquets accumulés et par conséquent souffrent d'un taux élevé de débordements de files d'attente.

2.2.3. Taux de réception

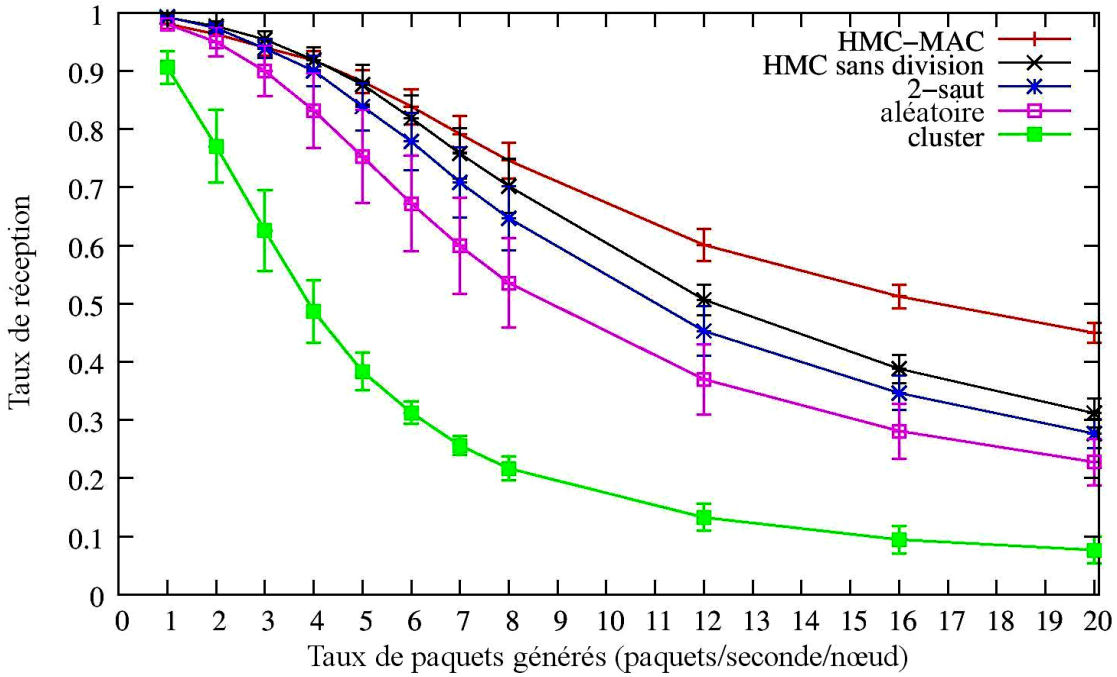


Figure 4.21 Taux de réception au niveau du puits pour le trafic continu.

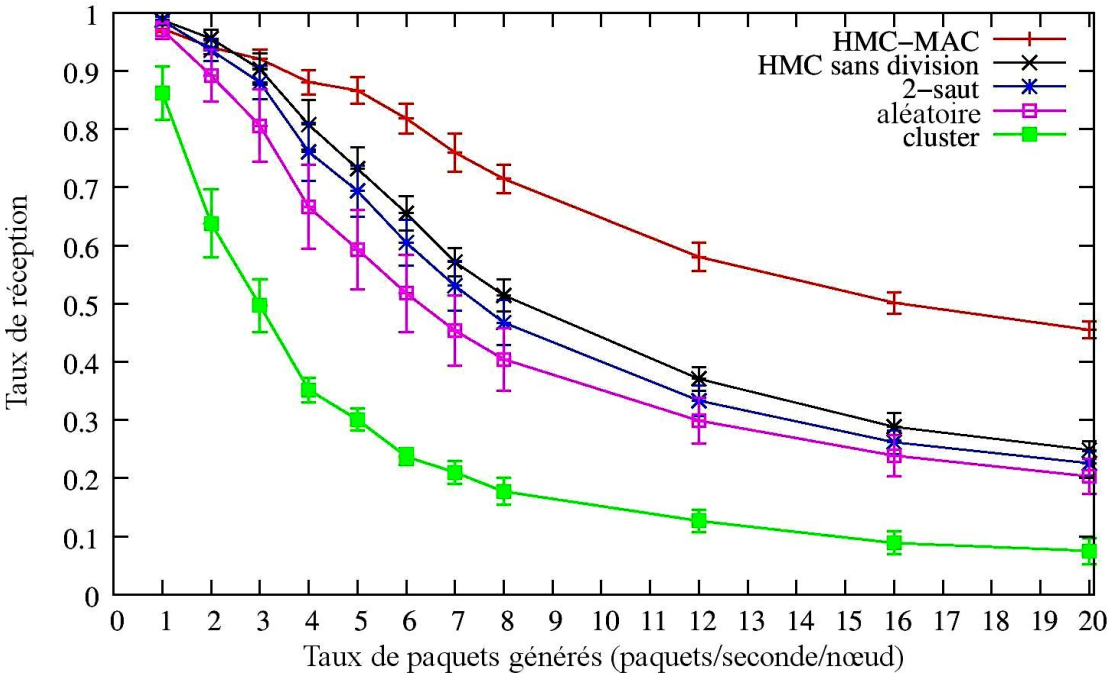


Figure 4.22 Taux de réception au niveau du puits pour le trafic en rafale.

Les figures 4.21 et 4.22 montrent le taux de réception de paquets au niveau du puits en fonction de la charge produite par chacun des nœuds. De façon similaire au débit, HMC-MAC offre le meilleur taux de réception par rapport aux autres protocoles sous forte charge et dans le cas d'un trafic en rafale.

2.2.4. Nombre de paquets reçus par l'ensemble des nœuds

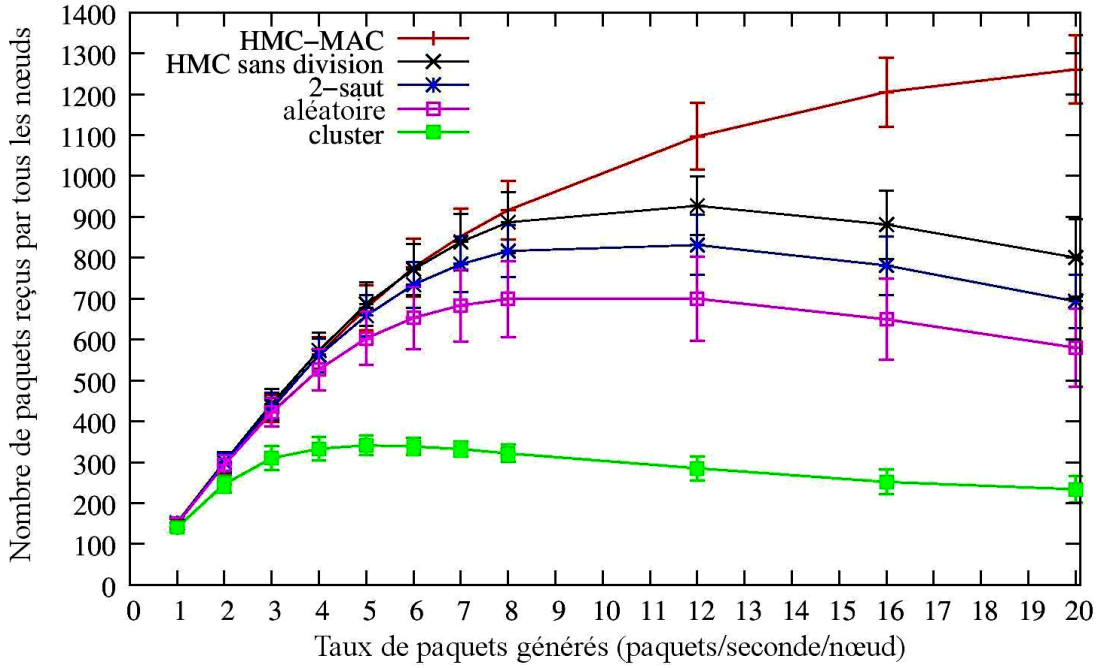


Figure 4.23 Nombre de paquets reçus par tous les nœuds pour le trafic périodique.

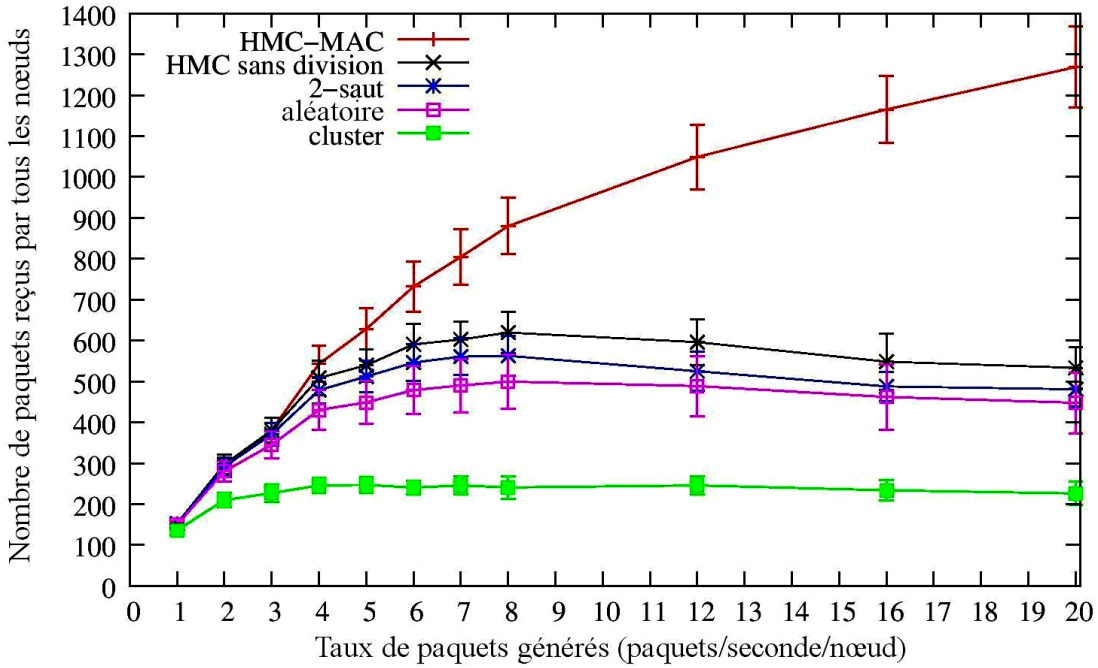


Figure 4.24 Nombre de paquets reçus par tous les nœuds pour le trafic en rafale.

Les figures 4.23 et 4.24 montrent le nombre de paquets reçus par tous les nœuds du réseau en fonction de la charge produite par chacun des nœuds. Pour HMC-MAC, le nombre de paquets reçus par tous les nœuds augmente avec le nombre de paquets générés selon les deux profils de trafic. En revanche, le nombre de paquets reçus par tous les nœuds pour les autres protocoles atteint la saturation à partir de 8 paquets par seconde et par nœud à la fois pour le trafic périodique et le trafic en rafale. Il est intéressant de noter que ce point de saturation est le même que pour le débit pour les deux types de profils de trafic (voir figure 4.17 et figure 4.18). Comme expliqué précédemment, cette saturation est due au nombre important de collisions.

Comme pour le débit, HMC-MAC améliore d'une façon significative le nombre de paquets reçus par tous les nœuds pour le trafic en rafale par rapport à celui correspondant au trafic périodique. Prenons l'exemple d'un taux de génération de 12 paquets par seconde et par nœud. Dans le cas d'une génération de trafic périodique, HMC-MAC augmente le nombre de paquets reçus par tous les nœuds d'environ 284,5%, 56,6%, 31.8% et 18,2% par rapport aux méthodes cluster, aléatoire, 2-sauts et HMC-MAC sans segmentation respectivement. Dans le cas d'un trafic en rafale, cette augmentation est d'environ 326.5%, 114.5%, 99,6% et 75,8% respectivement.

2.2.5. Nombre de collisions

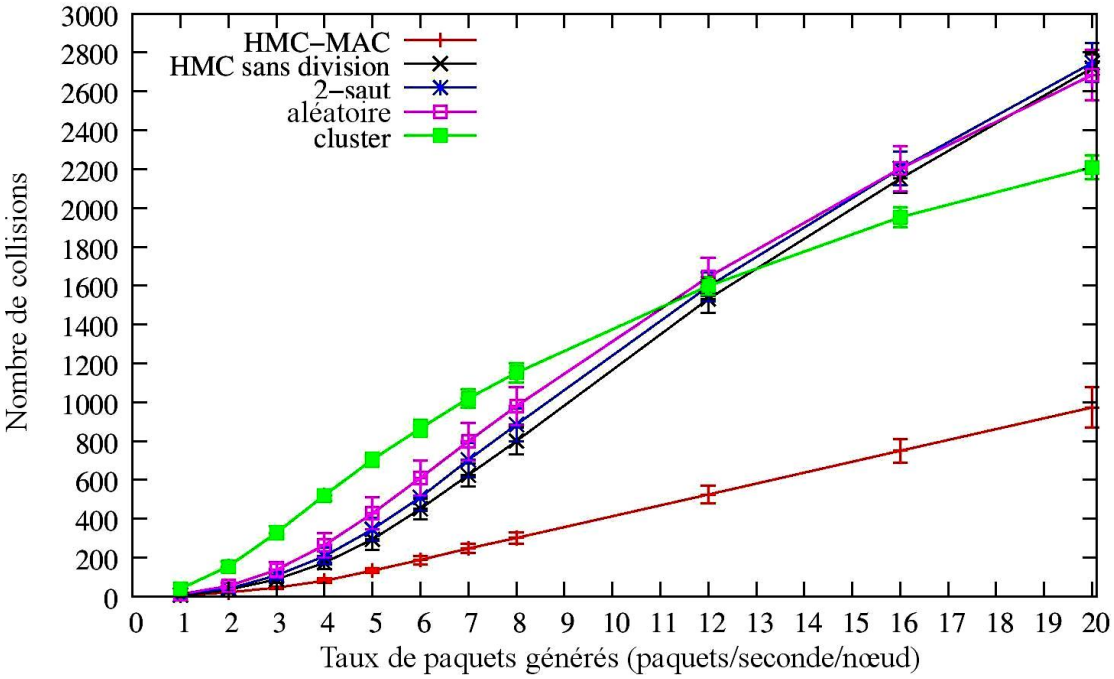


Figure 4.25 Nombre de collisions pour un trafic périodique.

