RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministére de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Batna 2
Faculté des Mathématiques et de l'Informatique
Département d'Informatique

Thèse

Pour l'obtention du diplôme de **Doctorat en informatique**
Spécialité : SYSTÈMES D'INFORMATION ET DE COMMUNICATION

Sous le titre

UN PROTOCOLE DE TRANSPORT FIABLE POUR LES RÉSEAUX DE CAPTEURS SANS FIL

Présentée par Abderrezak BENYAHIA

Soutenue publiquement le 04/05/2019 devant le jury :

Dr. Hamouma MOUMEN (MC-A)	**Président**	*Université de Batna 2*
Pr. Azeddine BILAMI (Professeur)	**Rapporteur**	*Université de Batna 2*
Pr. Rachid BEGHDAD (Professeur)	**Examinateur**	*Université de Bejaia*
Pr. Mohamed BENMOHAMED (Professeur)	**Examinateur**	*Université de Constantine 2*
Pr. Allaoua CHAOUI (Professeur)	**Examinateur**	*Université de Constantine 2*
Dr. Larbi GUEZOULI (MC-A)	**Examinateur**	*Université de Batna 2*

REMERCIEMENTS

Tout d'abord je tiens à remercier mon cher directeur de thèse Pr. Azeddine BILAMI, Professeur au département d'informatique de l'université de Batna 2, directeur du laboratoire LaSTIC, pour ses qualités scientifiques, professionnelles et pédagogiques qui m'ont permis d'apprendre beaucoup de leçons dans ma vie professionnelle. Grâce à ses conseils rigoureux et à sa disponibilité, j'ai atteint les objectifs principaux de ce travail. J'insiste sur ses qualités humaines, sa sympathie et son sens d'écoute qui m'ont donné à chaque fois un nouveau souffle pour continuer et progresser dans mes recherches.

Mes expressions et respects sincère au Dr Hamouma MOUMEN, maître de conférences -A- à l'université de Batna 2, qui m'a fait honneur de présider le jury. En plus, toutes mes gratitudes et remerciements aux membres de jury :

- *Rachid BEGHDAD, Professeur à l'université de Bejaia ;*
- *Mohamed BENMOHAMED, Professeur à l'université de Constantine 2 ;*
- *Alaoua CHAOUI, Professeur à l'université de Constantine 2 ;*
- *Larbi GUEZOULI, Maître de conférences -A- à l'université de Batna 2 ;*

qui ont accepté de juger le travail. Je leurs suis très reconnaissant pour l'intérêt qu'ils ont porté à ma thèse.

Je remercie également le Dr. Maamar SEDRATI pour ces conseils, son aide et sa collaboration dans cette thèse. Un grand remerciement pour sa contribution et ses recommandations qui ont amélioré convenablement ce travail. Mes remerciements les plus distingués à mon collègue Mr Mohamed Amine MERZOUG qui a contribué dans la formulation du manuscrit de la thèse.

Je remercie vivement mon cher père pour son soutien et son encouragement qui m'ont aidé à la réalisation de ce travail. Un grand merci à ma femme qui a contribué dans la théorie de cette thèse. En fait, je remercie ma petite famille (ma femme et mon petit-fils MOHAMED ABDENNOUR) qui m'ont supporté pendant ces dernières années de recherche, sans oublier mes frères et ma sœur.

A la fin, mes respects et mes remerciements à tous mes amis, mes collègues et à tous ceux qui m'ont encouragé et soutenu dans la réalisation de ce travail.

ملخص

النقل الموثوق للمعلومات يعتبر مطلب مهم بالنسبة للكثير من التطبيقات في شبكات الاستشعار اللاسلكية. في حقيقة الأمر، بروتوكول النقل الموثوق لهذه التطبيقات لا يجب أن يعتبر أساسا موثوقية النقل واستهلاك الطاقة فقط، لكن يجب أيضا اعتبار نسبة احتلال الذاكرة ومدة تسليم المعلومات. في الآونة الأخيرة، كثيرا من أعمال البحث وجهة نحو هذا الغرض، بينما البرتوكولات المقترحة تعالج بعض الجوانب لهذا المشكل وتهمل جوانب أخرى. على العكس في هذه الأطروحة نعرض حلا للنقل مبتكر صمم خصيصا للتزويد بموثوقية 100% دون تدهور العوامل الأخرى. عن طريق عدة آليات، نحاول أن نصل إلى هذا الهدف مع اجتناب الازدحام في الشبكة وضمان الأداء الحسن من حيث استهلاك الطاقة، مدة التسليم وتخزين الذاكرة. البروتوكول المقترح، المسمى (Congestion Avoidance with Reliable Transmission and Energy Efficiency) CARTEE، يحقق أهدافه من خلال آليات كثيرة، تعرف بـ: إرسال يعتمد على نافذة منزلقة مثبتة، اعتراف متناوب ضمني/صريح، تقنية جديدة لكشف لازدحام وتعديل نسبة الإرسال موزعة. لتقييم البروتوكول المقترح، قدنا تجارب معتمدة على المحاكات باستعمال برنامج المحاكات ns-3. النتائج المتحصل عليها تؤكد فعالية البروتوكول CARTEE وقابليته التطوير وكما تثبت أنه يتغلب أدائه على بروتوكولات النقل المقترحة مؤخرا من حيث الموثوقية، تجنب الازدحام، نسبة احتلال الذاكرة ومن حيث الكمون.

---كلمات افتتاحية: *تجنب الازدحام، تطبيقات تدفق المعلومات، تعديل نسبة الإرسال، نقل موثوق، نافذة منزلقة، شبكات الاستشعار اللاسلكية متعددة الوسائل.*

Abstract

Reliable data transport is an essential requirement for many multimedia applications in wireless sensor networks. Actually, an efficient transport protocol for these applications must take into account not only reliability and energy consumption factors but also memory occupancy and data delivery delay. Recently, many research works have been conducted in this area, however the proposed protocols treat some of these aspects and neglect others. Contrarily, in this thesis we present a novel transport solution designed to provide 100% reliability without making light of other factors. Through different mechanisms, we attempt to reach this objective with congestion avoidance and good performances in terms of energy consumption, delivery delay, and memory storage. The proposed protocol, called CARTEE (Congestion Avoidance with Reliable Transmission and Energy Efficiency), attains these goals through several mechanisms, namely: fixed sliding window transmission, alternative implicit/explicit acknowledgement, a new congestion detection technique, and distributed transmission rate adjustment. To evaluate the proposed protocol, we have conducted simulations using ns-3. The obtained results confirm the efficiency and scalability of CARTEE and demonstrate that it outperforms the recent proposed transport protocols in terms of reliability, congestion avoidance, data cache occupancy, and latency.
—*Keywords:* Congestion avoidance, data streaming application, transmission rate adaptation, reliable transport, sliding window, Wireless Multimedia Sensor Networks .

Résumé

Le transport fiable de données est une exigence essentielle pour la majorité des applications multimédia dans les réseaux de capteurs sans fil. Un protocole de transport adéquat pour ces applications ne doit pas seulement prendre en considération les facteurs de fiabilité et de consommation énergétique, mais aussi l'occupation de la mémoire et le délai de livraison de données. Récemment, plusieurs travaux de recherche ont été orientés vers cette tendance, cependant les protocoles proposés traitent quelques aspects du problème et négligent d'autres. Par contre dans cette thèse la solution de transport proposée offre une fiabilité de 100% sans dégrader les autres facteurs de performances en se basant sur plusieurs mécanismes, on essaie d'atteindre cet objectif tout en évitant la congestion et assurant de meilleures performances en termes de consommation énergétique, de délai de livraison et de stockage mémoire. Le protocole proposé, nommé CARTEE (Congestion Avoidance with Reliable Transmission and Energy Efficiency), atteint ses objectifs sur la base de plusieurs mécanismes, à savoir : une transmission à base de fenêtre glissante fixe, un acquittement alternatif implicite/explicite, une nouvelle technique de détection de congestion et un ajustement du taux de transmission distribué. L'évaluation des performances de ce protocole est faite en utilisant le simulateur ns-3. Les résultats obtenus confirment l'efficacité et l'évolutivité du protocole CARTEE et démontrent qu'il surpasse les performances des protocoles de transport récemment proposés en termes de fiabilité, d'évitement de la congestion, d'occupation de la mémoire et en termes de latence.
—*Mots clés :* Evitement de la congestion, application de flux de données, adaptation du taux de transmission, transport fiable, fenêtre glissante, Les réseaux de capteurs multimédia sans fil .

ACRONYMES

ADC Analog to Digital Conversion. 18, 21–24

AIMD Additive Increase/Multiplicative Decrease. 39

CARTEE Congestion Avoidance with Reliable Transport and Energy Efficiency. 13, 58–60, 62, 67, 69, 70, 72–81, 84, 89, 92, 95, 97, 98, 101, 102, 106–119

CODA Congestion Detection and Avoidance. 43–45, 48, 55, 56, 59, 61, 67–69, 77

DARPA Defense Advanced Research Projects Agency. 14

ERTP Energy-effecient and Reliable Transport Protocol. 50, 52, 55, 56, 60, 61, 63, 67, 74, 108–111, 113, 114, 116, 117

ESRT Event-to-Sink Reliable Transport. 27, 45, 46, 55, 56, 59, 61, 64, 67–69, 77, 108–111, 113, 114, 116, 117

EWMA Exponential Weighted Moving Average. 49, 75, 78

HDRTP Hybrid and Dynamic Reliable Transport Protocol. 53, 54, 60, 62, 108–111, 113, 114, 116, 117

IUSS Integrated Undersea Surveillance System. 14

LEACH Less Energy Adaptive Clustering Hierarchy. 28

MEMS Micro Electro Mechanical Systems. 22

NACK Implicit Acknowledgment. 67, 70

NACK Negative Acknowledgment. 55, 63

OOR Optimal Operating Region. 45, 46, 108–110

PEGASIS Power-Efficient Gathering in Sensor Information Systems. 28

PSFQ Pump Slowly Fetch Quickly. 27, 41, 42, 53–56, 59–64, 67, 108–111, 113, 114, 116, 117

RAIT Reliable Asynchronous Image Transfert. 52, 53, 60, 62

RCRT Rate-Controlled Reliable Transport. 48, 49, 59, 61–63, 67, 68, 77

RCSF Réseau de Capteurs sans fil. 9, 11–16, 20–24, 26–35, 37–44, 47, 50, 52, 53, 55, 57–60, 62, 67, 74, 76, 77, 79, 81–85, 89, 101, 103, 106, 107, 118, 119

RMST Reliable Multi-Segment Transport. 27, 47, 48, 55, 56, 59, 61, 63, 64, 67

RTO Retransmission Timeout. 50–52, 61, 74

SHARC Super Harvard Architecture. 24, 25

SIMD Single Instruction Multiple Data. 25

WSN Wirless Sensor Network. 14

GLOSSAIRE

acquittement implicite C'est un acquittement qui repose sur l'interception de la transmission du nœud voisin pour affirmer la bonne réception d'un segment sans utiliser un message de contrôle.. 34–36, 62, 67, 69, 70, 75, 76, 108, 113

congestion La congestion dans le réseau représente le fait d'un nœud qui reçoit un flux avec une fréquence qui excède sa capacité de stockage du flux en attente de traitement. Cette fréquence provoque une perte de données due à une congestion dans le réseau.. 9, 11, 15, 23, 33–37, 39, 40, 42–46, 48–50, 52, 53, 55–59, 61–64, 66–70, 72, 76–80, 95, 106, 109, 112, 116, 117

fenêtre glissante C'est un mécanisme de transmission utilisé dans TCP afin de réduire le délai de livraison de données. Ce mécanisme transmis des segments dans un intervalle de transmission et attend leurs acquittements pendant un intervalle d'acquittement.. 34, 35, 39, 52, 53, 95, 98, 103, 106, 108, 114

IEEE 802.11 Ensemble de normes qui spécifient le fonctionnement des réseaux locaux sans fil (WiFi).. 25, 27, 34

IEEE 802.15.1 Norme défini par IEEE qui spécifie des communications permettant un échange de données bidirectionnel à courte distance à base de radio fréquences.. 27

IEEE 802.15.4 Un protocole de communication défini par IEEE. Il inclus la spécification des réseaux personels sans fil à faible taux de transmission et de consommation ou LR WPAN (Low Rate Wireless Personal Area Network). 25, 27, 35, 62, 75, 78, 82, 83, 87, 88, 106, 107

LTE Un module du simulateur ns-3 qui fournit une implémentation de base d'un appareil LTE (Long term Evolution) des normes de téléphones mobiles GSM/EDGE, TD-SCDMA et UMTS. Il inclus le modèle de propagation ainsi que la couche MAC et physique.. 88, 93

NesC Un langage de programmation dérivé du langage C. Il est destiné pour des unitées de calcul possédant une capacité de stockage et de traitement limitée (i.e., nœuds capteurs). 83, 86, 93, 94

Nyquist Pour un signal à bande-limitée, le taux de Nyquist ajuste une bande faible dans la fréquence d'échantillonnage. D'où, le taux d'échantillonnage minimum doit être deux fois la bande passante du signal.. 23

Python Un Un language de programmation orienté interprété, multi-paradigme et multiplatformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet (similaire au language tcl et SmallTalk).. 87, 88, 94, 95

Send-And-Wait Appelé aussi Stop-And-Wait, est un mécanisme de transmission utilisé par la majorité des interfaces réseaux tel qu'IEEE 802.3 et IEEE 802.11. Il est basé sur un acquittement positif lors de la transmission de chaque trame.. 34, 35, 55, 61, 63–66, 72

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GÉNÉRALE

Les réseaux de capteurs sans fil (RCSF) occupent une place importante dans les réseaux informatiques. Les RCSF sont des systèmes distribués auto-organisés constitués de milliers de nœuds capteurs pour surveiller différents phénomènes. Les RCSF ont enregistré un grand succès avec l'apparition des applications multimédia et des applications de surveillance. Ces applications sont d'un grand intérêt pour les chercheurs dans différents domaines (i.e., médecine, géologie, environnement, ... etc.). Grâce à leurs caractéristiques d'autonomie et d'auto-organisation, l'utilisation de ces réseaux ne cesse de croitre. Pour atteindre leurs objectifs de déploiement, les nœuds capteurs effectuent deux tâches principales : détection et communication. La tâche de détection consiste à la transformation d'un évènement surveillé à une valeur numérique. Alors que la tâche de communication consiste à l'acheminement de l'information détectée vers un nœud particulier dans le réseau appelé « nœud puits » qui gère et collecte les données détectées pour l'utilisateur final.
La collection des données dans les RCSF nécessite une collaboration entre les nœuds du réseau, dont lequel chaque nœud envoie ses propres données et pourrait acheminer les données de ses voisins vers le puits. Ce comportement signifie qu'un nœud dans le réseau joue à la fois le rôle d'une source et d'un relai de données. Ces deux rôles intensifient le traitement et la communication de données au niveau des nœuds surtout lorsque la fréquence de détection est très élevée (i.e., big data). Les transmissions élevées augmentent le taux d'erreur dans les bits de données et la congestion, ce qui conduit à une dégradation des performances du réseau, ce qui n'est pas toléré par certaines applications (i.e., applications de flux de données et applications multimédia).
Les applications de flux de données [1] (i.e. audio et vidéo) où la donnée doit être fragmentée et réassemblée, exigent une livraison de données fiable, parce que la perte de données n'est pas tolérée et peut affecter le bon fonctionnement de l'application [2]. Dans un tel cas, les transmissions de bout-en-bout doivent être contrôlées pour s'assurer que le puits les a correctement reçues. Les données perdues doivent être retransmises pour compléter les trous dans le flux de données ; les causes (sources) de pertes de données doivent être détectées et évitées. Dans les RCSF, où le réseau est conçu pour être fonctionnel pour une longue période de temps en utilisant des ressources énergétiques limitées. En d'autres termes, les principaux objectifs de conception est d'utiliser raisonnablement les ressources des nœuds.
Au niveau des nœuds de détection, la communication est une tâche gourmande en termes d'énergie et dans la plupart des cas, les deux autres tâches de détection et de traitement sont négligées [3, 4, 5]. L'utilisation d'une solution de livraison de données fiable (i.e., la retransmission après congestion ou perte de données) génère une consommation énergétique supplémentaire. Pour alléger ce problème et réduire la communication, la solution de livraison de données fiable doit transmettre le flux sans trous en réduisant les retransmissions dans la mesure du possible. Un deuxième objectif consiste de concevoir une solution de transport fiable qui est indépendante des autres couches et qui peut fonctionner avec n'importe quel protocole de routage.

Motivations : Plusieurs solutions ont été proposées dans la littérature pour les applications de flux de données [1, 6, 7, 8] qui se focalisent sur certains aspects de fiabilité mais négligent d'autres. La première catégorie de solutions [1, 6] négligent la congestion et utilisent les acquittements implicites ou explicites négatifs pour régler le problème de perte de données. Dans ce cas, les congestions et les retransmissions non-nécessaires se produisent fréquemment. La seconde catégorie de solutions [8] considère la congestion en utilisant un mécanisme d'adaptation du taux de transmission centralisé, qui n'assure pas l'exploitation optimale des ressources du réseau. Le reste des solutions [7, 9] remédient à cet inconvénient (i.e., la non-utilisation optimale des ressources du réseau) et implémentent un mécanisme d'adaptation du taux de transmission distribué induisant une surcharge du réseau et une forte consommation énergétique.

Contribution : cette thèse présente une solution originale de transport fiable appelé Congestion Avoidance with Reliable Transport and Energy Efficiency (CARTEE) [10] conçue pour les applications de flux de données. Pour atteindre l'objectif tracé, notre proposition considère différents cas de gaspillage d'énergie (i.e., congestion, retransmissions non nécessaires, etc.). Le protocole CARTEE utilise un mécanisme de détection de congestion en conjonction avec un mécanisme d'adaptation du taux de transmission. Afin de réduire la latence, cette solution utilise un mécanisme de transmission à base de fenêtre glissante. Dans le souci de récupération des pertes de données d'une manière efficace, CARTEE utilise une combinaison de mécanismes d'acquittement implicite et explicite. En outre et afin d'alléger le mécanisme de transmission, cette solution renforce ce mécanisme avec un contrôle de congestion basé sur une adaptation distribuée du taux de transmission. Les mécanismes d'acquittement et d'adaptation du taux de transmission permettent au protocole CARTEE de réduire la charge de contrôle dans le réseau. L'évaluation des performances de la solution proposée a été effectuée en utilisant le simulateur ns-3 [11, 12].

Structuration de la thèse : cette thèse est organisée comme suit. Après une introduction générale, le premier chapitre présente un état de l'art sur les réseaux de capteurs sans fil et leurs applications. Le deuxième chapitre met l'accent sur le problème de transport dans les RCSF, et décrit les solutions proposées pour les applications de flux de données. L'étude préliminaire et la description détaillée de la solution proposée fera l'objet du troisième chapitre. Les approches utilisées pour évaluer les performances de notre proposition toute en justifiant le choix de l'approche utilisée dans l'évaluation feront l'objet du chapitre quatre. Ce chapitre abordera également l'implémentation de la solution dans le simulateur ns-3. Le dernier chapitre sera consacré à l'évaluation des performances du protocole CARTEE en se basant sur les résultats de simulation. En dernier lieu, on trouve une conclusion qui synthétise les objectifs de cette thèse et énonce les perspectives possibles à ce travail.

CHAPITRE 1

LES RÉSEAUX DE CAPTEURS SANS FIL

1.1 Introduction

L'utilisation des réseaux de capteurs sans fil (RCSF en anglais Wirless Sensor Network (WSN)) ne cesse d'accroitre durant cette dernière décennie. Plusieurs recherches ont été orientées vers le développement de ces réseaux pour élargir leurs champs d'utilisation. Selon leurs caractéristiques, les RCSF permettent de collecter des mesures (e.g. température, pression, lumière ou humidité) d'un tel ou tel phénomène à des endroits inaccessible (e.g. zone volcanique, zone de radiation nucléaire ou forte région montagneuse). En conjonction avec l'augmentation du taux d'utilisation des réseaux de capteurs, les recherches dans ce domaine visent à optimiser les protocoles et les algorithmes conçus pour ces réseaux (protocoles de transport, protocole de routage, algorithmes d'accès au support de transmission, ...etc.) pour améliorer leurs performances.
On peut considérer un RCSF comme un cas particulier des réseaux ad hoc sans fil, où les nœuds sont des capteurs (ou senseurs), déployés à l'intérieur. Ils peuvent être considérés aussi comme une extrapolation extrême de deux tendances générales en informatique : la miniaturisation (construire des ordinateurs plus petits) et l'interconnexion (réseau).
Les RCSF héritent des caractéristiques de leurs homologues réseaux mobiles ad hoc. Tout d'abord les capteurs sont très petits et donc plus sensible à une défaillance matérielle. Il est à noter que la batterie (ou l'énergie) est la ressource la plus importante dans un capteur, ce qui implique que les capteurs peuvent aussi échouer à cause d'une faible énergie. Deuxièmement, les capteurs sont déployés dans un champ avec une haute densité pour prolonger la durée de vie du réseau ce qui facilite la communication multi-saut entre capteurs, et donc les capteurs permettent d'économiser leur énergie en transmettant ou en acheminant leurs données captées sur de courtes distances. Troisièmement, la topologie du réseau peut changer fréquemment lorsque les capteurs rejoignent ou quittent le réseau (e.g. Mobilité des capteurs). Ainsi, les protocoles conçus pour les réseaux de capteurs doivent tenir compte de tous ces éléments, qui sont inhérent à ces types de réseaux, afin qu'ils restent opérationnels pour des durées de vie plus importantes.

1.2 Les domaines d'applications des réseaux de capteurs

Les réseaux de capteurs ont commencés à se développer dans le projet Defense Advanced Research Projects Agency (DARPA), qui s'est matérialisé sous forme Integrated Undersea Surveillance System (IUSS) déployé plus tard par l'US Navy dans la guerre sous-marine (détection des sous-marins ennemis) [13]. Aujourd'hui les réseaux de capteurs sont utilisés dans divers domaines tel que :

- Les applications militaires – le scannage du terrain, l'imagerie, la surveillance.
- La médecine – la télésurveillance des patients, notamment les personnes âgées, les implants de la peau pour la détection précoce de diverses maladies et mesure des paramètres sanguin, l'informatique portable, les capsules à avaler pour l'imagerie vidéo à l'intérieur du corps du patient.
- La circulation automobile – des capteurs dans les voitures et les infrastructures de transport, pour le suivi de la congestion et la prévention des accidents de la route.
- Domotique – mesure de la température et de l'humidité, le contrôle automatique de la climatisation, systèmes d'alarmes, . . . etc.
- Agriculture – mesure d'humidité et de température, contrôle automatique sur asperseurs, le suivi des mouvements du bétail, . . . etc. [14].
- L'environnement – surveillance du gaz dangereux, tremblement de terre précoce et la détection de l'incendie.

1.3 Caractéristiques des RCSF

Les RCSF appartiennent aux réseaux de communication sans fil, d'où ils sont considérés comme un cas particulier des réseaux ad hoc sans fil. Ces réseaux héritent des caractéristiques de leurs homologues réseaux ad hoc, tel que la mobilité, la topologie dynamique et les ressources énergétiques limitées. En outre, les RCSF possèdent quelques propriétés distinctes, parmi eux on peut citer [15] :

- Capacités de traitement : En plus de ressources énergétiques limitées, les RCSF possèdent aussi un espace mémoire de données et de programme ainsi qu'un processeur avec des capacités restreintes.
- Energie de la batterie : Les capteurs deviennent souvent invalides et abandonnés due à l'énergie épuisée. Donc, les protocoles et les algorithmes doivent prendre en considération la conservation de l'énergie. Ainsi, la communication consomme plus d'énergie par rapport aux traitements du processeur.
- Capacités de communication : La bande passante de communication des RCSF est étroite et changeante, ainsi que le rayon de portée qui peut atteindre les 100 mètres. La communication d'un capteur est plus vulnérable aux effets de l'environnement tel que l'affaiblissement, la réflexion et la dispersion du signal transmit. Donc, il est difficile de garder l'exécution régulière du réseau. Ceci signifie que les applications des RCSF doivent être robustes et tolérantes aux pannes.
- Réseau dynamique : Les nœuds capteurs peuvent quitter le réseau par épuisement de l'énergie ou due à d'autres échecs. Ceci entraine un changement dans la topologie du réseau, d'où il est nécessaire d'avoir des fonctions de reconfiguration et d'auto-organisation.
- Absence d'infrastructure : le déploiement des nœuds de capteurs sans fil ne nécessite pas d'installer une infrastructure réseau. Les nœuds de capteurs peuvent collaborer pour s'ajuster entre eux. Donc, ils peuvent rapidement et dynamiquement former un réseau autonome.
- Communication multi-saut : les nœuds capteurs peuvent uniquement communiquer avec leurs voisins dans un RCSF. Si un nœud veut communiquer avec des nœuds hors de sa portée, il doit établir un chemin multi-saut à travers des nœuds intermédiaires.

- Pertinence de l'application : Les RCSF sont des collecteurs centralisés de données, de communication multi saut et un modèle de trafic multipoint-à-point. Les RCSF sont différents des réseaux classiques et leurs mode de fonctionnement repose principalement sur les objectifs de l'application utilisée dans le réseau, ainsi, leurs ultime travail consiste à acquérir des données environnementales. Différents applications RCSF traitent les signaux physiques avec différentes manières, un protocole de routage ne peut être appliqué pour toutes les applications. La pertinence de l'application et l'un des importants défis des RCSF.

1.4 Technologie des RCSF

Les nœuds capteurs sont des petits appareils, qui utilisent une mémoire limitée, équipés d'un microcontrôleur utilisé pour les systèmes embarqués (typiquement 8 bits, 4-5 Mhz). En plus d'un microcontrôleur, un nœud capteur dispose généralement d'une petite quantité de mémoire RAM (quelque Kilo-octets), un émetteur-récepteur radio, un certain nombre de capteurs analogique reliés au microcontrôleur (par le biais d'un circuit convertisseur analogique/numérique) et une source d'énergie pour alimenter tous ces composants (typiquement deux batteries ; figure 1.1) [16].

Les capacités des nœuds capteurs sont limitées pour réaliser quelques tâches : ils peuvent seulement enregistrer

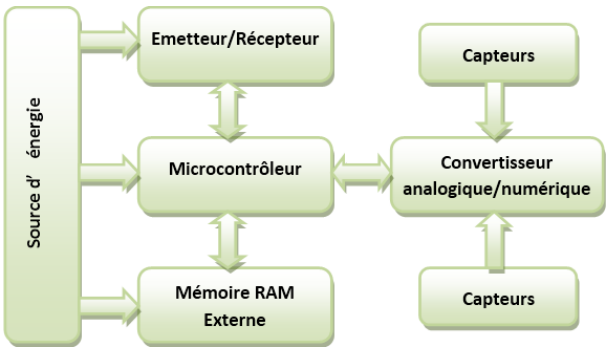

FIGURE 1.1 : Diagramme de block d'un nœud capteur

une certaine qualité de l'environnement (tel que la température, la pression atmosphérique, l'humidité… etc.), mais en raison de leurs limitations en terme de vitesse de calcul et d'affichage (la plupart des nœuds capteurs possèdent quelque diodes LED), les nœuds capteurs nécessitent d'autres interactions pour traiter et afficher les données qu'ils recueillent (figure 1.2).

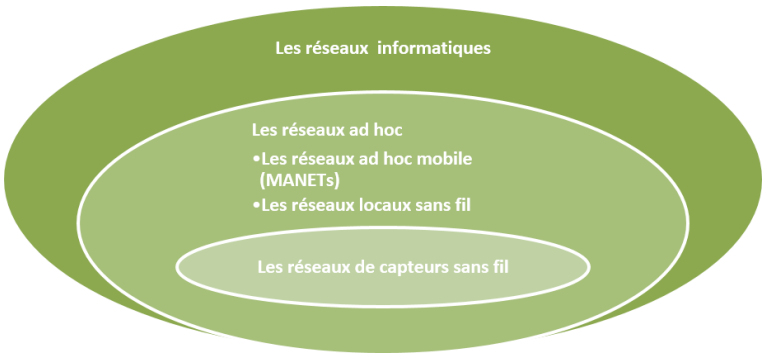

FIGURE 1.2 : La position des réseaux de capteurs dans les réseaux informatique

Un réseau de capteurs comprend un grand nombre de nœuds capteurs qui collectent des informations, et un ou plusieurs nœuds passerelles qui acheminent les données collectées à leur destination finale. Par exemple, un nœud passerelle capteurs peut être connecté à un ordinateur ou à un routeur Internet qui pourrait acheminer les données vers la destination finale (figure 1.3).

Les ressources énergétiques limitées des nœuds capteurs et le déploiement des réseaux de capteurs à grande

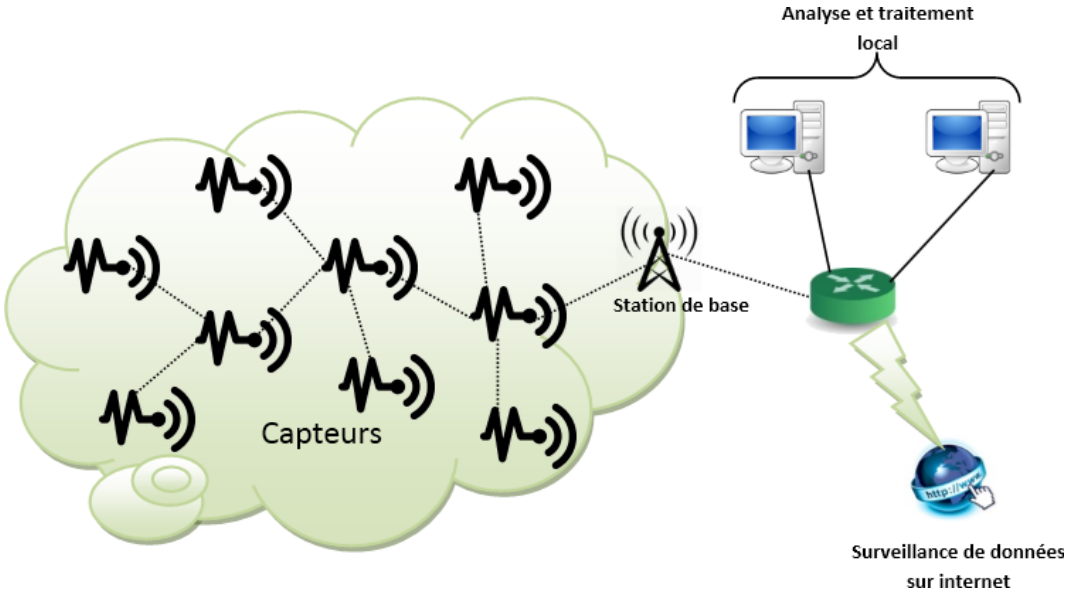

FIGURE 1.3 : Exemple d'utilisation d'un RCSF

échelle rend le changement des batteries épuisées sur les nœuds capteurs une opération compliquée, ce qui signifie que les implications suivantes sont importantes :

- Le coût élevé de la communication – le taux d'énergie utilisé par les nœuds capteurs pour la communication est beaucoup plus élevé que le taux utilisé pour la détection et le traitement. Cela signifie que des techniques telles que la compression des données et l'agrégation sont très importantes dans les réseaux de capteurs et les techniques de routage spéciales sont nécessaires pour éviter de mettre beaucoup de chemins de routage sur un seul nœud du réseau.
- La nature ad hoc des réseaux capteurs – éventuellement, certains nœuds qui ont épuisés leurs énergies chutent le réseau. Le réseau doit être conçu pour résister à ces événements, même si d'autres nœuds capteurs sont ajoutés au réseau, ces derniers doivent être intégrés d'une manière transparente sans intervention de l'opérateur. Ces considérations mettent fortement les réseaux de capteurs dans un contexte plus large des réseaux informatiques ad hoc.

1.5 Arrière-plan et définitions

Un nœud capteur sans fil peut réaliser des fonctionnalités principales en utilisant des ressources limitées. La première fonctionnalité s'intéresse à son environnement pour mesurer (détecter) le niveau d'un événement et traduire ce dernier sous forme d'une information numérique. La deuxième consiste à transmettre (communiquer) l'information détectée vers une station de base ou l'acheminer vers un autre nœud capteur. Ses fonctionnalités vont être décrites dans les prochaines sections.