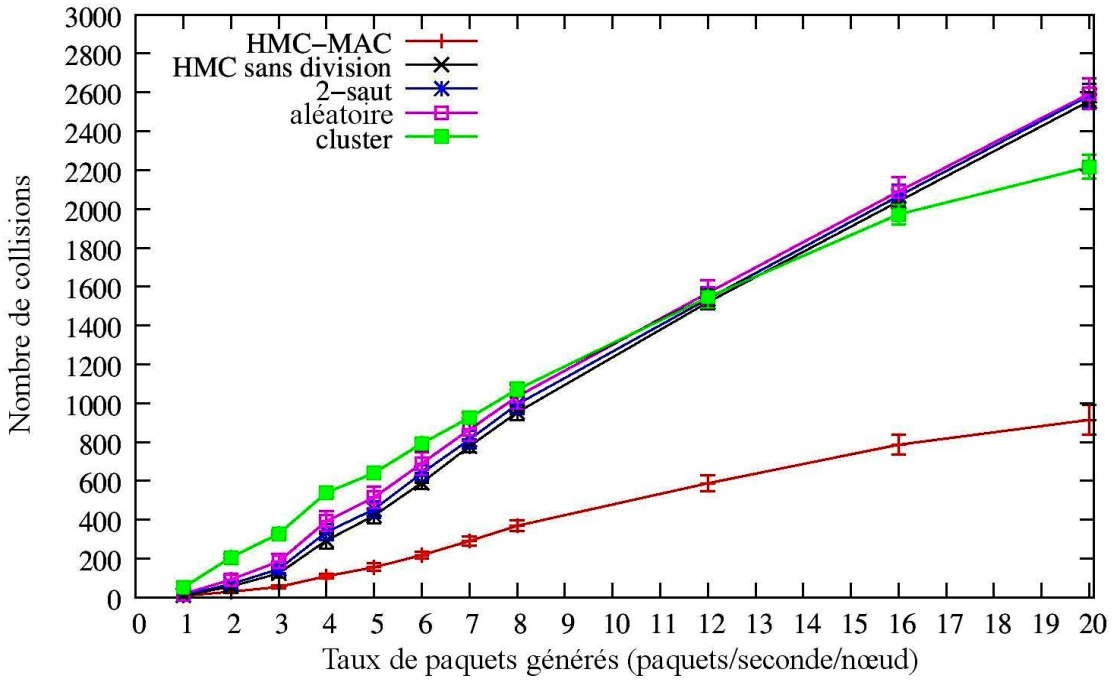


Figure 4.26 Nombre de collisions pour un trafic en rafale.

Les figures 4.25 et 4.26 montrent les résultats en termes de nombre de collisions. Nous observons que le nombre de collisions augmente avec le taux de génération de paquets. HMC-MAC réduit considérablement le nombre de collisions par rapport aux autres méthodes et ce fait est observé pour les deux profils de trafic. Comme mentionné précédemment, cette réduction est due au fait que HMC-MAC empêche d'une part les transmissions des trames aux nœuds qui sont occupés par un échange de données sur un autre canal et d'autre part offre une double utilisation des canaux. Ces deux figures nous permettent de remarquer qu'avant d'atteindre la saturation en nombre de paquets reçus par tous les nœuds (voir figure 4.23 et figure 4.24), les protocoles génèrent plus de collisions dans le cas du trafic en rafale par rapport au trafic périodique. Cela s'explique par le fait que dans le cas d'un trafic en rafale, les nœuds souffrent d'une contention plus forte durant la moitié du temps pour accéder au medium. En revanche lorsque la saturation est atteinte nous observons un nombre moins élevé de collisions dans le cas d'un trafic en rafale. Ceci résulte de l'effet d'écrêtage du trafic associé au profil du trafic (périodique ou rafale).

2.2.6. Nombre de paquets perdus dus aux collisions

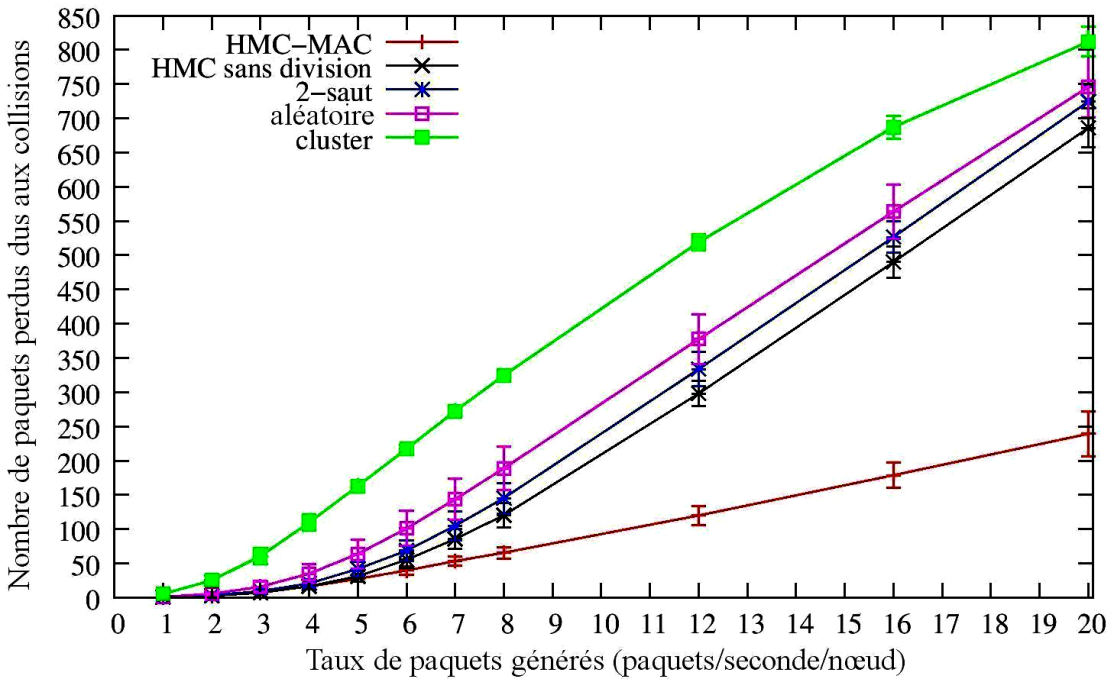


Figure 4.27 Nombre de paquets perdus dus aux collisions pour un trafic périodique.

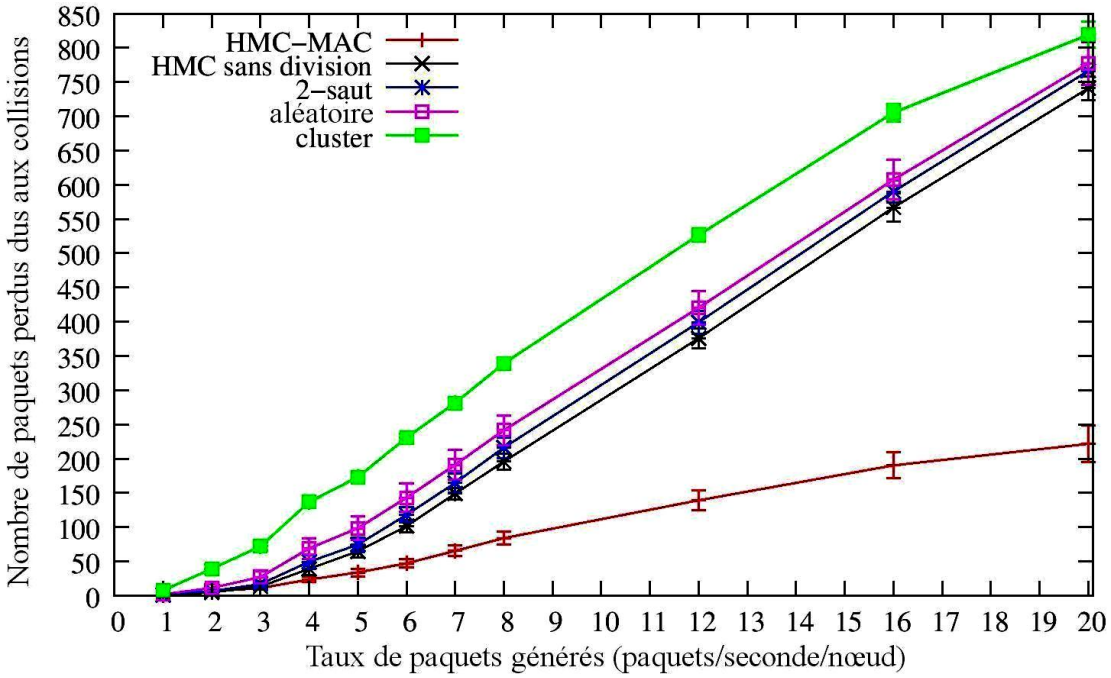


Figure 4.28 Nombre de paquets perdus dus aux collisions pour un trafic en rafale.

Les figures 4.27 et 4.28 présentent les résultats en termes de nombre de paquets perdus dus aux collisions. Le nombre de paquets perdus est plus important dans le cas d'un trafic en rafale que celui d'un trafic périodique à cause de l'augmentation de la contention entre les nœuds dans le cas du trafic en rafale. Pour les deux types de profils, HMC-MAC réduit significativement le nombre de paquets perdus par rapport aux autres méthodes. Ce qui prouve la robustesse de HMC-MAC en cas de génération sporadique de trafic qui peut représenter une génération d'alarme.

2.3. Evaluation des performances de HMC-MAC avec un nombre limité de canaux

Nous nous intéressons à l'évaluation de l'effet du nombre des canaux disponibles sur les performances de notre protocole par rapport aux autres méthodes étudiées. Le but de ces simulations est de montrer la robustesse de HMC-MAC face aux interférences qui peuvent être détectées sur certains canaux et nous empêchent de les utiliser. En effet, l'utilisation d'une liste noire, qui consiste à limiter le nombre des canaux disponibles, dans les protocoles multi-canaux permet de résister aux interférences externes. Le tableau 4.5 illustre les paramètres de simulation.

Simulateur utilisé	NS-2
Dimensions du réseau	100*100 m^2
Débit	250kbps
Portée de communication	20 m
Nombre de nœuds	49
Nombre des interfaces du puits	3
Nombre des canaux disponibles	16,10 et 6
Génération de trafic	Trafic périodique
Nombre de topologies générés	10 topologies aléatoires
Taille de la file d'attente	200
Paramètre des topologies	Lm = 7, Rm = Cm = 3
Taux de génération de paquets	1-20 paquets par seconde par nœud
Fenêtre d'observation	20 secondes
Nombre de répétition N	5 répétitions pour tous les protocoles
Modèle de propagation	*Two-ray ground*
Longueur des paquets	50 octets

Tableau 4.5 Paramètres de simulation utilisés pour l'évaluation des performances de HMC-MAC avec un nombre limité de canaux.

2.3.1. Débit agrégé

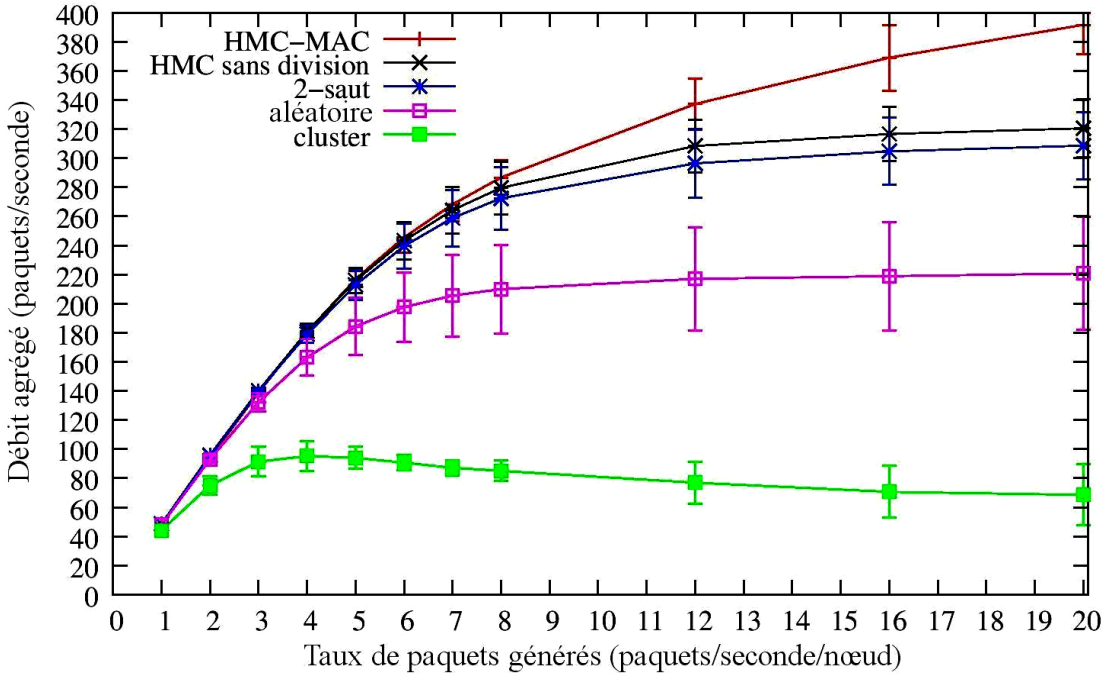


Figure 4.29 Débit agrégé lorsque 16 canaux sont disponibles.

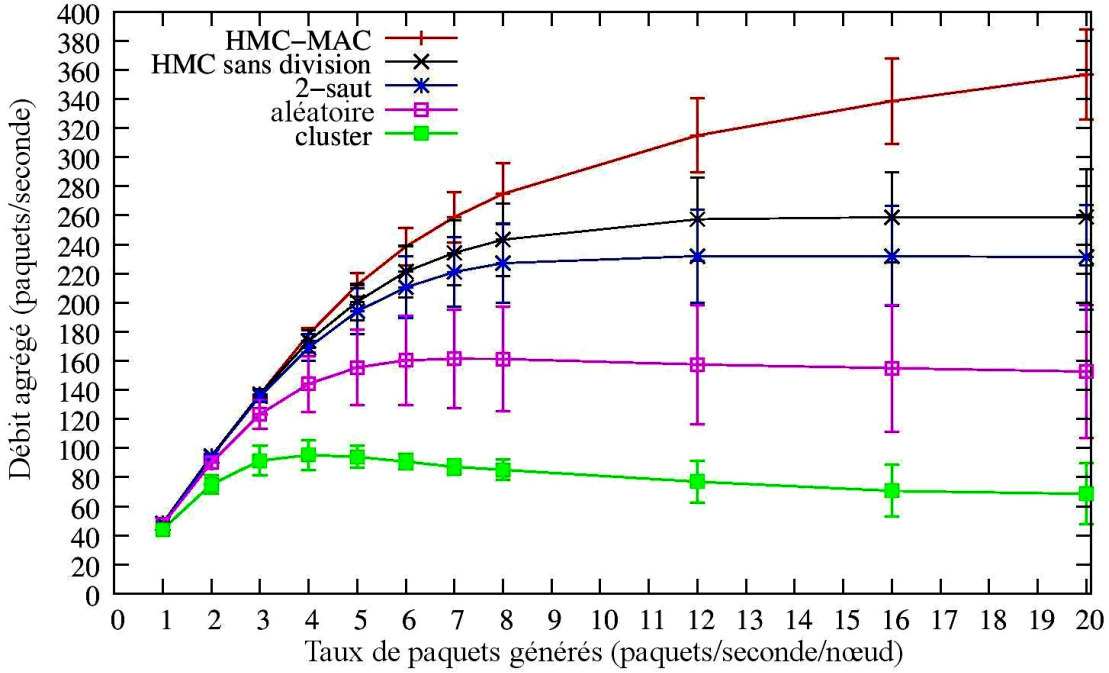


Figure 4.30 Débit agrégé lorsque 10 canaux sont disponibles.

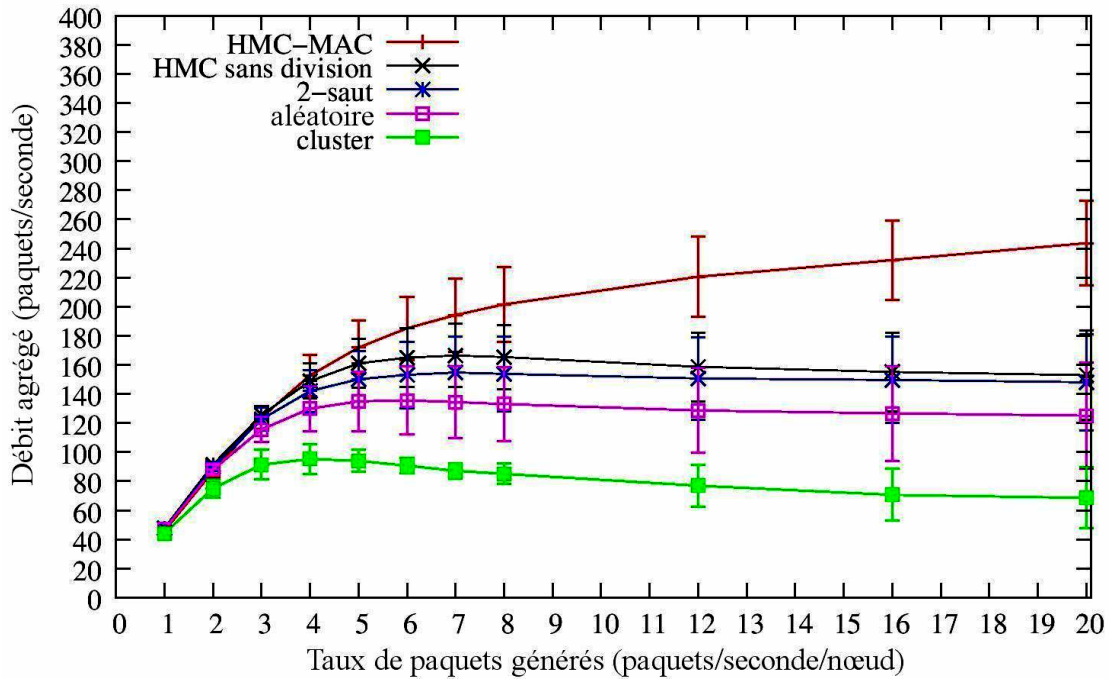


Figure 4.31 Débit agrégé lorsque 6 canaux sont disponibles.

Les figures 4.29, 4.30 et 4.31 montrent l'évolution du débit agrégé en fonction du taux de génération des paquets lorsque respectivement 16,10 et 6 canaux sont disponibles. Pour HMC-MAC, nous pouvons constater que quel que soit le nombre de canaux disponibles dans le réseau, le débit augmente avec le taux de génération de paquets dans la plage de trafic

exploré. En revanche, pour tous les autres protocoles, avec 16 canaux disponibles le débit atteint la saturation lorsque le nombre de paquets générés dépasse 12 paquets par seconde et par nœud. Avec 10 et 6 canaux disponibles, la saturation est atteinte respectivement à partir de 8 et 5 paquets par seconde et par nœud. Avec la méthode cluster, nous remarquons d'une part que le nombre de paquets reçus par le puits ne dépasse pas les 100 paquets par seconde et par nœud et d'autre part que la saturation est atteinte à partir de 4 paquets par seconde par nœud.

Les performances de notre proposition sont proches de celles obtenues avec HMC-MAC sans division sous faible charge de trafic, lorsque le nombre de paquets générés est inférieur à 6,4 et 3 paquets par seconde et par nœud pour respectivement 16,10 et 6 canaux disponibles. En revanche, sous forte charge, HMC-MAC améliore significativement le débit par rapport aux autres méthodes indépendamment du nombre de canaux disponibles. Cette amélioration est encore plus significative lorsque le nombre de canaux disponibles diminue, ceci est illustré dans le tableau 4.6 qui montre l'effet du nombre des canaux disponibles sur le débit de HMC-MAC par rapport aux autres méthodes. Prenons l'exemple de taux de génération de 16 paquets par seconde et par nœud. En utilisant 16 canaux, HMC-MAC augmente le débit d'environ 16.5%, 21% et 68,6% par rapport à HMC sans division, la méthode 2-sauts et la méthode aléatoire respectivement. Cependant, lorsque nous utilisons 10 canaux, cette augmentation est d'environ 30.8%, 45,8% et 118,4% respectivement. De même, lorsque nous utilisons 6 canaux, le gain est encore plus net par rapport aux autres méthodes sauf la méthode aléatoire où le gain est inférieur à celui de 10 canaux. Cela s'explique par le fait que HMC-MAC est le protocole le plus pénalisé par les pertes dues aux débordements de files d'attente. De plus, nous remarquons qu'en diminuant le nombre des canaux disponibles, le gain de HMC-MAC par rapport à la méthode cluster diminue. En effet, le débit de la méthode cluster reste constant car cette méthode fonctionne uniquement sur trois canaux en permanence, tandis que le débit de HMC-MAC diminue lorsque le nombre de canaux disponibles diminue.

Nous pouvons également remarquer que HMC-MAC sans division est plus performant que la méthode 2-sauts qui est à son tour plus performante que la méthode aléatoire. Cela peut être expliqué par le fait que la méthode 2-sauts ne prend pas en compte les canaux utilisés dans les voisinages à 3-sauts dans le cas où le nombre de canaux disponibles est suffisant pour les nœuds (pour 16 canaux) et par le fait que la méthode 2-sauts commence le choix aléatoire quand le nombre de canaux disponibles expire (pour 10 et 6 canaux).

Nombre de canaux	8 paquets par seconde par nœud				16 paquets par seconde par nœud			
	HMC sans division	2-sauts	aléatoire	Cluster	HMC sans division	2-sauts	aléatoire	Cluster
16	2.5%	5.2%	36.6%	237%	16.5%	21%	68.6%	422%
10	13%	21%	70.4%	223%	30.8%	45.8%	118.4%	379%
6	22%	31%	51.5%	137%	49.6%	55%	83%	228%

Tableau 4.6 Gain en débit de HMC par rapport aux autres méthodes.

2.3.2. Taux de débordement de files d'attente

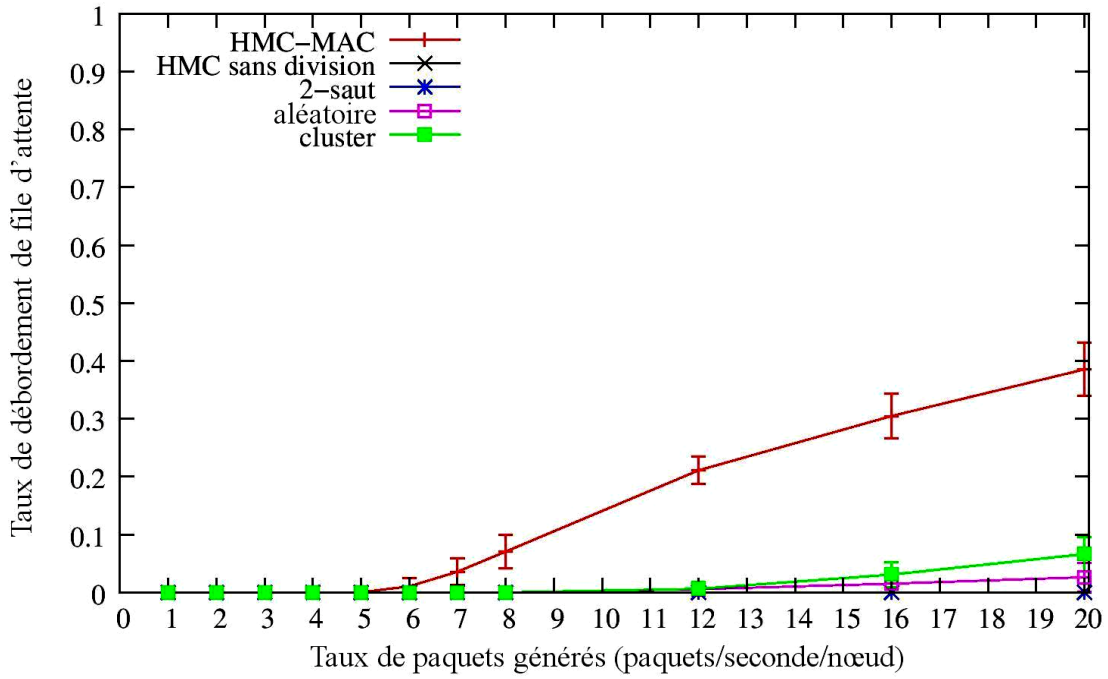


Figure 4.32 Taux de débordement de files d'attente lorsque 16 canaux sont disponibles.

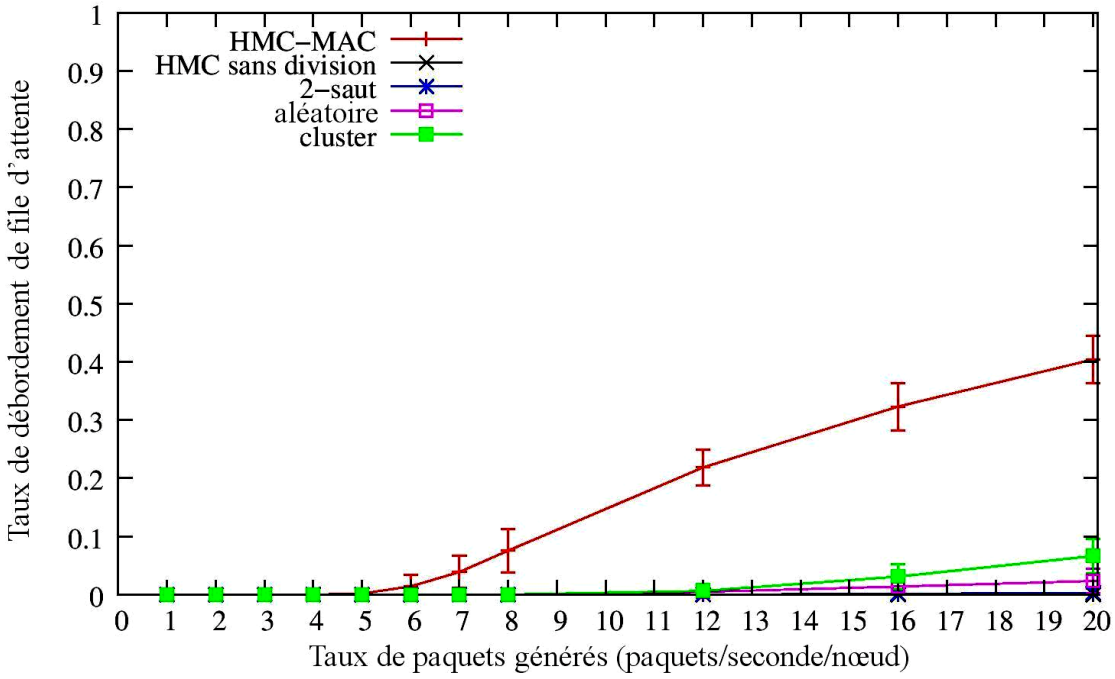


Figure 4.33 Taux de débordement de files d'attente lorsque 10 canaux sont disponibles.

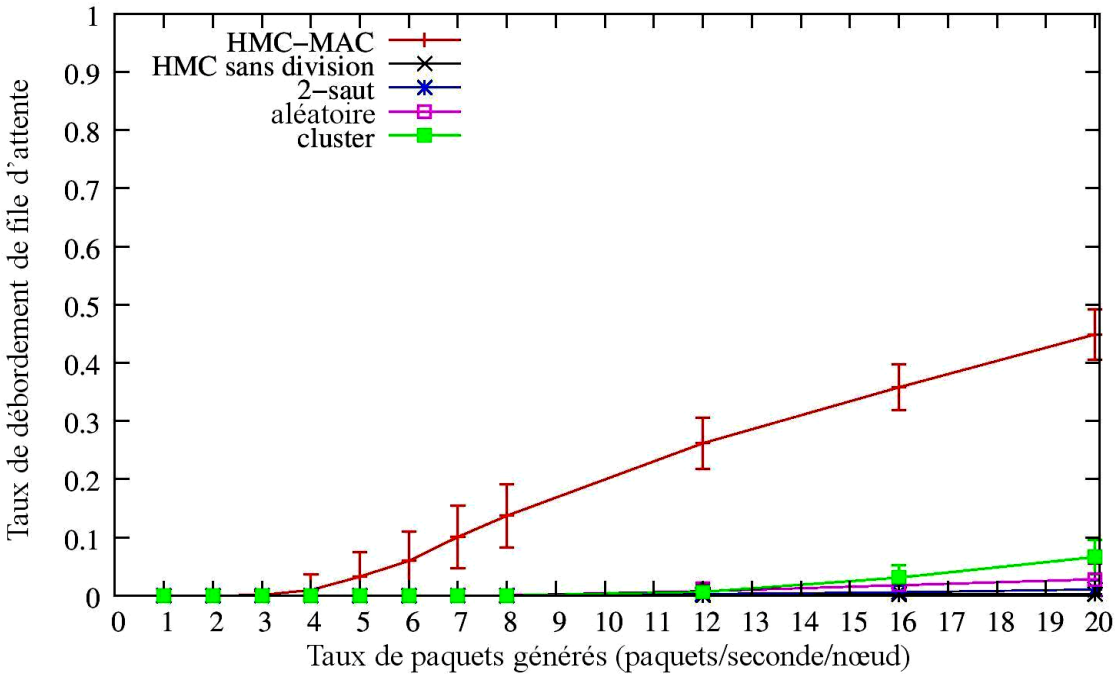


Figure 4.34 Taux de débordement de files d'attente lorsque 6 canaux sont disponibles.