1.5.1 Capteurs et détection

La détection est une technique utilisée pour recueillir des informations sur un objet physique, y compris l'apparition d'événements (par exemple, les changements dans l'état comme une dégradation dans la température ou la pression). Un objet qui effectue une tâche de détection est appelé capteur. Par exemple, le corps humain est équipé de capteurs qui sont capables de détecter des informations optiques de l'environnement (les yeux), des informations acoustiques tels que les sons (les oreilles), et les odeurs (nez). Ceci est un exemple de capteurs à distance, c.-à-d. qu'ils n'ont pas besoin de toucher l'objet surveillé pour recueillir des informations. D'un point de vue technique, un capteur est un dispositif permettant de traduire les paramètres ou les événements dans le monde physique en signaux qui peuvent être mesurés et analysés. Un autre terme couramment utilisé est le transducteur, qui est souvent utilisé pour décrire un dispositif qui convertit l'énergie d'une forme à une autre. Un capteur est donc un type de transducteur qui converti l'énergie dans le monde physique en énergie électrique qui peut être transmis à un système informatique ou à un contrôleur. Un exemple des mesures effectuées dans une tâche de détection (ou d'acquisition de données) est illustrée dans la figure 1.4. Les phénomènes du monde physique sont observés par un dispositif de détection. Les signaux électriques qui en résultent ne sont souvent pas prêts pour un traitement immédiat, donc ils passent par une phase de conditionnement du signal, ici, une série d'opérations peut être appliqué au signal détecté pour le préparer à une utilisation ultérieure. Par exemple, les signaux ont souvent besoin d'amplification (ou atténuation) pour modifier l'amplitude du signal afin d'être correctement converti d'un signal analogique vers un signal numérique. En plus, le conditionnement du signal nécessite souvent des opérations de filtrage pour éliminer les bruits indésirables dans certains intervalles de fréquences. Après le conditionnement, le signal analogique sera transformé en signal numérique en utilisant un convertisseur analogique-numérique (Analog to Digital Conversion (ADC)). A la fin, le signale numérique et prêt pour un traitement ultérieur (stockage ou visualisation) [5].

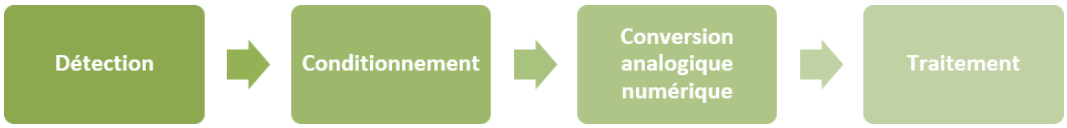

FIGURE 1.4 : Acquisition des données

Caractéristiques de détection

Il est important de comprendre les caractéristiques des détecteurs pour choisir le détecteur adéquat à une tâche de surveillance donnée. Les caractéristiques suivantes sont nécessaires dans le choix du détecteur [16].

La fonction de transfert : Elle représente la relation entre le signale physique d'entrée et le signale électrique de sortie

Hystérèse : Elle représente la capacité d'un capteur de suivre les changements du signal d'entré que ce soit une augmentation ou une diminution.

Linéarité : Représente la déviation de la courbe à partir de la fonction idéale de transfert.

Sensibilité : Représente le rapport entre un petit changement dans le signal d'entré et le changement résultat du signal de sortie.

Précision : Représente la plus grande erreur prévue entre le signal idéal de sortie et celui obtenu.

Plage dynamique : Représente la plage de signal d'entré qui peut être convertie avec précision en signal de sortie.

Bruit : Tous les capteurs produisent un bruit en plus du signal de sortie. Pour les applications qui nécessitent une haute précision, la détection de la quantité du bruit introduit par un capteur peut être plus importante.

Résolution : C'est la variation du signal d'entré qui peut être détectée.

Bande passante : Le changement dans le signal d'entré nécessite un certain temps pour que le changement soit détecté dans le signal de sortie. Ce temps est appelé le temps de réponse. Plusieurs capteurs possèdent un temps de décroissance, ce qui représente le temps après une modification progressive du signal physique pour que la sortie diminue à sa valeur originale. Les valeurs réciproques au temps de réponse et au temps de désintégration sont appelées des fréquences de coupure inférieure et supérieure. La bande passante d'un capteur est la plage de fréquence entre ces deux fréquences.

Classification des capteurs

Les capteurs doivent être choisis pour une application selon les propriétés physiques surveillés. Par exemple, de telles propriétés comprennent la température, la pression, la lumière ou l'humidité. Outre les propriétés physiques, la classification des capteurs peut être basée sur une variété d'autres méthodes, par exemple, s'ils ont besoin d'une alimentation externe. Si les capteurs nécessitent une alimentation externe, ils sont considérés comme des capteurs actifs. Autrement dit, ils doivent émettre une sorte d'énergie (ex, microondes, lumière, son) pour déclencher une réponse ou détecter un changement dans l'énergie du signal transmis. D'autre part, les capteurs passifs détectent l'énergie dans l'environnement et tirent leur puissance de cet apport énergétique – par exemple, les capteurs passifs infrarouge (PIR) mesure le rayonnant de la lumière infrarouge à partir de la proximité de l'objet [5].
Le classement des capteurs peut être également basé sur les méthodes qu'ils appliquent et les phénomènes électriques qu'ils utilisent pour convertir les propriétés physiques en signaux électriques. Dans cette classification, on distingue les catégories suivantes :

- Les capteurs résistifs : s'appuient sur des changements à la résistivité électrique d'un conducteur basé sur les propriétés physiques telles que la température.
- Les capteurs capacitifs : Un principe similaire peut être appliqué à des capteurs capacitifs, qui peuvent être utilisés pour mesurer le mouvement, la proximité, l'accélération, la pression, les champs électriques, la composition chimique et la profondeur liquide.
- Les capteurs inductifs : sont basés sur le principe de l'inductance électrique, qui s'explique par une force électromagnétique qui s'est induit par un courant fluctuant. L'inductance est déterminée par les dimensions du capteur (la surface de section transversale, la longueur de la bobine), le nombre de spires de la bobine et la perméabilité du noyau. Toute variation de ces paramètres (par exemple, causés par les mouvements du noyau dans la bobine) modifie l'inductance. Les capteurs inductifs sont souvent utilisés pour mesurer la proximité, la position, la force, la pression, la température et l'accélération.
- Les capteurs piézo-électriques : utilisent l'effet piézoélectrique de certains matériaux (par exemple, des cristaux et de certaines céramiques) pour mesurer la pression, la force, l'intensité et l'accélération. Lorsqu'une pression est appliquée à un tel matériau, il provoque une déformation mécanique et un déplacement de charge, proportionnel à la quantité de pression. Le principal avantage de dispositifs piézoélec-

trique par rapport aux autres approches est que l'effet piézo-électrique n'est pas sensible aux champs électromagnétiques ou aux radiations.

1.5.2 Communication et réseau

Les capteurs nécessitent une communication sans fil pour qu'ils puissent transférer les données collectées à une station de traitement centrale, ceci est important parce que les applications réseau nécessitent des centaines ou des milliers de nœuds capteurs, souvent déployés dans des zones distantes et inaccessibles. Cependant, un nœud capteur sans fil n'a pas seulement un composant de détection, mais il possède aussi une capacité du traitement de la communication et de stockage. Avec ces améliorations, un nœud capteur n'est pas souvent responsable de la collection des données, mais aussi le responsable de l'analyse du réseau, de la corrélation et de la fusion de ces propres données et les données des autres nœuds capteurs. Lorsque plusieurs capteurs surveillent en collaboration un grand environnement physique, ils forment un réseau de capteurs sans fil (RCSF). Les nœuds capteurs communiquent non seulement entre eux mais aussi avec une station de base (Sink) en utilisant leurs ondes radio, qu'ils lui permettent de diffuser leurs données à des systèmes de traitement distant, de visualisation, d'analyse et de stockage. Par exemple la figure 1.5 montre deux champs de détection surveillant deux régions géographiques différentes et se connectent à internet en utilisant leurs stations de bases.
Les capacités des nœuds capteurs sont différentes dans un RCSF allant d'un simple phénomène de surveillance

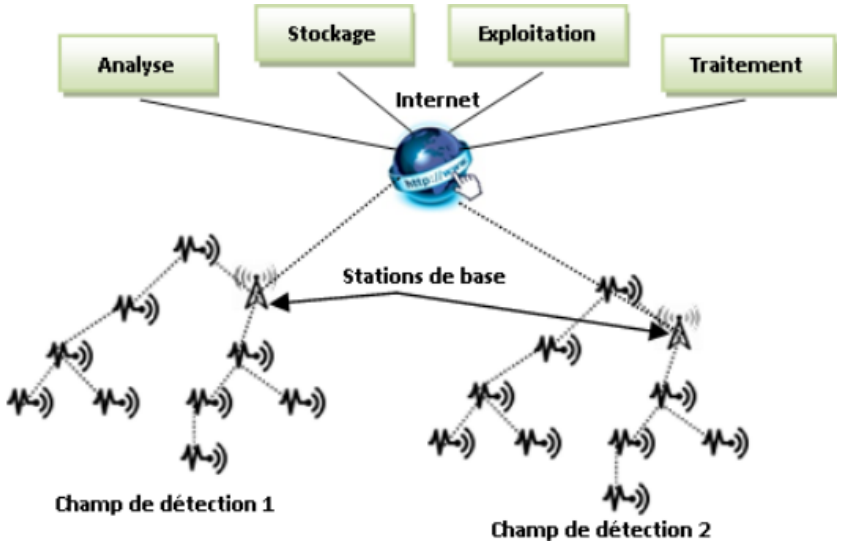

FIGURE 1.5 : Réseaux de capteurs sans fil

à des dispositifs plus complexes de détection (i.e. acoustique, optique, magnétique). Ils peuvent également être différents dans leurs capacités de communication, ils utilisent par exemple les technologies telles que l'ultrason, l'infrarouge ou les fréquences radio avec une latence et un taux de données différent. Quand des nœuds simples peuvent seulement collecter et communiquer les données d'un environnement à observer, d'autres dispositifs plus puissants (ex, les dispositifs avec une large capacité de traitement, d'énergie et de stockage) peuvent également effectuer des fonctions de traitement et d'agrégation. Ces dispositifs prennent souvent des responsabilités supplémentaires dans un réseau RCSF, par exemple, ils peuvent former des squelettes de communications qui peuvent être utilisés par d'autres dispositifs de détection ayant des ressources limitées pour atteindre la station de base. Enfin, d'autres dispositifs peuvent avoir accès à des supports technologiques supplémentaire, par exemple, un récepteur GPS, qui leurs permettent de déterminer avec précision leurs positions. Cependant, ces systèmes consomment souvent beaucoup d'énergie pour être réalisable dans les nœuds capteurs à faible-coût et à faible-puissance.

1.6 Architecture d'un nœud capteur

Un nœud capteur est l'élément actif dans un RCSF, parce qu'il est le responsable de la détection, du traitement et de la communication. Donc, il inclut et exécute des algorithmes de traitement ainsi qu'un ensemble de protocoles de communication. Les ressources disponibles dans un capteur influent sur la qualité, la taille et la fréquence d'informations détectées. D'où, la conception et l'implémentation d'un RCSF est une tâche décisive.

Un nœud capteur est constitué de quatre composants (sous-systèmes) : détection, traitement, communication et gestion d'énergie. Un concepteur possède un ensemble d'options pour construire et mettre ces sous-systèmes dans un nœud programmable unifié. Le processeur est l'élément central dans un nœud et le choix d'un processeur détermine le compromis entre la flexibilité et l'efficacité. Il existe plusieurs processeurs optionnels : microcontrôleur, processeur de signal numérique, circuit intégré d'une application spécifique et le FPGA (Field Programmable Gate Array).

Le sous-système de détection est connecté au processeur en utilisant un ADC. Il existe plusieurs façons pour réaliser cette connexion, parmi eux, on peut connecter plusieurs détecteurs numériques avec un ADC multicanal qui inclut plusieurs ADC de grande vitesse dans un seul circuit intégré. Cependant, ce type de circuits intégrés produit une diaphonie et augmente le bruit. D'autres capteurs incluent un ADC intégré qui peut être connecté directement avec le processeur à travers un protocole standard puce-à-puce.

Le sous-système de communication peut interagir aussi avec le processeur. D'autres émetteur/récepteur possèdent leurs propres processeurs pour effectuer le traitement de signal de la couche physique et de la liaison de données. Ceci est pour alléger le sous-système de traitement. Le sous-système de communication est le sous-système le plus énergivore et sa consommation énergétique doit être régularisée. Presque tous les émetteur/récepteur disponibles fournissent des fonctions de contrôle pour basculer l'émetteur/récepteur entre plusieurs niveaux d'opérations : état repos et sommeil. Le sous-système d'énergie fournit l'énergie pour tous les sous-systèmes pour alimenter leurs composants électroniques (tel que les oscillateurs, les amplificateurs, les registres et les compteurs).

La figure 1.6 présente les différents sous-systèmes d'un nœud capteur sans fil et les différentes techniques

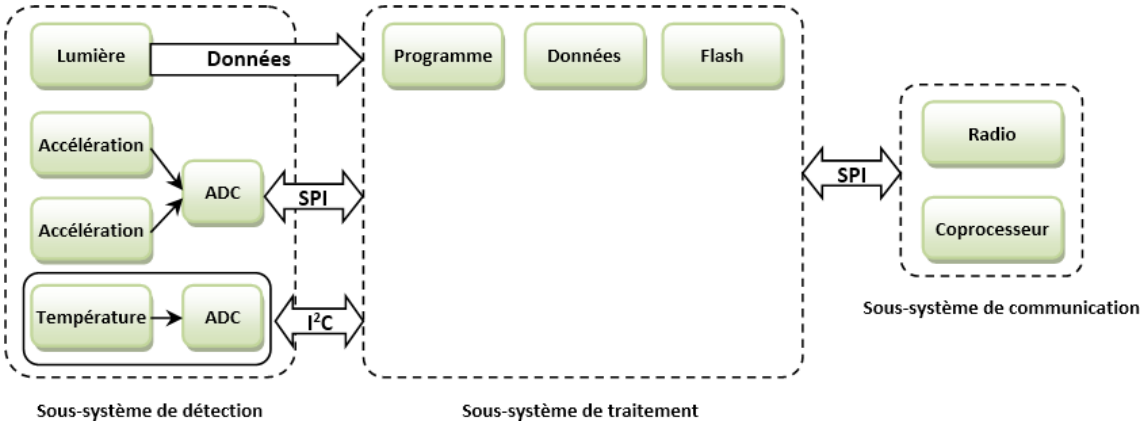

FIGURE 1.6 : Architecture d'un nœud capteur sans fil

d'intégration. Le sous-système d'énergie et leurs relations ne sont pas présentés dans cette figure [5].

1.6.1 Le sous-système de détection

Le sous système de détection intègre un ou plusieurs capteurs physiques et fournit aussi des convertisseurs ADC ainsi que des mécanismes de multiplexage pour les partager. Les détecteurs présentent une interface

entre le monde virtuel et le monde réel. Avec l'apparition des systèmes micro-électromécanique (Micro Electro Mechanical Systems (MEMS)), la détection est devenue un processus omniprésent. Actuellement, il existe un excédent de détecteurs qui mesurent et quantifient des attributs physiques. Un détecteur physique contient un transducteur qui converti une forme d'énergie typiquement à une énergie électrique (voltage). La sortie de se transducteur est un signal analogique possédant une magnitude continue selon la fonction de temps. Donc, un convertisseur ADC est nécessaire pour traduire le signal analogique du sous-système de détection vers un signal numérique reconnu par le processeur.
Le tableau 1.1 fournit un résumé des types de détecteurs utilisés dans les RCSF. Il fournit aussi un résumé concis des événements détectés et leurs aspects.

Détecteur	**Données détectés**	**Evénements détectés**
Détecteur d'émission acoustique	Les ondes élastiques générées par l'énergie libérée lors de la propagation des fissures	Pour Mesurer les changements et les déplacements structurels
Détecteur acoustique	La vibration de la pression acoustique	– Détection de véhicules – Mesurer les irrégularités structurelles – Contamination des gaz au niveau ppm – Mesurer la teneur en eau d'un sol
Accéléromètre	L'accélération 2D et 3D des objets	– Les activités volcaniques et les ondes sismiques de 1er et 2em degrés – La rigidité des structures due aux changements modaux des structures – La rigidité des os, des membres et les articulations – La fluctuation motrice dans la maladie de Parkinson – L'irrégularité dans le rail, l'essieu ou les roues d'un train – Le défaut dans les objets fragiles pendant leurs déplacements
Détecteur de capacité	Concentration de soluté	
ECG	Rythme cardiaque	
EEG	Activité électrique du cerveau	
EMG	Activité du muscle	
Détecteur électrique/électromagnétiqu	Capacité de Résistivité/conductivité électrique ou l'inductance affectée par la composition du sol testé	Mesure le contenu et la distribution des éléments nutritifs
Gyroscope	Vitesse angulaire	La détection de la position angulaire au niveau des appareils rotatifs
Détecteur électromagnétique		Présence, vitesse et densité des véhicules dans une rue ; congestion

Détecteur	Données détectés	Evénements détectés
Oxymètre	L'oxygénation du sang pour l'hémoglobine d'un patient	L'effort cardiovasculaire et les tendances d'effort relatif à une activité Détecteur pH Concentration d'ions hydrogène Indique le contenu d'acide et d'alcalin dans l'eau – mesure de la propreté
Spectroscopie photo acoustique	Détection du gaz	Pour détecter les fuites du gaz dans une canalisation
Cylindre piézoélectrique	La rapidité du gaz	Une fuite qui produit un bruit de haute fréquence générant une vibration
Détecteur d'humidité du sol	Humidité du sol	Gestion des engrais et de l'eau
Détecteur de température	Température	
Détecteur baromètre	Pression exercé sur on fluide	
Détecteur infrarouge passif	Rayonnement infrarouge des objets	Détection des mouvements
Détecteur séismique	Mesure les ondes séismiques primaires et secondaires	Pour détecter le tremblement de terre
Détecteur d'oxygène	La quantité et proportion d'oxygène dans le sang	

TABLE 1.1 : Résumé des détecteurs utilisés dans les RCSF

Le convertisseur analogique-numérique

Le convertisseur analogique numérique (ADC) convertit l'entrée d'un détecteur (signal analogique continue) vers un signal numérique. Ce processus nécessite deux étapes :

1. Le signal analogique doit être quantifié (e.g. convertit du signale estimé continu vers un signal estimé discret que ce soit dans le temps et dans la magnitude). La décision la plus importante dans cette étape est de déterminer le nombre de valeurs discrètes permises. Deux facteurs influencent cette décision : (a) la fréquence de la magnitude du signal ; et (b) les ressources de stockage et de traitement disponibles.

2. La fréquence d'échantillonnage. Dans l'ingénierie de communication et le traitement de signal numérique, cette fréquence est décidée par le taux de Nyquist[1]. Cependant, dans les RCSF, le taux de Nyquist n'est pas suffisant. Le sur-échantillonnage est nécessaire à cause du bruit.

Un ADC est caractérisé par une résolution de conversion de signal exprimé en nombre de bits pouvant codés le signal numérique en sortie. Dans la sélection d'un ADC, la connaissance du processus ou de l'activité à surveiller est importante. Par exemple, dans une zone de surveillance où l'intervalle de la propriété thermique est entre [-20°C,+80°C], il est adéquate de choisir une résolution de l'ADC et un détecteur garantissant à mesurer correctement le changement de ses valeurs thermiques. Si on considère un changement de 0.5°C, il

[1]Pour un signal à bande-limitée, le taux de Nyquist ajuste une bande faible dans la fréquence d'échantillonnage. D'où, le taux d'échantillonnage minimum doit être deux fois la bande passante du signal.

est suffisant de choisir un ADC avec une résolution de 8 bits. D'autre part, si le changement est de 0.0625°C, alors l'ADC doit avoir une résolution de 11 bits [5].

1.6.2 Le sous-système de traitement

Le sous-système de traitement rassemble tous les autres sous-systèmes et quelques périphériques additionnels. Sa raison principale est d'exécuter les instructions appartenant à la détection, la communication et l'auto-organisation. Parmi beaucoup de choses, ce sous-système se compose d'un circuit processeur, d'une mémoire non-volatile (généralement une mémoire flash interne) pour sauvegarder des instructions d'un programme, une mémoire dynamique pour stocker temporairement des données détectées et une horloge interne.
Vu la variété de processeur disponible pour construire un RCSF, le choix du processeur est crucial, parce qu'il affecte le coût, la flexibilité, les performances et la consommation énergétique d'un nœud. Si la tâche de détection est bien définie au début et elle ne change pas dans le temps, le concepteur peut choisir entre le FPGA ou le processeur de signal numérique. Ces processeurs sont très efficaces en termes de consommation énergétique. Cependant, ceci n'est pas toujours le processeur à usage général, d'où la conception et l'implémentation peuvent être compliquées et coûteuses.
Actuellement les nœuds capteurs disponibles utilisent des microcontrôleurs. Les RCSF sont des technologies émergentes et la communauté de recherche est toujours active pour développer des protocoles de communication et des algorithmes de traitement de signal efficaces en termes de consommation énergétique. Ceci nécessite une installation et une mise à jour de code dynamique, d'où l'utilisation du microcontrôleur est l'option la plus convenable [5].

Aperçu sur l'architecture processeur

Les concepteurs des nœuds capteurs concentrent principalement sur des architectures de processeurs permettant d'optimiser le temps d'exécution des algorithmes. L'intérêt de cette optimisation vise à réduite le transfert de données entre le processeur et la mémoire. Donc, un effort considérable est consacré pour minimiser le temps d'exécution des instructions destinées pour le transfert de données et les opérations arithmétique et logique.
Le sous-système de traitement peut être conçu en utilisant l'un des trois architectures des ordinateurs suivantes : Von Neumann, Harvard, et Super-Harvard (Super Harvard Architecture (SHARC)). L'architecture Von Neumann un seul espace mémoire utilisé par les instructions et les données du programme. Elle fournit un seul bus pour le transfert de données entre le processeur et la mémoire. Cette architecture possède relativement une vitesse de traitement lente parce que chaque transfert de données nécessite un intervalle d'horloge séparé. La figure 1.7 illustre l'architecture de Von Neumann.

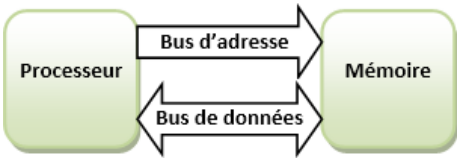

FIGURE 1.7 : L'architecure de Von Neumann

L'architecture de Harvard a modifié l'architecture de Von Neumann en fournissant un espace mémoire séparé pour stocker les instructions et les données du programme. Chaque espace interagit avec le processeur avec un bus séparé. De cette manière, les instructions et les données du programme peuvent être accédées simultanément. Cette architecture supporte également l'opération une seule instruction, plusieurs données (Single Instruction Multiple Data (SIMD)). Elle peut facilement supporter les opérations multitâches. La figure 1.8 schématise l'architecture de Harvard.

La nouvelle génération des architectures de processeurs est l'architecture super-Harvard, connue comme

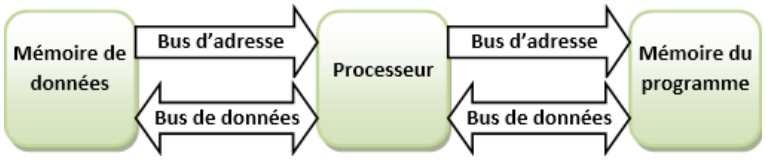

FIGURE 1.8 : L'architecture de Harvard

SHARC. C'est une extension de l'architecture Harvard en ajoutant deux composants principales à son prédécesseur, ces composants fournissent des alternatives pour accéder aux dispositifs d'E/S à partir du sous-système de traitement. L'un des composants est une instruction cache interne qui améliore les performances de l'unité de traitement. Il est utilisé pour stocker les instructions fréquemment utilisées pour réduire les opérations de recherche d'instructions répétées. La figure 1.9 présente l'architecture de Super-Harvard.

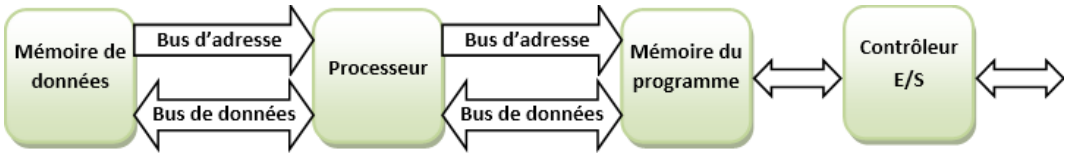

FIGURE 1.9 : L'architecture de Super-Harvard

1.6.3 Le sous-système de communication

Le transfert de données rapide et à faible consommation entre les sous-systèmes d'un nœud capteur sans fil est critique pour l'efficacité global du réseau mis en place. Cependant, la taille pratique d'un nœud capteur pose des restrictions dans les bus de transfert. La communication à travers un bus parallèle est plus rapide que la transmission série, mais la transmission parallèle nécessite plus d'espace. En raison de la taille du nœud, les bus parallèles n'ont jamais été supportés dans la conception d'un nœud capteur sans fil.

Le choix, est donc, entre les interfaces séries tel que l'interface de périphérique série (SPI), l'E/S d'objectif général (GPIO), l'E/S de données sécurisées (SDIO), le circuit inter-intégré (I2C) et le bus série universel (USB). Parmi ceci, les bus les plus utilisés sont le SPI et l'I2C [5].

Interfaces de communication

La famille de la norme IEEE 802.11 a été introduite en 1997 et c'est la technologie de réseau sans fil la plus courante pour les systèmes mobiles. Elle utilise différentes bandes de fréquences, par exemple, la bande 2.4 Ghz est utilisée par IEEE 802.11b et IEEE 802.11g, alors que le protocole IEEE 802.11a utilise une bande de fréquence de 5 Ghz. IEEE 802.11 a été fréquemment utilisée dans les premiers réseaux de capteurs sans fil et elle peut être encore trouvée dans les réseaux actuels lorsque les exigences en termes de bande passante sont élevées (par exemple les capteurs multimédia). Cependant le taux d'énergie élevé des réseaux basés sur IEEE 802.11 rend cette norme inadaptée pour les réseaux de capteurs à faible énergie. Le taux de données

typiquement nécessaire dans les réseaux de capteurs est comparable à la bande passante fournit par le modem, par conséquent, le taux de données fournit par IEEE 802.11 est typiquement plus élevé que le nécessaire. Pour cette raison le standard IEEE 802.15.4 [17] a été conçu spécialement pour les communications à courte portée dans les réseaux de capteurs de faible puissance, et qui est actuellement supportée par la plupart des développeurs de nœuds capteurs commerciales et académiques.
Lorsque les plages de transmission radio des nœuds capteurs sont assez larges et les capteurs peuvent transmettre leurs données à la station de base, ils peuvent former une topologie en étoile comme il est montré sur la figure 1.10 (à gauche). Dans cette topologie, chaque capteur communique directement avec la station de base en utilisant un seul saut. Cependant, les réseaux de capteurs couvrent souvent des zones géographiques plus larges et la puissance de transmission radio devrait être maintenue au minimum afin de conserver l'énergie, par conséquent, la communication multi-saut est le cas le plus fréquent dans les réseaux de capteurs comme illustré dans la figure 1.10 (à droite). Dans cette topologie maillée, les nœuds capteurs doivent non seulement détecter et diffuser leurs données, mais aussi servir autant que relais pour d'autres nœuds capteurs, qui doivent collaborer pour propager les données détectées vers la station de base (i.e. opération de routage). Le routage consiste à trouver un chemin multi-saut à partir du nœud expéditeur jusqu'à la station de base, c'est l'une des issues les plus importantes qui ont attiré l'attention de plusieurs communautés de recherches. Quand un nœud sert comme un relai pour plusieurs chemins, il a souvent l'occasion d'analyser et de prétraiter les données détectées dans le réseau, ce qui conduit à l'élimination des informations redondantes ou l'agrégation des données qui peuvent être plus petit que les données d'origines.

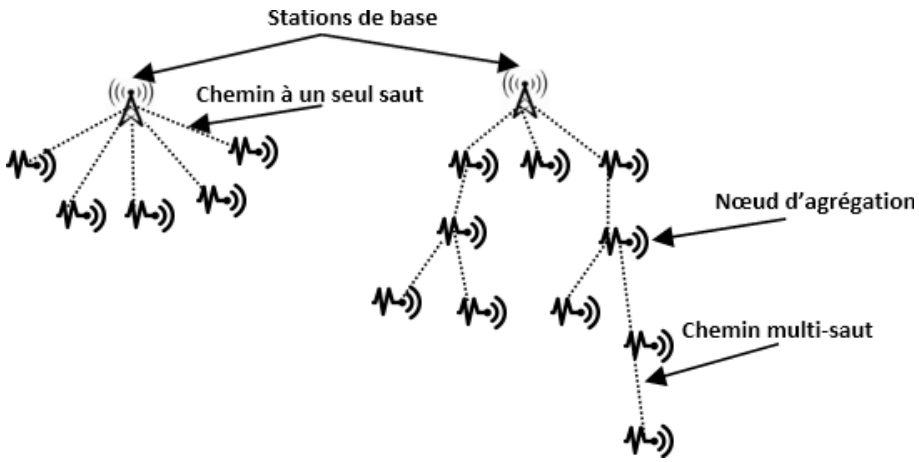

FIGURE 1.10 : La communication à un seul saut contre le multi-saut

Pile protocolaire d'un nœud

Les chercheurs ont développé plusieurs protocoles spécifiques aux RCSF avec l'économie d'énergie comme principale préoccupation. Le tableau 1.2 représente un modèle de la pile protocolaire qui peut être utilisé pour décrire les entités de la couche ainsi que les protocoles typiques qui sont applicables aux RCSF [18].
La communication dans les RCSF est reliée aux protocoles de la couche transport, réseau et la liaison de données.

Couche	Tâche	Protocoles
Application	• Application de traitement • Agrégation • Traitement des requêtes • Base de données	Une variété de protocoles de la couche application
Transport	• Dissémination et accumulation • Fragmentation et réassemblage • Cache • Stockage • Contrôle de fiabilité	• PSFQ • RMST • ESRT • CODA • RCRT
Réseau	• Gestion des topologies adaptatives • Routage	• Diffusion dirigée • LEACH • GAF • SPEED
Liaisons de données	• Partage du canal • Timing • Localisation	• WiFi IEEE 802.11 b/g • Bluetooth IEEE 802.15.1 • ZigBee IEEE 802.15.4
Physique	• Canal de communication • Détection • Actionnement • Traitement de signal	

TABLE 1.2 : Pile protocolaire d'un nœud capteur sans fil

Protocole de transport Généralement, les protocoles de transport s'intéressent à la disposition d'un service de communication fiable pour la couche application. Ceci est l'objectif du protocole Pump Slowly Fetch Quickly (PSFQ) [6]. C'est un protocole adaptatif qui réalise une correction locale des erreurs en utilisant un acquittement saut-par-saut. Un autre protocole qui vise la fiabilité de communication est Reliable Multi-Segment Transport (RMST) [2] qui repose aussi sur des acquittements saut-par-saut. Cependant, ce protocole est conçu pour fonctionner en conjonction avec la diffusion dirigée.
Une approche intéressante est utilisée par le protocole Event-to-Sink Reliable Transport (ESRT) [8]. Ce protocole a été conçu pour les réseaux de capteurs à base d'événement et il change la focalisation des protocoles de transports traditionnels. Les auteurs affirment que la fiabilité dans les RCSF dépend de la tâche de détection des événements. ESRT suppose qu'un événement doit être détecté lorsque la station de base reçoit un nombre minimum de rapport d'événements à partir des nœuds capteurs. Si ce seuil n'est pas atteint, la station de base ne reconnait (conçoit) pas l'événement. Donc, ESRT ajuste le taux de transmission de chaque nœud d'une façon que le seuil désiré est atteint et les événements sont faiblement détectés.

Protocoles de routage Le routage est le processus de transmission d'un paquet de données à partir d'une source vers une destination, peut-être par le biais de nœuds intermédiaires. Ceci est connu comme une communication point-à-point. La communication dans les RCSF peut être divisée en trois cas : à partir des nœuds capteurs vers un nœud de surveillance, entre des nœuds voisins et à partir du nœud de surveillance vers les nœuds capteurs. Le premier cas est utilisé pour envoyer les données détectées par des nœuds capteurs vers une application de surveillance. Le deuxième cas arrive souvent lorsqu'une collaboration est nécessaire entre des nœuds. Le troisième cas est souvent utilisé pour publier un important fragment d'informations aux nœuds capteurs (e.g. le changement du mode de fonctionnement, la diffusion d'un nouveau intérêt au réseau, activer/désactiver un ou plusieurs nœuds ou envoyer des requêtes au réseau).
Les algorithmes de routage dans les RCSF peuvent être divisés en trois types : routage plat, routage hiérarchique, et routage adaptative. Dans le routage plat les rôles des nœuds sont supposés tous identiques. Un routage simple pour le routage plat est de former un routage arborescent. Les auteurs [19] ont fourni une analyse des frontières sur le coût de l'énergie qui peut être obtenu avec l'agrégation des données en utilisant une topologie en arbre. L'algorithme InFRA [20] a construit un routage arborescent en établissant une organisation hybride du réseau dans laquelle les nœuds sources sont organisés en clusters. La communication du cluster vers la station de base se passe de façon multi-saut. La topologie résultante est une heuristique distribuée du problème de l'arbre de Steiner[2].
Les nœuds dans le routage hiérarchique peuvent avoir un rôle statique ou dynamique. Plusieurs algorithmes ont été proposés pour le routage hiérarchique dans la littérature. Less Energy Adaptive Clustering Hierarchy (LEACH) [21] est un protocole basé sur des clusters qui tourne aléatoirement les têtes du cluster pour distribuer uniformément la charge d'énergie sur le réseau. Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [22] est une amélioration de LEACH dans laquelle les nœuds forment des chaînes, et chaque nœud communique uniquement avec le voisin le plus proche qui à son tour, transmet les paquets à la station de base. Le routage adaptatif change son comportement selon différentes conditions d'application et du réseau, tel que les ressources énergétiques disponibles. La diffusion dirigée [23] est le protocole pionner qui essaie de chercher un meilleur chemin à partir des sources vers la station de base qui peut recevoir des données à partir de plusieurs chemins avec des fréquences de livraison de données différentes. Si le meilleur chemin échoue, d'autres chemins avec des fréquences de livraison faible assurent la livraison de données. Les auteurs [24] proposent une solution de routage évolué de la diffusion dirigée, qui essaie de découvrir et maintenir des chemins alternatifs, pour connecter les sources à la station de base, pour rendre le réseau plus tolérant aux pannes [25].