Les figures 4.32, 4.33 et 4.34 présentent le taux de paquets perdus dû aux débordements de files d'attente en fonction de la charge produite par chacun des nœuds. Nous constatons que HMC-MAC subit un taux élevé de débordements de files d'attente lorsque 16, 10 et 6 canaux sont disponibles. Ces résultats s'interprètent de la même manière que ceux obtenus dans la partie 2.2.2 de ce chapitre. Nous remarquons aussi qu'en diminuant le nombre de canaux disponibles, les pertes dues aux débordements de files d'attente augmentent. Cela est dû à l'augmentation du nombre de nœuds utilisant le même canal. Notons que l'apparition des pertes provoquées par le débordement de files d'attente, pour la méthode aléatoire et la méthode cluster, est due à la forte contention entre les nœuds.

2.3.3. Taux de réception

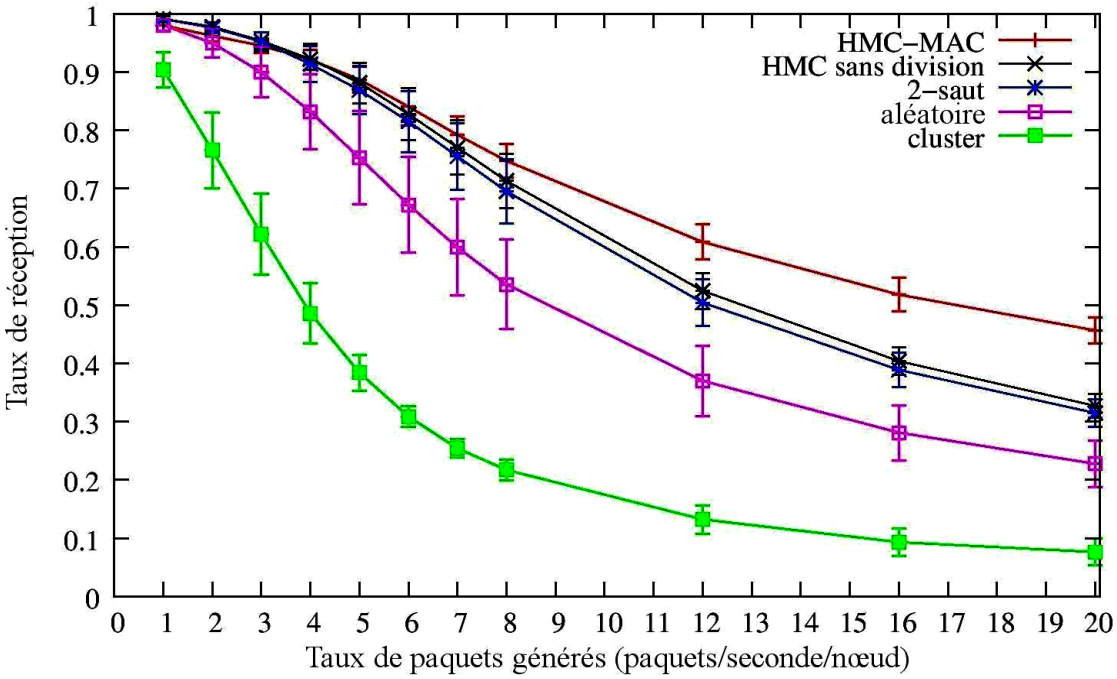


Figure 4.35 Taux de réception au niveau de puits lorsque 16 canaux sont disponibles.

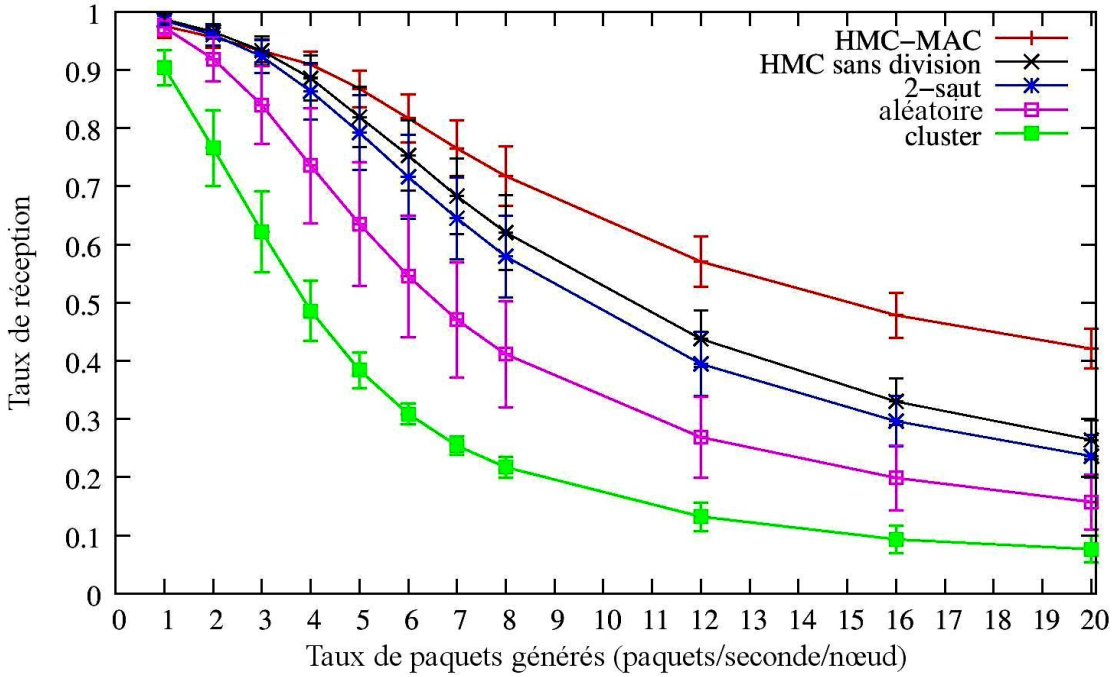


Figure 4.36 Taux de réception au niveau de puits lorsque 10 canaux sont disponibles.

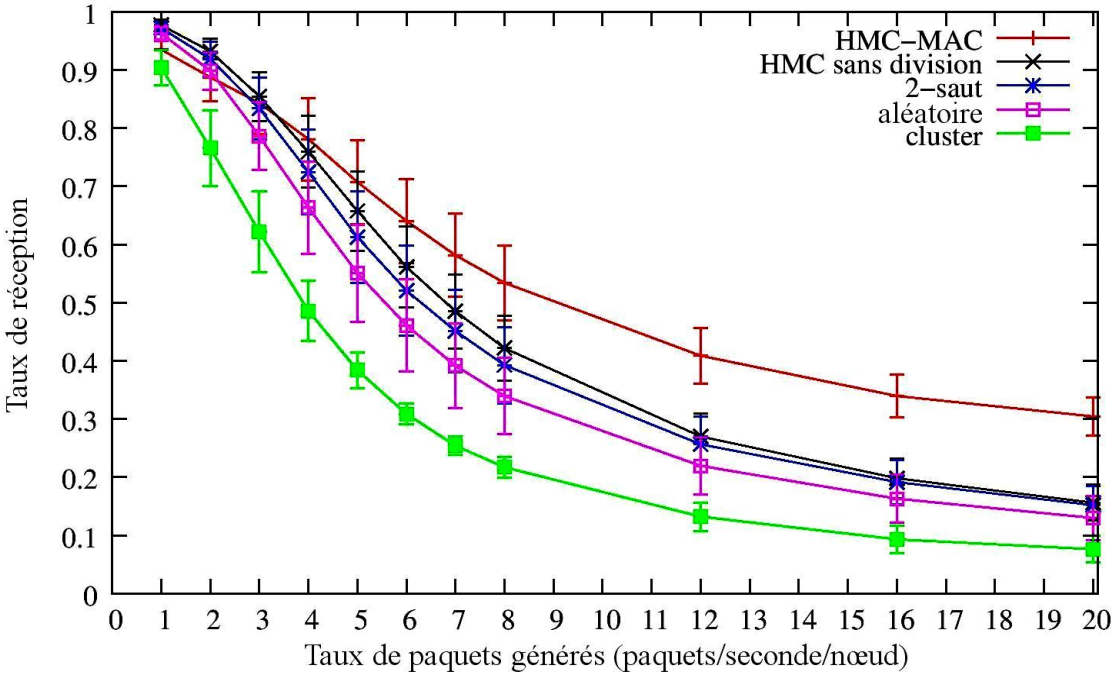


Figure 4.37 Taux de réception au niveau de puits lorsque 6 canaux sont disponibles.

Les figures 4.35, 4.36 et 4.37 présentent les résultats en termes de taux de réception au niveau de puits. Comme pour le débit, HMC-MAC donne de meilleures performances comparé à celles des autres protocoles quel que soit le nombre des canaux disponibles. Ceci est plus significatif lorsque le nombre de canaux disponibles diminue.

2.3.4. Nombre de paquets reçus par l'ensemble des nœuds

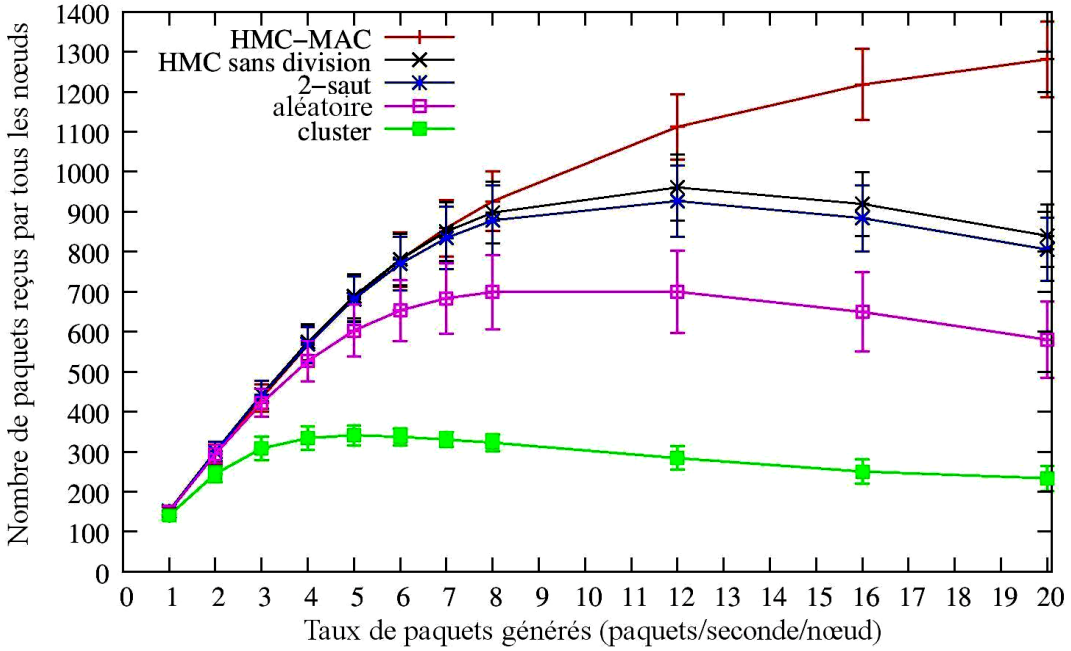


Figure 4.38 Nombre de paquets reçus par tous les nœuds lorsque 16 canaux sont disponibles.

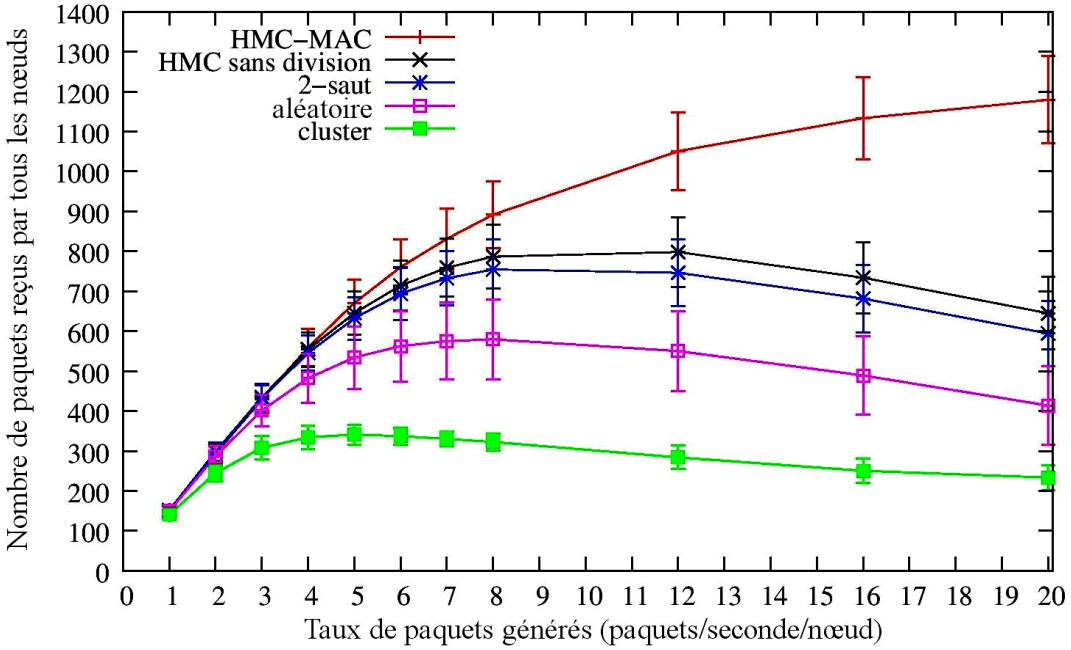


Figure 4.39 Nombre de paquets reçus par tous les nœuds lorsque 10 canaux sont disponibles.