Protocoles de la couche MAC La couche liaison ou contrôle d'accès au média (MAC) contrôle l'accès au média de communication d'un nœud en utilisant plusieurs techniques (tel que la contention et la division du temps). Fondamentalement, la couche MAC doit gérer les canaux disponibles au niveau d'un nœud pour éviter les collisions et les erreurs de communications [25].
La majorité des solutions essaient de fournir une communication fiable à faible consommation. Dans ce sens, les auteurs [26] ont utilisés des techniques pour prévoir une meilleure taille de la trame pour réduire la taille du paquet et la consommation d'énergie. Pour éviter la transmission des paquets sous des conditions non fiables, les auteurs [27] ont appliqués des techniques de filtrage pour estimer le bruit ambiant et déterminer si le canal est libre pour la transmission. Les auteurs [28] proposent un protocole de la couche MAC basé sur une solution d'ordonnancement de la logique flou pour améliorer les protocoles existants à faible consommation. Les variables d'entrées de ce protocole sont les taux des nœuds qui ont (a) des tampons surchargés (b) un taux élevé de transmissions erronés (c) rencontrés une transmission échouée.

[2] L'arbre de Steiner est un problème d'optimisation combinatoire de « Jakob Steiner ». Cette optimisation est similaire au problème de l'arbre couvrant minimal dans la recherche opérationnelle

1.7 Défis et contraintes

Les réseaux de capteurs sans fil partagent plusieurs similarités avec les systèmes distribués, ces similitudes rendent les réseaux de capteurs soumises à des contraintes et des issues communs. Ces contraintes influent sur la conception des RCSF conduisant à des protocoles et des algorithmes qui diffèrent de leurs homologues dans d'autres systèmes distribués. Dans cette section on va décrire les contraintes de conceptions les plus importants des RCSF [5].

1.7.1 Contrainte d'énergie

L'utilisation des ressources énergétiques imitées au niveau des nœuds capteurs sans fil est la contrainte la plus souvent associée à la conception des réseaux de capteurs. Généralement, ils sont alimentés avec des batteries, qui nécessitent un chargement ou un replacement lorsque l'énergie est épuisée. La batterie influe sur le fonctionnement d'un nœud capteur, surtout lorsqu'elle n'est pas rechargeable. Ceci influe aussi la durée de vie du réseau, ainsi que la tâche de surveillance. La durée de la tâche de surveillance dépend de la nature de l'application utilisée pour la surveillance. Par exemple, les scientifiques qui surveillent les mouvements glacials exigent des capteurs qui fonctionnent pour des années alors que les détecteurs dans un champ de bataille nécessitent quelques heures ou quelques jours de fonctionnement.
En conséquence, le premier défi de conception des RCSF (souvent le plus important) est de garantir une consommation énergétique efficace. En fait, cette exigence est pertinente dans tous les aspects de conception de solutions destinées pour ces réseaux. Par exemple les choix et les décisions prisent au niveau de la couche physique affectent la consommation énergétique du dispositif global ainsi que la conception des couches supérieurs [29].
Vu que, la couche MAC qui est responsable de fournir le canal aux nœuds capteurs, on trouve des stratégies de communication MAC basées sur la contention. Dans ces stratégies, un nœud capteur peut accéder au canal de communication à tous moment, ceci conduit potentiellement à des collisions entre plusieurs nœuds. Ces défis doivent être adressés par la couche MAC pour assurer que les transmissions sont éventuellement réussies. Les défauts de ces approches résident dans la consommation énergétique et le délai engendré par les collisions et les mécanismes de récupération. Donc, d'autres protocoles de la couche MAC des réseaux de capteurs utilisent la contention-libre, où l'accès au média est strictement réglementé pour éliminer les collisions et pour permettre aux nœuds capteur de fermer leurs radios lorsqu'aucune communication n'est prévu.
Sachant que, la couche réseau est responsable de la découverte des chemins à partir des nœuds capteurs vers la station de base, les caractéristiques du chemin (tel que la longueur en nombre de saut, l'énergie nécessaire de transmission et l'énergie disponible au niveau des nœuds intermédiaires) déterminent le coût de la communication multi-saut. L'exploitation efficace des ressources énergétiques au niveau de la couche réseau influe sur la conception du système d'exploitation, du middleware et aussi les applications. Par exemple, l'agrégation dans la couche réseau est utilisée pour éliminer les données redondantes ou pour mesurer plusieurs lectures détectées. Ceci est réalisé en utilisant un compromis entre le traitement et la communication, ce qui apporte un gain considérable dans la consommation énergétique [3, 30].

1.7.2 Autogestion

Les applications RCSF sont conçues pour fonctionner dans un environnement dur, contrôlé à distance sans aucune infrastructure ou une possibilité de maintenance ou de réparation. Donc, les nœuds capteurs doivent s'organiser automatiquement, se collaborer et s'adapter aux échecs sans aucune intervention humaine.

Déploiement ad hoc

Une variété d'applications RCSF ne tolère pas un endroit déterminé pour gérer les nœuds capteurs. Ceci est le cas d'un RCSF déployé à une zone distante ou inaccessible. Par exemple, les capteurs qui servent à surveiller un champ de bataille ou une région sinistrée pourront être jetés dans les zones d'intérêt, mais plusieurs capteurs ne peuvent pas être actifs pour une longue durée de temps. Cependant, d'une manière autonome, les nœuds survivant doivent réaliser des configurations qui incluent la communication avec les nœuds voisins, la détermination de leurs positions et l'établissement de leurs responsabilités de détection. Le mode de fonctionnement des nœuds capteurs peut différer en fonction de ces informations, par exemple, la position du nœud et le nombre de ces voisins peut déterminer la quantité et le type d'information qui doit générer ou acheminer pour les autres nœuds du réseau.

Fonctionnement incontrôlé

Plusieurs RCSF fonctionnent d'une manière autonome sans aucune intervention humaine. Un tel dispositif autogéré doit surveiller ses alentours, s'adapter aux changements de l'environnement et coopérer avec les dispositifs voisins pour former des topologies ou pour réaliser les stratégies de détection, de traitement et de communication [31]. L'autogestion se retrouve dans les RCSF sous plusieurs formes. La première forme est l'auto-organisation qui se réfère à la capacité du réseau pour s'adapter aux paramètres de configuration du système et son état environnemental. Par exemple, un dispositif de détection choisi sa puissance de transmission pour maintenir un certain degré de connectivité (i.e., l'augmentation de la puissance de transmission permet à un nœud d'atteindre plus de voisins). La deuxième, c'est l'auto-optimisation qui se réfère à la capacité d'un dispositif de détection pour utiliser ces ressources d'une manière optimale. La troisième forme est l'auto-protection ; elle représente la capacité d'un dispositif de détection pour se protéger contre les attaques et les intrusions. La dernière forme, l'auto entretien, qui permet aux dispositifs de détection à découvrir, identifier et réagir aux perturbations. Dans les RCSF, toutes ces formes d'autogestion doivent être prises en considération dans la phase de conception et d'implémentation pour éviter le gaspillage énergétique.

1.7.3 Communication sans fil

La nécessité des communications sans fil pose plusieurs défis dans la conception des RCSF. Par exemple, l'atténuation du signal impose des limitations dans la portée de la transmission radio due aux obstacles pouvant dégrader l'amplitude du signal durant sa propagation. En conséquence, une distance importante entre les nœuds capteurs et la station de base engendre plus de puissance de transmission. Donc, il est raisonnable de diviser les longues distances en plusieurs petites distances, ce qui conduit aux défis des communications et du routage multi-saut. La communication multi-saut nécessite une coopération inter-nœuds pour identifier des chemins efficaces. Ce défi persiste aussi dans les réseaux qui utilisent le principe du rapport cyclique pour conserver l'énergie, où plusieurs nœuds utilisent une politique de conservation d'énergie où la radio bascule en veille si elle n'est pas utilisée. En conséquence, un nœud peut rater des messages transmissent de ses voisins où il est considéré autant que relai pour ces messages. Il existe aussi des réseaux qui proposent des stratégies de réveille à la demande [32] pour assurer que les nœuds peuvent être réveillés au besoin. Généralement, ce fonctionnement nécessite l'utilisation de la radio en deux modes. La premier mode, radio à puissance faible, est utilisée pour recevoir des messages de réveille, et le deuxième mode qui exploite la radio à une puissance forte utilisé pour répondre à un appel de réveille. Une autre stratégie qui est le rapport cyclique adaptif [33], sur laquelle quelques nœuds du réseau qui rentrent en veille pendant la période de pause et un autre sous-ensemble de nœuds du réseau restent actifs pour former le réseau principal.

1.7.4 Gestion décentralisée

La densité du réseau et les contraintes d'énergie empêchent l'utilisation des algorithmes centralisés afin d'implémenter des protocoles pour la gestion du réseau (i.e., protocoles de routage, protocoles d'accès au support de communication). Au lieu de cela, les nœuds capteurs doivent collaborer avec leurs voisins pour prendre des décisions localisées, c.-à-d. sans connaissances globales. Les résultats de ces algorithmes décentralisés (ou distribués) ne peuvent pas être optimales, mais ils peuvent être plus efficaces dans la consommation énergétique par rapport aux solutions centralisés. Si on considère le routage comme un exemple de solutions centralisées et distribuées. Dans une approche centralisée, la station de base s'occupe de la collecte d'informations à partir des nœuds capteurs du réseau, établi des chemins optimales (e.g. en terme d'énergie) et informe chaque nœud de son chemin. Alors qu'une approche distribuée permet à chaque nœud du réseau à prendre des décisions de routage basées sur les informations locales limitées (e.g. la liste de ces nœuds voisins, les distances vers la station de base). Bien que ces approches décentralisées puissent conduire à des chemins non optimaux, elles peuvent réduire significativement le coût de gestion.

1.7.5 Contraintes de conception

Bien que les capacités des systèmes de traitement traditionnel augmentent rapidement, l'objectif principal de la conception des capteurs sans fil et de créer des dispositifs très petits, moins cher et plus efficace. Mais le besoin d'une petite forme et d'une faible consommation énergétique empêche l'intégration d'autres composants désirables, tel que le récepteur GPS. Ces exigences et contraintes influent également sur plusieurs niveaux de la conception logicielle, par exemple, les systèmes d'exploitation doivent avoir une petite zone mémoire et doivent gérer leurs ressources d'une manière efficace. Cependant le manque de fonctionnalités matérielles avancées (e.g., le support d'exécution parallèle) facilite la conception des petits systèmes d'exploitation efficace. Les contraintes du matériel de détection affectent aussi la conception de plusieurs algorithmes et protocoles exécutés dans les RCSF. Par exemple, la table de routage qui conclue des entrées pour chaque éventuelle destination dans un réseau peut être plus large pour être stockée dans la mémoire d'un nœud capteur. Au lieu de cela, uniquement une petite quantité de données (tel que la liste des voisins) peut être stockée dans la mémoire d'un nœud. Donc, plusieurs solutions et architectures logicielles (système d'exploitation, middleware, protocoles de routage) doivent être conçues pour fonctionner efficacement sur un matériel à ressources limitées.

1.7.6 Sécurité

Plusieurs réseaux de capteurs collectent des informations sensibles. Le fonctionnement à distance non surveillé des nœuds capteurs augmente leurs expositions aux intrusions et aux attaques malveillantes. En outre, les communications sans fil ont rendu la tâche facile aux intrus pour écouter les transmissions des nœuds capteurs. Par exemple, l'un des défis dans la sécurité dans RCSF est l'attaque DoS (Dénie de Service), où son but est de perturber le bon fonctionnement d'un réseau de capteurs. Ceci peut être réalisé en utilisant plusieurs variétés d'attaques, parmi eux on trouve le brouillage radio (jamming attack) qui utilise les signaux à puissance forte pour empêcher les communications des nœuds capteurs. Les conséquences peuvent être sévères et ils dépendent de l'application du réseau de capteurs. Bien que de nombreuses techniques et solutions pour les systèmes distribués qui préviennent les attaques, beaucoup d'entre elles comportent des exigences importantes en matière de calcul, de communication et de stockage, qui ne peuvent souvent pas être satisfaites par des nœuds capteurs à ressources limitées. En conséquence, les réseaux de capteurs nécessitent de nouvelle solution pour l'établissement et la distribution des clés, l'authentification des nœuds et la confidentialité.

1.7.7 Autres défis

A partir des défis cités précédemment, il est claire que plusieurs choix de conception dans un RCSF se différent des autres réseaux et systèmes. D'autres variétés additionnelles de défis affectent la conception des nœuds capteurs et les RCSF. Par exemple, d'autres capteurs peuvent être montés sur des objets mobiles, tel qu'un véhicule ou un robot, qui mènent à un changement continue de la topologie du réseau qui nécessite des adaptations fréquentes au niveau de plusieurs couches du système, qui incluse le routage (e.g., le changement de la liste des voisins), contrôle d'accès au média (e.g., le changement de la densité du signale) et l'agrégation de données (e.g., le changement des régions de détections qui se chevauchent). Un réseau de capteur hétérogène est constitué de dispositifs avec une capacité de matériel varié, par exemple, des nœuds capteurs peuvent avoir plusieurs ressources matérielles si leurs tâches de détection nécessitent plus de traitement et de stockage ou s'ils sont responsables de la collecte et du traitement d'informations pour d'autres capteurs du réseau. Aussi, d'autres applications de capteurs peuvent avoir des performances et des qualités spécifiques, par exemple, faible latence, pour la détection des événements critiques ou un haut débit pour collecter la détection des vidéos. Les deux exigences d'hétérogénéité et de performances affectent la conception des capteurs sans fil et leurs protocoles. Enfin, puisque les réseaux informatiques traditionnels sont basés sur des normes préétablies, de nombreux protocoles et mécanismes dans les réseaux de capteurs sans fil sont des solutions propriétaires, alors que les solutions basées sur des normes s'émergent lentement. Les standards sont importants dans l'interopérabilité et facilite la conception et le déploiement des applications RCSF.

1.8 Conclusion

Ce chapitre a mis l'accent sur les réseaux de capteurs sans fil, leurs constituants, leurs tendances de recherche et leurs défis et contraintes. A cause des ressources limitées et la nature distribuée des RCSF, plusieurs recherches ont été lancées durant ces deux dernières décennies dans le souci d'améliorer leurs performances où la préoccupation principale est de prolonger la durée de vie du réseau le plus longtemps possible. Les recherches dans le domaine ont ciblés tous les composants et les niveaux de développement des RCSF. D'un côté, plusieurs recherches été orientées pour miniaturiser les composants matériels et améliorer l'architecture processeur pour optimiser l'exploitation des ressources énergétiques. D'un autre côté, les recherches n'ont exclus aucune couche de la pile protocolaire ni d'algorithmes de traitements. Ce chapitre a cité également des approches accréditées pour faire face à certains défis (tel que les méthodes d'accès au média, les stratégies de réveille, les mécanismes de fiabilité, . . . , etc.), mais ces approches n'agissent pas parfaitement avec les sources des problèmes. D'où les axes de recherches dans les RCSF ne cessent de croitre.
Le transport fiable dans les RCSF devient plus intéressant ces dernières années, surtout avec l'apparition des détecteurs et les applications multimédia. Malheureusement cette caractéristique rencontre un défi majeur dans les réseaux ad hoc et les RCSF due au taux d'erreurs élevé qui caractérise la transmission sans fil. Ces dernières années plusieurs recherches ont abordé ce problème pour remplir les exigences des applications multimédia dans les RCSF. En conséquence, une panoplie de solutions de transport est apparue permettant de développer des applications multimédia dans les RCSF. Le prochain chapitre met en évidence ce défi pour définir ces caractéristiques, analyser ces lacunes, décrire ces solutions et présenter ces perspectives.

CHAPITRE 2

PROTOCOLES DE TRANSPORTS DANS LES RCSF

2.1 Introduction

Le transport fiable a gagné une importance fondamentale dans les RCSF suite à-sa capacité d'établissement d'une connexion de bout-en-bout, obtenue grâce aux mécanismes de contrôle de flux et de congestion, ainsi à l'allocation équitable de la bande passante. La nécessité d'un transport fiable de données accroit avec le développement des détecteurs et des applications multimédia dans les RCSF[34]. L'utilisation des protocoles de transport fiable classique dans les RCSF est inappropriée due à plusieurs contraintes ayant un impact sur le débit et la capacité des ressources. En fonction des caractéristiques de la transmission sans fil et les contraintes d'énergie, la conception d'un transport fiable dans les RCSF ne comprend pas uniquement des mécanismes de contrôle de flux et de congestion, mais aussi elle prend en considération l'exploitation de la capacité énergétique d'une manière efficace dans le but de prolonger la durée de vie du réseau le maximum possible.
Plusieurs variantes ont été proposées dans la littérature pour améliorer les performances des protocoles de transport dans les RCSF [6, 7, 35, 8]. Chaque variante est caractérisée par le type d'application et les facteurs de performances prisent en considérations. Ce chapitre traite le problème et les défis de transport fiable dans les RCSF et il relate également quelques solutions proposées dans la littérature pour améliorer les performances de transport dans ces réseaux.

2.2 Définition du problème de transport dans les RCSF

Un protocole de transport dans les réseaux de capteurs dépend des exigences et du type d'application utilisée dans le réseau. Certaines applications exigent un transport fiable de données sans se préoccuper de la consommation énergétique et le délai de livraison (tel que les applications orientées requêtes), alors que d'autres applications nécessitent un certain taux de fiabilité en assurant une gestion efficace de l'énergie (tel que les applications de data-streaming). Il existe aussi des applications trop exigeantes en fiabilité, en un délai court et un contrôle de congestion tout en exploitant l'énergie d'une manière efficace (tel que les applications multimédia). Donc, les objectifs principaux d'un protocole de transport pour les RCSF sont [36, 37] :

- Fiabilité de transport : Représente un taux qui estime les données reçues par rapport aux données réellement transmissent par les sources. Pour assurer la fiabilité, un protocole de transport doit inclure un mécanisme de contrôle de flux et de retransmission. Certains applications exigent un taux de fiabilité et peuvent tolérer des pertes au-delà de ce taux. D'autres applications nécessitent une livraison fiable de toutes les données, les commandes, et les requêtes transmissent.

- Contrôle de congestion : Défini la capacité du protocole pour réagir aux situations de congestion au niveau des nœuds sources et intermédiaires. Pour garantir cet objectif, un protocole de transport utilise un mécanisme de détection de congestion en conjonction avec un mécanisme d'ajustement de taux de transmission des données pour éviter les situations de congestion. Cet objectif est généralement exigé par des applications qui génèrent un grand flux de données comme le data streaming.
- Efficacité de l'énergie : C'est la capacité d'un protocole à optimiser l'utilisation des ressources énergétiques. Cet objectif peut être garanti en éliminant toute sorte de gaspillage de l'énergie, i.e., la minimisation de la perte pour éviter la retransmission de données et la réduction des paquets de contrôle pour minimiser le coût de la communication. Les protocoles de transport font recourt aux mécanismes de contrôle de congestion, d'acquittement implicite et de retransmission saut-par-saut pour garantir l'efficacité énergétique.
- Court délai de livraison : Mesure l'intervalle entre le débit de transmission du premier segment au niveau de la source jusqu'à la réception du dernier segment au niveau de la station de base. Certaines applications, i.e., les applications temps réel (applications multimédia), nécessite une livraison de données très courte. Cet objectif dépend du mécanisme de transmission utilisé par le protocole de transport. La transmission continue et la transmission à fenêtre glissante sont les mécanismes les plus souvent utilisés pour offrir un court délai de livraison.

2.2.1 Mécanismes de transport

Les protocoles de transport peuvent utiliser plusieurs mécanismes pour garantir une livraison fiable de données avec une gestion efficace de l'énergie. Chaque mécanisme vise à améliorer certains facteurs de performances (i.e., la fiabilité, le contrôle de congestion, l'économie de l'énergie ou le délai de livraison), d'où, le choix de ces mécanismes est une décision cruciale dans la conception d'un protocole de transport pour les RCSF [10].

Mécanismes de transmission

Durant la transmission d'un flux de données, un protocole de transport repose sur un algorithme de transmission, qui est utilisé pour contrôler la transmission du flux. Un concepteur de protocole peut choisir entre plusieurs mécanismes proposés dans la littérature, mais le choix de ce mécanisme influe directement sur le délai de livraison des données.

Transmission continue Dans ce mécanisme, un protocole de transport transmet tous le flux, segment par segment, sans attendre un acquittement entre chaque transmission. Généralement, les protocoles utilisent une fréquence de transmission pour séparer entre deux transmissions successives. Pour assurer la fiabilité, ce mécanisme est renforcé par un mécanisme d'acquittement négatif pour détecter d'éventuelles lacunes dans le flux. Cependant, le mécanisme d'acquittement négatif est insuffisant en termes de notification, surtout pour aviser l'émetteur à propos des segments correctement reçus. En conséquence, ce mécanisme peut engendrer un gaspillage considérable en espace mémoire.
La transmission continue est simple à implémenter et elle peut atteindre un délai de livraison optimale, mais l'utilisation d'une haute fréquence de transmission augmente l'occurrence des situations de congestion et de collisions due à une surcharge dans le réseau.

Transmission Send-And-Wait Le mécanisme Send-And-Wait (appelé aussi Stop-And-Wait) présenté dans la figure 2.1, est utilisé au niveau de la couche MAC de la plupart des interfaces réseau (i.e., IEEE 802.11,

IEEE 802.15.4). En utilisant ce mécanisme, un protocole transmet un segment et se met en attente d'un acquittement. S'il ne reçoit pas d'acquittement pendant une période de temps bien déterminée, il effectue la retransmission du segment. Donc, la transmission du prochain segment dépend de l'acquittement du segment actuel. D'une manière plus précise, un segment est considéré en cours de transmission jusqu'à la réception d'un acquittement ou l'épuisement du nombre de tentative de retransmission.
Ce mécanisme doit inclure un mécanisme d'acquittement, i.e., les protocoles de la couche MAC préfèrent l'utilisation d'un acquittement explicite positif. Certains protocoles de transport utilisent un mécanisme d'acquittement implicite, où l'émetteur écoute la transmission d'un segment par son voisin, mais la présence des liens asymétrique dans les RCSF peut conduire à des retransmissions non nécessaires.

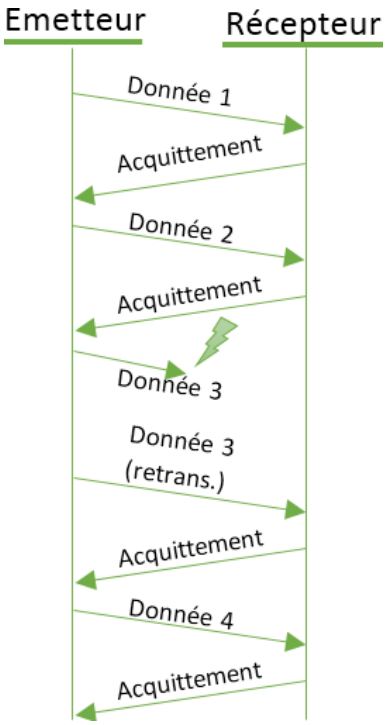

FIGURE 2.1 : Mécanisme stop-and-wait

Le mécanisme Send-And-Wait garanti un transport fiable de données en minimisant la consommation de l'énergie, mais le temps d'attente d'un acquittement ralenti considérablement le délai de livraison, surtout dans un environnement avec un taux de perte élevé.

Transmission à fenêtre glissante Le mécanisme de transmission à fenêtre glissante (présent dans TCP figure 2.2) est utilisé comme solution hybride qui profite des avantages des deux mécanismes précédents. Donc, pour réduire le délai de livraison, ce mécanisme utilise une transmission pseudo continue où un nœud se met en attente d'un acquittement après la transmission d'un certain nombre de segments défini par la taille de la fenêtre glissante. Dans ce type de transmission, les acquittements doivent être positifs, mais ils peuvent être implicites ou explicites.
Les performances de la transmission à fenêtre glissante dépondent de la taille de la fenêtre, tel que l'utilisation d'une fenêtre réduite peut minimiser les frais de retransmission mais augmente considérablement le délai de livraison, alors que l'utilisation d'une fenêtre large réduit le délai de livraison mais provoque des situations de congestion.

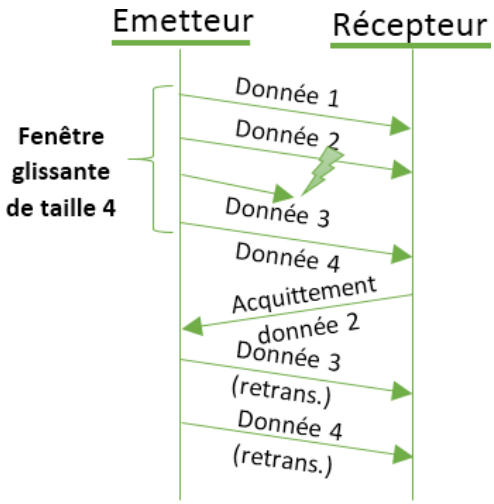

FIGURE 2.2 : Mécanisme de tranmsission à base de fenêtre glissante

Mécanismes d'acquittemen

La perte dans la transmission sans fil est de nature intermittente, tel que la fiabilité doit être renforcé avec un mécanisme de retransmission. Malheureusement, la retransmission dans un réseau qui est soumis à des contraintes de ressources énergétiques peut engendrer un gaspillage d'énergie nécessitant le déclenchement de cette retransmission uniquement aux situations appropriées. Les solutions de transport fiable reposent sur un mécanisme d'acquittement pour réduire les frais de retransmission dans le réseau. Ce mécanisme est une information qui notifie un émetteur à propos de l'échec ou la réussite de transmission.
Dans un mécanisme d'acquittement, on peut choisir entre une notification positive ou négative des segments transmis. Dans un acquittement positif, l'information de notification inclus les numéros de segments reçus par succès. Lors de la réception de cette notification, l'émetteur peut déterminer la suite des segments qui doivent être transmis ou à retransmettre.
Les acquittements négatifs sont utilisés par les algorithmes de retransmission sélective, où l'émetteur est notifié à propos des lacunes détectés dans un flux. En utilisant cette information, l'émetteur retransmet uniquement les segments manquants, ce qui réduit considérablement les retransmissions non nécessaires.
Un protocole peut utiliser les deux techniques de notification suivantes :

- Acquittement explicite : Dans une notification explicite, un mécanisme d'acquittement utilise un message de contrôle qui inclut l'information d'acquittement. Ce message est transmis à partir du récepteur vers l'émetteur pour le notifier à propos des segments manquants ou correctement reçus.
- Acquittement implicite : Pour réduire le coût des communications, d'autres protocoles préfèrent l'utilisation des acquittements implicites. Cette technique ne nécessite pas la transmission d'un message d'acquittement. Donc, après la transmission d'un segment, un émetteur écoute le canal pour d'éventuelles transmissions du segment à partir du nœud récepteur.

Il faut prendre en considération que l'acquittement implicite est utilisé uniquement entre les nœuds intermédiaires, par ce qu'il n y a pas de transmission à partir de la destination finale. A cet effet, la station de base doit utiliser uniquement un acquittement explicite.

2.1.3. Mécanismes de contrôle de congestion

La congestion est un effet d'une saturation au niveau des tampons d'un nœud dans le réseau lors de la réception de nouveaux segments. Les nœuds capteurs agissent avec cet effet en exécutant quelques politiques de

suppression pour libérer l'espace dans les tampons, en permettant ainsi d'inclure des nouveaux segments. Malheureusement, ces politiques de suppression conduisent à une perte de données due à une congestion dans le réseau. Cependant, ce genre de perte est intolérable dans un protocole de transport fiable, parce qu'il provoque des retransmissions inutiles. Ces dernières sont considérées comme une source de gaspillage d'énergie au niveau d'un réseau avec des ressources énergétiques restreintes. Certains protocoles prennent en considération cet effet, en intégrant un contrôle de congestion pour faire face aux pertes de données dues à une congestion dans le réseau.
Le contrôle de congestion repose principalement sur deux mécanismes. Le premier mécanisme, est un mécanisme de détection de congestion utilisé pour détecter d'éventuelles congestions dans le réseau. Le deuxième est un mécanisme d'ajustement du taux de transmission. Ce mécanisme est déclenché lors de la détection de la congestion pour ajuster le taux de transmission au niveau de la source dans le but de réduire les situations de congestion.
Les protocoles de transport peuvent compter sur plusieurs approches pour contrôler la congestion. Dans cette section on cite uniquement les deux approches les plus largement utilisées dans les RCSF.

Les approches distribuées Dans ces approches, le contrôle de congestion est exécuté au niveau de chaque nœud du réseau. Lorsqu'un nœud détecte une congestion, il essaye de déterminer la source ayant un taux de transmission élevé et notifie cette dernière pour réduire son taux de transmission. Les mécanismes de détection de congestion utilisent plusieurs techniques pour déduire une situation de congestion, parmi eux, on trouve :

Taux d'occupation de la file d'attente dans cette technique, un nœud estime le taux d'occupation de la file d'attente dans la prochaine réception. Si ce dernier sature la file d'attente, le nœud suppose que la prochaine réception cause une congestion au niveau du nœud.

Taux de contention et/ou de réception la technique repose sur l'estimation du taux de réception et/ou de contention au niveau d'un nœud. Lorsqu'un nœud détecte que le taux estimé dépasse un certain seuil, il suppose que cette fréquence de réception peut causer une situation de congestion.

Les approches centralisees Ces approches préfèrent effectuer le contrôle de congestion au niveau de la station de base et relèvent cette responsabilité des autres nœuds du réseau. A ce fait, la station de base estime quelques paramètres pour faire le rapport entre une perte due aux erreurs de transmission ou due à une congestion dans le réseau. Lorsque la station de base détecte une situation de congestion, et à base de ces paramètres, elle identifie les sources suspectées pour les notifier à propos d'un ajustement du taux de transmission. Les techniques utilisées pour la détection de congestion sont :

Délai de récupération de l'erreur Cette technique doit être conjointe avec des acquittements de bout-en-bout. Lorsque la station de base détecte des lacunes dans le flux (fragments manquants), transmet un acquittement négative vers la source et estime le délai de récupération du fragment manquant. Dans le cas où le délai dépasse un certain seuil, la station de base considère que le fragment est perdu due à une congestion.

Notification de réaction en utilisant cette technique, la station de base récupère une notification de congestion à partir des segments transmis par les sources (un bit drapeau indiquant la congestion). Donc, c'est une notification transmise par les nœuds ayant détecté la congestion.

2.3 La relation transport/application

Le choix des mécanismes utilisés et les objectifs ciblés dans un protocole de transports nécessite de savoir le type de l'application de surveillance ciblée. Le type de l'application est lié au modèle de communication utilisé par cette application. Donc, suivant le modèle de livraison de données utilisé, le concepteur de protocole de transport décide dans le choix des mécanismes appropriés pour l'application. Dans les RCSF, il existe trois grandes familles d'applications qui se distinguent par leurs modèles de communication, où on va discuter les caractéristiques et les contraintes de chaque modèle dans les sections qui suit [38].

2.3.1 Applications orientées temps

Dans ce type d'application, les nœuds capteurs possèdent un intervalle d'activité et un intervalle d'inactivité défini par l'application. Dans l'intervalle d'activité, les nœuds capteurs collectent les données requises et les transmet vers la station de base, ensuite ils reviennent à l'état inactif jusqu'au prochain intervalle d'activité. La période entre deux intervalles d'activité est appelée « le taux d'échantillonnage de données ». L'utilisation du temps d'inactivité vise à prolonger la durée de vie du réseau, alors on conclut que ces applications se préoccupent de l'économie de l'énergie. En outre, une transmission de données pendant un intervalle d'inactivité ne trouvera pas un nœud à l'écoute, d'où il est nécessaire de transmettre les données dans une période d'activité. Donc, le délai de livraison de données ne doit pas dépasser la durée d'activité des nœuds du réseau. En revanche, les applications orientées temps exigent un court délai de livraison de données en assurant une exploitation efficace des ressources énergétiques. La fiabilité dans la communication orientée temps est parfois non prise en compte pour certaines applications, mais d'autres exigent qu'un certain taux de fiabilité soit assuré.