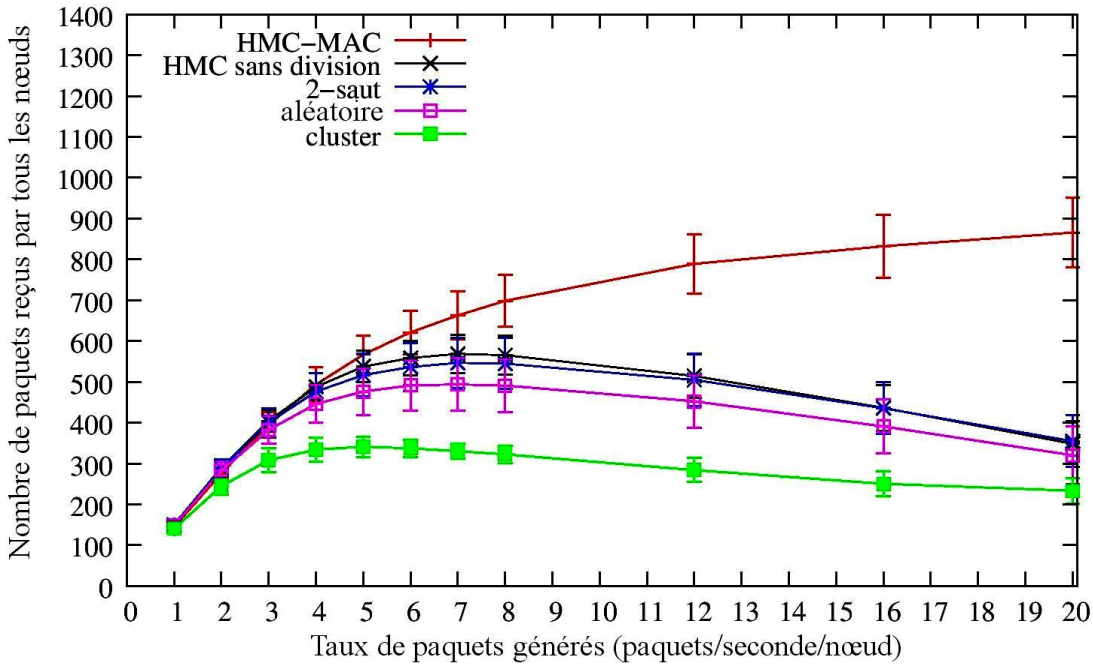


Figure 4.40 Nombre de paquets reçus par tous les nœuds lorsque 6 canaux sont disponibles.

Les figures 4.38, 4.39 et 4.40 présentent les résultats en termes de nombre de paquets reçus par tous les nœuds du réseau. Le tableau 4.7 montre l'effet du nombre de canaux disponibles sur le nombre de paquets reçus par tous les nœuds avec notre protocole par rapport à celui des autres protocoles. Nous constatons que HMC-MAC est plus efficace que tous les autres protocoles comparés, cette efficacité est plus significative lorsque le nombre de canaux disponibles diminue. Prenons l'exemple d'un taux de génération de 8 paquets par seconde et par nœud. En utilisant 16 canaux, HMC-MAC augmente le nombre de paquets reçus par tous les nœuds d'environ 3.2%, 5.4% et 32% par rapport aux méthodes HMC-MAC sans division, 2-sauts et aléatoire respectivement. D'autre part, lorsque nous utilisons 10 canaux, l'augmentation est d'environ 13.4%, 18.2% et 54% respectivement. Cela est dû au fait que la segmentation du réseau permet une double utilisation des canaux. C'est la raison pour laquelle notre protocole a la capacité de travailler avec un nombre limité des canaux. Il est également intéressant de remarquer que le gain de HMC-MAC par rapport aux autres méthodes augmente avec le taux de génération de paquets. Cela prouve que HMC-MAC gère mieux l'accès au medium sous forte charge et avec un nombre limité des canaux disponibles. Cependant, les autres méthodes souffrent plus de la perte de paquets due essentiellement au nombre important de collisions provoquant la saturation. Notons que les points de saturation sont les mêmes que ceux observés pour le débit, ils sont indiqués précédemment pour les différents nombres de canaux disponibles (voir figures 4.29, 4.30 et 4.31). Notons aussi que HMC-MAC est significativement meilleur que la méthode cluster, mais comme mentionné précédemment le nombre de paquets reçus par tous les nœuds diminue lorsque le nombre de canaux disponibles diminue.

Nombre de canaux	8 paquets par seconde par nœud				16 paquets par seconde par nœud			
	HMC sans division	2-sauts	aléatoire	Cluster	HMC sans division	2-sauts	aléatoire	Cluster
16	3.2%	5.4%	32%	187%	32.4%	37.8%	87.6%	387%
10	13.4%	18.2%	54%	177%	54.4%	66.3%	131.8%	353%
6	23.5%	28.2%	42.3%	117%	90.8%	91%	113%	232%

Tableau 4.7 Gain en nombre de paquets reçus par tous les nœuds de HMC par rapport aux autres méthodes.

2.3.5. Nombre de collisions

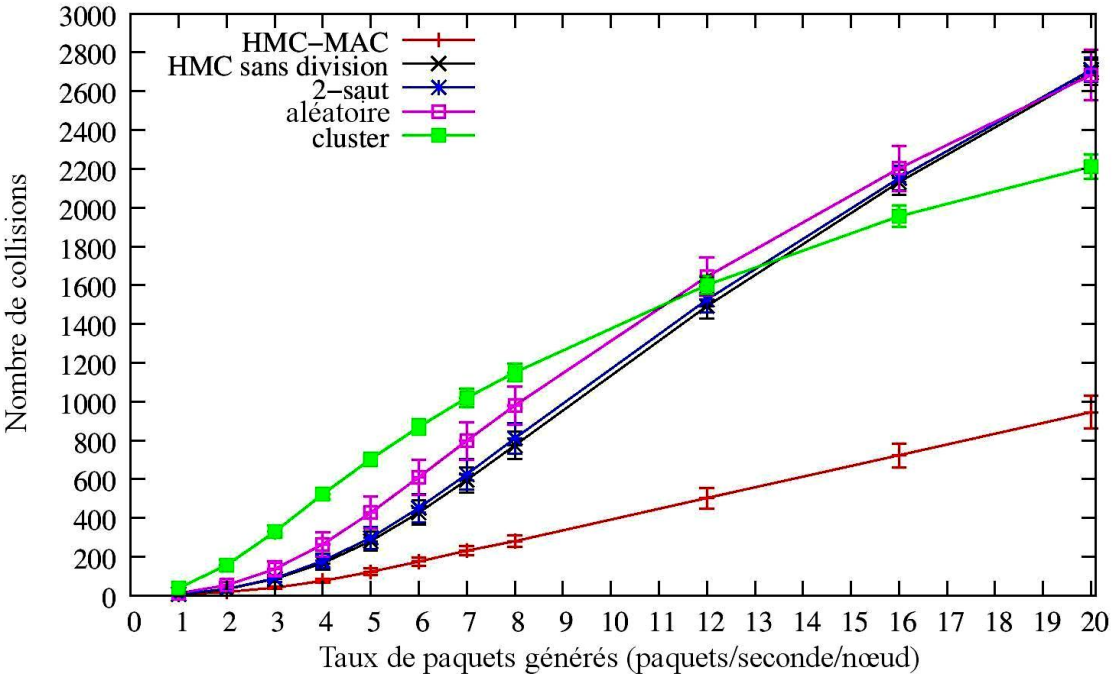


Figure 4.41 Nombre de collisions lorsque 16 canaux sont disponibles.

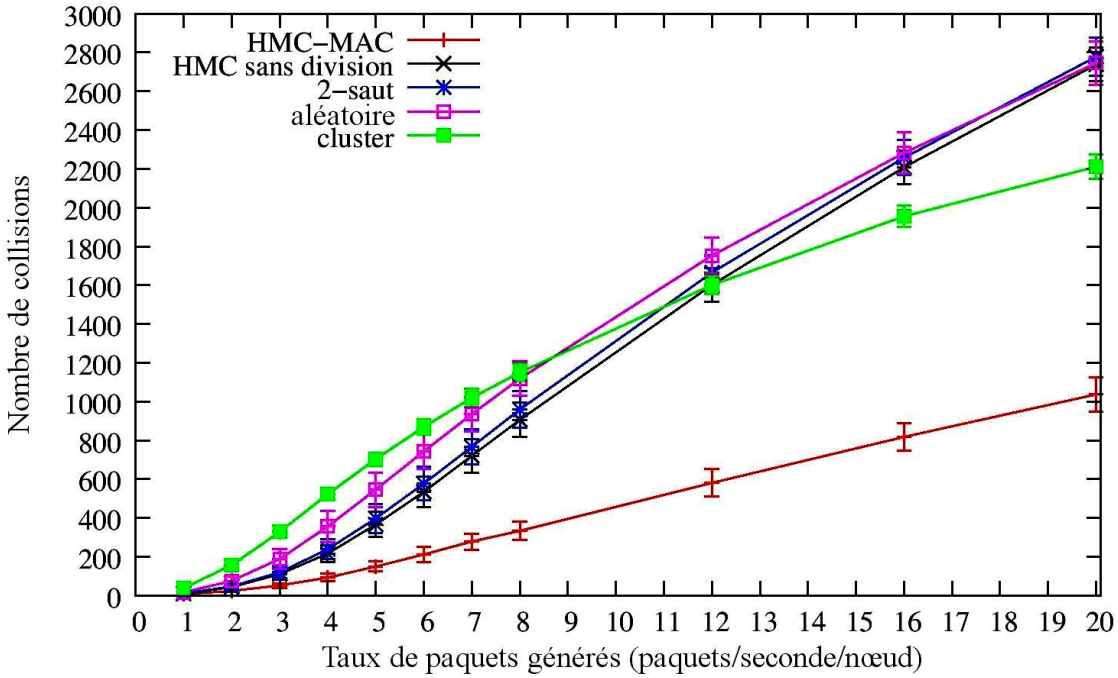


Figure 4.42 Nombre de collisions lorsque 10 canaux sont disponibles.

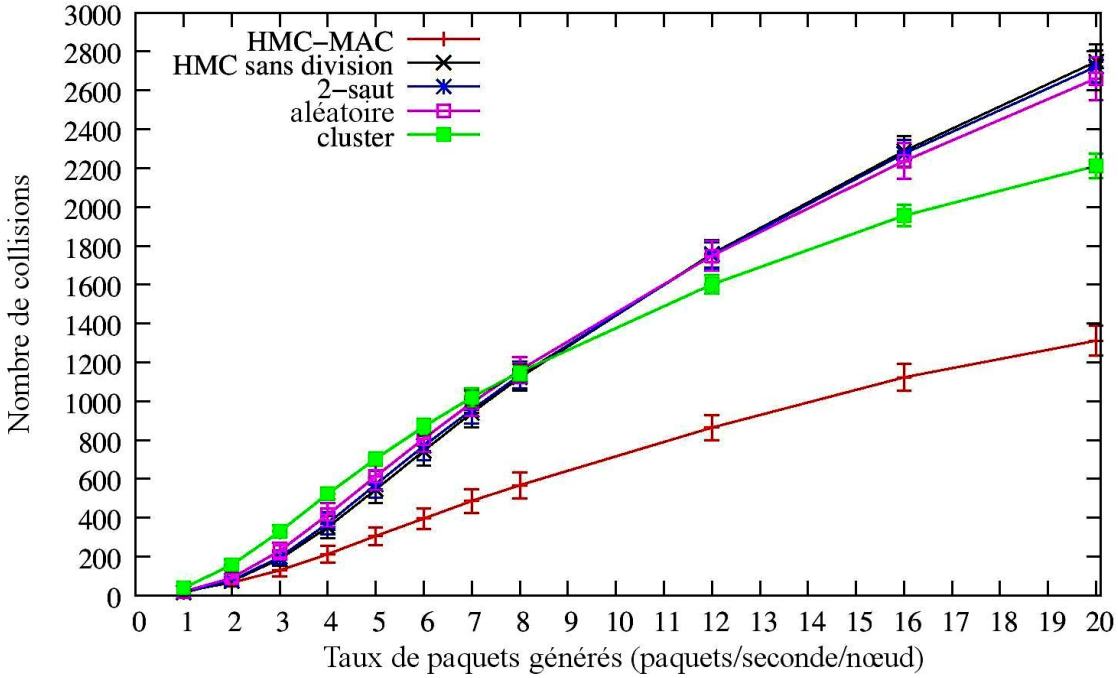


Figure 4.43 Nombre de collisions lorsque 6 canaux sont disponibles.

Les figures 4.41, 4.42 et 4.43 montrent le nombre de collisions en fonction du taux de génération de paquets lorsque respectivement 16, 10 et 6 canaux sont disponibles. Ces résultats nous permettent de remarquer que le nombre de collisions augmente quand le nombre de canaux disponibles diminue, ceci pour tous les protocoles. Ce comportement est dû à l'augmentation du nombre de nœuds en compétition sur le même canal. Il est intéressant de

constater que quel que soit le nombre de canaux disponibles, HMC-MAC réduit considérablement le nombre de collisions par rapport aux autres méthodes grâce à l'utilisation efficace des canaux et à la segmentation du réseau qui permet d'éviter les pertes de paquets provoquées par le problème du nœud sourd.

2.3.6. Nombre de pertes de paquets dues aux collisions

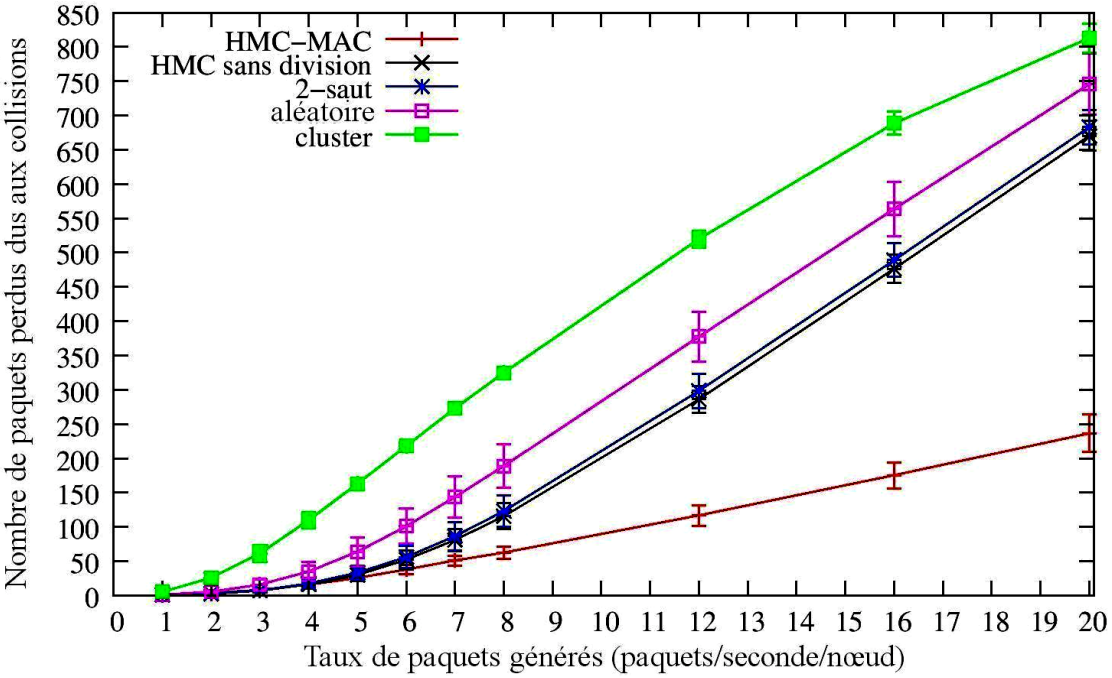


Figure 4.44 Nombre de pertes de paquets dues aux collisions lorsque 16 canaux sont disponibles.