2.3.2 Applications orientées événements

Les nœuds capteurs dans ces applications, interviennent uniquement lors de la détection d'un événement. Donc, à l'occurrence d'un événement, les nœuds capteurs transmettent cet événement vers la station de base. La surveillance dans ce cas-ci, se passe en temps réel, tel que le délai de notification d'un événement doit respecter l'échéancier précisé par l'application. En conséquence, la détection de l'événement est une tâche importante pour le bon fonctionnement de l'application. Les données qui s'écoulent de ces détecteurs sont fortement liées, tel que les lacunes de données dues aux transmissions erronées sont intolérable. Cependant, les ressources énergétiques incluent dans les détecteurs des applications orientées événements (tel que la surveillance d'un événement dans un établissement) sont souvent non soumises aux contraintes d'énergie autonome. En conclusion, les applications orientées événements nécessite une fiabilité de livraison de données en respectant l'échéancier de livraison, mais sans se préoccuper de la consommation énergétique.

2.3.3 Applications orientées requêtes

Le déclenchement d'une transition dans les applications orientées requêtes commence à partir de la station de base (par fois appelé le lanceur de requête) pour parcourir tout le réseau, et ce pour collecter les événements détectés par l'ensemble des nœuds capteurs. A la fin, les événements collectés doivent être acheminés vers la station de base. Dans ce modèle de communication, l'entité principale est la requête qui inclut non seulement la requête elle-même, mais aussi des données collectées au niveau de chaque nœud qui a reçus cette requête. L'échec de la requête est causé par une transmission erronée, ce qui signifie que les erreurs de transmission ne sont pas souhaitables dans ce type d'applications. D'où, la livraison fiable de données est un facteur pertinent pour accomplir une requête. Le délai de livraison est de moindre importance dans les applications orientée

requête, mais il est souhaitable d'atteindre un court délai de livraison. Par contre, les détecteurs dans ce modèle de communication sont alimentés par des batteries autonomes, ce qui implique, la nécessité de la gestion efficace des ressources énergétiques.
Le tableau 2.1 récapitule les exigences des applications en termes de fiabilité, de délai de livraison et de consommation énergétique, selon le modèle de communication utilisé au niveau application.

	Fiabilité	**Délai de livraison**	**Energie efficace**
Applications orientées temps	Importante	Nécessaire	Nécessaire
Applications orientées événements	Nécessaire	Nécessaire	Importante
Applications orientées requêtes	Nécessaire	Important	Nécessaire

TABLE 2.1 : Facteurs de perfrmances par modèle de communication

2.4 Défis de transport dans les RCSF

Les concepteurs des applications RCSF sont confrontés à de nombreux challenges dus aux ressources limités utilisées par les nœuds capteurs. Ces défis s'étalent même à la conception d'un protocole de transport pour les RCSF. En outre, et suite aux exigences des applications spécifiques, d'autres défis apparaissent et compliquent considérablement la conception du protocole de transport pour les RCSF. A cet égard, les objectifs et les principaux défis de la couche transport vont être discutés dans cette section [4].

2.4.1 Mécanisme d'acquittement

Le protocole TCP le bien connu, utilise un mécanisme de contrôle de flux basé sur la retransmission de bout-en-bout en conjonction avec un mécanisme de contrôle de congestion basé sur une fenêtre glissante variable (Additive Increase/Multiplicative Decrease (AIMD)). Plus spécialement, la réduction de la perte des paquets et la congestion s'effectue en utilisant une communication entre la source et la destination sans aucune intervention des nœuds intermédiaires. Donc, les entités de transport résident uniquement dans la source et la destination. En outre, chaque flux est considéré indépendant pour fournir une communication point-à-point.
Les mécanismes utilisés par les protocoles conçus pour les réseaux filaires mènent souvent à un gaspillage énergétique dans les RCSF, où les informations collectés par le groupe des capteurs sont beaucoup plus important que les informations détecté par un seul nœud capteur. Ceci implique, qu'un protocole de transport à base de communication point-à-point conduit à un gaspillage de ressources au niveau d'un RCSF. Cependant, les mécanismes de contrôle de flux (fiabilité) et de congestion sont utilisés pour améliorer l'efficacité de l'énergie, d'où il est indispensable de les considérer dans un protocole de transport pour les RCSF.

2.4.2 Exigences dépendantes de l'application

Les RCSF sont déployés avec des applications de surveillances spécifiques, tel que chaque application reposent sur un modèle de communication (discuté dans les sections précédentes).En conséquence, une application spécifique possède ces propres exigences et facteurs de performances. Donc, la conception d'un protocole de transport doit être conçue spécifiquement à une telle catégorie d'application. En plus, une application peut être développée pour cibler plusieurs finalités (militaire, environnemental, santé, découverte spatiale et zone de secours de catastrophe). L'importance des facteurs de performances se varie aussi selon le champ d'application.

En conclusion, les objectifs spécifiques des RCSF influent sur les exigences de conception d'un protocole de transport.

2.4.3 Contraintes de l'énergie

Dans les RCSF, La majorité des concepteurs se préoccupent de l'efficacité de l'énergie durant la phase de conception d'une solution, parce que les ressources énergétiques limitées disponibles dans les nœuds capteurs affectent la conception d'une solution. Donc, un protocole de transport doit être conçu de tel sort, qu'il soit économique, i.e., les objectifs de fiabilité et de contrôle de congestion doivent être atteints avec un faible cout énergétique. Par exemple, si le taux de fiabilité au niveau de la station de base dépasse les exigences de l'événement détecté (due à une redondance de données), le nœud source peut conserver l'énergie en réduisant la quantité d'information transmise. De même, les mécanismes de fiabilité de bout-en-bout, qui ont données leurs preuves dans les réseaux conventionnels, engendrent une consommation énergétique considérable dans un routage multi-saut (due à la retransmission). Ce qui signifie que cette solution n'est pas évolutive pour un RCSF. Donc, un protocole de transport doit être conçu de tel sort que le mécanisme de fiabilité réduit la consommation énergétique.

2.4.4 Implémentation biaisée

Un réseau de capteurs est généralement déployé avec un grand nombre de nœuds ayant des ressources limitées, connectés à une station de base sans limitation de ressources. Les limitations de traitement et de capacité mémoire des nœuds capteurs empêchent l'exécution des algorithmes sophistiqués. Donc, les algorithmes de transport doivent être conçus de tel sort que la majorité du traitement soit exécuté au niveau de la station de base pour réduire le traitement au niveau des nœuds capteurs. Ceci optimise l'exploitation des ressources au niveau des nœuds capteurs.

2.4.5 Contraintes de routage

L'adressage des nœuds capteurs dans un RCSF varie selon le protocole de routage utilisé. Dans ces réseaux, une variété de protocoles de routage a été proposée et qui est répertoriée en quatre classes selon leurs topologies (Routage plat, hiérarchique, centré sur les données, et géographique). Chacune de ces classes possède son propre technique d'adressage. Par exemple dans le routage centré sur les données, on peut avoir un adressage à base d'attributs nommés, alors que dans le routage hiérarchique, on trouve un adressage à base de deux adresses physiques (adresse courte des nœuds capteurs et adresse large des coordinateurs).
Dans la littérature, plusieurs protocoles de transport ont été développés pour les RCSF pour faire face à ce défi, mais, leur conception reste toujours orientée vers une telle classe de routage. Les sections qui suivent présentent une description de quelques solutions proposées dans la littérature qui prennent en considération le défi de routage.

2.5 Transport dans les RCSF (solutions proposées)

Dans les RCSF, plusieurs protocoles de la couche transport ont été conçus pour adresser divers questions tel que la fiabilité de transport et le contrôle de la congestion. La liste des protocoles proposés sont mentionnées dans la figure 2.3. Pour mettre en évidence le problème de transport dans les RCSF, on va présenter dans cette section un résumé à propos des protocoles les plus étudiés dans la littérature.

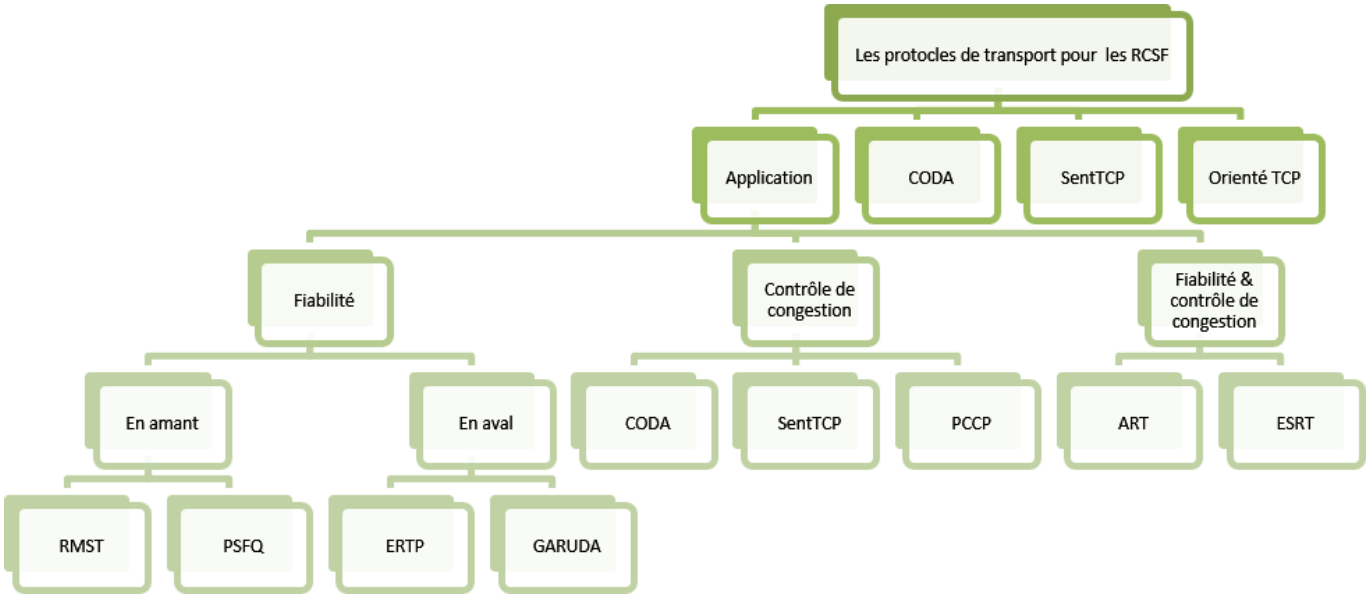

FIGURE 2.3 : Les protocole de transport dans les RCSF

2.5.1 PSFQ (Pump Slowly Fetch Quickly)

PSFQ [6] a était conçu pour une livraison fiable de segments de codes dans les environnements de capteurs sans fil reprogrammables. Dans ces environnements, la station de base transmet des segments de codes vers les nœuds capteurs du réseau afin de reprogrammer leurs activités de détection. Donc, la communication utilisée dans ces environnements est de type point à multipoints, et le protocole doit assurer une fiabilité de livraison tout en minimisant le coût des communications dans le réseau. Ce protocole doit être évolutif dans un RCSF dense.

PSFQ est constituée de trois fonctions : transmission de message (l'opération « pump »), transmission-initiative de récupération d'erreur (l'opération « fetch ») et rapport d'état sélective (l'opération « report »). La source injecte des messages dans le réseau et les nœuds intermédiaires mettent les messages dans leurs tampons, afin de les transmettre avec l'ordonnanceur approprié pour atteindre les limites du délai libre. Le nœud de transmission maintien un cache de données et utilise les informations en cache pour détecter les données perdues, s'il est nécessaire, le nœud de transmission déclenche une récupération de données. Comme la plupart des systèmes d'acquittement négatif, il n'y a aucune façon pour que la source puisse savoir la bonne réception des messages au niveau des récepteurs. Ceci présente plusieurs inconvénients. Premièrement, les segments de données doivent être conservés indéfiniment au niveau de la source pour une possibilité de retransmissions. Ensuite, il est nécessaire pour les sources d'obtenir des statistiques à propos de l'état de dissémination (ex. le pourcentage des nœuds qui ont obtenue l'image d'exécution complète pour une tâche d'application) dans le réseau comme base pour la prise de décisions ultérieures. Donc, il est nécessaire d'incorporer un mécanisme d'évaluation dans PSFQ flexible (ex, adaptatif à l'environnement) et évolutif (ex. minimise la surcharge).

L'opération « pump »

Rappelant que PSFQ n'est pas une solution de routage mais c'est une solution de transport. Dans le cas où un nœud spécifique doit être adressé, PSFQ peut fonctionner au-dessous du système de routage ou de diffusion de données, pour atteindre la fiabilité. Un nœud source utilise les méthodes basées-TTL pour contrôler la portée de son opération de réactivité. PSFQ crée un cache au niveau des nœuds intermédiaires pour activer la récupération locale de perte et de livraison de données en séquence.

PSFQ défini un « message injecté » associé avec l'opération « pump », il possède quatre champs : identificateur de fichier, taille du fichier, numéro de séquence et TTL. La charge utile du message porte le fragment de donnée.
L'opération « pump » est importante pour contrôler quatre facteurs de performance associés aux exigences des applications. Premièrement, un temps de diffusion convenable des segments de code pour tous les nœuds cibles utilisés pour refaire les activités des nœuds capteurs. Deuxièmement, pour fournir un contrôle de flux basique de telle sorte que l'opération de réactivité n'écrase pas les opérations régulières du RCSF. Ensuite, pour les RCSF déployés à une forte densité où les nœuds sont généralement dans la portée de transmission de plusieurs nœuds voisins, elle doit éviter les messages redondants pour économiser l'énergie et minimiser la collision et la contention sur les canaux sans fil. Finalement, pour localiser la perte toute en évitant la propagation des messages de notifications de la perte vers les nœuds en aval. Cette opération nécessite des mécanismes qui assurent un acheminement de données en séquence au niveau des nœuds intermédiaires. Les deux facteurs de performance discutés ci-dessus nécessitent un ordonnanceur propre à l'acheminement de données.

L'opération « fetch »

Vue que la plupart des applications des RCSF génèrent, dans la plupart de temps un trafic concurrent, la perte de paquets se produit généralement à cause des erreurs de transmission due à la mauvaise qualité du lien sans fil et ce n'est pas la cause d'une congestion du trafic. Ceci ne signifie pas que l'occurrence de la congestion n'existe pas dans les RCSF mais la majorité de perte de paquets dans ces réseaux est associée aux erreurs de transmission. Ceci est particulièrement vraie en tenant compte de l'environnement fortement imprévisible dans lequel les réseaux de capteurs fonctionnent, ainsi que la qualité des liens de communication qui peuvent variés considérablement due aux obstacles où les conditions ambiantes hostiles.
Un nœud passe en mode « fetch » une fois qu'un trou de numéro de séquence dans les fragments de fichier et détecté au niveau du nœud récepteur. PSFQ utilise le concept de « l'agrégation de perte » lorsque la perte est détectée, il tente, dans la mesure du possible, de rassembler tous les messages perdus dans une seule opération « fetch »

L'opération « report »

PSFQ prend en charge l'opération « report » qui est conçu spécifiquement pour réagir aux états de livraison des données d'information utilisateurs d'une manière simple et évolutive. Dans les communications sans fil, le coût de la transmission d'un message complet (incluant une large quantité d'informations) est considérablement réduit par rapport à la fragmentation du même message en petits segments [39]. Suite au grand nombre potentiel des nœuds cible dans les RCSF ainsi que les chemins potentiellement long (c-à-dire, les longues chemins multi-saut augmente considérablement le coût de livraison de données) le réseau deviendra saturé, principalement lorsque chaque nœud transmet une réaction sous forme de messages rapport. Donc, il est nécessaire de réduire le nombre de messages utilisés pour des raisons de réaction. Le mécanisme de réaction et le message « report » du PSFQ sont conçus pour adresser ces défis. Le message « report » est conçu pour traverser à partir du plus loin nœud cible en arrière vers l'utilisateur en« saut-par-saut ». Chaque nœud dans le chemin vers l'utilisateur est capable d'acheminer son message « report » de manière regroupée. Lorsque le message se propage vers l'utilisateur qui a demander le rapport, les nœuds intermédiaires peuvent ajouter leurs propres informations de réaction au message « report » original transmit par le nœud cible le plus loin.
En conclusion, PSFQ garantie un meilleur taux de fiabilité de livraison de données, évolutif dans un environnement dense et minimise le coût de communication, mais malheureusement, la fiabilité n'est pas complètement assurée. Il existe des situations où le protocole n'arrive pas à compléter tous les segments du codes (tel que

la perte de tous le message). En outre, ce protocole néglige complètement la congestion et la collision se produisant fréquemment dans réseau dense.

2.5.2 CODA (Congestion Detection and Avoidance)

Congestion Detection and Avoidance (CODA) [7] est l'un des protocoles ayant adressé le problème de congestion dans les RCSF. Les phénomènes de collision et de surcharge sont considérés dans CODA comme une situation de congestion qu'il essaye de les éliminer. Pour faire face à ces situations, CODA ajuste l'intervalle de transmission des segments au niveau de chaque nœud ayant détecté ce genre de situation. Pour atteindre ces finalités, CODA repose principalement sur deux mécanismes qui seront discutés dans les prochaines sections. Les nouds « hotspots » (ou plus clairement les nœuds intermédiaires reliant plusieurs source) peuvent se produire dans différents régions du champ de détection due à différents scénarios de congestion qui peuvent survenir. Ce problème motive la création des mécanismes « open-loop hop-by-hop backpressure » et « closed-loop multi-source regulation » de CODA. Ces deux mécanismes de contrôle sont parfaitement complets entre eux surtout dans le cas où la congestion est persistante. Donc, il est nécessaire d'utiliser différents fonctions de contrôle du taux d'information dans les différents nœuds du RCSF qui dépendent de la nature du nœud (source, puits ou intermédiaire). Donc, le mécanisme de contrôle utilisé dépond de la nature du nœud qui exécute le mécanisme, suite à l'information disponible au niveau de chaque nœud. Par exemple, les nœuds sources possèdent des informations à propos du trafic généré, alors que cette information n'est pas disponible au niveau des nœuds intermédiaires, par contre les « puits » sont parfaitement placés pour comprendre le taux d'exactitude du signal reçu, parce que les « puits » sont des nœuds puissants capables d'effectuer des heuristiques compliqués. L'objectif de la solution CODA est de maintenir un coût faible ou nul des opérations pendant les conditions normales, mais cette solution reste capable de répondre assez-rapidement pour allégir la congestion autours des « hotspots » lorsqu'ils sont détectés. Les sections qui suivent discutent les deux mécanismes principaux de CODA.

Open-loop hop-by-hop backpressure

La contre-pression (« backpressure ») est le mécanisme primaire de contrôle de l'échelle de temps rapide lorsqu'une congestion s'est produite. Lorsqu'une congestion est détectée, le récepteur diffuse un message de refoulement vers ces voisins et en même temps applique des réglages locaux pour prévenir la propagation de la congestion en aval.
Un nœud diffuse un message de contre-pression lorsqu'il détecte une congestion. Les signaux contre-pression se propagent en amont vers la source. Dans le cas des événements d'impulsion de donnée dans les réseaux dense, il est fortement probable que la contre-pression se propage directement vers les sources. Les nœuds recevant les signaux de contre-pression accélèrent leurs taux de transmission ou enlèvent les paquets basés sur quelques politiques de congestion.
Lorsqu'un nœud en amont (vers la destination) reçoit un message de « backpresssure », et en se basant sur ces propres conditions locales relatives au réseau, il peut déterminer s'il doit propager ce message vers l'amont. Par exemple, en fonction de la politique locale de congestion, un nœud peut simplement enlever ces paquets de données entrant lors de la réception d'un message de « contre-pression », ce qui empêche la constitution de sa file d'attente, au lieu de propager en plus le message de « contre-pression » en amont à cause d'une saturation de la file d'attente. Cependant, dans ce cas-ci, le contrôle de congestion « closed-loop » deviendrait nécessaire pour traiter n'importe qu'elle congestion persistante à cause de la politique du nœud qui agisse localement pour détecter la congestion, sans propager un signal « contre-pression ».
Le terme « profondeur de la congestion » signifie dans CODA le nombre de saut qu'un message de « contre-

pression »a parcouru avant qu'un nœud non-congestionné est rencontré. Cette information (« profondeur de la congestion ») peut être utilisée par le protocole de routage et la politique locale de suppression de paquets pour équilibrer la consommation de l'énergie pendant la congestion au travers les différents chemins. Deux solutions peuvent être utilisées :
Envisager la profondeur instantanée de congestion comme un indicateur du protocole de routage pour sélectionner les meilleurs chemins, en réduisant ainsi le trafic sur les chemins qui souffrent d'une profonde congestion.
Alternativement, au lieu de coupler le contrôle de congestion et le routage, les nœuds peuvent silencieusement enlever des messages de signalisation associés avec les protocoles de routage ou de transmission. Ces actions contribueront de pousser les flux d'événements en dehors des régions de congestion et loin des « hotspots » d'une façon plus transparente.

Closed-loop multi-source regulation

Dans les RCSF il est nécessaire d'affirmer un contrôle de congestion dans le cas d'événement d'une congestion persistante à travers les multiples sources vers un« puits » unique, où le « puits » joue un rôle important comme un contrôleur de 1 à n à travers les multiples sources. Le coût du contrôle de flux « closed-loop » est typiquement élevé par rapport au contrôle « open-loop » à cause de la nécessité de signalisation des réactions. CODA propose une approche qui ajuste dynamiquement toutes les sources associées avec un événement de données particulier. Dans un fonctionnement normal, les sources s'ajustent elles-mêmes à un taux prédéfini sans intervention de l'ajustement des« puits closed-loop ».
Lorsque le taux d'événement de la source (τ) est inférieure à une fraction η au débit maximum théorique (S_{max}) du canal, la source ajuste sa fréquence de transmission. Si cette valeur dépasse ($\tau \geq \eta S_{max}$) [40], la source est plus susceptible de contribuer à la congestion et donc le contrôle « closed-loop » est déclenché. Le seuil "η" ici n'est pas identique au seuil utilisé dans la détection locale de la congestion, en fait "η" doit être beaucoup plus petit à cause du résultat suggéré dans [40]. La source rentre dans l'état d'ajustement du « puits » uniquement lorsqu'un dépassement du seuil est détecté. Dans cet état la source nécessite une réaction constante (ex. ACK) à partir du « puits » pour maintenir son taux (τ). La source déclenche un ajustement du « puits » lorsqu'elle détecte ($\tau \geq \eta S_{max}$) en mettant un bit de régularisation dans les paquets d'événement acheminer vers le « puits ». La réception des paquets avec un bit de régularisation ajusté, force le « puits » à envoyer un acquittement (ex. 1 ACK par 100 événements reçus au niveau du « puits ») pour ajuster toutes les sources associées avec un événement de données particulier. Les acquittements peuvent être transmissent d'une manière spécifique à l'application. Par exemple, un « puits » doit transmettre un acquittement uniquement aux chemins qu'il désire renforcer, dans le cas des applications de diffusion direct. La réception d'un acquittement au niveau des sources devra servir autant qu'un mécanisme d'auto-synchronisation permettant aux sources de maintenir le taux actuel des événements (τ).
Lorsqu'un nœud ajuste son bit de régularisation, il prévoit la réception d'un acquittement à partir du « puits » à un taux prédéfini, ou bien, il s'attend à recevoir un certain nombre d'acquittements sur une période prédéfinie qui indique une perte occasionnelle des acquittements due à une congestion transitoire. Si la source reçoit un nombre prescrit d'acquittements pendant cet intervalle, elle maintient son taux (τ). Quand la congestion persiste, les acquittements peuvent être perdus en forçant les sources d'enlever leurs taux d'événements (τ) selon une fonction de décrémentation du taux (ex. décrémentation multiplicatif, ...etc.). Un « puits » peut arrêter la transmission des acquittements à base de son avis sur les conditions du réseau. Le « puits » est capable de mesurer ses propres conditions de charge locale du canal ($rtho$) et si ceci est excessif ($\rho \geq \gamma S_{max}$), il peut arrêter la transmission des acquittements vers les sources.

Tant que le nœud « puits » possède des informations à propos du taux de transmission pour chaque source de données, il peut donc, prendre quelques actions spécifiques à l'application si ce taux est toujours moins que le rapport du taux désirable. Dans ce cas-ci, le « puits » déduit que les paquets ont été perdus à travers le chemin dû à une congestion persistante et arrête la transmission des acquittements vers les sources. Lorsque la congestion est enlevée, le « puits » peut continuer la transmission des acquittements, et comme résultat, le taux d'événements des nœuds sources devrait augmenter selon une fonction d'incrémentation du taux (ex. incrémentation additif).
Puisque dans la majorité des applications le « puits » possède des capacités de stockage et de traitement beaucoup plus puissants par rapport aux nœuds capteurs et les points de collecte d'informations, donc, ce dernier pourrait maintenir des informations d'état associés aux types de données spécifiques. En observant le flux de paquets à partir des sources, si la congestion est prétendue, le « puits » pourrait transmettre des signaux de contrôle explicite vers ses sources pour abaisser leurs valeurs de seuil "η" pour les forcés de déclencher la régulation des événements du « puits » à un taux faible. Ceci fournit un mécanisme de priorité implicite comme partie du contrôle de congestion « closed-loop ».
Lorsque le taux des événements est réajusté au niveau des sources à une valeur (τ) qui est moins qu'un certain"η" du débit maximum théorique (S_{max}) du canal, les sources commencent à s'ajuster elles-mêmes sans nécessiter des acquittements à partir du « puits ».
Comme a été mentionné au début de cette section, CODA assure proportionnellement un transport fiable de données en minimisant les situations de collision et de congestion. Malgré sa réaction efficace avec les situations de congestion, mais la fiabilité n'est complètement assuré. Ainsi, CODA ne gère pas d'une manière efficace les intervalles de transmissions au niveau des nœuds source. En conséquence, le délai de livraison de données est considérablement élevé.

2.5.3 ESRT (Event-to-Sink Reliable Transport)

ESRT vise à assurer une fiabilité désiré par l'application en exploitant d'une manière efficace les ressources énergétique tout en minimisant la congestion. Pour atteindre ces objectifs, et à partir d'une analyse statistique, les concepteurs du protocole supposent que les performances du réseau puissent être divisées en cinq régions (à discuter par la suite). Selon les intervalles de transmission ajustés au niveau de chaque source de données, le réseau peut être situé dans l'une de ces cinq régions pour bien contrôler les performances du réseau.
La première motivation de l'ESRT [8] est pour atteindre et maintenir les opérations réseau dans l'état de la région d'exploitation optimal (Optimal Operating Region (OOR)). A partir d'ici, l'objectif est de configurer le rapport de taux de fréquence "f" pour atteindre la précision de détection des événements désirée avec une consommation énergétique minimale. Pour réaliser ceci, ESRT utilise un mécanisme de contrôle de congestion qui vise à atteindre une fiabilité de détection tout en économisant l'énergie.
D'une manière générale, un réseau peut résider dans l'un des cinq états

$$S_i \in \{(NC, LR), (NC, HR), (C, HR), (C, LR), OOR\}$$

où :

- No Congesion Low Reliability (NC, LR) : C'est un état de non congestion avec faible fiabilité. Dans cet état, il n'y a pas de congestion confirmé et la fiabilité atteinte est inférieure à celle requise.
- No Congesion High Reliability (NC, HR) : C'est un état de non congestion avec forte fiabilité. Dans cet état, la fiabilité atteinte dépasse la fiabilité requise et il n'y a pas de congestion dans le réseau.

- Congesion Low Reliability (NC, LR) : C'est un état de congestion avec faible fiabilité. Dans cet état, la fiabilité atteinte est supérieure à celle requise mais il y'a une congestion confirmée.
- Congesion Low Reliability (NC, HR) : C'est un état de congestion avec forte fiabilité. Dans cet état, la fiabilité observée est inadéquate et il y'a une congestion confirmée.
- Optimal Operating Region (OOR) : C'est un état d'exploitation optimale de la région. Dans cet état, le réseau est exploité dans la tolérance du point optimal, où la fiabilité requise est atteinte avec une consommation énergétique minimale.

Selon l'état S_i, ESRT calcule et met à jour un journal de taux de fréquence f_{i+1}, qui sera diffusé ensuite vers les nœuds sources. Par exemple, si $S_i \in \{(NC, LR), (C, LR)\}$, les niveaux de fiabilité observés sont inadéquats pour détecter les caractéristiques de l'événement désiré. Dans une telle situation, ESRT met à jour agressivement le journal de taux de fréquences pour suivre de manière fiable l'événement.
Ce type d'auto-configuration du protocole ESRT, permet au protocole de s'adapter avec les topologies dynamiques et le déploiement aléatoire. Une autre caractéristique importante d'ESRT est sa capacité de conservation des ressources énergétiques limitées lorsque les niveaux de fiabilités pour détecter un événement dépassent ceux requises, évidement c'est la situation où $S_i \in \{(NC, HR), (C, HR)\}$. La capacité de réduire le rapport taux de fréquences est obtenu grâce à la conservation de l'énergie. Pour la capacité de détecter les événements d'une manière fiable, ESRT adopte une approche prudente pour réduire f d'une manière contrôlée.
Les algorithmes d'ESRT fonctionnent principalement au niveau du nœud « puits », avec des fonctions minimales au niveau des nœuds sources. D'une manière plus précise, les nœuds capteurs nécessitent uniquement les deux fonctionnalités additionnelles suivantes :
Les nœuds capteurs doivent écouter les diffusions du nœud « puits » à la fin de chaque intervalle de décision est mettent à jour leurs rapports taux.
Les nœuds capteurs doivent déployer un mécanisme simple pour maintenir la détection de la congestion.
ESRT utilise la diffusion du nœud « puits » pour communiquer le rapport taux de fréquence mis à jour aux nœuds capteurs afin d'éviter toutes les problèmes de réaction de la latence et également pour économiser les ressources énergétique limitées du capteur. En outre, ESRT fonctionne sous le principe d'identification collectif et ne nécessite pas d'identifiant unique pour la ressource.

Opérations du protocole ESRT

En utilisant des limites de décision bien définie, ESRT identifie l'état courant S_i à partir :

- D'un indicateur de fiabilité η_i calculé par le nœud « puits » pour un intervalle de décision "i".
- D'un mécanisme de détection de congestion.

Selon l'état courant S_i et les valeurs de f_i et η_i, ESRT calcule le rapport taux de fréquences f_{i+1} pour qu'il soit diffusé aux nœuds sources. A la fin du prochain intervalle de décision, le nœud « puits » déduit un nouveau indicateur de fiabilité "η_{i+1}" correspondant au rapport taux de fréquences mit à jour f_{i+1} des nœuds sources. En conjonction avec n'importe quel rapport de congestion, ESRT détermine ensuite le nouvel état du réseau S_{i+1}. Ce processus se répète jusqu'à ce que la région d'exploitation optimale est atteinte (état OOR). La figure 2.4, montre un diagramme d'états/transitions du protocole ESRT.

ESRT assure la fiabilité désiré par l'application et atteint la région optimale d'exploitation du réseau, mais en échangeant des événements de petite taille (constituées d'un seul segment). Lorsqu'on utilise des événements constitués de plusieurs segments, les performances du réseau se dégradent et le protocole ne peut pas atteindre la région d'exploitation optimale du réseau.

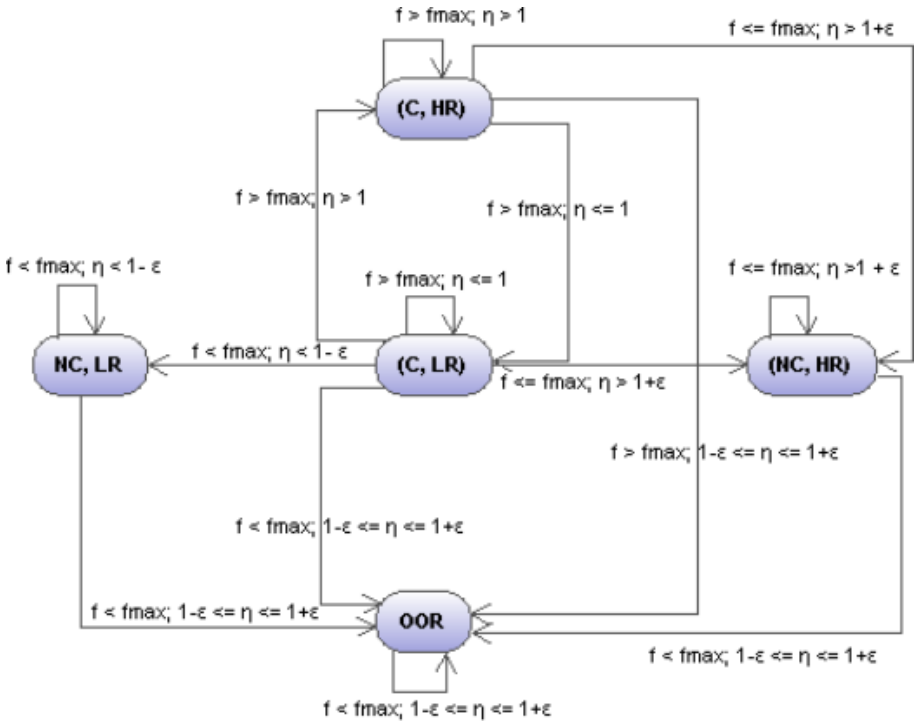

FIGURE 2.4 : Diagramme d'états/transitions du protocole ESRT

2.5.4 RMST (Reliable Multi-Segment Transport)

Le protocole RMST [2] est conçu essentiellement pour s'exécuter en conjonction avec la diffusion dirigée. La diffusion dirigée [23, 41] fournit une communication multipoints à multipoints pour les RCSF un peu semblable au multicast du réseau filaire. Le traitement de la sensibilité à la conservation de l'énergie, le traitement du routage de données central et le traitement des limitations du volume de trafic sont des exemples de motivations qui ont contribué à réaliser la diffusion dirigée.