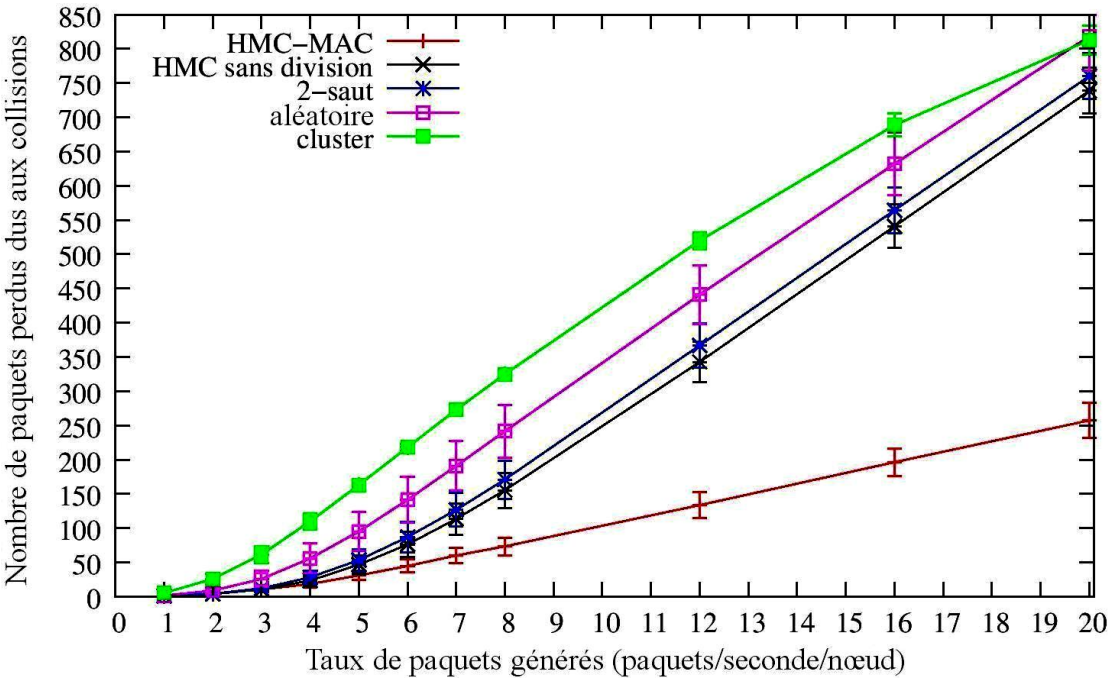


Figure 4.45 Nombre de pertes de paquets dues aux collisions lorsque 10 canaux sont disponibles.

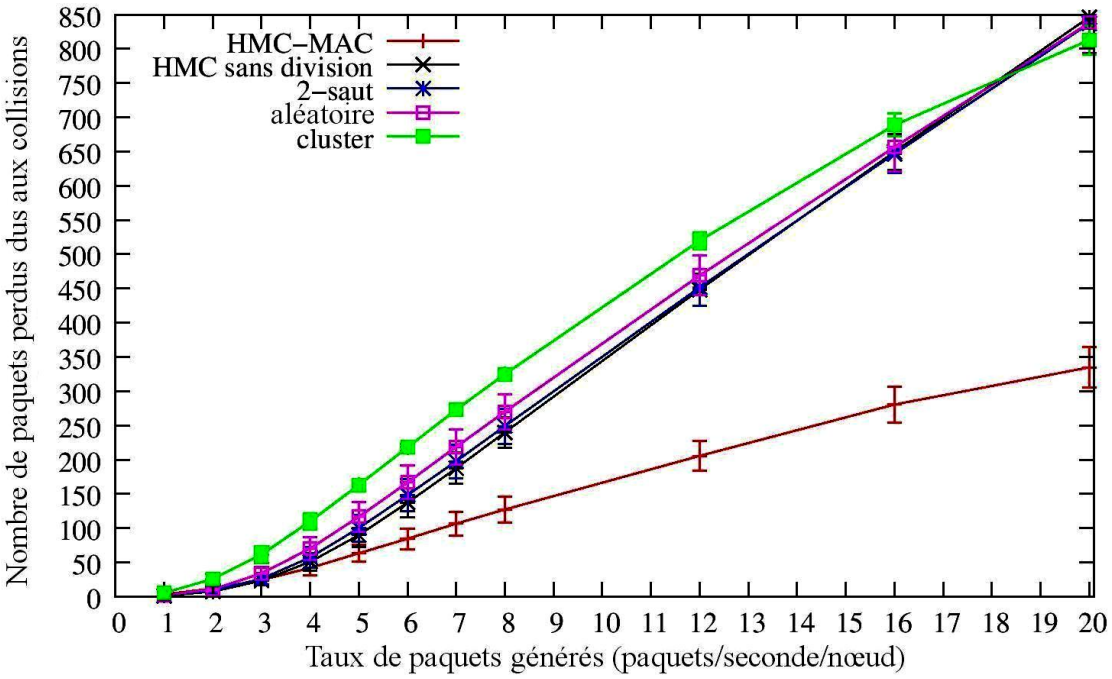


Figure 4.46 Nombre de pertes de paquets dues aux collisions lorsque 6 canaux sont disponibles.

Les figures 4.44, 4.45 et 4.46 présentent les résultats de simulation qui concernent l'évolution du nombre de paquets perdus dus aux collisions en fonction du nombre de paquets générés. Le tableau 4.8 montre l'effet du nombre de canaux disponibles sur ce nombre de paquets perdus avec HMC-MAC par rapport aux autres protocoles. HMC-MAC réduit significativement le nombre de paquets perdus par rapport aux autres protocoles. Prenons l'exemple de 8 paquets par seconde et par nœud. Avec 16 canaux disponibles, HMC-MAC réduit le nombre de paquets perdus d'environ 46%, 49% 67% et 81% par rapport aux méthodes HMC-MAC sans division, 2-sauts, aléatoire et cluster respectivement. De même, avec 10 canaux disponibles, cette réduction est d'environ 52.7%, 57%, 69.6% et 77.4% respectivement. D'autre part, avec seulement 6 canaux, HMC-MAC reste plus performant que les autres protocoles mais le gain obtenu est inférieur à celui avec 10 canaux. En effet, avec seulement 6 canaux, les interfaces du puits se voient contraintes à exploiter un canal déjà utilisé dans leur voisinage à 1-saut ou à 2-sauts, ce qui augmente le nombre de paquets perdus dans le réseau. Nous pouvons également remarquer que le nombre des canaux disponibles a un impact faible sur le nombre de paquets perdus lorsque la saturation est atteinte.

Nombre de canaux	8 paquets par seconde par nœud				16 paquets par seconde par nœud			
	HMC sans division	2-sauts	aléatoire	Cluster	HMC sans division	2-sauts	aléatoire	cluster
16	46%	49%	67%	81%	63%	64%	69%	74.5%
10	52.7%	57%	69.6%	77.4%	63.7%	65.2%	69%	71.5%
6	47%	49%	53%	61%	56.8%	56.7%	57.3%	59.2%

Tableau 4.8 Gain en nombre de paquets perdus de HMC par rapport aux autres méthodes

2.3.7. Délai de bout-en-bout

L'objectif de cette partie est d'évaluer le délai de bout-en-bout des paquets correctement reçus par le puits dans une topologie de réseau arborescente à plusieurs profondeurs. Cette évaluation est effectuée en fonction de la profondeur des sources pour les protocoles comparés. Notons que le délai de transit dans un réseau multi-sauts est défini par le temps de sérialisation pour chaque saut et le temps du séjour dans chacune des files d'attente des émetteurs. Pour des raisons de simplicité, nous avons décidé de présenter les résultats uniquement avec un taux de génération de 12 paquets par seconde et par nœud.

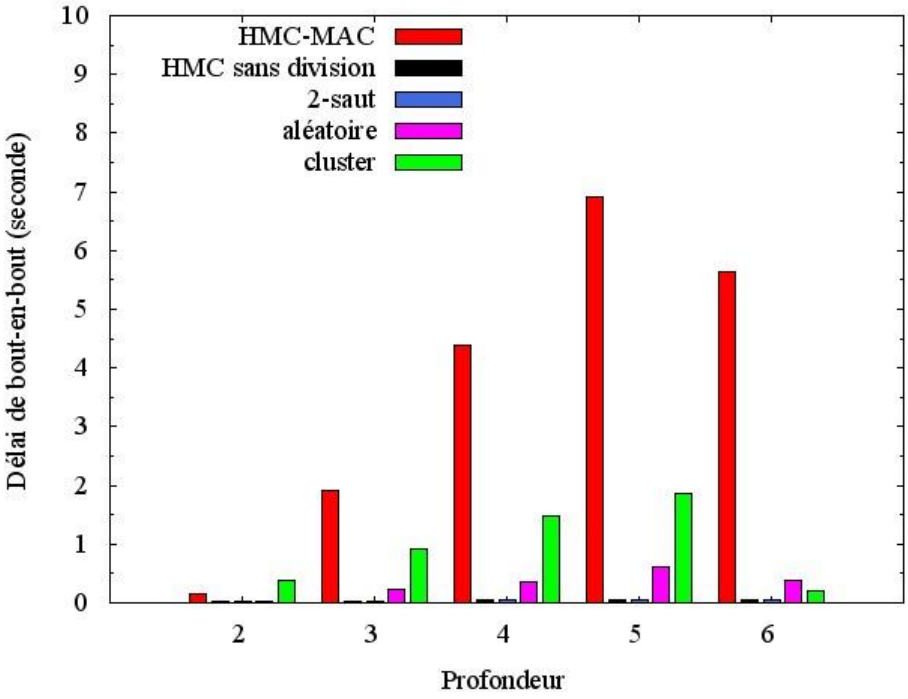


Figure 4.47 Délai de bout-en-bout lorsque 16 canaux sont disponibles.

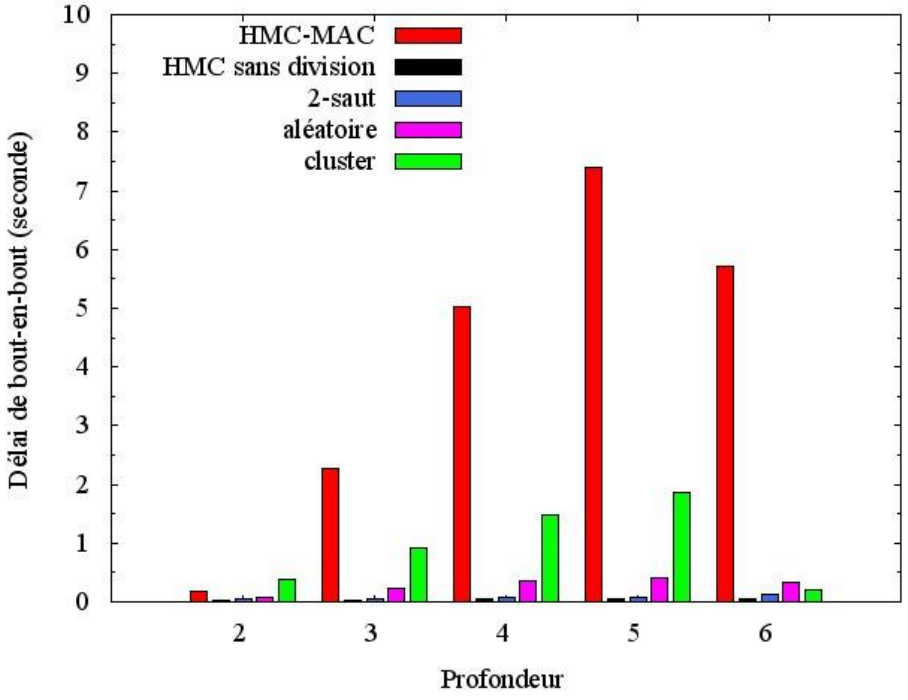


Figure 4.48 Délai de bout-en-bout lorsque 10 canaux sont disponibles.

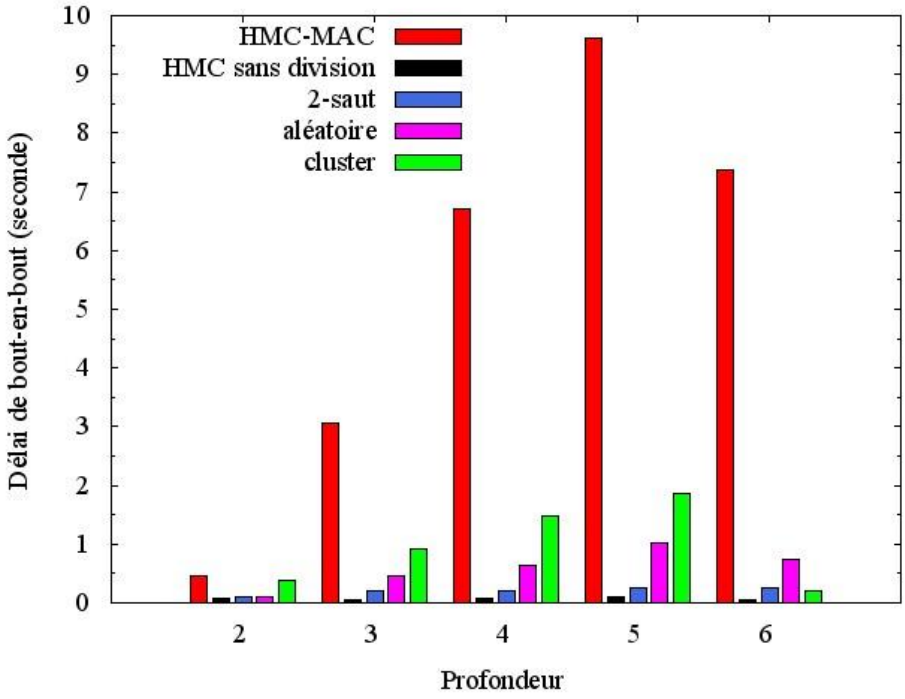


Figure 4.49 Délai de bout-en-bout lorsque 6 canaux sont disponibles.

Les figures 4.47, 4.48 et 4.49 présentent le délai de bout-en-bout des paquets correctement reçus par le puits, en fonction de la profondeur des sources, pour respectivement 16,10 et 6 canaux disponibles. Nous observons que le délai de HMC-MAC est plus grand que celui des autres méthodes. Cela est dû au fait que les paquets acheminés souffrent d'un long temps d'attente dans les files d'attentes avant d'être envoyés. En effet, HMC-MAC est capable d'acheminer beaucoup plus des paquets que les autres protocoles. Cependant, la couche MAC n'arrive pas à écouler tous les paquets en attente d'envoi, ce qui provoque une accumulation de trafic dans les files d'attente des nœuds intermédiaires et surtout les nœuds proches de puits. Nous remarquons aussi que le délai de bout-en-bout augmente lorsque nous diminuons le nombre des canaux disponibles, ceci est dû à l'augmentation de la contention entre les nœuds utilisant les mêmes canaux.