Le protocole RMST est implémenté au tant que filtre qui pourrait être attaché à un nœud de diffusion selon le besoin de base, sans recompilation du noyau de diffusion ou du filtre de dégradation. Lors de l'exécution, le protocole RMST peut être configuré en mode cache ou en mode sans cache.

La fiabilité dans le protocole RMST se réfère à la livraison éventuelle à tous les « puits » souscrites dans un flux liée à une entité RMST unique. Une entité RMST unique est un ensemble de données constitué d'un ou plusieurs fragments provenant de la même source. L'ordre de livraison, qui n'est pas garanti, est transparent pour le client RMST.

Deux services de transport distinct doivent être ajoutés à la diffusion : la gestion effective de la fragmentation et le réassemblage des unités basées sur la sémantique des applications et la livraison garantie. Bien que ces exigences soient orthogonales, plusieurs applications nécessitent les deux exigences à la fois.

Dans le protocole RMST, c'est le récepteur qui s'occupe de la détection d'un fragment s'il nécessite une retransmission. Mais, l'utilisation du terme « récepteur » ne signifie pas nécessairement le « puits ». Dans le mode « sans mise en cache », uniquement le « puits » qui gère l'intégrité d'une entité RMST en termes de fragments reçus. Dans le mode « mise en cache » un nœud RMST collecte les fragments et il est capable d'initier la récupération des fragments manquants au nœud suivant à travers le chemin vers la source.

Il existe de type de pertes détectées par un « récepteur » : un trou dans la séquence de fragments et une séquence tronqué. Lorsqu'un trou dans la séquence de fragments est détecté, les segments manquant doivent être spécifiquement demandés, ceci est équivalent à un comportement basé sur la demande ARQ sélectif. La troncature d'une séquence est réellement un cas particulier d'une séquence manquante, détecté par le récepteur

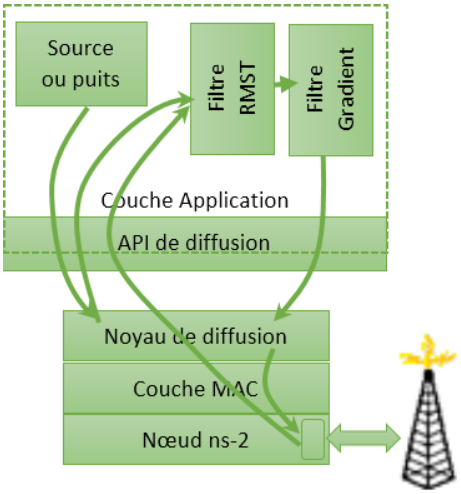

FIGURE 2.5 : Architecture du nœud

via un délai d'attente adapté au temps de réception attendue du prochain fragment.
Lorsqu'un nœud est défaillant, le comportement normal de la diffusion est de rétablir un nouvel ensemble de gradients de données via un intérêt exploratoire. Dans cette mesure, les réseaux de capteurs sont autoréparables. RMST bénéficie du comportement sous-jacent lié aux nœuds défaillants. RMST peut compter sur les mécanismes de diffusion qui garantissent la découverte éventuelle d'un chemin de la source vers le « puits ».
Dans le mode « cache », les fragments en cache à travers les chemins renforcés sont utilisés pour limiter la consommation énergétique due à la retransmission de bout-en-bout. Dans le mode « sans cache », la couche MAC sous-jacente est exploitée pour limiter la surcharge de la couche transport.
RMST assure une fiabilité de livraison de donnée, réduit aussi le coût des communications dans le mode « cache », mais il dépend principalement sur le routage à diffusion dirigé. Ainsi, RMST ignore les situations de congestion.

2.5.5 RCRT (Rate-Controlled Reliable Transport)

Le protocole Rate-Controlled Reliable Transport (RCRT) [9] a été conçu pour assurer une fiabilité de livraison de bout-en-bout tout en évitant les situations de congestions. Ce protocole admet de transmettre une collection de données à partir de plusieurs nœuds capteurs vers la station de base. L'objectif de conception du protocole RCRT est d'assurer un équilibre d'attribution du taux de transmission entre les nœuds sources pour exploiter équitablement la bande passante et vise à combler les insuffisances du protocole CODA. Pour atteindre ces fins, RCRT repose sur quatre composants principaux : fiabilité de bout-en-bout, détection de congestion, adaptation du taux de transmission et allocation du taux de transmission.

Fiabilité de bout-en-bout

La détection et la demande de récupération des segments manquants sont effectuées au niveau puits. Ce nœud crée pour chaque flux de données une entrée qui inclut la séquence des segments reçus dans l'ordre, ainsi que les segments reçus en désordre. A partir de ces informations, le nœud puits peut établir une liste des séquences manquantes pour chaque flux de données. Lors de la présence de séquences insuffisantes, le puits initie un acquittement négatif vers la source correspondante en lui demandant de retransmettre les segments manquants. RCRT utilise l'acquittement négatif pour éviter la surcharge du réseau due aux messages de contrôle. A la réception d'un acquittement négative, le nœud source retransmit les segments insuffisants vers le puits. Donc la récupération des segments manquants est effectuée de bout-en-bout (entre les nœuds source et puits).

Détection de congestion

La détection de la congestion dans le protocole RCRT est basée sur l'estimation du temps de réponse aux acquittements négatif. D'une manière plus précise, lorsque le nœud puits lance une demande de récupération des segments manquants, il estime leurs temps de réception à partir du nœud source correspondant. Si ce temps dépasse un certain seuil, le nœud puits considère que le réseau est congestionné et lance une requête d'adaptation du taux de transmission qui implique la source concerné par les segments manquants. Pour chaque source de données i le nœud puits fixe la valeur du seuil C_i en estimant son temps d'aller-retour correspondant. Cette estimation repose sur la technique Exponential Weighted Moving Average (EWMA)[1] [42].

Adaptation du taux de transmission

Le composant d'adaptation du taux de transmission est déclenché lorsque le nœud puits détecte que le réseau est congestionné en se basant sur l'estimant du temps de récupération des segments manquants. Pour faire face à la congestion, RCRT utilise l'algorithme d'incrémentation additive et de décrémentation multiplicative proposée par les auteurs dans [43]. D'une manière plus précise, RCRT incrémente le taux de transmission en utilisant l'équation suivante :

$$R\left(t+1\right) = R\left(t\right) + A$$

Où A est un constant ajusté à 0.5 paquets/second et $R\left(t\right)$ est le taux de transmission à l'instant t. Lors de la détection d'une congestion, RCRT décrémente multiplicativement le taux de transmission selon l'équation suivante :

$$R\left(t+1\right) = M\left(t\right) R\left(t\right)$$

Où $M\left(t\right)$ est un facteur de décrémentation multiplicatif dépendant du temps, tel que $M\left(t\right)$ est obtenu comme suite :

$$M\left(t\right) = \frac{p_i\left(t\right)}{2 - p_i\left(t\right)}$$

Sachant que p_i est le taux de perte des segments estimé en utilisant la technique EWMA avec

$$w = [1, 1, 1, 1, 0.8, 0.6, 0.4, 0.2]$$

Allocation du taux de transmission

Une fois le taux de transmission $R\left(t\right)$ est estimé, le rôle du composant d'allocation du taux de transmission est d'attribuer ce taux aux nœuds sources appropriées en utilisant une politique d'allocation. RCRT offre trois politiques différentes d'allocation du taux de transmission, à savoir :

- Demande proportionnelle : dans cette politique, chaque flux spécifie un taux désiré notée d_i, tel que le taux r_i attribué au nœud i est proportionnel à la demande d_i de tel sorte que $P\left(t\right) = \frac{r_i(t)}{d_i}, \forall i$.
- Demande limitée : Dans cette politique le taux $R\left(t\right)$ est divisé sur tous les flux de données en s'assurant qu'aucun flux ne dépasse d_i.
- Equitable : Cette politique assigne un taux de transmission $R\left(t\right)$ équitable à tous les flux de données.

[1] Exponential Weighted Moving Average est une technique utilisé pour estimer la pondération exponentielle d'une variable à partir de ses données historiques

2.5.6 ERTP (Energy-effecient and Reliable Transport Protocol)

Energy-effecient and Reliable Transport Protocol (ERTP) [1] est un protocole de transport destiné pour les applications « data-streaming » dans les RCSF, où les lectures détectées sont transmis à partir de plusieurs capteurs (sources) vers une station de base (ou « puits »). ERTP cible deux objectifs :

- Une fiabilité de bout-en-bout : le premier but d'ERTP est d'atteindre une fiabilité de bout-en-bout de toutes les données transmises par chaque nœud capteur.
- Une efficacité énergétique : Puisque la latence de bout-en-bout n'est pas une préoccupation urgente dans plusieurs applications data streaming des RCSF, l'efficacité énergétique est souvent importante dans ces applications. Pour un fonctionnement à long terme du réseau, le protocole de transport devrait minimiser la consommation énergétique de la détection.

ERTP fait trois hypothèses à propos de la couche liaison au-dessous et la couche application au-dessus :

- Faible taux de données : ERTP suppose que si le taux de transmission de données soit faible, alors la congestion dans le réseau est négligeable. C'est une hypothèse raisonnable dans la plupart des applications data streaming déployées dans la pratique.
- Faible taux d'intrusions : un nœud est capable de suspendre la transmission des paquets des voisins à un saut. L'estimation des instruisons deviennent coûteuses, car le un nœud doit écouter les paquets qui ne sont pas destiné pour lui (écoute en repos).
- Faible transmission concurrente : les collisions de transmission se produisent sauf si au moins deux nœuds voisins qui se trouvent dans la même portée viennent de transmettre des données en même temps. Cependant, pour des applications à faible taux de données, la collision est négligeable parce que la probabilité que deux nœuds voisins transmettent dans le même moment est petite. Par exemple, s'il existe N nœuds voisins en communication et M paquets qui peuvent être transmit dans une période, la probabilité que deux nœuds ou plus transmettent un paquet simultanément est de $1 - \prod_{k=1}^{N-1} \frac{M-k}{M}$ [1].
- L'ERTP est constitué de deux composants : fiabilité de bout-en-bout et l'expiration du délai de retransmission de bout-en-bout (Retransmission Timeout (RTO) saut-par-saut).

Fiabilité de bout-en-bout

Assure la fiabilité de bout-en-bout nécessaire pour l'application en contrôlant dynamiquement le nombre de transmissions d'un paquet au niveau de chaque nœud intermédiaire. De toute évidence, un nœud capteur ne peut pas permettre un très grand nombre de retransmissions à cause de la préoccupation de la fraîcheur et l'équité des paquets. Dans la plupart des protocoles de transport, un nombre de transmission prédéfini est utilisé. Pour atteindre une fiabilité de bout-en-bout et une économie énergétique, ERTP détermine dynamiquement le nombre maximum de retransmissions au niveau de chaque nœud. Un nombre maximum de retransmissions insuffisant peut causer une perte de paquets pour les acheminer vers le nœud « puits », un gaspillage en termes d'énergie et de ressources réseau ainsi qu'une dégradation de la fiabilité de bout-en-bout. En revanche, l'énergie devient inefficace lorsque le nombre maximal de retransmission est trop élevé. Pour équilibrer la consommation énergétique et la fiabilité de bout-en-bout, le composant de « *fiabilité de bout-en-bout* » détermine et actualise une valeur proche à l'optimale du nombre maximal de retransmissions pour les paquets de données au niveau de chaque nœud.

RTO saut-par-saut

Assure la fiabilité de bout-en-bout nécessaire pour l'application en ajustant dynamiquement le délai d'expiration de la retransmission (RTO) au niveau de chaque nœud. Le mécanisme saut-par-saut "iACK" est exécuté au niveau de l'émetteur en écoutant les paquets transmis par le récepteur vers le prochain saut est considéré comme un "iACK". L'émetteur doit retransmettre le paquet s'il ne reçoit pas un iACK après expiration d'un intervalle de temps. Il n'est pas évident de déterminer combien de temps un nœud doit attendre un "iACK", cette attente dépend du temps qu'un paquet peut occuper pour être acheminer par les nœuds en aval. La figure 2.6 présente les opérations normales du protocole HBH iACK.

Lorsque le nœud "i" achemine un paquet du nœud $i-1$ au nœud $i+1$, le nœud $i-1$ intercepte cet ache-

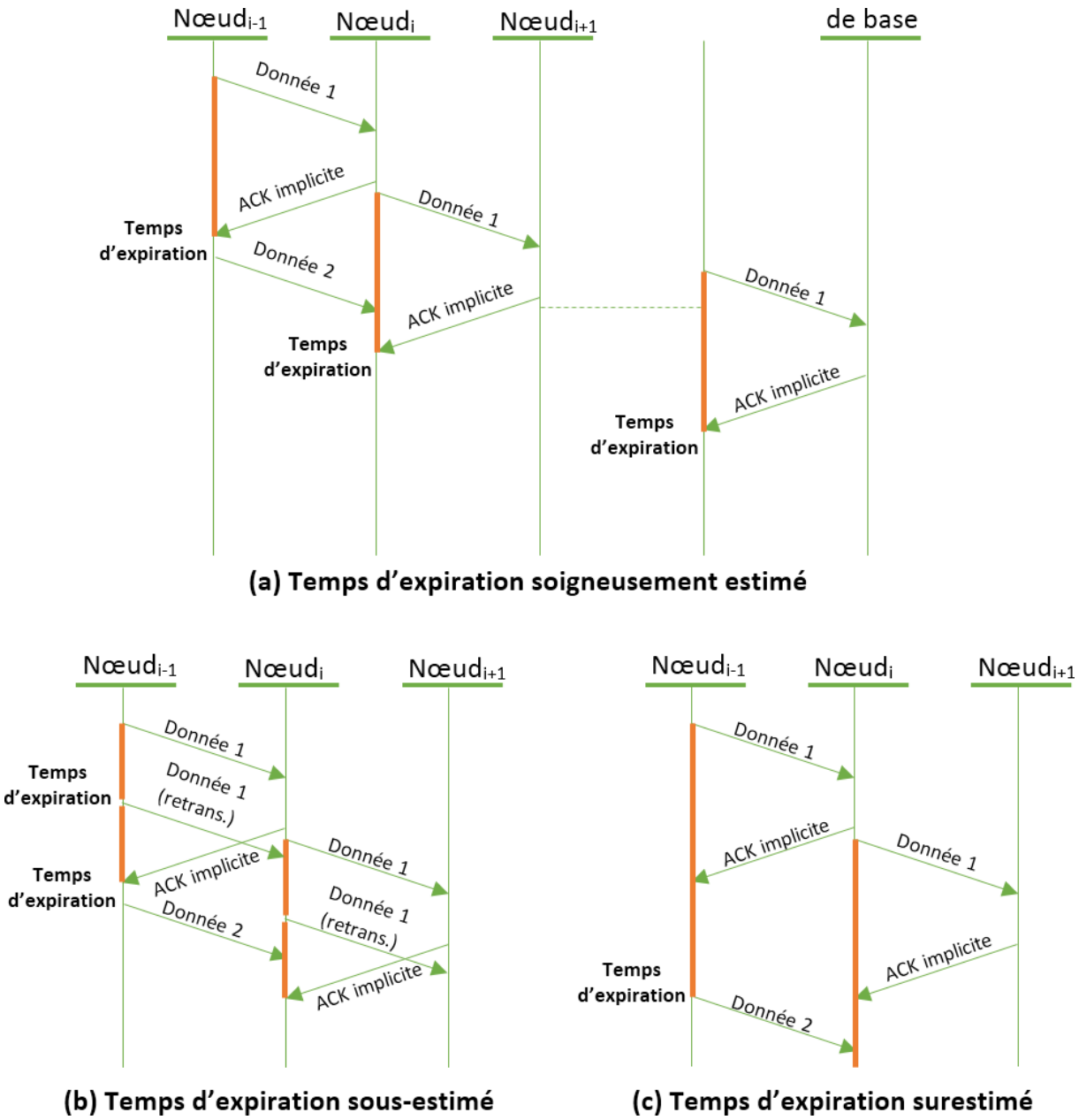

FIGURE 2.6 : Les opérations HBH iACK

minement et le considère comme un "iACK". Une valeur RTO « prématuré » pour HBH iACKaugmente la consommation énergétique parce que les émetteurs vont envoyer des paquets dupliqués. En conséquence, une exploitation énergétique inefficace par ce que le paquet a déjà été envoyé (figure 2.6 (b)). D'autre part, une

grande valeur RTO tend vers l'augmentation de la latence de transmission ainsi qu'une réduction au débit du réseau. Donc, pour atteindre une exploitation énergétique efficace, le composant RTO saut-par-saut de l'ERTP est le responsable de l'ajustement dynamique du temps d'expiration de la retransmission. Évidemment, lorsqu'un paquet atteint le« puits », il n'y aura pas d'autres acheminements. Donc, le nœud « puits » doit envoyer un "eACK". Pendant la transmission immédiate du "eACK" par récepteur, le temps d'expiration du "eACK" est principalement basé sur le temps d'aller-retour saut-par-saut. Chaque nœud maintien une liste de détection de paquet dupliqué pour empêcher la propagation des paquets dupliqués dans le réseau.

2.5.7 RAIT (Reliable Asynchronous Image Transfert Protocol in WSNs)

Reliable Asynchronous Image Transfert (RAIT) est un protocole de transport fiable destiné pour transférer des images d'une manière asynchrone [44]. Ce protocole introduit la transmission de segments à fenêtre glissante en utilisant une méthode à double fenêtrage. La première est utilisée pour la transmission quant à la deuxième pour la réception des segments. En utilisant cette méthode de transmission, RAIT peut contrôler à la fois, les paquets perdus due aux transmissions erronées (détectées par la fenêtre de réception), et les paquets perdus due à des situations de congestion (détectées par la fenêtre de transmission).
Pour présenter le comportement de ce protocole, on considère un réseau de capteurs sans fil dense où les nœuds de détection sont équipés d'une caméra. Pour construire des chemins à partir des nœuds de détection vers la station de base, le routage hiérarchique est utilisé. Dans ce type de routage, les nœuds inférieurs hiérarchiques participent avec les nœuds supérieurs hiérarchiques pour communiquer leurs informations de détection. La perte de paquets due aux transmissions erronées est récupérée en utilisant le mécanisme de retransmission. Dans cette situation, chaque nœud capteur gère deux files d'attentes : la première pour la transmission et la deuxième pour la réception de données [45, 46] (voir figure 2.7).

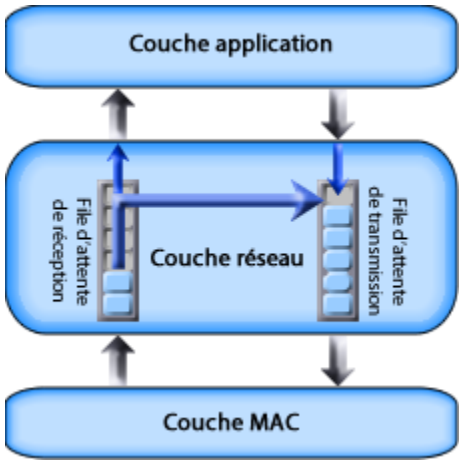

FIGURE 2.7 : Flux de paquets dans un nœud capteur

Dans un RCSF, un nœud capteur joue le rôle d'un générateur de données et en même temps un routeur, parce que le nœud doit acheminer les données détectées vers la station de base. Donc, la file d'attente de transmission d'un nœud capteur est partagée entre le nœud lui-même et les nœuds inférieurs hiérarchiques. En outre, la couche réseau dans un nœud capteur de l'état des files d'attentes de transmission et de réception. A partir de ces observations, RAIT suppose qu'il n'est pas suffisant d'utiliser un seul mécanisme de fenêtre glissante pour transmettre les segments, mais aussi d'ajouter une autre fenêtre pour la réception.

Transmission à fenêtre glissante

Comme il a été mentionné dans les sections précédentes, la transmission à base de fenêtre glissante est utilisée pour réduire le délai de livraison de données. RAIT exploite ce type de transmission (utilisé par le TCP classique) en lui rajoutant une deuxième fenêtre pour gérer la réception de données. Cependant, le mécanisme de transmission à fenêtre glissante souffre du problème de la saturation des files d'attentes, surtout dans un environnement à contraintes de ressources (tel que RCSF). En fait, cet évènement engendre une élimination des paquets au niveau des files d'attentes, ce qui se traduit par une perte de donnée due à une situation de congestion. En conséquence, ce phénomène augmente le coût de livraison de données en termes de consommation énergétique.
Pour prévenir à ce genre de situation, RAIT utilise la deuxième fenêtre pour assurer la disponibilité au niveau de la file d'attente de transmission. En se basant sur la file d'attente de transmission, un nœud émetteur ne transmet pas de segments, que si la file d'attente peut inclure les segments à transmettre. En cas d'espace suffisant dans la file d'attente, le récepteur envoie un acquittement à l'émetteur pour lui notifier la disponibilité de l'espace libre dans la file d'attente. Donc, le nœud émetteur peut reprendre sa transmission en se basant sur l'espace disponible spécifié dans l'acquittement.

Communication inter-couche

Le protocole RAIT utilise un mécanisme à base de jeton au niveau de la fenêtre glissante de transmission. Un nœud émetteur demande un jeton du récepteur avant de commencer sa transmission. Si ce dernier reçoit le jeton, il peut transmettre ses segments vers le nœud récepteur. Lors de la délivrance d'un jeton, un nœud récepteur vérifie l'espace disponible au niveau de sa file d'attente pour délivrer un jeton qui inclut la capacité de la file d'attente. Cependant, dans les RCSF généralement chaque couche dans la pile protocolaire fonctionne d'une manière indépendante sans aucunes connaissances de l'état des autres couches. Donc, une couche ne peut pas contrôler le fonctionnement d'une autre couche. En conséquence, les protocoles de transport existants ne peuvent pas contrôler l'état des files d'attentes. RAIT adresse ce problème en implémentant une solution à base de communication inter-couche, tel que la couche transport peut communiquer directement avec la couche MAC pour vérifier l'état des files d'attentes.

2.5.8 HDRTP (a hybrid and dynamic transport protocol for WSN)

Hybrid and Dynamic Reliable Transport Protocol (HDRTP) [35] a été conçu pour compléter les insuffisances du protocole PSFQ [6] (présenté dans les sections précédentes). Donc, les auteurs de cette solution ont réalisé une analyse approfondie du protocole PSFQ pour améliorer ses performances. L'objectif principal de cette amélioration est pour garantir une fiabilité complète de livraison de données, qui n'a pas été atteinte par le protocole PSFQ.
On rappelle que le protocole PSFQ utilise une communication multipoint à point, où le mécanisme d'acquittement repose sur des acquittements négatifs saut-par-saut. Ces avantages principaux sont : la fiabilité, l'évolutivité et la robustesse. Ce protocole repose sur trois mécanismes de base : pump (envoie), fetch (récupération), report (statistiques). Ces fonctions ont été discutées en détail dans la section 2.5.1 La transmission des segments de code dans PSFQ utilise une horloge aléatoirement choisi entre T_{min} et T_{max}, où un nœud transmet un segment dans un intervalle aléatoire entre 0 et T_{max} ($T_{tr} \in [T_{min}, T_{max}]$, et T_{tr} est le temps de transmission). A partir de cette horloge, on peut estimer la durée maximale de livraison d'une donnée en utilisant l'équation suivante :

$$D(n) = T_{max} * n * h$$

Où n est le nombre de fragments constituant la données et h est le nombre de saut pour atteindre la station de base.
Dans l'opération de récupération réactive, PSFQ utilise aussi une horloge $T_r < T_{min}$ pour récupérer rapidement un segment notifié perdu lors de la réception d'un acquittement négative. La récupération réactive n'est pas suffisante pour agir avec tous les cas de pertes de segment. Parce que, si les derniers segments sont perdus, la récupération réactive ne détecte pas cette perte. Afin d'éviter la perte du dernier segment, PSFQ rajout un mécanisme de récupération proactif. Alors, si un nœud ne reçoit pas la totalité des segments dans une période T_{pro}, le nœud diffuse un acquittement négatif vers ces voisins, en notifiant les segments insuffisants. En conséquence, la conjonction des récupérations réactive et proactive permettent de répondre parfaitement aux pertes de segments. Maintenant, si on pose la question « que ce passe-t-il, si un nœud échoue pour transmettre tous les segments ? ». En fait, cette question est considérée par le protocole HDRTP (discuté dans la section suivante).

Perte de tous les segments

PSFQ ne répond pas à la situation où tous les segments de données sont perdus, due aux insuffisances de l'opération de récupération (réactive et proactive). Dans ce cas-ci, un nœud récepteur qui n'a reçu aucun segment de données, se mit en attente d'un nouveau flux de donnée, alors que le flux est totalement perdu lors de sa transmission.
Pour remédier à ceci, HDRTP ajoute deux messages de contrôles, le premier message, qui est `INFMSG`, pour notifier le nœud récepteur à propos de la transmission d'un nouveau flux. Le deuxième message (`ACKMSG`) est un acquittement pour confirmer à l'émetteur la bonne réception du message `INFMSG`. En mentionnant, que l'acquittement de chaque message permet de détecter la perte d'un message, HDRTP utilise ces deux messages pour confirmer que le nœud récepteur est informé d'une nouvelle transmission de flux, et s'est mis en attente de ce flux. Donc, s'il n'a reçu aucun segment, la récupération proactive sera lancée pour demander les segments insuffisants, et évidemment, la perte de tous les segments est évitée.

Transmission lente

L'opération de transmission dans PSFQ ralenti la livraison du flux de données, due à l'attente aléatoire entre T_{max} et T_{min}. Tant que ces deux paramètres sont constants, le délai de livraison peut être assez grand même dans un réseau qui n'est pas dense. Puisque le protocole PSFQ n'ajuste pas ces deux paramètres d'une manière dynamique, cette configuration se traduit à des insuffisances dans le protocole PSFQ. Donc, HDRTP a considéré cet inconvénient en ajustant dynamiquement T_{max} et T_{min} selon la qualité du lien de chaque nœud. A l'envoie d'un message `INFMSG`, le nœud inclus l'estimation des deux paramètres dans le message `INFMSG`.

Echec du protocole

Dans PSFQ, si la station de base envoie des segments de codes vers les nœuds capteur au moment où quelques nœuds sont dans un état de redémarrage ou réinitialisation, ces dernières ne reçoivent aucun segment de codes pour changer leurs fonctionnements. Donc c'est un échec du protocole, parce que PSFQ soufre de la perte de la totalité du message. Cependant, HDRTP répond parfaitement à ce genre de question en réagissant avec cette perte. En conséquence, dans HDRTP, et en utilisant les notifications `INFMSG` et `ACKMSG`, la station de base peut énumérer les nœuds qui n'ont pas reçu les segments de code, due à leurs redémarrage.
Donc, on peut conclure que le protocole HDRTP a répondu parfaitement aux insuffisances du protocole PSFQ, surtout en termes de délai de livraison. Mais malheureusement, les solutions proposées ignorent complètement

les situations de congestion qui influent sur la perte et la consommation énergétique.

2.5.9 Autre solutions

Les auteurs Long Jun et al [47] ont proposé une architecture pour traiter et envoyer des images à travers un RCSF d'une manière fiable. L'image dans cette architecture doit être traité et compressé afin de détecter les mouvements. La communication de l'image est réalisée en se basant sur la transmission Send-And-Wait. Pour garantir la fiabilité de livraison, cette architecture ajoute un mécanisme d'acquittement négatif de bout-en-bout. Puisque le mécanisme de traitement des images n'a pas été considéré dans cette thèse, cette solution et loin d'être étudié dans les travaux d'analyse et de comparaison.
Les auteurs [48] ont transformé le problème de transport en un problème d'optimisation à trois paramètres : nombre de nœuds déployés, position des nœuds, et structure de transmission. Sur la base de cette optimisation, les autres ont prouvé que l'optimisation des performances du réseau (fiabilité, consommation énergétique, unité de transmission de donnée, durée de vie du réseau) peut être résolue mathématiquement. A partir d'une étude analytique, et après l'optimisation du problème, la solution obtenue est une configuration optimale permettant de garantir une fiabilité en exploitant l'énergie d'une manière efficace

2.6 Comparaison des protocoles

Cette section présente une partie de la comparaison des protocoles de transports étudiés. Le tableau 2.2 et le tableau 2.3 présentent une comparaison basée sur des critères mentionnés au-dessous [49].

Protocole	Fiabilité								Efficacités de l'énergie
	Catégorie	Direction	Type	Détection et notification de la perte				Récupération de la perte	
				NACK	IACK	N° de séquence	Temps d'expiration	Augmentation du taux de détection	Retransmission du paquet	
PSFQ	**Paquet**	**En aval**	**Saut-par-saut**	✓		✓	✓		✓	**Non**
CODA										**Bon**
ESRT	**Evénement**	**En amont**	**Evénement-au-puits**				✓	✓		**Equitable**
RMST	**Paquet**	**En amont**	**Saut-par-saut**	✓			✓		✓	**Bon**
ERTP	**Paquet**	**En amont**	**Bout-en-bout**		✓		✓		✓	**Bon**

TABLE 2.2 : Comparaison de la fiabilité et l'efficacité de l'énergie

Le Tableau 2.2 présente des résumés sur les protocoles de transport fournissant un mécanisme de fiabilité. La fiabilité est la fonction principale de la couche transport qui assure une livraison appropriée d'informations à partir de la source vers la destination ou le « puits ». Il existe différents mécanismes de fiabilité pour différents protocoles parce que la majorité des protocoles sont conçu pour résoudre des problèmes basés sur l'application. Par rapport à ceci, la majorité des protocoles de transport utilisent des acquittements négatifs (Negative Acknowledgment (NACK)) et un temps d'expiration dans l'étape de détection et de notification de

la perte et ils utilisent une retransmission de paquets pour l'étape de récupération de la perte. Chaque méthode proposée possède ses avantages et ses inconvénients qui dépendent de l'application elle-même.

Protocole	Contrôle de congestion									
	Détection de la congestion					Notification		Ajustement du taux de trans.		
	Paquets envoyés par succès	Taille de la file d'attente	Temps de service (TS)	TS/TA	Charge du canal	Explicite	Implicite	Démarrer et arrêter	AIMD	Ajustement exacte du taux
PSFQ										
CODA		✓			✓	✓			✓	
ESRT	Evén.-au-puits	✓					✓			✓
RMST										
ERTP										

TABLE 2.3 : Comparaison du contrôle de congestion

D'autres protocoles n'offrent aucun mécanisme de fiabilité et ils possèdent uniquement un mécanisme de contrôle de congestion comme il est présenté dans le tableau 2.1. La majorité de ces protocoles utilisent une méthode basée sur la taille de file d'attente pour détecter la congestion, une méthode implicite pour l'étape de notification de la congestion et ils ajustent leurs taux de transmission pour éviter la congestion.

2.7 Conclusion

Dans ce chapitre, un état de l'art sur les protocoles de transport RCSF les plus étudiés dans la littérature a été réalisé. Un protocole de transport doit fournir une fiabilité de transport de données dans n'importe quel réseau. Dans les RCSF, un protocole de transport ne doit pas fournir uniquement une fiabilité de transport de données, mais il doit utiliser également les ressources énergétiques d'une manière efficace. Pour remédier à ceci et améliorer les performances du réseau, les solutions étudiées dans ce rapport ne ciblent pas uniquement la récupération des paquets perdus, mais aussi l'optimisation du mécanisme du contrôle de congestion pour réduire la retransmission des paquets perdus, ce qui mène vers une consommation énergétique efficace. Pour améliorer les performances du réseau, chaque protocole de transport dans les RCSF repose sur l'un des mécanismes de la retransmission saut-par-saut ou la retransmission de bout-en-bout. Les performances du premier mécanisme montrent qu'ils sont mieux que le second mécanisme, suite à la réduction du nombre de retransmission des paquets perdus, ce qui rapporte un gain considérable dans l'exploitation des ressources énergétiques. Les optimisations qui utilisent les interactions entre-couches sont aussi nécessaires pour augmenter la fiabilité et améliorer la qualité du réseau. On utilise dans ces optimisations les coopérations entre les différentes couches de la pile protocolaire pour qu'elles partagent des informations. La majorité des protocoles de transport construisent des statistiques à propos des canaux pour ajuster dynamiquement le taux de transmission lorsqu'une congestion est détectée.

Quoi qu'on propose de solutions pour assurer un transport fiable de données dans les RCSF, ces solutions ne s'adaptent qu'à une certaines variantes d'applications pour les RCSF. La conception d'un tel protocole se focalise toujours sur un ensemble de paramètres qui dépendent principalement sur les exigences de l'application et les contraintes du réseau.

CHAPITRE 3

LE PROTOCOLE DE TRANSPORT CARTEE

3.1 Introduction

Avec l'apparition des applications multimédia dans les RCSF, le besoin en solutions de transport croit de jour en jour, surtout avec le développement des détecteurs acoustique et de télésurveillance. Les exigences de ces applications varient selon la nature des évènements à détecter ou une catégorie exige une fiabilité de livraison de données et ne s'intéresse pas à la consommation énergétique et au délai de livraison, alors que d'autres sollicitent un court délai de livraison et une exploitation efficace des ressources énergétiques. Dans le chapitre précédent nous avons cité une variété de solutions de transport destinée pour les applications multimédia, où chaque solution vise à répondre aux besoins d'une application spécifique dans les RCSF. Cependant, les solutions de transport proposées dans la littérature ne répondent pas à la totalité de ces exigences à cause des défis rencontrés dans la conception. En conséquence, les travaux de recherche ouvrent de nouvelles perspectives pour répondre aux besoins des applications multimédia pour les RCSF.
Ce chapitre présente un nouveau protocole de transport baptisé CARTEE [10] (pour Congestion Avoidance and Reliable Transport and Energy Efficiency) assurant une fiabilité de livraison de données dans les RCSF. Ce protocole est conçu pour les applications multimédia où il vise à garantir une fiabilité et un court délai de livraison de données avec l'exploitation efficace des ressources énergétiques. Pour atteindre ces fins, CARTEE prend en considération plusieurs sources de gaspillage de l'énergie (i.e. la congestion, les retransmissions non nécessaires, ...etc.). Il repose sur quatre principaux mécanismes utilisés pour optimiser un tel ou tel facteur de performance. Le premier mécanisme est la transmission à fenêtre glissante fixe utilisé pour minimiser le délai de livraison. Le deuxième est le mécanisme d'acquittement qui fonctionne en conjonction avec la fenêtre glissante est utilisé pour assurer la fiabilité et réduire les messages de contrôle. Le troisième et le quatrième est un couple qui permet d'éviter la congestion, évidement, un mécanisme de détection de congestion avec un mécanisme d'ajustement de la fréquence de transmission. Ces mécanismes vont être discutés en détail dans les prochaines sections.

3.2 Motivation

Le développement progressif des applications destinées pour les RCSF a motivé les chercheurs à réfléchir sur des solutions de transport spécifiques à ces réseaux. Un protocole de transport est devenu indispensable surtout lors de l'apparition des applications multimédia. Cependant, les défis de transport de données dans les RCSF ont orientés la conception des solutions vers la concentration sur l'amélioration des facteurs de performance

exigées par certaines applications spécifiques. Les solutions proposées dans la littérature sont dépendantes soit de l'application ou du protocole de routage sous-jacent ce qui a motivée la conception du protocole CARTEE. D'une manière générale, CARTEE a été conçu pour répondre aux exigences des applications multimédia dans les RCSF leur permettant de transférer un flux de données toute en assurant sa fiabilité de livraison. Donc la préoccupation principale de cette solution est d'offrir une certaine autonomie entre les couches de la pile protocolaire du réseau. Cette solution ne s'intéresse pas uniquement au facteur de dépendance, mais encore elle se focalise sur l'optimisation de la majorité des facteurs de performances exigés par les applications multimédia. Donc, CARTEE permet d'alléger la couche transport en réduisant ses entités protocolaires grâce à son champ d'application élargie par rapport aux solutions existantes.
Pour bien éclairer les motivations derrière la conception du protocole CARTEE, on va présenter dans la section suivante une étude de performances des solutions existantes et une critique des insuffisances pour chaque solution.

3.2.1 Etude des solutions existantes

Plusieurs solutions de transport ont été proposées dans la littérature visant à améliorer les performances du transport dans les RCSF. Chaque solution a été conçue pour répondre aux besoins d'une application spécifique dans ces réseaux. Le tableau 3.1 présente les performances et les applications ciblées par les propositions de protocole de transport dans les RCSF.

Solution	**Tâche**	**Performances**
PSFQ [6] (Pump Slowly Fetch Quickly)	– Assure une fiabilité saut-par-saut – Evolutivité	Destiné pour les détecteurs reprogrammables. Il permet de transférer des segments de codes à partir du nœud « puits » avec une communication de type point à multi point.
ESRT [8] (Event to Sink Reliable Transport)	– Assure un certain taux de fiabilité (désiré par l'application) – Gestion efficace de l'énergie – Minimise la congestion – Minimise les caches de données	Destiné pour les applications nécessitant un certain niveau de fiabilité, tel que : – Surveillance des zones volcaniques – Surveillance du niveau d'eau dans un barrage
RMST [2] (Reliable Multi-Segment Transport)	– Assure une fiabilité saut-par-saut – Fournit deux mode de fonctionnement (cache ou sans cache) pour minimiser les caches de données	Utilisé par les applications qui repose sur l'adressage à diffusion dirigée [23] (intérêt-ingrédients).
CODA [7] (Congestion Detection and Avoidance)	– Gestion efficace de l'énergie – Elimination de la congestion	Utilisé par les applications qui repose sur l'adressage à diffusion dirigée

Solution	Tâche	Performances
RCRT [9] (Rate Controlled Reliable Transport)	– Assure une fiabilité de bout-en-bout – Elimine la congestion – Elimine les caches de données	Conçu pour les applications multimédia (détecteurs acoustiques) qui ne tolèrent pas la perte de données.
ERTP [1] (Energy Efficiency Reliable Transport Protocol)	– Assure Un certain taux de fiabilité (spécifié par l'application) – Gestion efficace de l'énergie – Elimine les caches de données	Soigneusement conçu pour les applications qui exigent un certain taux de fiabilité
RAIT [44] (Reliable Asynchronous Image Transfer)	– Assure une fiabilité saut-par-saut – Gestion efficace de l'énergie – Minimise les caches de données	Conçu pour les applications multimédia (transfert d'images) qui exige la fiabilité de livraison et l'efficacité de l'énergie sans se préoccupé du délai de livraison
HDRTP [35] (A Hybrid And Dynamic Reliable Transport Protocol)	– Assure une fiabilité saut-par-saut (en remédiant les insuffisances de PSFQ) – Evolutive – Réduit le délai de livraison par rapport à PSFQ	Même champ d'applications du **PSFQ** (i.e. amélioration du protocole PSFQ)

TABLE 3.1 : Estimation des performances des solutions proposées

3.2.2 Critique des solutions existantes

Les facteurs de performances ciblés dans le problème de transport varient selon la nature d'application et le phénomène à surveiller. Par exemple, les applications [1] RCSF destinées à surveiller des phénomènes naturelles (i.e., les zones volcaniques ou le niveau d'eau dans un barrage) exigent un certain niveau de fiabilité parce que la redondance de l'information détectée peut récupérer la perte. Par contre, d'autres applications [50] multimédia (i.e., la surveillance des frontières, la télésurveillance) insistent sur la fiabilité de livraison de données en négligeant la consommation énergétique et le délai de livraison, ceci est dû à l'importance de l'information détectée. D'autres applications multimédia (i.e., la détection dans les champs de batailles) considèrent aussi l'efficacité de l'énergie toute en réduisant le délai de livraison.
En effet, les solutions de transport proposées pour les RCSF s'intéressent uniquement aux exigences des applications apparues durant leurs périodes de conception donnant une panoplie de solutions proposées au problème de transport dans les RCSF. Pour répondre aux exigences des applications RCSF, les concepteurs de détecteurs sans fil se retrouvent face à une variété de solutions qui doivent être implémentées dans leurs systèmes d'exploitation. Mais malheureusement les concepteurs de détecteurs sans fil évitent cette variété de propositions parce que leurs dispositifs imposent des contraintes en termes d'espace mémoire. En revanche, une solution qui élargie son champ d'application dans les RCSF reçoit une grande appréciation de la part des concepteurs, ce qui a motivé la conception du protocole CARTEE.
CARTEE a été proposé pour répondre aux exigences des applications multimédia, et également les applications exigeant une fiabilité de livraison de données. Donc, CARTEE est conçu pour combler aux insuffisances des protocoles existants. Pour faire apparaitre clairement les objectifs ciblés par ce protocole, le tableau 3.2

présente les insuffisances de chaque solution proposée pour le problème de transport dans les RCSF.

Solution	Insuffisances
PSFQ	– Conçu pour les communications point-à-multipoint – Absence du contrôle de la congestion – Un gaspillage des ressources énergétique surtout dans la communication multipoint-à-point – Le délai de livraison est important – Les nœuds capteurs nécessitent un cache de données suffisamment grand
ESRT	– La fiabilité n'est pas garantie (sauf la fiabilité désirée par l'application qui est partiellement assurée) – Efficacité de l'énergie uniquement lors de l'absence du mécanisme de fiabilité au niveau routage. – Minimise la congestion avec des données constituées d'un seul segment. Lors de l'utilisation de plusieurs segments de données, la congestion perturbe les performances du réseau et la région d'exploitation optimale n'est jamais atteinte – Dans le cas des données constituées de plusieurs segments, l'ajustement de la fréquence de transmission peut ralentir considérablement le délai de livraison de données.
RMST	– Le mode non-cache cause un gaspillage de l'énergie – Absence du contrôle de congestion – Le délai de livraison est important – Dans le mode cache, les nœuds capteurs nécessitent un cache de données suffisamment grand – Basé sur le routage à diffusion dirigée
CODA	– La fiabilité n'est pas assurée – Le délai de livraison est important – Basé sur le routage à diffusion dirigée
RCRT	– La consommation énergétique est considérablement grande due à la fiabilité de bout-en-bout – Délai de livraison suffisamment grand due à la centralisation de l'ajustement du taux de transmission au niveau du nœud « puits » – Le cache de données au niveau source doit inclure toutes les segments de données tout au long de la connexion. Ceci est dû au mécanisme d'acquittement de bout-en-bout qui repose sur les acquittements négatifs.

Solution	Insuffisances
ERTP	– Un certain taux de fiabilité (spécifié par l'application) est assuré – Absence du contrôle de congestion – Le délai de livraison est important dû au mécanisme Send-And-Wait – Retransmissions non nécessaires due à la sous-estimation du RTO
RAIT	– Délai de livraison suffisamment grand – Absence du contrôle de congestion – Malgré l'optimisation de la consommation énergétique, les situations de congestion peuvent causer un gaspillage de ces ressources.
HDRTP	Possède les mêmes insuffisances de PSFQ

TABLE 3.2 : Insuffisances des solutions proposées dans la litérature

La prochaine section présente une étude théorique sur laquelle est basée la conception du protocole CARTEE qui vise à combler les insuffisances des solutions proposées.

3.3 Support théorique

Comme il est mentionné dans la section précédente, CARTEE a été conçu pour fournir un transport fiable de données pour les applications multimédia RCSF. Dans ces applications, les données détectées (audio, vidéo, ...) se caractérisent de la taille importante due au dispositif de détections (microphone, camera, ...). Vu que les interfaces réseau dans les RCSF possèdent un MTU (unité de transmission maximale) limité et puisque les transmissions reposent sur ces interfaces, la segmentation est indispensable. Par exemple, une image d'1 Ko ne peut être transmise directement dans IEEE 802.15.4 [17] qui possède un MTU de 127 octets. En conséquence, les données multimédia dans ces réseaux nécessitent un mécanisme de segmentation et de réassemblage. Donc, lorsqu'une donnée est détectée, le nœud source doit la segmenter en petit segments (supportable par l'interface réseau) pour les transmettre vers le nœud « puits ». Pour assurer une fiabilité de transport de segments, CARTEE implémente un mécanisme de transmission à base de fenêtre glissante afin de transmettre ces segments vers le puits. En conjonction avec le mécanisme de transmission, CARTEE repose sur un mécanisme d'acquittement implicite combiné avec un mécanisme d'acquittement explicite. La récupération des segments perdus est basée sur une fiabilité saut-par-saut et les segments perdus seront récupérés directement à partir du nœud voisin en avale vers le puits.
Vu que les RCSF sont caractérisés par une communication multipoints-à-point, les flux de données causent des situations de congestion. Pour faire face à ce problème, CARTEE utilise un mécanisme d'ajustement du taux de transmission distribué au niveau de chaque nœud du réseau. En fait, chaque nœud du réseau repose sur une transmission point-à-point pour détecter et acheminer une information vers la station de base. Puisque CARTEE n'est pas une solution de routage, il repose sur la couche transport pour s'assurer de la transmission à partir du nœud source jusqu'au nœud puits.
Les prochaines sections présentent les obstacles et les défis du problème de transport dans les RCSF et proposent des solutions à ces problèmes.

3.3.1 Fiabilité de saut-par-saut

Les auteurs dans [2] montrent que lorsqu'on utilise le transport saut-par-saut au lieu du transport de bout-en-bout, la retransmission est réduite. Le nombre de retransmissions diminue lorsque le nombre de saut augmente (il atteint 61% avec 10 sauts). Le transport de bout-en-bout est plus adéquat pour les réseaux filaires (pour le contrôle de flux). Cependant ce mécanisme n'est pas approprié pour les RCSF et il peut être considéré comme une source de gaspillage d'énergie. Le protocole RCRT utilise le transport de bout-en-bout afin d'assurer la fiabilité et éviter le maintien d'un cache de données au niveau des nœuds intermédiaires. Néanmoins, si on le compare avec le transport saut-par-saut tel que ERTP, RCRT consomme énormément l'énergie. Par conséquent, pour assurer une fiabilité de livraison et en réduisant la consommation de l'énergie, la solution proposée repose sur le transport saut-par-saut.
Pour assurer un transport fiable de données, il est indispensable de faire recours à la retransmission. Donc à partir de cela on peut déduire le lemme suivant :

Lemme 1. *Les retransmissions augmentent le taux de fiabilité*

Démonstration. Le taux de fiabilité r_R peut être obtenu à partir de l'équation suivante :

$$r_R = \frac{n_{reception}}{n_{transmission}} \tag{3.1}$$

Où $n_{transmission}$ représente le nombre de segments réellement transmis à partir de la source vers le puits et $n_{reception}$ est le nombre de segments reçus au niveau du puits. Dans le cas d'une communication à un seul saut $n_{reception} = (1-p) * n_{transmission}$, où p est le taux d'erreur dans le lien de communication. Alors que dans le cas d'une communication de n saut $n_{reception} = ((1-p) * n_{transmission})^n$. Si on considère une communication à un seul saut qui inclus un mécanisme de retransmission, $n_{reception} = \sum_{i=1}^{retrans} (1-p)^i * n_{transmission}$. Avec $retrans$ est le nombre maximum de retransmissions pour une transmission erronée. De ce fait, la fiabilité dans une communication à un seul saut peut être estimée comme suite :

$$r_R = \sum_{i=1}^{retrans} (1-p)^i \tag{3.2}$$

A partir de l'équation 3.2, les retransmissions augmentent le taux de fiabilité dans un seul saut. Par récurrence, la fiabilité dans une communication de n saut est estimée comme suite :

$$r_R = \left(\sum_{i=1}^{retrans} (1-p)^i \right)^n \tag{3.3}$$

□

3.3.2 Maintien du cache de données

Les protocoles RMST et PSFQ utilisent les acquittements négatifs pour retrouver les segments insuffisants. Dans ce mécanisme d'acquittement, chaque nœud maintien un cache local de données et vérifie les séquences de données pour trouver d'éventuelles trous de données. Si un trou est détecté, le nœud génère un acquittement négatif (NACK) pour ses voisins pour demander la retransmission des segments insuffisants. Ce mécanisme semble parfait pour récupérer les segments insuffisants, mais en fait, ce mécanisme soulève le problème du maintien d'un cache de données au niveau de chaque nœud. Un autre problème réside dans le paquet NACK lui-même puisqu'il est un message de contrôle qui consomme de l'énergie. En outre, il n'y a aucune garantie que le message atteint sa destination, ce qui engendre des retransmissions supplémentaires. De plus, la durée d'occupation du cache est indéterminée par manque de notification des segments correctement reçus. Par

exemple, PSFQ et RMST gardent les segments dans le cache jusqu'à la réception d'un nouveau flux. En conséquence, ce mécanisme peut conduire à des situations de congestion ou des pertes de données dans d'autres situations. Une solution qui semble idéale à ce problème est l'utilisation du mécanisme Send-And-Wait, aussi connue sous l'appellation Stop-And-Wait dans ERTP. C'est vrai que cette solution permet d'éliminer le cache de données (provoquant la congestion), mais en revanche, elle augmente la latence. Par ailleurs, il n'est pas évident d'estimer le temps d'attente d'un acquittement, surtout dans un environnement présentant un taux élevé d'erreurs de bit dans les paquets. ESRT utilise une transmission continue sur la base d'une fréquence de transmission. Dans ce cas-ci, cette solution offre une bonne exploitation de la bande passante et ne nécessite pas de cache de données, mais la perte de segments peut ne pas être récupérable.
Pour mieux analyser le problème, le mécanisme de cache de données peut être représenté en utilisant la théorie des files d'attente. La figure 3.1 représente un diagramme d'état transition de la taille de la fille d'attente en utilisant un mécanisme de transmission Send-And-Wait.

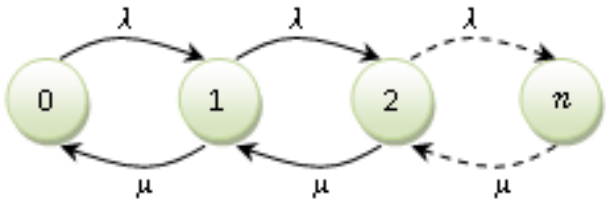

FIGURE 3.1 : Transmission send-and-wait

Dans cette figure, chaque état représente le nombre de segments en attente d'un éventuel acheminement. Le paramètres λ représente le taux d'arrivé des segments et μ c'est le taux d'occupation du segment dans la file d'attente qui nécessite un acquittement. La taille nécessaire de la file est estimée en utilisant la formule d'Erlang [51] suivante :

$$A = \lambda\mu \tag{3.4}$$

Tel que A représente la taille estimée du tampon nécessaire pour stocker les segments nécessitant un acheminement. La probabilité de la perte de segments due à une congestion peut être obtenue à partir de la formule de perte d'Erlang :

$$B\left(A, N\right) = \frac{\frac{A^N}{N!}}{\sum_{j=0}^{N} \frac{A^j}{j!}} \tag{3.5}$$

Où N est la taille actuelle choisie pour le tampon.

Transmission send-and-wait

Dans le mécanisme Send-And-Wait, le taux de transmission d'un segment correcte est estimé comme suite :

$$\mu_{sucee} = \mu_{transmission} + \mu_{ACK} \tag{3.6}$$

En utilisant l'équation 3.6, μ peut être obtenu à partir de l'équation suivante :

$$\mu = \mu_{sucee} + MaxRentrans * \mu_{sucee} * p \tag{3.7}$$

Où $MaxRentrans$ est le nombre maximum de retransmissions tolérées par le mécanisme et p est le taux d'erreur du lien entre le nœud et son voisin en amont vers le puits (on suppose que le lien entre le nœud et son voisin en avale est parfait). En remplaçant l'équation 3.7 dans 3.4, on obtient :

$$A = \lambda\left(\mu_{sucee} + MaxRentrans * \mu_{sucee} * p\right) \tag{3.8}$$

Transmission continue avec NACK

Si on analyse les solutions qui reposent sur la transmission continue en conjonction avec un mécanisme d'acquittement négatif (i.e., PSFQ et RMST), on peut schématiser leurs comportements par un diagramme d'état/transition dans la figure 3.2. Dans ce mécanisme, la taille nécessaire de la file d'attente peut être estimée

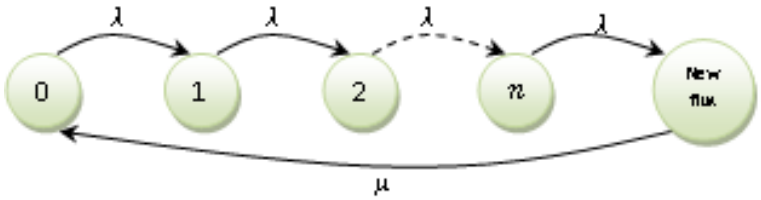

FIGURE 3.2 : Transmission continue avec des acquittements négatifs

comme suit :

$$A = \lambda\,(\mu * n) \tag{3.9}$$

Où n est la taille du flux de données (nombre de segments), et μ peut être obtenu comme suite :

$$\mu = \mu_{transmission} + MaxRentrans * \mu_{transmission} * p \tag{3.10}$$

En remplaçant l'équation 3.10 dans 3.9, on obtient :

$$A = \lambda\,((\mu_{transmission} + MaxRentrans * \mu_{transmission} * p) * n) \tag{3.11}$$

Transmission à base de fenêtre glissante

La figure 3.3 représente le diagramme d'état transition de la transmission à base de fenêtre glissante en utilisant une taille de fenêtre notée w. Dans ce mécanisme, la taille nécessaire de la file d'attente est estimée comme

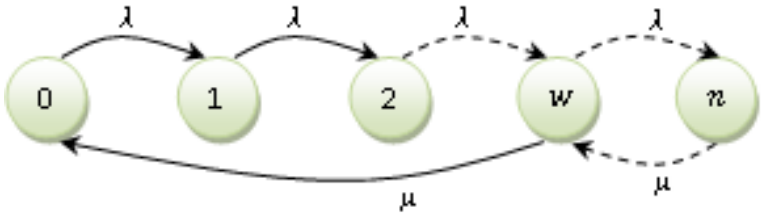

FIGURE 3.3 : Transmission à base de fenêtre glissante

suite :

$$A = \lambda\,((\mu_T * w) + \mu_{ACK}) \tag{3.12}$$

Où μ_T est calculé par la formule :

$$\mu_T = \mu_{transmission} + MaxRentrans * \mu_{transmission} * p \tag{3.13}$$

En remplaçant l'équation 3.13 dans l'équation 3.12, on obtient :

$$A = \lambda\,[((\mu_{transmission} + MaxRentrans * \mu_{transmission} * p) * w) + \mu_{ACK}] \tag{3.14}$$

Par exemple, si on considère un nœud ayant un taux de réception $\lambda = 20$ segments/seconde, un taux de transmission $\mu_{transmission} = 2$ segments/seconde, un taux d'acquittement $\mu_{ACK} = 2$ segments/second et

$MaxRentrans = 4$. En variant p (taux des erreurs dans le lien de communication) entre 0.01 et 0.9, la figure 3.4 démontre le changement de la latence selon la qualité de liens en appliquant les équations 3.8, 3.11 et 3.14. La transmission continue avec des acquittements négatifs surpasse les performances du mécanisme Send-And-Wait en termes de latence. Cependant, la transmission à base de fenêtre glissante a atteint des performances plus proches à la transmission continue.

En appliquant l'équation 3.5 dont A est affecté aux équations 3.8, 3.11 et 3.14, la figure 3.5 représente la

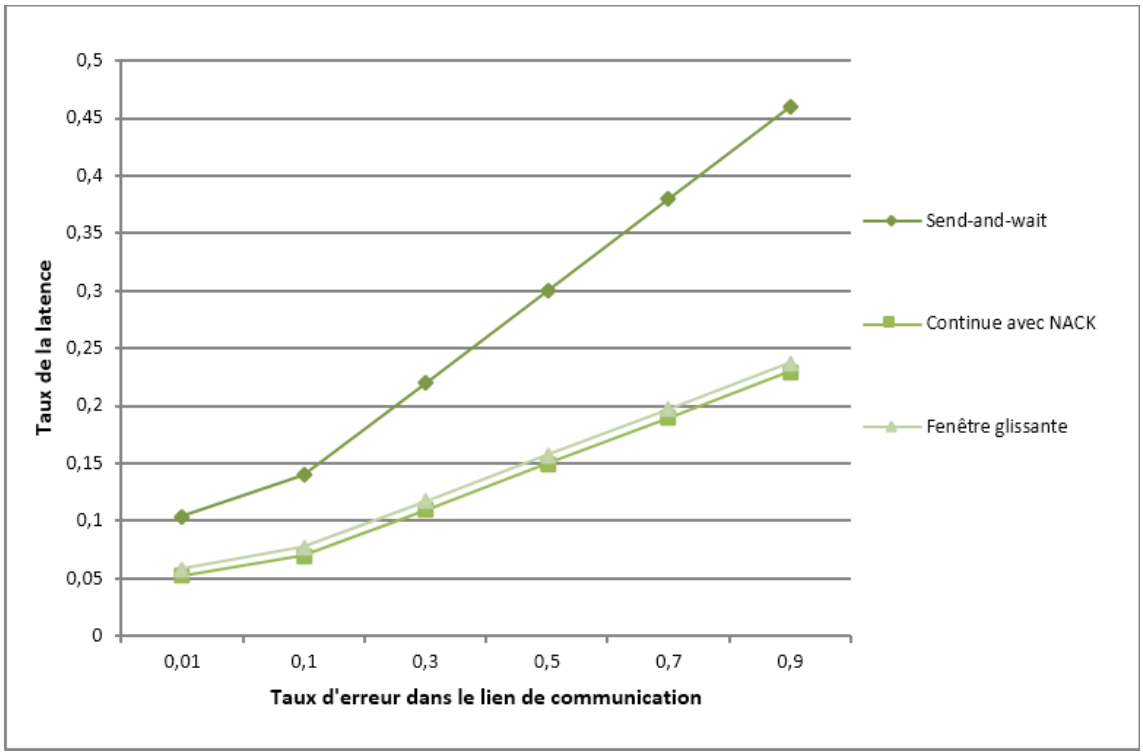

FIGURE 3.4 : La latence selon la qualité du lien

probabilité de congestion selon la qualité du lien en utilisant un cache de données d'une taille 20 segments.

Malgré les meilleures performances de la transmission continue avec un mécanisme d'acquittement négatif en termes de latence, les congestions dans ce mécanisme sont vraiment présentes. D'autre part, la congestion dans le mécanisme Send-And-Wait est inexistante dans la plupart du temps, mais il est médiocre en termes de latence. La motivation principale pour adopter un mécanisme de transmission à base de fenêtre glissante dans CARTEE est pour minimiser la latence et la congestion que les applications multimédia dans les RCSF ne supportent pas. CARTEE rajoute un mécanisme d'ajustement du taux de transmission pour éliminer les situations de congestion.

3.3.3 Acquittement

La plupart des solutions proposées utilisent des mécanismes d'acquittement explicite pour notifier un nœud à propos des segments insuffisants (tel que PSFQ et RMST). Dans ce mécanisme, chaque nœud qui a détecté un trou dans la séquence des segments reçus, envoie un acquittement négatif ou positif au nœud qui est capable à récupérer cette séquence insuffisante. Lorsque le nœud en question reçoit cet acquittement, il retransmet les segments insuffisants. Malheureusement, la bonne réception de l'acquittement n'est pas garantie dans cette situation, et les trous au niveau de la séquence peuvent persister pendant une longue durée de temps, surtout dans les liens de mauvaises qualités. Donc, la retransmission de l'acquittement dans cette situation est inévitable (par exemple, Dans le cas où la séquence insuffisante ne s'est pas récupérée après l'expiration du temps

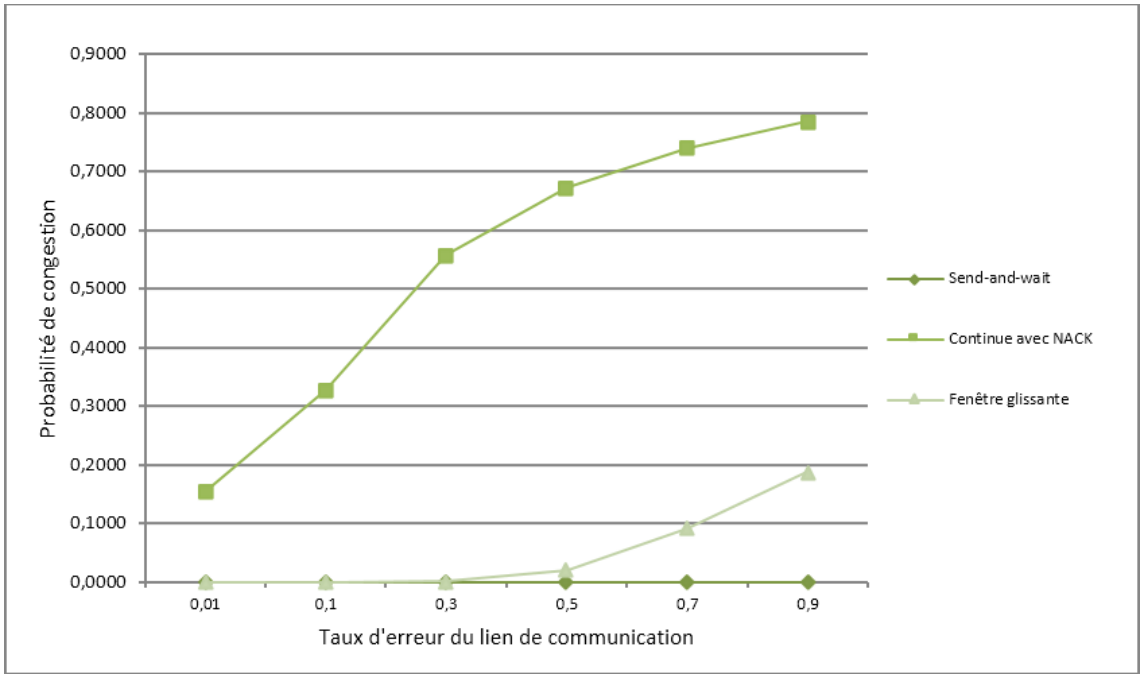

FIGURE 3.5 : Probabilité de congestion par rapport à la qualité du lien

d'acquittement, PSFQ retransmet un autre acquittement). En conséquence, la transmission et la retransmission des acquittements génèrent un gaspillage de l'énergie ce qui peut réduire la longévité du réseau.
Pour faire face à ce gaspillage, ERTP utilise un mécanisme d'acquittement implicite (Implicit Acknowledgment (NACK)). Dans ce mécanisme, un nœud ayant envoyé un message, écoute sa transmission à partir du prochain nœud pour affirmer sa réception. En d'autres termes, si le message est écouté, la transmission est considérée comme achevée, sinon la transmission est supposée comme échouée. ERTP utilise un mécanisme d'écoute à faible puissance (LPL) [52] pour réduire l'énergie de l'écoute. Donc, en utilisant ces deux mécanismes, l'énergie peut être considérablement conservée.
Pour assurer une consommation efficace de l'énergie, CARTEE est basé principalement sur les deux mécanismes cités en haut. Effectivement, CARTEE écoute un NACK à la fin de chaque intervalle de fenêtre glissante, ce qui réduit en plus la consommation énergétique.