Les tableaux 4.9, 4.10 et 4.11 montrent l'impact de la profondeur des sources sur le nombre de paquets reçus par le puits. Nous remarquons que le gain de HMC-MAC par rapport aux autres méthodes augmente avec la profondeur des sources. Par exemple, pour 16 canaux disponibles, HMC-MAC augmente le nombre des paquets reçus par les nœuds de la profondeur 2 d'environ 4.8%, 9% et 39.6% par rapport aux méthodes HMC-MAC sans division, 2-sauts et aléatoire respectivement. Cette augmentation est d'environ 15.6%, 25.2% et 123% respectivement pour la profondeur 4. De même, elle est d'environ 36.5%, 54.4% et 236% respectivement pour les nœuds de profondeur 6. Ce qui nous permet de conclure que HMC-MAC achemine beaucoup mieux les paquets provenant des nœuds plus profonds que les autres protocoles. Il est également intéressant de remarquer que le gain de HMC-MAC par rapport aux autres méthodes augmente en diminuant le nombre des canaux disponibles.

	HMC sans division	2-sauts	Aléatoire	Cluster
2-saut	4.8%	9%	39.6%	236%
3-saut	7.8%	12.1%	75.1%	1098%
4-saut	15.6%	25.2%	123%	2204%
5-saut	29.5%	36%	162%	10026%
6-saut	36.5%	54.4%	236%	25572%

Tableau 4.9 Gain de HMC-MAC par rapport aux autres méthodes en fonction de la profondeur de la source lorsque 16 canaux sont disponibles.

	HMC sans division	2-sauts	Aléatoire	Cluster
2-saut	12.1%	25.1%	74.7%	220%
3-saut	28%	44.6%	157%	998%
4-saut	50.1%	73.5%	253%	1948%
5-saut	64.6%	91.7%	316%	8646%
6-saut	96.4%	129%	460%	22068%

Tableau 4.10 Gain de HMC-MAC par rapport aux autres méthodes en fonction de la profondeur de la source lorsque 10 canaux sont disponibles.

	HMC sans division	2-sauts	Aléatoire	cluster
2-saut	32.5%	37.1%	60%	146%
3-saut	64.9%	85.2%	147.3%	583%
4-saut	112.6%	123.8%	190.8%	912%
5-saut	124.3%	151.5%	260%	405%
6-saut	266%	208%	307%	854%

Tableau 4.11 Gain de HMC-MAC par rapport aux autres méthodes en fonction de la profondeur de la source lorsque 6 canaux sont disponibles.

Ces résultats présentés nous permettent de conclure qu'il y a toujours un compromis entre le délai de transit et le nombre de paquets acheminés au puits. En effet, HMC-MAC est capable d'acheminer plus de paquets que les autres protocoles mais souffre d'un délai de transit plus grand que les protocoles concurrents. Cependant, les autres protocoles assurent de courts délais de transit mais acheminent un nombre limité de paquets.

3. Conclusion

Dans ce chapitre, nous avons présenté une évaluation des performances par simulation de HMC-MAC et nous l'avons comparé à d'autres méthodes utilisées dans la littérature.

Dans la première partie, nous avons montré l'efficacité de la méthode d'allocation de HMC-MAC par rapport aux autres méthodes en termes de nombre de conflits et de taux d'interférence. Ces métriques sont obtenues en se basant sur l'étude des conséquences de la réutilisation des canaux dans les voisinages jusqu'à 3-saut.

Dans la deuxième partie, nous avons évalué les performances de HMC-MAC dans des scenarios où les nœuds génèrent des paquets dans le réseau. D'abord, nous avons considéré que le puits est équipé d'une seule interface. Nous avons montré que HMC-MAC est plus performant que les autres protocoles mais le puits n'est pas capable d'absorber tous les paquets provenant de ses fils. Ensuite, nous avons utilisé un puits équipé de plusieurs interfaces. Nous avons évalué en premier temps l'impact du type de trafic et en second temps l'impact du nombre de canaux disponibles sur les performances de HMC-MAC par rapport aux autres méthodes. Les résultats ont montré que, sous forte charge, HMC-MAC améliore significativement les performances par rapport aux autres méthodes. Cette amélioration est plus significative dans le cas de trafic en rafale, et aussi lorsque le nombre de canaux disponibles diminue.

D'autre part, les résultats ont montré que HMC-MAC souffre de problème de débordement de files d'attente essentiellement dû à l'accumulation des paquets dans les files d'attente des nœuds qui sont proches de puits. Cette accumulation provoque un long temps de séjour des paquets dans les files d'attente avant d'être envoyés. Ceci engendre des délais de bout-en-bout significatifs par rapport aux autres protocoles. Cependant, le nombre de paquets acheminés avec HMC-MAC est plus important que celui avec les autres protocoles.

Chapitre 5 Conclusion et perspectives

Dans les premières applications des réseaux de capteurs sans fil, le besoin du haut débit n'a pas été considéré comme la principale préoccupation. Cependant, dans ces dernières années, l'émergence des nouvelles applications qui nécessitent le déploiement d'un nombre important des nœuds et la transmission d'une forte charge de trafic nous oblige à considérer le besoin des réseaux de capteurs sans fil à haut débit.

1. Conclusion

Dans le cadre de cette thèse, nous avons étudié les protocoles MAC multi-canaux les plus représentatifs qui ont été proposés pour les réseaux de capteurs sans fil. Nous avons présenté les avantages ainsi que les inconvénients de chaque protocole. Les protocoles qui utilisent la technique TDMA ne sont pas bien adaptés aux changements de topologies et surchargent le réseau. En revanche, les protocoles qui utilisent la technique CSMA/CA sont plus flexibles et plus adaptés aux changements du réseau. Cependant, ces protocoles souffrent d'un nombre élevé de collisions qui dégradent les performances du réseau.

Pour cela, nous avons étudié les conséquences de l'utilisation du même canal dans les voisinages jusqu'à 3-sauts de chaque nœud. Nous avons formalisé que le risque de collision augmente lorsque le nombre de sauts diminue entre les nœuds utilisant le même canal pour la réception de données. Ensuite, nous avons abordé les pertes de paquets dues au problème du nœud sourd, les collisions dues au problème du terminal caché et celles dues au problème des interférences externes.

Cela nous a amené à proposer HMC-MAC, un protocole MAC multi-canal qui répond aux exigences des RCSF à haut débit dans une topologie multi-sauts. HMC-MAC utilise la technique CSMA/CA afin de convenir un réseau évolutif et d'éviter les difficultés de coopération entre les nœuds. Le but du protocole HMC-MAC est de minimiser le risque de collisions et d'interférences provoqué par l'utilisation de CSMA/CA et d'augmenter le débit du réseau. Pour cela ce protocole est caractérisé par les actions suivantes. D'abord, il exploite des transmissions parallèles en utilisant les canaux autorisés par la couche physique de la norme IEEE 802.15.4. En effet, nous avons proposé une technique qui permet aux nœuds de choisir d'une façon distribuée le canal le plus convenable dans leur voisinage jusqu'à 3-sauts. Ensuite, il utilise une approche de segmentation temporelle afin d'organiser les activités des nœuds, ce qui diminue l'effet de la forte densité et évite les pertes provoquées par le problème des nœuds sourds. Enfin, il utilise un puits multi-interface dans le réseau permettant la réception simultanée des données de plusieurs nœuds sur des canaux différents. De ce fait,

nous avons proposé une méthode d'ordonnancement des slots de temps et des canaux qui permet au puits de recevoir en continu des trames de données.

Des simulations faites en utilisant le simulateur NS-2 ont permis de montrer l'intérêt de notre proposition. D'abord, nous avons présenté une évaluation des performances de HMC-MAC et nous l'avons comparé aux autres méthodes utilisées dans la littérature. Les résultats ont montré l'efficacité de la méthode d'allocation de HMC-MAC par rapport aux autres méthodes en termes de nombre de conflits et de taux d'interférences. Ensuite, nous avons évalué les performances de HMC-MAC en fonction du taux de génération de paquets dans des scenarios divers. Les résultats ont montré que sous forte charge, HMC-MAC offre de meilleures performances par rapport aux autres protocoles en termes de débit agrégé, taux de livraison de paquets, nombre de paquets reçus par l'ensemble des nœuds du réseau, nombre de collisions et nombre de paquets perdus. Cependant, HMC-MAC souffre des pertes de données dues aux débordements de files d'attente et d'un délai de transit de bout-en-bout significatif dû à l'accumulation des paquets dans les files d'attente des nœuds situés autour de puits. Dans la suite de ce chapitre, nous abordons des nouvelles perspectives pour accompagner ce travail.

2. Perspectives

Les travaux effectués durant de cette thèse nous ont permis de proposer de nouvelles perspectives de recherche. Ces perspectives se divisent en 2 groupes : les perspectives qui permettent de consolider notre travail et les perspectives qui représentent une extension de notre travail.

2.1. Perspectives permettant la consolidation de notre travail

Mobilité des nœuds

Nous avons montré les performances de notre proposition dans des topologies aléatoires dans lesquelles les nœuds sont fixes, ceci en utilisant un modèle de propagation non probabiliste. Il serait intéressant d'évaluer notre proposition en utilisant un modèle de propagation plus réaliste, ce qui revient à considérer une certaine instabilité des liens et/ou une légère mobilité des nœuds.

De plus, plusieurs applications envisagées pour le RCSF exigent la mobilité des nœuds. Pour cela, nous envisageons d'étudier les performances de notre proposition dans des scenarios où les nœuds sont mobiles.

Validation par maquettage

L'évaluation des performances de notre proposition est effectuée en utilisant un simulateur réseau. Les résultats de simulation ont montré le bon fonctionnement de notre proposition dans divers scenarios. Cependant, il serait intéressant d'évaluer notre proposition sur une plateforme réelle afin de confirmer ou consolider les résultats obtenus par simulation.

Evaluation globale du trafic de contrôle

Le trafic de contrôle est utilisé afin de recueillir et partager les informations nécessaires pour les échanges de données entre les nœuds. Dans cette thèse, nous avons évalué les performances réalisées une fois que toutes les informations nécessaires sont échangées. Une évaluation globale du trafic de contrôle échangé serait un bon complément à ce travail. Notons que dans notre protocole, l'échange de ce trafic est fait en utilisant la technique TDMA.

2.2. Perspectives représentant une extension de notre travail

Répartition de trafic

Comme expliqué précédemment, HMC-MAC souffre de problèmes d'accumulation des paquets dans les files d'attente surtout pour les nœuds proches du puits qui ont un grand nombre de descendants. Une piste serait de proposer un mécanisme équitable de répartition de trafic autour des voisins du puits. Cette solution permettrait de réduire les délais de bout-en-bout et les pertes de paquets dus aux débordements de files d'attente pour les nœuds proches du puits et ainsi d'apporter un gain important en termes de débit. Cette perspective a fait l'objet du sujet d'une nouvelle thèse entreprise dans notre équipe.

Plusieurs puits

Dans cette thèse, nous avons considéré qu'il existe un seul puits dans le réseau. Comme montré précédemment, HMC-MAC offre des meilleures performances en termes de la charge offerte au niveau de la couche MAC mais souffre de problème de débordement de files d'attente dû à l'accumulation des paquets dans les files d'attente des nœuds proches de puits. Nous pouvons supposer que HMC-MAC donnera de meilleurs résultats lorsque nous considérons l'existence de plusieurs puits dans le réseau. Cependant, un nouveau protocole de routage et une nouvelle méthode de segmentation du réseau doivent être mis en place. Cet aspect sera traité dans nos travaux futurs.

Nombre d’interfaces de puits et longueur des paquets

Les protocoles multi-canaux visent à améliorer les performances globales du réseau. Cependant, le débit est souvent limité par la capacité de réception des interfaces de puits. Ainsi, l'utilisation des canaux multiples perd une partie de son intérêt. Il serait intéressant d’évaluer les performances en termes de nombre d’interfaces de puits afin de tester le débit agrégé maximal atteint par notre protocole.

De plus, dans nos simulations nous avons considéré que la longueur des paquets est de 50 octets pour un réseau soumis à une forte charge. D’autres simulations pourraient être envisagées pour évaluer l’impact de la charge et de la longueur des paquets sur les performances des protocoles comparés.

Liste des abréviations

Annexe

Afin d'étudier le comportement des paquets restants dans les files d'attente, nous avons prolongé le temps de simulation jusqu'à ce que tous les paquets existants dans les files d'attente soient transmis. Dans ces simulations, nous considérons que 16 canaux sont disponibles.

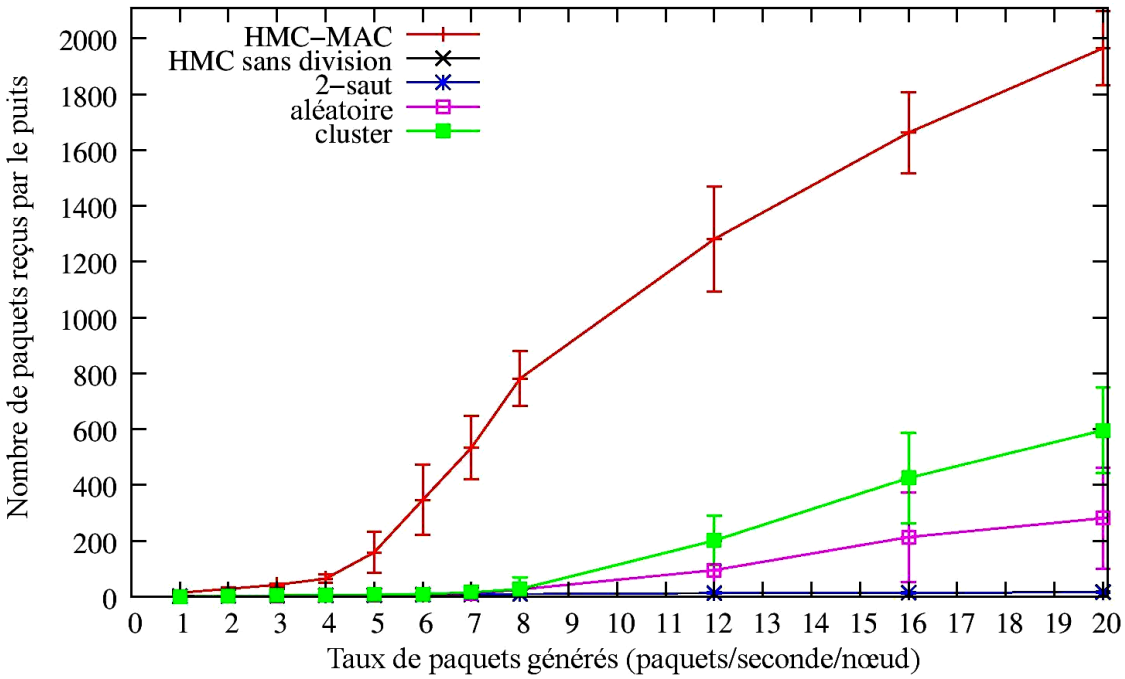


Figure 5.1 Nombre de paquets reçus par le puits après l'arrêt de génération de trafic dans des conditions de dégorgement des files d'attente.