3.3.4 Adaptation du taux de transmission

Le mécanisme d'ajustement du taux de transmission est utilisé pour réagir aux situations de congestion. On trouve dans la littérature deux types d'adaptation : centralisée et distribuée. Le premier type, qui est l'adaptation centralisée du taux de transmission (utilisée par ESRT et RCRT) est effectué au niveau du puits où ce dernier estime le taux de transmission et le diffuse vers les nœuds sources. A la réception d'un paquet d'ajustement, les nœuds sources ajustent leurs fréquences de transmission suivant la valeur estimée afin de transmettre les segments vers le puits selon cette fréquence. Le deuxième type est l'adaptation distribuée du taux de transmission (implémenté par CODA) est utilisé pour éviter la congestion dans le réseau et optimiser l'exploitation de la bande passante. En fait, le choix entre un mécanisme d'ajustement centralisé et distribuée est crucial dans la conception d'une solution de transport, et il a un impact sur les facteurs de performances du réseau. Pour clarifier en mieux et étudier chaque mécanisme, un RCSF peut être considérer comme un problème de flow tel qu'il est présenté dans la figure 3.6. Dans cette figure, $G = (V, U)$ est un graphe de flow où S_{ij} sont des nœuds sources et I_j sont les nœuds intermédiaires. Les arcs c_{ij} sont les capacités assignées aux arcs reliant les sources aux sommets intermédiaires, et les arcs c_j sont les capacités affectés aux arcs intermédiaires per-

mettant de relier les nœuds intermédiaires au nœud puits $(0 < i \leq n; 0 < j \leq m)$. Le flux attribué à un arc $u_{ij}; u_{ij} \in U$ est noté par f_{ij}. Pour exploiter la bande passante du réseau d'une manière optimale :

1. f_{ij} doit être ajusté selon la capacité c_{ij} $(f_{ij} \leq c_{ij})$
2. c_j doit être égale à la somme des f_{ij} $(c_j = \sum_{i=1}^{n} f_{ij})$

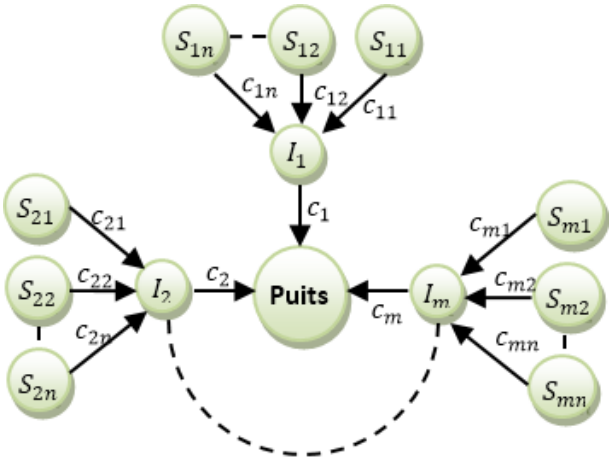

FIGURE 3.6 : Problème d'adaptation du taux de transmission (problème de flow)

Contrôle du taux de transmission centralisé

Le protocole ESRT utilise une adaptation du taux de transmission centralisée, attribue une valeur unique f_{ij} (notée f) et ajuste cette valeur jusqu'à atteindre la région d'exploitation optimale (OOR). Dans cette solution, l'exploitation optimale de la bande passante n'est atteinte que si les capacités c_j sont égales. Mais dans la plupart des cas, les capacités c_j possèdent différentes valeurs et elles dépondent des performances de leurs sauts suivants. Donc, on peut conclure que l'adaptation centralisée ne peut pas atteindre l'exploitation optimale de la bande passante. ESRT repose sur ce mécanisme uniquement pour réduire le traitement au niveau des nœuds capteurs. En outre, dans l'adaptation centralisée, la nouvelle estimation du taux de transmission nécessite une diffusion vers les sources, ce qui augmente considérablement le coût des communications.
RCRT utilise un mécanisme d'adaptation centralisé pour ajuster la fréquence de transmission au niveau des sources. Exactement, dans RCRT, le puits estime le taux de transmission de chaque flux de donnée et envoie ce taux vers le nœud source concerné. En plus, le mécanisme de détection de congestion est exécuté au niveau du nœud puits dans ce protocole. L'objectif principale de cette solution est que (1) les fréquences f_{ij} sont estimées sur la base de la capacité c_{ij}, et (2), la formule $c_j = \sum_{i=1}^{n} f_{ij}$ est vérifiée. Cependant, l'attribution de la nouvelle fréquence nécessite plus de communication entrainant un gaspillage considérable de l'énergie des nœuds capteurs.

Contrôle du taux de transmission distribué

CODA implémente deux composants principales pour ajuster le taux de transmission. Le premier composant (open-loop hop-by-hop back pressure) est déclenché lorsqu'un nœud détecte une situation de congestion. L'insuffisance de ce composant réside dans l'ajustement de la fréquence f_j qui est calculée sur la base de la capacité c_j au lieu de c_{ij}, et la condition $c_j = \sum_{i=1}^{n} f_{ij}$ n'est pas vérifiée. Une fois, le taux est calculé, le nœud en question le diffuse vers ces voisins. Le deuxième composant (close-loop multi-source regulation) est déclenché lorsqu'un nœud source détecte un taux de transmission élevé. Dans ce cas-ci, le nœud source ajoute

une information dans les paquets envoyés signalant une situation de congestion au niveau du puits. Afin d'éviter cette congestion, et lors de la réception de ces paquets, le puits envoie un paquet de contrôle demandant l'ajustement du taux de transmission au niveau des nœuds sources concernés. L'avantage de ce composant est que 1 la fréquence f_{ij} est assignée à chaque arc u_{ij} selon la capacité c_{ij}, et 2 l'équation $c_j = \sum_{i=1}^{n} f_{ij}$ est vérifiée. CODA exploite la bande passante d'une manière optimale en utilisant le composant close-loop multi-source regulation, mais le coût de ce composant en terme de communication est élevé. D'un autre côté, le coût des communications du composant open-loop hop-by-hop back pressure est faible mais il n'exploite pas la bande passante d'une manière optimale.
Dans le souci de réduire le coût de communications, CARTEE implémente un contrôle du taux de transmission distribué, dans lequel chaque nœud ayant détecté un taux de transmission élevé par rapport au taux de réception, doit inclure le taux de transmission approprié dans les segments transmis. Chaque nœud qui a écouté ces fragments doit ajuster son taux de transmission selon le taux inclus dans le segment. Dans ce mécanisme, CARTEE 1 attribue des fréquences f_{ij} sur la base des capacités c_{ij} (tel que $f_{ij} \leq c_{ij}$) et 2 l'équation $c_j = \sum_{i=1}^{n} f_{ij}$ est vérifiée. En conséquence, CARTEE ne nécessite pas des traitements et des espaces mémoires supplémentaires au niveau des nœuds capteurs.

3.4 Description

CARTEE invite tous les nœuds du réseau (sources, intermédiaires et puits) à collaborer pour assurer la fiabilité. Les solutions centralisées de transport fiable (tel que ESRT) nécessite plus de communications par rapport aux solutions distribuées. Aussi, il est évident que la consommation énergétique durant la détection et le traitement est négligée par rapport à la consommation lors des communications [5, 3, 4]. Donc, afin de réduire la consommation énergétique, les communications doivent être réduites dans la mesure du possible. Pour atteindre ces finalités, CARTEE implémente une solution distribuée dont laquelle un nœud contribuant à la communication, doit être impliqué dans la livraison fiable de transport de données.
Pour offrir un transport fiable, CARTEE est essentiellement basée sur quatre mécanismes : transmission à base de fenêtre glissante, acquittement implicite/explicite, détection de congestion et ajustement du taux de transmission distribué. Ce comportement peut être résumé dans le diagramme d'état/transition (figure 3.7). A chaque moment CARTEE peut être dans l'un des états suivants :

- Inactif : Initialement, ce protocole est dans un état inactif jusqu'à la réception des fragments à partir de la couche réseau ou un flux à partir de la couche application.
- Envoie des segments : Dans cet état, CARTEE estime le taux de réception. En cas de réception d'un acquittement implicite ou explicite, CARTEE enlève les segments correctement reçus des tampons du cache et envoie les segments correspondant à l'intervalle de la fenêtre glissante tout en estimant, le délai de transmission des segments transmis et vérifie la probabilité de situation de congestion à partir des taux de transmission et de réception estimés.
- Attente d'un acquittement : Dans cet état, CARTEE se met en attente de la réception d'un acquittement implicite ou explicite afin de mettre à jour son tampon.
- Attente des segments : S'il n'y a pas de segment à transmettre et le flux de données est encore insuffisant, CARTEE se met en attente du reste des segments. Dans cet état, CARTEE transmet un acquittement explicite à chaque expiration de l'intervalle d'acquittement.

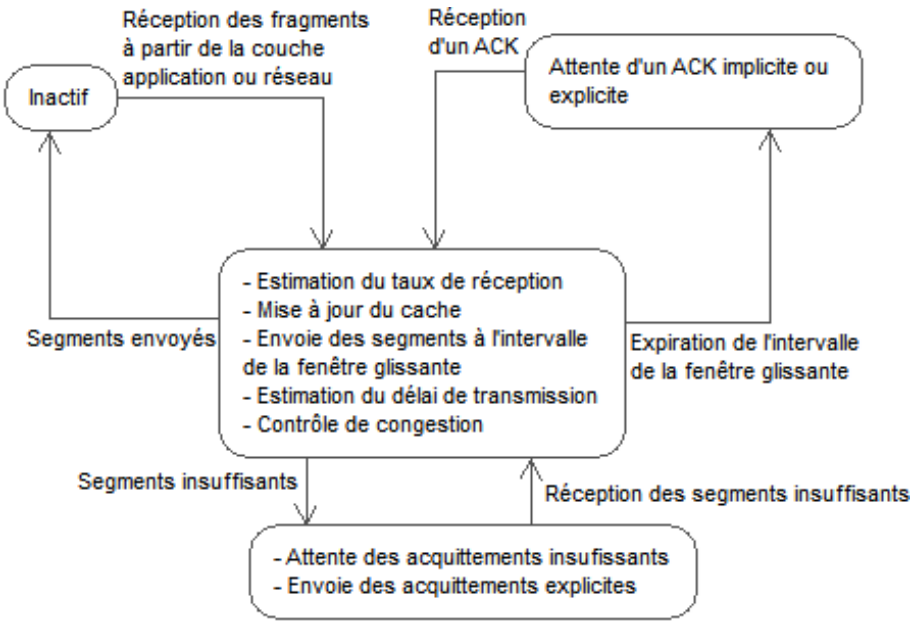

FIGURE 3.7 : Comportement du protocole CARTEE

3.4.1 Mécanisme de transmission

Fonctionnement au niveau du nœud source

Dans CARTEE, lorsque la couche transport reçoit un flux de donnée à partir de la couche application, elle le divise en segments pouvant être transmis par l'interface réseau. Les segments obtenus après fragmentation sont insérés dans le tampon de transmission pour être transmis par la suite vers le puits en utilisant une transmission à base de fenêtre glissante. A la fin de chaque intervalle de la fenêtre glissante, la couche transport attend un NACK à partir de la couche MAC. A la réception d'un NACK, le nœud source ou intermédiaire vérifie la séquence de données et exécute un test de détection de congestion, et réagit selon le cas approprié (retransmission ou ajustement du taux de transmission). Si aucun NACK n'est détecté pendant un intervalle $IACKWaitingTime$, le nœud suppose que le puits est inatteignable. La figure 3.8 représente un diagramme d'activité du nœud source.

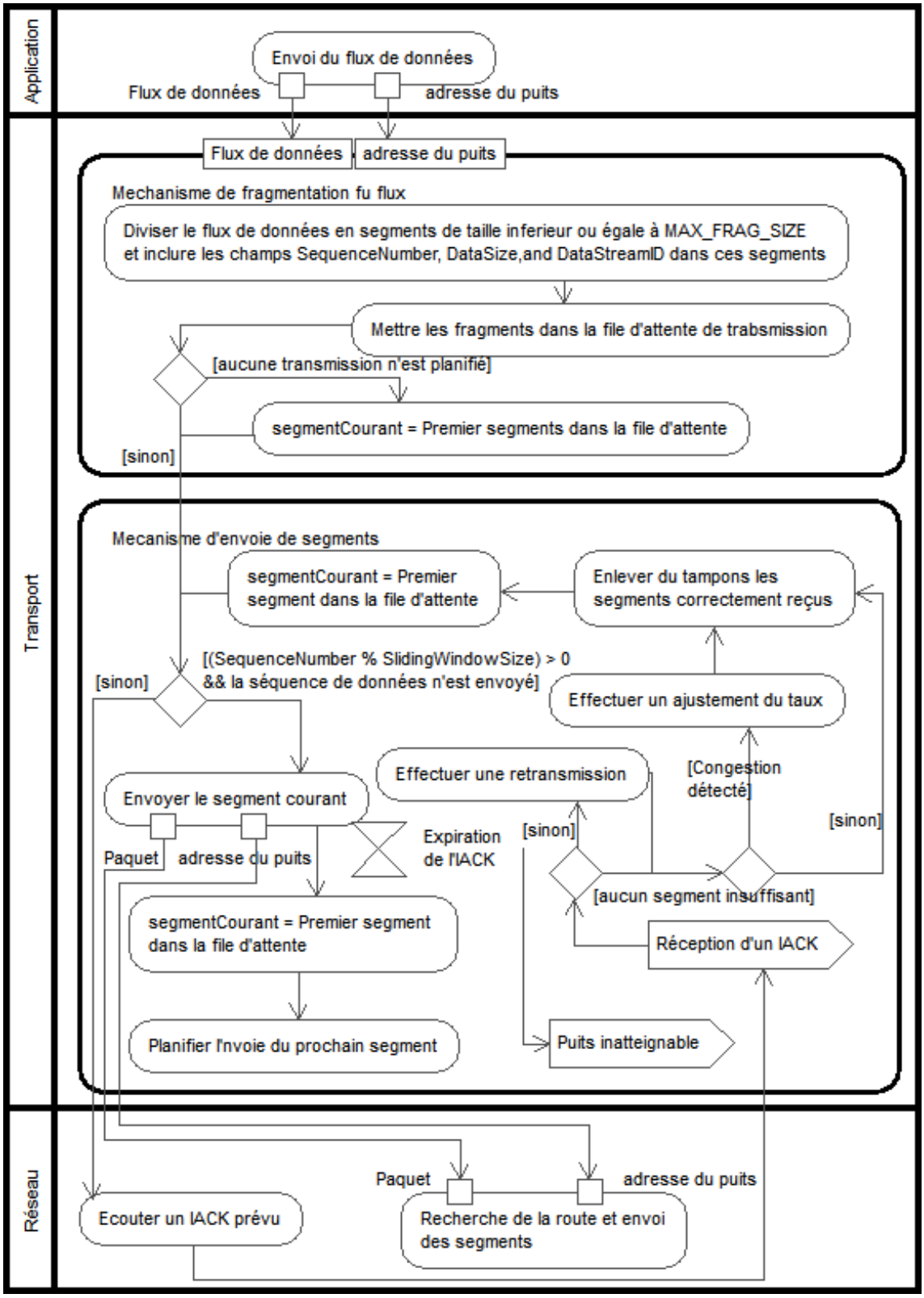

FIGURE 3.8 : Diagramme d'activité du nœud source

Fonctionnement au niveau du nœud intermédiaire

A la réception d'un segment, le nœud intermédiaire vérifie la séquence de segments reçus pour détecter d'éventuels trous. En présence de trou, la séquence insuffisante est incluse dans le segment reçu et le segment est mis dans la file d'attente de transmission. Afin de transmettre le segment, le nœud intermédiaire exécute le méca-

nisme d'envoi des segments décrit dans la figure 3.8. Avant d'acheminer les segments, le mécanisme d'envoi déclenche le mécanisme de détection de congestion. En présence d'une congestion et après l'estimation du taux de transmission, le nouveau taux de transmission est inclus dans le segment à acheminer. Dans la solution CARTEE, le champ des séquences insuffisantes et celui du taux de transmission sont utilisés pour être écoutés par le voisin en amont pour lui permettre de retransmettre les segments insuffisants ou ajuster son taux de transmission.

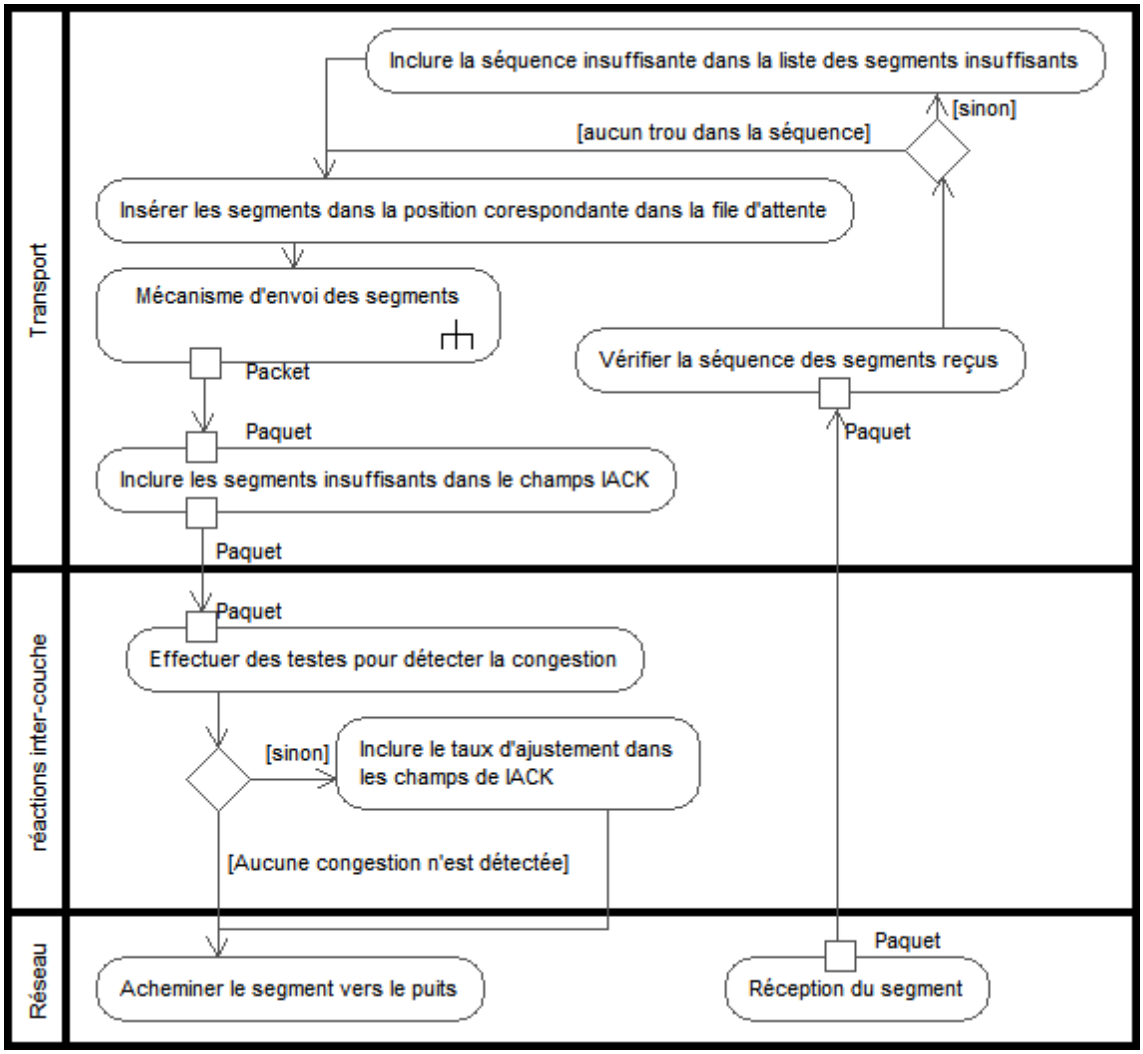

FIGURE 3.9 : Diagramme d'activité du nœud intermédiaire

Transmission à base de fenêtre glissante fixe

Afin de réduire la latence et minimiser l'occupation des caches de données au niveau des nœuds intermédiaires, la transmission à base de fenêtre glissante fixe est utilisée dans ce protocole. Le choix de la taille de la fenêtre est crucial parce qu'elle influe profondément sur la latence et l'occupation du cache de données. L'utilisation d'une taille réduite pour la fenêtre augmente la latence, alors qu'une plus grande taille nécessite un cache de données de taille importante.

Selon notre étude (équation 3.14), l'utilisation d'une taille pour une fenêtre glissante fixée à la valeur 1 ($w = 1$), effectue une latence est similaire à celle du mécanisme Send-And-Wait. En incrémentant la valeur, la latence converge vers la transmission continue. On observe également qu'après la valeur $w = 7$, aucune amélioration de la latence n'est constatée (i.e. devient négligeable). Donc, dans le protocole CARTEE, la taille

de la fenêtre w est fixée à la valeur sept (07) pour réduire la latence et l'occupation du cache de données. De plus, le choix de la valeur 7 est lié aussi au champ d'acquittement qui est lui aussi de taille 7 bits. Ce champ de contrôle sera discuté dans la section suivante.
Le nœud puits collecte les segments reçus afin de les rassembler dans un flux donné complet. Lorsque le flux est complet, il sera acheminé vers la couche application.

3.4.2 Mécanisme d'acquittement

CARTEE utilise la combinaison des acquittements implicites et explicites pour notifier un nœud émetteur à propos de l'état de réception des segments. Dans ce protocole, les paquets de données ainsi que les acquittements possèdent un champ ACK permettant de signaler l'état de réception des segments. La structure générale d'un paquet CARTEE est présentée dans la figure suivante :

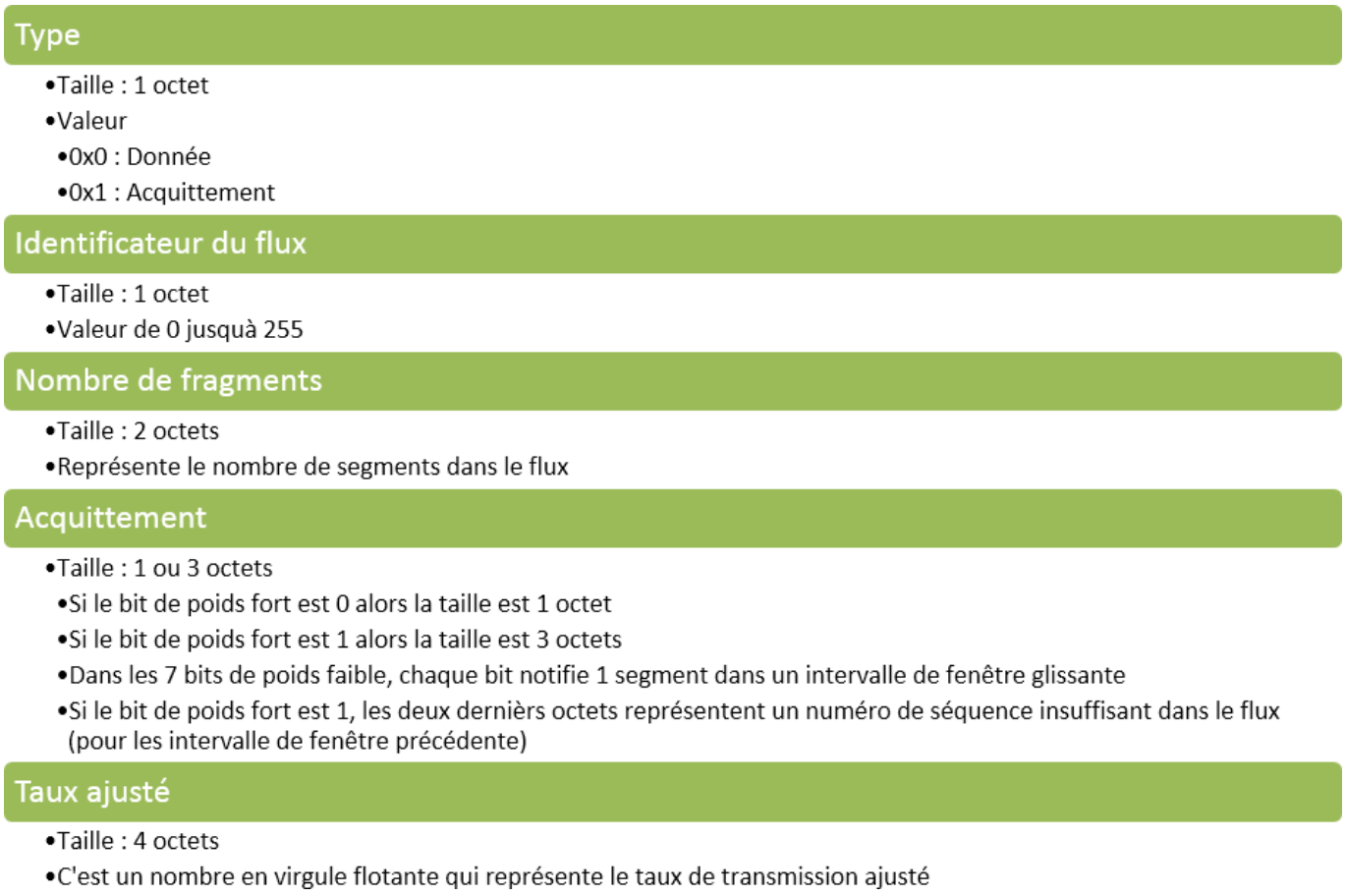

FIGURE 3.10 : Structure du paquet CARTEE

Acquittement implicite

Chaque nœud source ou intermédiaire transmet n segments dans chaque intervalle de fenêtre glissante, où n est obtenu comme suite :

$$n \leq w \left\{ \begin{array}{ll} n = w & , n \textit{ paquets transmets} \\ n \leq w & , n \textit{ paquets retransmets} \end{array} \right\} \tag{3.15}$$

Un intervalle de fenêtre glissante est divisé en deux intervalles : transmission (I_{tx}) et acquittement I_{ACK}. L'intervalle de transmission est variable et peut être calculé en utilisant l'équation suivante :

$$I_{tx} = n * f \tag{3.16}$$

Où f est le taux de transmission du nœud. Après la transmission de n segments, chaque nœud émetteur doit attendre un temps I_{ACK} pour qu'il écoute les segments transmis par son voisin (le prochain nœud vers le puits). Si un segment est écouté durant cet intervalle, le nœud récupère le champ acquittement à partir du segment afin de vérifier l'état de réception des segments. Les segments correctement reçus seront éliminés de la file d'attente de transmission par contre les segments insuffisants restent dans cette file d'attente. Après cette étape, le nœud reprend sa transmission à partir du début de la file d'attente.
Le numéro de séquence des segments reçus (qui doivent être enlevés de la file d'attente) peuvent être obtenu à partir de l'équation suivante :

$$\begin{array}{lcl} seqNo & = & w_{index} * w \\ w_{index} & = & \frac{F_{seqNo}}{w}; 0 \leq w_{index} < \frac{F_{fragCount}}{w} \\ ACK_{seqNo} & = & seqNo + i \end{array} \tag{3.17}$$

Avec i, l'index de chaque bit ayant la valeur 1 dans le champ d'acquittement ($0 \leq i < w$), et $w_{index} \in N$, l'index de la fenêtre glissante utilisée pour transmettre les segments. F_{seqNo} est le numéro de séquence du segment écouté, $F_{fragCount}$, représente le nombre de segments constituants le flux de données, $seqNo$, le premier numéro de séquence dans la fenêtre glissante ayant l'index w_{index} et ACK_{seqNo}, le numéro de séquence du segment correctement reçu.

Acquittement explicite

La taille du champ d'acquittement est souvent 1 octet, et il peut être 3 octets lorsque le segment insuffisant possède un numéro de séquence inferieur à $seqNo_s$. Dans une telle situation, le premier octet doit ajuster son bit de poids fort à 1 et les deux derniers octets indiquent le numéro de séquence du segment manquant. Puisque rien ne garantit qu'un nœud émetteur puisse clairement écouter son voisin, le mécanisme d'acquittement explicite doit être introduit. Exactement, lorsqu'un nœud (e.g., N_{i+1} dans la figure 3.12) observe un retard dans la livraison de données (après un durée t_{EACK}), il génère un acquittement explicite au nœud précédant (e.g., N_i dans la figure 3.12). Si aucun segment n'est reçu, le nœud N_{i+1} tente de retransmettre des acquittements explicites pendant chaque intervalle I_{EACK}. Les tentatives s'arrêtent après un nombre de retransmissions ($MaxRetries$).

Le temps d'expiration de retransmission (RTO) estimé par le protocole ERTP est une source de gaspillage de l'énergie qui provoque des retransmissions non nécessaires, ceci est à cause de la bande passante fixée par cette solution. La bande passante d'une interface de communication destinée pour les RCSF ne peut pas être fixe, et elle dépend du canal sélectionné au niveau de la couche physique. Pour éviter cette estimation, CARTEE délègue cette tâche au nœud récepteur qui possède des informations supplémentaires (comme le taux de réception des segments pour estimer t_{EACK} et I_{EACK}. Ces deux paramètres ont un impact sur la surcharge dans le réseau (dû aux paquets de contrôles EACK). CARTEE prolonge l'attente d'acquittement explicite en utilisant un temps noté MAX_WAIT_TIME (fixé à une valeur suffisamment grande) pour garantir que les acquittements explicites seront effectués dans cette période.

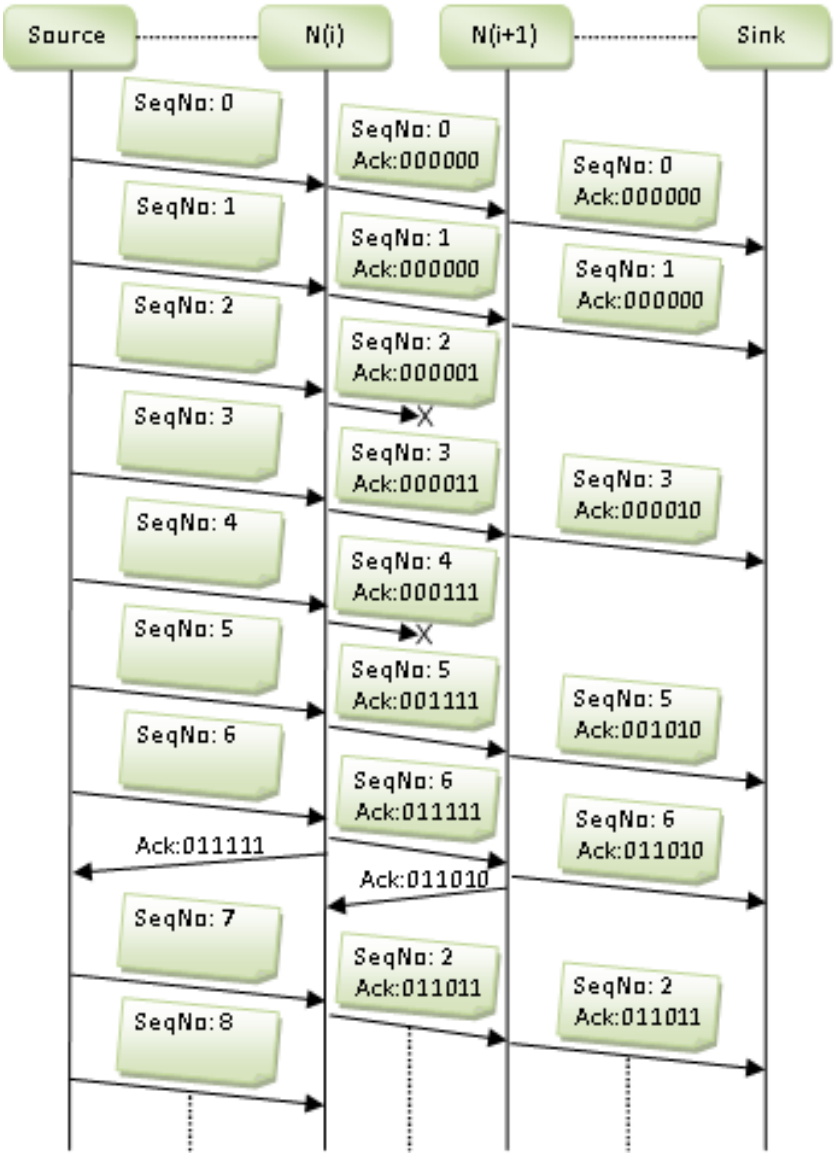

FIGURE 3.11 : Transactions des acquittements implicites

Estimation du temps de l'acquittement explicite

L'acquittement explicite est utilisé lorsqu'un acquittement implicite échoue parce qu'il possède plus de chances pour être reçu par rapport à l'acquittement implicite. Donc, la transmission d'un acquittement explicite s'effectue après la détection d'un échec pour écouter un acquittement implicite. CARTEE calcule t_{EACK} en utilisant une technique d'estimation de la moyenne du temps de réception des segments (la technique EWMA [42]).

$$t_{EACK} = \frac{\sum_{i=1}^{w} W_i * TempsReception_i}{\sum_{i=0}^{w} W_i} * a \tag{3.18}$$

Avec $TempsReception_i$ représente le temps estimé entre le segment F_i et le segment F_{i-1} d'une fenêtre glissante, a, le nombre de fragments insuffisants dans une fenêtre glissante ($0 < a \leq w$) (i.e. calculé en se basant sur les segments reçus au niveau du nœud récepteur). Les poids W_i utilisés dans CARTEE sont représentés dans le vecteur $Weight = \{1, 1, 1, 0.8, 0.6, 0.4, 0.2\}$.
Pour estimer l'intervalle I_{EACK} et réduire la surcharge dans le réseau, le temps d'aller-retour (RTT) entre l'émetteur et le récepteur doit être calculé. Le RTT peut être obtenue à partir de la couche MAC de l'interface IEEE 802.15.4 en utilisant la primitive PD-DATA-Indication.