La figure 5.1 présente le nombre de paquets reçus par le puits après l'arrêt de génération de trafic en fonction du taux de génération de paquets. Nous remarquons que ces résultats ont la même allure que ceux obtenus pour le taux de débordement de files d'attentes (voir figure 4.32). Ces résultats montrent que HMC-MAC est capable d'acheminer un nombre important des paquets restants dans les files d'attente par rapport aux autres protocoles.

Liste des publications

1. Rana Diab, Gérard Chalhoub, and Michel Misson, "Hybrid Multi-Channel MAC Protocol for Wireless Sensor Networks: Interference rate evaluation", in IEEE 78th Vehicular Technology Conference: VTC2013-Fall, Las Vegas, USA, Septembre 2013.

2. Rana Diab, Gérard Chalhoub, and Michel Misson, "Channel Allocation Evaluation for a Multi-Channel MAC Protocol", in IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, London, UK., Septembre 2013.

3. Rana Diab, Gérard Chalhoub, and Michel Misson, "Overview on Multi-Channel Communications in Wireless Sensor Networks", Special Issue on Network Protocols and Algorithms Surveys and Tutorials ISSN, vol. 5, no. 3, pp. 1943–3581, 2013.

4. Rana Diab, Gérard Chalhoub, and Michel Misson, "Evaluation of a Hybrid Multi-Channel MAC protocol for Perodic and Burst Traffic", in The 39th IEEE Conference on Local Computer Networks (LCN), Edmonton, Canada, Septembre 2014 (short paper).

5. Rana Diab, Gérard Chalhoub, and Michel Misson, "Enhanced Multi-Channel MAC protocol for Multi-Hop Wireless Sensor Network", in Wireless Days, Rio de Janeiro, Brésil, Novembre 2014.

Bibliographie :

[1] IEEE 802.15, « Part 15.4 : Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (WPANs) », ANSI/IEEE, Standard 802.15.4 R2006, 2006.

[2] I. Howitt et J. A. Gutierrez, « IEEE 802.15.4 low rate - wireless personal area network coexistence issues », in *2003 IEEE Wireless Communications and Networking, 2003. WCNC 2003*, 2003, vol. 3, p. 1481-1486 vol.3.

[3] ZigBee, « ZigBee Specification », ZigBee Standards Organization, Standard ZigBee 053474r17, janv. 2008.

[4] HART Communication., « http://www.hartcomm2.com/ ». .

[5] « ISA 100.11a - Wireless Standards for Industrial Automation - 2009 », *Scribd*. [En ligne]. Disponible sur: http://www.scribd.com/doc/59551018/ISA-100-11a-Wireless-Standards-for-Industrial-Automation-2009.

[6] « Van_Hoesel_Thesis.pdf ».

[7] « CC2420 | Proprietary 2.4 GHz | Wireless Connectivity | Description & parametrics ». [En ligne]. Disponible sur: http://www.ti.com/product/cc2420. [Consulté le: 10-avr-2015].

[8] Jayasuriya A., Perreau S., Dadej A., and Gordon S., « Hidden vs. exposed terminal problem in ad hoc networks », in *Proceedings of the Australian Telecommunication Networks and Applications Conference*, 2004.

[9] O. D. Incel, « A survey on multi-channel communication in wireless sensor networks », *Comput. Netw.*, vol. 55, n° 13, p. 3081-3099, sept. 2011.

[10] G. EkbataniFard et R. Monsefi, « A Detailed Review of Multi-Channel Medium Access Control Protocols for Wireless Sensor Networks », *Int. J. Wirel. Inf. Netw.*, vol. 19, n° 1, p. 1-21, mars 2012.

[11] Rappaport T. S., *Wireless Communications PRINCIPLES AND PRACTICE*, Second edition ed. Bernard M. Goodwin. 2002.

[12] X. Chen, P. Han, Q.-S. He, S.-L. Tu, et Z.-L. Chen, « A Multi-Channel MAC Protocol for Wireless Sensor Networks », in *The Sixth IEEE International Conference on Computer and Information Technology, 2006. CIT '06*, 2006, p. 224□224.

[13] M. Salajegheh, H. Soroush, et A. Kalis, « HYMAC: Hybrid TDMA/FDMA Medium Access Control Protocol for Wireless Sensor Networks », in *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007*, 2007, p. 1-5.

[14] A. Rowe, R. Mangharam, et R. Rajkumar, « RT-Link: A Time-Synchronized Link Protocol for Energy- Constrained Multi-hop Wireless Networks », in *2006 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, 2006. SECON '06*, 2006, vol. 2, p. 402-411.

[15] « CC2420 Data Sheet - zigbee.pdf ». .
[16] B. Priya et S. S. Manohar, « EE-MAC: Energy Efficient Hybrid MAC for WSN », *Int. J. Distrib. Sens. Netw.*, vol. 2013, p. e526383, déc. 2013.
[17] Y. Wu, J. A. Stankovic, T. He, et S. Lin, « Realistic and Efficient Multi-Channel Communications in Wireless Sensor Networks », in *IEEE INFOCOM 2008. The 27th Conference on Computer Communications*, 2008, p. -.
[18] K. S. J. Pister et L. Doherty, « TSMP: Time Synchronized Mesh Protocol », présenté à Parallel and Distributed Computing and Systems.
[19] T. Watteyne, A. Mehta, et K. Pister, « Reliability Through Frequency Diversity: Why Channel Hopping Makes Sense », in *Proceedings of the 6th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks*, New York, NY, USA, 2009, p. 116–123.
[20] R. Soua, P. Minet, et E. Livolant, « MODESA: An optimized multichannel slot assignment for raw data convergecast in wireless sensor networks », in *Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International*, 2012, p. 91-100.
[21] S. Lohier, A. Rachedi, I. Salhi, et E. Livolant, « Multichannel access for bandwidth improvement in IEEE 802.15.4 Wireless Sensor Networks », in *Wireless Days (WD), 2011 IFIP*, 2011, p. 1-6.
[22] O. D. Incel, S. Dulman, et P. Jansen, « Multi-channel Support for Dense Wireless Sensor Networking », in *Smart Sensing and Context*, P. Havinga, M. Lijding, N. Meratnia, et M. Wegdam, Éd. Springer Berlin Heidelberg, 2006, p. 1-14.
[23] van Hoesel L. F. W., « Sensors on speaking terms: Schedule-based medium access control protocols for wireless sensor networks », Ph.D. dissertation, University of Twente, Enschede, 2007.
[24] M. D. Jovanovic et G. L. Djordjevic, « TFMAC: Multi-channel MAC Protocol for Wireless Sensor Networks », in *8th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services, 2007. SIKS 2007*, 2007, p. 23-26.
[25] H. K. Le, D. Henriksson, et T. Abdelzaher, « A Practical Multi-channel Media Access Control Protocol for Wireless Sensor Networks », in *International Conference on Information Processing in Sensor Networks, 2008. IPSN '08*, 2008, p. 70-81.
[26] Y. Kim, H. Shin, et H. Cha, « Y-MAC: An Energy-Efficient Multi-channel MAC Protocol for Dense Wireless Sensor Networks », in *International Conference on Information Processing in Sensor Networks, 2008. IPSN '08*, 2008, p. 53-63.
[27] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, et T. F. Abdelzaher, « MMSN: Multi-Frequency Media Access Control for Wireless Sensor Networks », in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, 2006, p. 1 13.
[28] J. L. Hill et D. E. Culler, « Mica: a wireless platform for deeply embedded networks », *IEEE Micro*, vol. 22, n° 6, p. 12-24, nov. 2002.
[29] G. P. Halkes et K. G. Langendoen, « Crankshaft: An Energy-Efficient MAC-Protocol for Dense Wireless Sensor Networks », in *Wireless Sensor Networks*, K. Langendoen et T. Voigt, Éd. Springer Berlin Heidelberg, 2007, p. 228-244.

[30] Y. Wu, M. Keally, G. Zhou, et W. Mao, « Traffic-Aware Channel Assignment in Wireless Sensor Networks », in *Wireless Algorithms, Systems, and Applications*, B. Liu, A. Bestavros, D.-Z. Du, et J. Wang, Éd. Springer Berlin Heidelberg, 2009, p. 479-488.
[31] O. D. Incel, L. van Hoesel, P. Jansen, et P. Havinga, « MC-LMAC: A multi-channel MAC protocol for wireless sensor networks », *Ad Hoc Netw.*, vol. 9, n° 1, p. 73-94, janv. 2011.
[32] J. Borms, K. Steenhaut, et B. Lemmens, « Low-Overhead Dynamic Multi-channel MAC for Wireless Sensor Networks », in *Wireless Sensor Networks*, J. S. Silva, B. Krishnamachari, et F. Boavida, Éd. Springer Berlin Heidelberg, 2010, p. 81-96.
[33] M. Buettner, G. V. Yee, E. Anderson, et R. Han, « X-MAC: A Short Preamble MAC Protocol for Duty-cycled Wireless Sensor Networks », in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, New York, NY, USA, 2006, p. 307–320.
[34] « ZigBee Alliance > Home ». [En ligne]. Disponible sur: http://www.zigbee.org/.
[35] « La domotique en ZigBee par CityGrow, comment ça marche ? », *Blog Eavs Groupe*. .
[36] « zigbee alliance Archives », *News Domotiques by Domadoo*. .
[37] « ZigBee Health Care Certified Products ». [En ligne]. Disponible sur: http://www.zigbee.org/Products/ByStandard/ZigBeeHealthCare.aspx. [Consulté le: 06-nov-2014].
[38] M. J. Miller et N. H. Vaidya, « On-demand TDMA scheduling for energy conservation in sensor networks », 2004.
[39] Ed Callaway, « MAC Proposal for the Low Rate 802.15.4 Standard », *MOTOROLA*, 2001.
[40] « La technologie WirelessHART ». [En ligne]. Disponible sur: http://fr.hartcomm.org/hcp/tech/wihart/wireless_how_it_works.html. [Consulté le: 06-oct-2014].
[41] « untitled - AS-74_3199_wirelesshart_vs_isa100.11a.11a_-_the_format_war_hits_the_factory_floor.pdf ». .
[42] « ISA | The International Society of Automation ». [En ligne]. Disponible sur: https://www.isa.org/. [Consulté le: 07-oct-2014].
[43] « ISA-100.11a Wireless Standard for Industrial Automation Approved ». [En ligne]. Disponible sur: http://www.ihs.com/news/wireless/2009/isa-100-industrial-automation-091109.htm.
[44] « Web Exclusive: Analysis of wireless industrial automation standards: ISA-100.11a and WirelessHART - ISA ». [En ligne]. Disponible sur: https://www.isa.org/standards-publications/isa-publications/intech-magazine/2012/december/web-exclusive-analysis-wireless-industrial-automation-standards-isa-100-11a-wirelesshart/. [Consulté le: 07-oct-2014].
[45] Z. Shelby et C. Bormann, *6LoWPAN: The Wireless Embedded Internet*. Chichester, U.K: Wiley-Blackwell, 2009.
[46] « IEEE 802.15.4e WPAN Task Group ». [En ligne]. Disponible sur: http://www.ieee802.org/15/pub/TG4e.html.

[47] G. Chalhoub et M. Misson, « Cluster-tree based energy efficient protocol for wireless sensor networks », in *2010 International Conference on Networking, Sensing and Control (ICNSC)*, 2010, p. 664-669.
[48] P. Minet, S. Mahfoudh, G. Chalhoub, et A. Guitton, « Node Coloring in a Wireless Sensor Network with Unidirectional Links and Topology Changes », in *2010 IEEE Wireless Communications and Networking Conference (WCNC)*, 2010, p. 1-6.

Résumé

L'utilisation des canaux multiples améliore significativement les performances globales des réseaux de capteurs sans fil (RCSF) en permettant des transmissions parallèles sur plusieurs canaux. Cependant, la mise en place d'un protocole MAC multi-canal dans un réseau multi-sauts nécessite une méthode efficace d'allocation des canaux pour permettre une coordination entre les nœuds afin de partager les canaux disponibles. Dans cette thèse, nous nous intéressons à la conception d'un protocole MAC multi-canal qui répond aux exigences des applications de collecte de données à haut débit dans un réseau multi-saut. Pour ce faire, nous abordons d'abord les principales façons d'utiliser plusieurs canaux pour réaliser le partage du médium. Ensuite, nous proposons un protocole MAC multi-canal, répondant aux exigences des RCSF à haut débit, qui combine les trois techniques TDMA, CSMA et FDMA. En effet, nous utilisons une nouvelle méthode d'allocation des canaux qui permet aux nœuds de choisir le canal de réception le plus convenable dans leurs voisinages jusqu'à 3-sauts d'une façon distribuée afin de minimiser les effets des interférences et des collisions. Enfin, nous évaluons par simulation les performances de notre protocole et nous le comparons à d'autres protocoles proposés dans la littérature. Les résultats obtenus montrent l'efficacité de notre proposition dans les différents scenarios étudiés.

Mots clés : réseaux de capteurs sans fil, protocole MAC, multi-canal, multi-saut.

Abstract

The use of multi-channel significantly improves the overall network performance of wireless sensor networks (WSNs) by allowing parallel transmissions over multiple channels. However, the design of a multi-channel MAC protocol in a multi-hop network requires an efficient channel allocation method that allows the coordination between the nodes in order to share available channels. In this thesis, we focus on the conception of a multi-channel MAC protocol that meets the requirements of high data collection applications in a multi-hop network. In order to achieve this goal, we first present the main techniques to use multiple channels to realize medium access sharing. Then, we propose a multi-channel MAC protocol that meets the requirements of high data rate WSNs, which combines three techniques TDMA, FDMA and CSMA. Indeed, we use a new channel assignment method that enables nodes to choose the most convenient channel in their 3-hop neighborhood in a distributed manner in order to minimize the effects of interferences and collisions. Finally, we evaluate by simulation the performance of our protocol and we compare it to other protocols proposed in the literature. The results show the efficiency of our proposition in the different studied scenarios.

Keywords : wireless sensor network, MAC protocol, multi-channel, multi-hop.