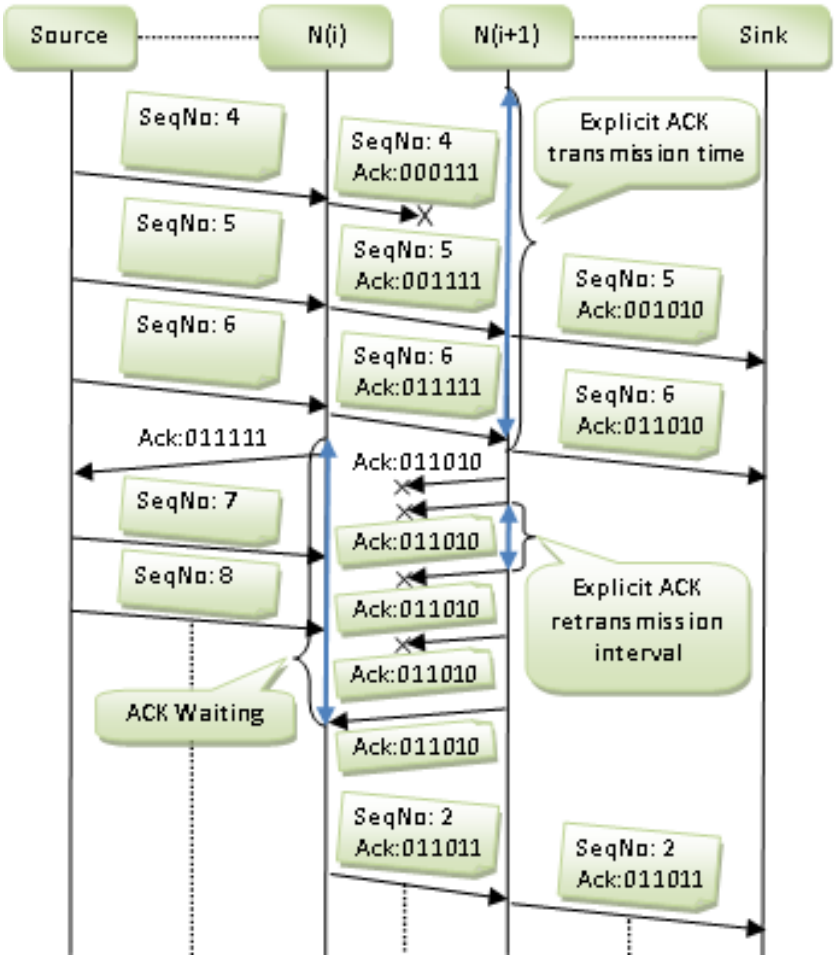

FIGURE 3.12 : Transactions des acquittements explicites

3.4.3 Mécanisme de détection de congestion

La source principale de la perte de données dans les RCSF provient soit des liens momentanément ou durablement erronés, soit causé par des situations de congestion. Pour récupérer cette perte, un mécanisme de retransmission est un indispensable, malgré son coût en fonction de consommation énergétique. Les RCSF sont conçus pour fonctionner avec une source d'énergie autonome pour une période de temps suffisamment large (i.e. des jours pour ne pas dire des mois), et parfois, ces sources d'énergie sont irremplaçables ou non-rechargeables. Donc, le coût des communications doit être optimisé et les retransmissions doivent être réduites. Les situations des liens erronés sont causées par des facteurs externes tels que l'atténuation du signal, le bruit, les interférences et d'autres obstacles physiques. Le contrôle de ces facteurs n'est pas abordé par CARTEE qui est dédié uniquement aux retransmissions.

La disparition d'un lien dans les RCSF est due à une demande de l'application ou à d'autres problèmes techniques (un crash ou redémarrage du nœud, batterie épuisée, ...). Dans tous les cas, les communications utilisant ce lien doivent être arrêtées et les nœuds sources doivent être informés (une requête de découverte d'un autre chemin doit être lancée). Après la transmission de $n \leq w$ segments en utilisant CARTEE, chaque nœud bascule vers l'état d'attente jusqu'à la réception d'un acquittement implicite ou explicite. Si aucun acquittement n'est reçus après $MAX\_WAIT\_TIME = 400mS$, le nœud suppose que le puits est inatteignable, suspend la communication et lance la découverte d'un nouveau chemin à partir de la couche réseau. Une fois le chemin est établit, le nœud reprend la communication à partir des segments considérés comme non reçus.

Solutions existantes

La congestion est produite lorsque le taux de réception excède les capacités d'acheminement des paquets. Pour alléger ce problème, la couche réseau maintien une file d'attente des paquets reçus. Lorsque la congestion est persistante, la file d'attente devient saturée et les paquets reçus doivent être retirés de cette file (selon la politique de gestion de la file), ce qui provoque une perte de paquets due à la congestion. Plusieurs solutions de transport ont abordés ce défi dans les RCSF. Le protocole ESRT est l'un des premiers protocoles qui a abordé ce problème en implémentant un mécanisme de détection de congestion basé sur l'occupation de la file d'attente. D'une manière plus précise, après chaque transmission dans ESRT, chaque nœud capteur doit calculer le taux d'occupation de sa file d'attente. Lorsqu'un nœud détecte que ce taux cause une situation de congestion pendant la prochaine transmission, il ajuste le champ de notification de congestion à la valeur 1 dans chaque fragment reçu, puis il l'achemine vers le puits. De cette façon et après la réception de ce fragment, le puits est notifié qu'une éventuelle congestion se produit, ce qui lui permettra de réduire et diffuser la fréquence de transmission au niveau des nœuds sources.
En fait, CODA suppose que le taux d'occupation de la file d'attente ne reflète pas vraiment les situations de congestion parce que cette notification est insuffisante, et la saturation au niveau du canal de communication est considérée aussi comme une situation de congestion. Dans cette optique, CODA propose un nouveau mécanisme de détection de congestion basé sur l'estimation au niveau du canal de transmission où les nœuds détectent la congestion par écoute du canal. En cas d'une congestion, le nœud diffuse un paquet de contrôle pour demander aux nœuds voisins de réduire leurs taux de transmission.
Dans le protocole RCRT, la congestion est détectée au niveau du puits en estimant le délai de récupération d'une perte de données. A l'apparition d'une perte de données, le nœud puits demande les données perdues et estime leurs durée de récupération. Si la durée excède une certaine période de temps (fixée par RCRT), le puits suppose qu'une congestion s'est produite dans le réseau, et il déclenche le mécanisme d'adaptation du taux de transmission pour ajuster le taux de transmission des sources suspectées par cette congestion.

Analyse de la situation de congestion

Les solutions ESRT, CODA et RCRT possèdent des avantages et des inconvénients. Parce que toutes ces solutions ne réagissent pas parfaitement aux situations de congestion. Pour remédier à ce problème, CARTEE propose un nouveau mécanisme de contrôle de congestion basé sur l'estimation du taux d'émission et de réception. Tous simplement, lorsqu'un nœud détecte que son taux de réception excèdent celui de la transmission, il prévoit une congestion. Pour éviter cette situation, le taux de transmission doit être ajusté selon le taux de réception.

Lemme 2. *Au niveau d'un nœud, la congestion est probable lorsque le taux de transmission est faible par rapport au taux de réception.*

Démonstration. Le processus de transmission/réception au niveau d'un nœud peut être représenté en un processus de naissance et de mort Markovien [53]. Chaque état dans ce processus représente le nombre de paquets présents dans un nœud. On suppose que la capacité de la file d'attente est $n-2$ paquets (donc un nœud peut inclure $n-1$ paquets), ce qui signifie que le nombre d'états est $n+1$ (indexés de 0 à n). Lorsque le processus atteint l'état n, ceci signifie que la congestion s'est produite au niveau du nœud. La figure 3.1 représente le diagramme d'état transition reflétant ce processus. Dans une chaine de Markov, on note le taux de réception de paquets par $\lambda\epsilon + o(\epsilon)$ et le taux de transmission par $\mu\epsilon + o(\epsilon)$. La matrice de transition peut être représentée

comme suite :

$$P(\epsilon) = \begin{pmatrix} 1-\lambda\epsilon & \lambda\epsilon & 0 & \cdots & 0 \\ \mu\epsilon & 1-(\lambda+\mu)\epsilon & \lambda\epsilon & \cdots & 0 \\ 0 & \mu\epsilon & 1-(\lambda+\mu)\epsilon & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1-(\lambda+\mu)\epsilon \end{pmatrix}$$

Chaque vecteur ligne dans la matrice $P(\epsilon)$ est noté $\pi_i(\epsilon)$ tel que $0 \leq i \leq n$. Un vecteur $\pi_i(\epsilon)$ où $i \geq 1$ s'interprète par $i-1$ paquets dans la file d'attente et un 1 paquet en cours de transmission. Dans le cas où $i = 0$, cela veut dire que la file d'attente est vide et aucun paquet n'est à transmettre (état inactif). Chaque élément $\pi_{ij}(\epsilon)$ du vecteur $\pi_i(\epsilon)$ représente la probabilité que le contenu de la file d'attente change de $i-1$ à $j-1$ (Avec $0 \leq j \leq n$). Un nœud est congestionné lorsqu'il est à l'état $n-1$ ($N = \pi_{n-1}(\epsilon)$) et il va basculé vers l'état n. La probabilité de cette transition est :

$$Prob\{N = \pi_n(\epsilon)\} = Prob\{Congestion\} = \lambda\epsilon + o(\epsilon) \tag{3.19}$$

Si le taux de réception augmente la probabilité de congestion augmente. Le vecteur $\pi_{n-1}(\epsilon)$ est stochastique, i.e.,$\sum_{j=0}^{n} \pi_{(n-1)j} = 1$. Donc l'équation suivante est vérifiée :

$$\lambda \geq \mu \Rightarrow Prob\{Congestion\} \in [\mu\epsilon + o(\epsilon), 1] \tag{3.20}$$

Ce qui reflète la propriété du lemme. □

La nouvelle solution de CARTEE

CARTEE détecte la congestion en estimant le taux de transmission (noté f_{tx}) et de réception (noté f_{rx}). Dans le cas où ces taux sont différents, CARTEE déclenche le mécanisme d'adaptation du taux pour assurer un état stationnaire au niveau de chaque nœud ($f_{tx} = f_{rx}$). En effet, ces taux sont estimés en utilisant la technique EWMA. Pour estimer le taux de réception CARTEE calcule la moyenne du temps de réception des fragments, quant à la transmission, les primitives de la couche MAC sont utilisées. Le taux f_{tx} est calculé comme suite :

$$f_{tx} = \frac{1000mS}{t_{tx}} \tag{3.21}$$

Où t_{tx} est le temps prévu pour la transmission d'une trame. Dans l'interface IEEE 802.15.4, t_{tx} est estimé par l'équation suivante :

$$t_{tx} = \left(\sum_{i=1}^{p*(MaxR-1)} t_{mTr} + t_{maxAckWait} \right) + t_{mTr} + (t_{maxAckWait} * p) \tag{3.22}$$

Où p est le taux d'erreur du lien entre le nœud émetteur et récepteur, $MaxR$ est le nombre de tentatives de transmission d'un paquet, t_{mTr} est le temps de transmission d'un paquet au niveau MAC et $t_{maxAckWait}$ est la durée maximale d'attente d'un acquittement. A partir de l'équation 3.22, on observe que t_{tx} dépend de p. Lorsque le taux p augmente, t_{tx} augmente et f_{tx} augmente. Le temps t_{mTr} est calculé par :

$$t_{mTr} = t_{CCA} + t_{phModulation} \tag{3.23}$$

Où t_{CCA} et $t_{phModulation}$ représentent respectivement les temps de d'estimation du canal libre et de modulation de la trame au niveau de la couche physique.

Initialement, chaque nœud capteur ajuste son taux de transmission à $f_{tx} = f_{rx}$ et transmet ses segments. A la réception d'un segment, le nœud récepteur estime son taux de réception f_{rx} et s'il constate que $f_{tx} \neq f_{rx}$, il exécute le mécanisme d'adaptation du taux de transmission afin d'éviter la congestion et/ou la sous exploitation de la bande passante. Ce mécanisme est discuté dans la prochaine section.

3.4.4 Mécanisme d'adaptation du taux de transmission

CARTEE assure la fiabilité en utilisant deux mécanismes principaux : retransmission basée sur des acquittements implicites/explicites et une adaptation distribuée du taux de transmission. Le premier mécanisme traite la perte de données et améliore la fiabilité et le deuxième pour éviter la congestion dans le souci d'économie d'énergie. D'une manière générale, Il faut souligner que le mécanisme d'adaptation du taux de transmission est un complément du mécanisme de transmission permettant de réduire le nombre de retransmissions. Puisque les transmissions consomment plus d'énergie, le mécanisme d'adaptation du taux de transmission permet de réduire la perte de données provoquée par une congestion.

Théorème 1. *Le contrôle de congestion réduit les retransmissions, mais à un moment donnée, l'un des deux contrôles (retransmission ou contrôle de congestion) est exécuté à la fois.*

Démonstration. Le taux de fiabilité entre le nœud émetteur et le nœud récepteur est estimé en utilisant l'équation 3.2, où $p = p_{le} + p_c$, Avec p_{le} et p_c sont respectivement les probabilités de la perte causée par un lien erroné ou une situation de congestion. A partir du lemme 2, p_c est estimé par l'équation suivante :

$$\begin{array}{lll} p_c = 0 & ; & f_{tx} > f_{rx} \\ p_c = 1 - \left(\frac{f_{tx}}{f_{rx}}\right) & ; & f_{tx} \leq f_{rx} \end{array} \tag{3.24}$$

La congestion est évitée en ajustant f_{rx} à une valeur égale à f_{tx}. Pour prouver ce théorème, on applique un contrôle commutant [54] au temps t approprié en utilisant deux contrôles u_1 et u_2 définis par :

$$\begin{array}{lll} u_1(t) = \sum_{i=0}^{MaxRetries} (1 - p_{le})^i & ; & p_c = 0 \\ u_1(t) = (1 - p_c)^n & ; & p_{le} = 0 \end{array} \tag{3.25}$$

Où $MaxRetries$ est le nombre maximum de tentatives de transmissions. A chaque instant t, $u_1 * u_2 = 0$. □

CARTEE déclenche le mécanisme d'adaptation du taux de transmission dans deux cas, le premier cas est la situation de congestion ($f_{tx} < f_{rx}$), et le deuxième cas permet d'éviter la sous-estimation de la bande passante ($f_{tx} > f_{rx}$). Dans les deux cas, le nœud ajuste le champ d'ajustement du taux de transmission de chaque segment à la valeur f_{tx}/n_{source} (où n_{source} est le nombre de nœuds sources présents dans la file d'attente de transmission au niveau du nœud émetteur). Chaque segment envoyé peut être écouté par le nœud précédant (le nœud qui a causé un excès ou un faible taux de réception). En fait, chaque nœud ayant écouté un segment incluant un champ d'ajustement du taux de transmission, doit réajuster son taux de transmission à la valeur incluse dans le segment et continuera sa transmission. De cette manière, tous les nœuds sources peuvent utiliser un taux de transmission équitable. Pour assurer un contrôle commutant, CARTEE active l'adaptation du taux de transmission et désactive le mécanisme d'acquittement dans l'intervalle de transmission (estimé par l'équation 3.16) et après cet intervalle, CARTEE inverse l'activation des contrôles.

3.5 Conclusion

Dans ce chapitre, une description détaillée d'un nouveau protocole de transport nommée CARTEE a été réalisé. Ce protocole a été conçu pour assurer un transport fiable de données dans les RCSF. Les défis qui ont motivés la conception de ce protocole ont été aussi présentés avec les insuffisances des solutions proposées dans la littérature. Parmi les insuffisances qui ont motivés à la conception du CARTEE est le champ d'application restreint caractérisant les solutions existantes que ce protocole tente de répondre aux exigences des applications multimédia dans les RCSF. Une étude théorique sur laquelle est basée la conception de CARTEE a été présentée. Les décisions qui ont été prises dans cette étude ont permis de choisir les quatre mécanismes

(transmission, acquittement, détection de congestion et adaptation du taux de transmission) utilisés par le protocole CARTEE. Ces mécanismes ont été décrits en utilisant des diagrammes et des équations mathématiques pour mieux expliquer leurs comportements.
En fait, toute cette étude exposée fournit n'est juste qu'un encre sur papier ne promettant une solution performante, parce que sa validation doit être concrétisée dans la réalité. Donc, il est nécessaire de valider le comportement du protocole CARTEE dans un système réel afin d'analyser ses performances. Dans le monde réel plusieurs méthodes d'analyse et de validation existent pour valider le comportement de CARTEE. La discussion de ces méthodes fera l'objet du prochain chapitre.

CHAPITRE 4

EVALUATION DES PERFORMANCES DANS LES RCSF

4.1 Introduction

Après avoir décrit dans le chapitre précédent, la solution proposée (CARTEE) qui théoriquement surpasse les performances des propositions existantes dans la littérature. Cependant, cette théorie manque de l'expérimentation qui permet d'établir un lien entre la théorie et le monde réel. En fait, l'expérimentation réelle permet de valider le comportement de chaque mécanisme du protocole CARTEE afin d'analyser et estimer ses performances dans un réseau de capteur sans fil réel (monde réel). La mise en œuvre de cette solution pour des objectifs de validation et d'évaluation de performances peut être réalisée en deux approches. La première approche consiste à effectuer cette évaluation dans des capteurs réels (expérimentation dans un système réel) et la deuxième, est la plus souvent utilisée, se focalise sur l'expérimentation à base de simulation.
Ce chapitre présente les approches utilisées pour évaluer les performances d'une solution, ainsi que les lacunes et les insuffisances de chaque approche. Vue l'importance des outils d'évaluation, ce chapitre dresse une comparaison entre ces outils afin de choisir l'outil approprié pour l'évaluation de performances.

4.2 Approches d'évaluation de performances

L'évaluation de performance est un ensemble d'hypothèses et de processus analytiques dont le but est d'estimer l'efficacité de quelques paramètres de performances. Cette évaluation est divisée en deux classes : la modélisation et les mesures directes. Dans la classe de modélisation (l'objet du chapitre précédent), l'évaluation de performance est effectuée lors de la conception d'une solution. Elle est utilisée pour obtenir des estimations lors de l'implémentation du système. La classe des mesures directe est la plus évidente pour évaluer les performances d'une solution concrétisée [55].
L'évaluation des performances d'un protocole dans les RCSF consiste à mesurer un ensemble de paramètres permettant d'estimer ces performances [56]. La solution proposée vise à améliorer un ensemble de paramètres tel que la fiabilité, la gestion de l'énergie, le délai de livraison et le taux d'occupation de la file d'attente. L'évaluation à base de modèles mathématique (modélisation) ne suffit pas pour valider une solution, d'où il est nécessaire de faire recours à des expérimentations réelles. Alors on doit extraire les paramètres permettant d'estimer ces facteurs de performances dans un environnement réel (évaluation directe).
Deux approches pouvant être exploités pour réaliser une évaluation directe pour une solution dans un RCSF. La première évaluation repose sur l'utilisation d'un RCSF réel existant pour mesurer les paramètres estimant les performances de la solution. La deuxième approche consiste à imiter le fonctionnement d'un RCSF en uti-

lisant un simulateur, ce qui nous permet de tracer et mesurer ces performances. L'évaluation des performances se diffère d'une approche à l'autre, et peut être un défi dans une telle approche qui a ses propres avantages et insuffisances.

4.2.1 Evaluation directe à base d'un système réel

Dans cette évaluation, la solution doit être implémentée directement dans des nœuds capteur réel et exécutée par la suite dans un RCSF réel. Pour mesurer les facteurs de performances, il est nécessaire de créer des fichiers journaux permettant de tracer les opérations de communications et les paramètres de chaque facteur de performances. A la fin d'exécution, ces fichiers journaux peuvent être analysés en utilisant des outils d'analyse telle que WireShark [57] et TCPDump [58].

Plateformes de nœud capteurs

L'avancement dans la technologie des systèmes de détection Micro-Electro Mécanique (MEMS) a conduit au développement des nœuds capteurs miniatures dotés d'une capacité de détection, de traitement et de communication sans fil. Un nœud capteur sans fil est composé d'un microcontrôleur, d'un émetteur/récepteur, d'une horloge, d'une mémoire et d'un convertisseur numérique analogique. Les nœuds capteurs sont déployés pour surveiller la multitude d'un phénomène naturel ou d'un événement généré par l'homme. Les ressources principales les plus souvent critiques sont l'énergie et la mémoire. Le microcontrôleur utilisé par un nœud capteur exécute des instructions à faible fréquence par rapport à un processeur d'un ordinateur [59]. Plusieurs plateformes de nœuds capteurs ont été développées dont chacune repose sur une architecture spécifique. Les prochains points présentent quelques plateformes les plus utilisées dans les RCSF.

MICA Ce détecteur permet de développer des applications pour des RCSF. Il est conçu spécialement pour optimiser la consommation énergétique dans le réseau. Il est basé sur le système d'exploitation TinyOS et fournit une communication réseau ad hoc maillé en permettant la reprogrammation des nœuds de détection. Ce détecteur peut être utilisé dans des applications destinées pour la surveillance à l'intérieur d'un bâtiment, tel que la détection acoustique, vidéo, vibration, . . . etc. Inclus une interface réseau IEEE 802.15.4 de 2.4 Ghz.

TELOS C'est une plateforme à source libre conçue pour faire des expérimentations destinées pour les communautés de recherche développée par l'université de Berkeley. Donc, c'est une plateforme à faible consommation d'énergie et inclus une interface réseau IEEE 802.15.4.

AVR Le détecteur Atmel AVR est un microcontrôleur 8-bits basé sur l'architecture RISC améliorée. Il permet d'exécuter des instructions en puissance d'un seul cycle d'horloge et atteint une vitesse de 1 MIPS par MHz. Ce microcontrôleur optimise les consommations énergétiques par rapport à la vitesse du traitement.

FireFly Ce détecteur a été conçu pour des plateformes hardware à faible consommation énergétique. Il utilise un microcontrôleur Atmel Atmega32L de 8 bits et une interface réseau sans fil IEEE 802.15.4. Le microcontrôleur fonctionne à une vitesse de 8 MHz, une taille de la RAM de 2 Ko et celle de la ROM est de 32 Ko. Il inclue des détecteurs de lumière, de température, audio, accélération duel-axes et un détecteur de mouvement infrarouge. Le détecteur FireFly peut communiquer avec un PC en utilisant un USB externe.

Systèmes d'exploitations pour les nœuds capteur

Un nœud capteur doit inclure un système d'exploitation qui fournit un fonctionnement simple et une gestion efficace des ressources du nœud. Les ressources typiques dans un nœud capteur sont le processeur, la mémoire, l'horloge et l'interface réseau. Le système d'exploitation utilise le multiplexage de ressources par deux manières : temps et espace. Le multiplexage temporel implique différents programmes qui utilise les ressources à tour de rôle mais le multiplexage spatial implique plusieurs programmes qui accèdent à une partie de la ressource au même temps. En considérant les contraintes de ressources des nœuds capteur dans un RCSF, une nouvelle approche est nécessaire pour concevoir des systèmes d'exploitation pour les RCSF [59].
Plusieurs systèmes d'exploitation ont été conçus pour les nœuds capteur sans fil où chacun repose sur architecture et un paradigme de programmation différente. Les prochains points citent quelques systèmes d'exploitation utilisés dans les RCSF.

TinyOS [60] C'est un système d'exploitation source libre destiné pour les nœuds capteurs. Ce système est caractérisé par sa flexibilité, sa conception est à base de composants et désigné pour des applications spécifiques. TinyOS peut supporter des programmes concurrents avec des exigences mémoire plus faible, où le système d'exploitation occupe uniquement 400 octets d'espace mémoire. La bibliothèque des composants du TinyOS inclue la pile protocolaire du réseau, les services distribués, les pilotes des détecteurs et les outils d'acquisition de données. Au niveau de la couche MAC, TinyOS implémente le protocole TDMA, TDMA/CSMA, Z-MAC, B-MAC et la couche MAC IEEE 802.15.4 en option.

Langage utilisé peut exécutés des applications développées dans le langage de programmation NesC qui est le dialecte du langage C.

Plateformes supportées peut être installé sur les plateformes : Mica, Mica2, Micaz, Telos, Tmote, . . . etc.

Contiki [61] C'est un système d'exploitation source libre écrit en C, conçu pour les nœuds capteurs. Il est caractérisé par sa portabilité incluant un noyau orienté événement. Contiki fournit le multitâche préemptive utilisé au niveau d'un processus et sa configuration occupe 2 Ko de RAM et 40 Ko de ROM. L'installation du Contiki fournit les fonctionnalités de multitâche, de multithreading préemptive, de proto-threads, la pile protocolaire TCP/IP, l'IPv6, une interface utilisateur graphique, un navigateur web, un serveur web, un client telnet et un réseau virtuel. Contiki fournit aussi l'extension 6LowPAN au niveau de la couche MAC.

Langage utilisé peut exécute des applications développées dans le langage C.

Plateformes supportées supporte les plateformes : Tmote, AVR série MCU.

MANTIS [62] C'est l'abréviation de "MultimodAl system for NeTworks for In-situ wireless Sensors" qui représente un système d'exploitation écrit en C destiné pour les nœuds capteurs fournissant un nouveau mécanisme multithreading pour les RCSF. MANTIS est un système d'exploitation de poids faible permettant d'économiser l'énergie occupant 500 octets d'espace incluant le noyau, l'ordonnanceur et les tâches du réseau. La portabilité est la fonctionnalité la plus importante apportée par ce système d'exploitation, tel qu'on peut tester des applications MANTIS dans des ordinateurs de bureau ou PDA, et par la suite on les charge dans un nœud capteur. MANTIS permet la gestion à distance du nœud capteur à travers la programmation dynamique.

Langage utilisé supporte des applications écrites en langage C.

Plateformes supportées peut être exécuté sur les platformes : Mica2, Micaz et Telos.

Nano-RK [63] C'est un système d'exploitation développé pour les nœuds capteur assurant un multitâche préemptive en temps réel. Les objectifs de conception du Nano-RK sont le multitâche, le routage multi-saut, ordonnancement à base de priorités, la rapidité et planification, la prolongation de la durée de vie du réseau, l'utilisation limitée des ressources de l'application avec l'utilisation d'un système d'exploitation occupant une petite taille de stockage. Ce système utilise 2Ko de RAM et 18Ko de ROM. Nano-RK supporte des applications temps réel (software et hardware) en utilisant différents algorithmes d'ordonnancement temps réel, tel que, le l'ordonnancement monotone et l'ordonnancement à temps harmonisé.

Langage utilisé exécute des applications écrites en langage C.

Plateformes supportées peut être installé sur la plateforme MicaZ et FireFly.

LiteOS [64] C'est un exemple du système d'exploitation Unix conçu pour les nœuds capteurs à l'université Illinois. L'objectif de conception est d'offrir un système d'exploitation pour les RCSF similaire à l'Unix. Ce système fournit une programmation système familière au paradigme de programmation (programmation à base de thread, bien qu'il fournit un support d'enregistrement des gestionnaires des événements en utilisant le callback). LiteOs fournit également un système de fichier hiérarchique, un support de programmation orienté objet (sous forme de LiteC++) et un noyau similaire à Unix. Il peut s'exécuter dans un MicaZ incluant un processeur 8 Mhz, 128 octets de mémoire flash et 4 Ko de RAM. Ce système se compose de trois composants : LiteShell, LiteFS et le noyau.

Langage utilisé exécute des applications développées dans le langage LiteC++.

Plateformes supportées supporte les plateformes MicaZ et AVR série MCU.

Avantages et inconvénients

L'avantage de cette évaluation c'est qu'elle permet d'effectuer des expérimentations en utilisant le système réel ce qui garantit la crédibilité de l'évaluation. Malgré ces avantages, en termes de crédibilité de résultats, cette évaluation devient plus coûteuse lorsque la densité du réseau est élevée. Aussi l'utilisation d'une horloge distribuée (i.e. chaque nœud possède son propre horloge) influe sur l'ordonnancement des événements.

4.2.2 Evaluation directe à base de simulation

Dans ce type d'évaluation on va imiter le fonctionnement d'un RCSF en utilisant la simulation par ordinateur. Il existe plusieurs variétés de simulateur réseau dans la littérature qui se distingue par leurs domaines d'application, leurs précisions et les modèles implémentés. Afin de tirer profit de ces simulateurs, ns-3 [12] est utilisé pour évaluer les performances du protocole CARTEE. Pour effectuer des simulations et analyser les résultats obtenus, CARTEE doit être implémenté dans ce simulateur. Avant d'exécuter les résultats, il est nécessaire de choisir un ensemble de configuration réseau (i.e., nombre de nœud, topologie, taille des paquets, ... etc.) permettant d'exécuter la solution sous différentes conditions (i.e., réseau dense, donnée de taille importante, plusieurs sources de données, ... etc.).

Dans cet évaluation, le choix du simulateur est crucial parce qu'il affecte la crédibilité de l'évaluation. Les sections qui suivent présentent quelques simulateurs de RCSF les plus utilisés dans la littérature.

TOSSIM

TOSSIM [65] est un simulateur de réseaux de capteurs sans fil utilisant le système d'exploitation TinyOS. Il est basé sur la simulation à événements discrets. Ce simulateur compile tous les modèles réseau sous le Framework Tiny OS pour des milliers de nœuds capteurs. TOSSIM a modifié quelques niveaux bas du système d'exploitation TinyOS et le hardware d'un nœud capteur. Ces modification ont permet d'imiter fidèlement le comportement du nœud capteur ce qui a élargi le champ d'expérimentation sous TOSSIM. L'exécution orienté évènement du TinyOS s'adapte soigneusement à la simulation à événements discrets ce qui nécessite des modifications légère au niveau du noyau TOSSIM. La compilation entière du TinyOS peut être utilisée pour inclure plusieurs applications dans le simulateur. En exécutant le système d'exploitation TinyOS sur un ordinateur de bureau, TOSSIM facilite le passage entre l'exécution dans un RCSF réel et l'exécution d'une simulation. TOSSIM fournit des interfaces graphiques permettant de visualiser en détaille l'exécution de la simulation.

Architecture du simulateur L'architecture du simulateur est constituée de cinq composants : un support de compilation des composants (Framework TOSSIM), une file d'attente des événements discrets, un ensemble de composants hardware TinyOS ré-implémentés, un mécanisme du modèle d'extension pour la radio fréquence ainsi que le convertisseur ADC et un service de communication pour les programmes externes y compris le simulateur. La figure 4.1 schématise l'architecture des composants TOSSIM. Ce simulateur profite des avantages de la structure TinyOS et la compilation du système entier pour générer des simulations à événements discrets directement à partir du graphe des composants. Il exécute le même code qui s'exécute dans un nœud capteur. En remplaçant quelques niveaux faibles des composants, TOSSIM traduit les interruptions hardware à des événements discrets du simulateur. La file d'attente des événements associée au simulateur délivre les interruptions pour gérer l'exécution d'une application TinyOS. L'avantage du simulateur TOSSIM c'est qu'il garde la majorité du code TinyOS sans changement.

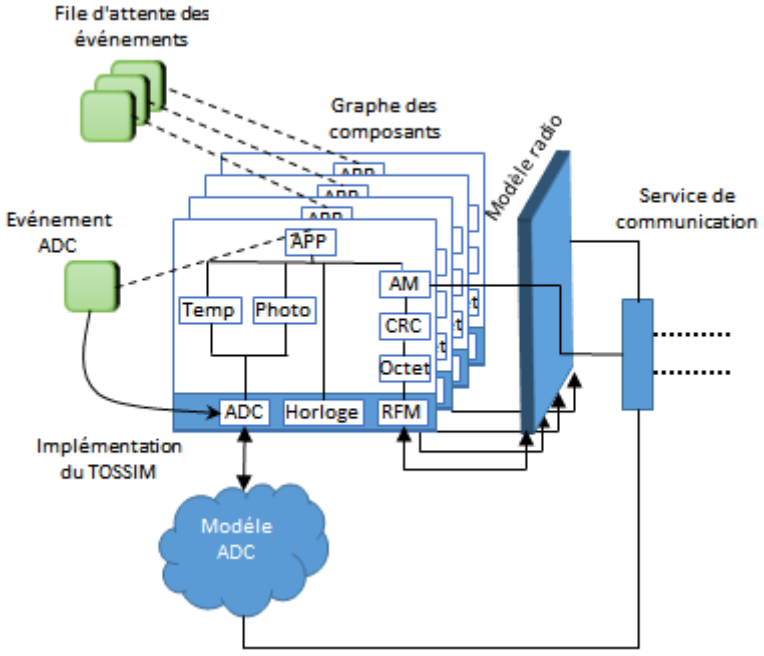

FIGURE 4.1 : Architecture du simulateur TOSSIM

Support de compilation TOSSIM modifie le compilateur NesC pour permettre la compilation à partir du graphe des composants TinyOS vers le Framwork du simulateur. En ajoutant ces modifications, une application peut être compilée pour le simulateur TOSSIM ou bien pour un nœud capteur.

Modèles de communication TOSSIM fournit des mécanismes aux développeurs TinyOS pour choisir la précision nécessaire du modèle radio pour leurs simulations. Il est constitué d'un graphe orienté de probabilités pour les erreurs de bit. Chaque arc (u,v) dans le graphe représente une probabilité d'erreur lorsque le nœud u envoie des paquets vers le nœud v. Pour simuler le comportement des liens asymétriques, la probabilité de l'arc (v,u) doit être différente de celle de l'arc (u,v). Les probabilités du graphe peuvent changer au moment de la spécification de la configuration du réseau ou bien lors de l'exécution de la simulation. Lors de la transmission d'un paquet, un évènement de transmission se propage vers le canal d'entrée de chaque nœud connecté au réseau.

Outils de visualisation et d'analyse Il offre l'outil TinyViz pour visualiser, contrôler et analyser les résultats de simulation.

COOJA

Le simulateur COOJA [66] est conçu pour les réseaux de capteurs sans fil qui fonctionnent avec le système d'exploitation Contiki. Il permet simultanément la simulation au niveau réseau, au niveau système d'exploitation et au niveau des jeux d'instructions de la machine. COOJA simule des réseaux constitués de différents types de nœuds capteurs (i.e., différents softwares et différents hardwares). Le simulateur est flexible où plusieurs parties du simulateur peuvent être facilement remplacées ou étendues avec des fonctions supplémentaires. Dans COOJA, un nœud possède trois propriétés : ses données mémoire, son type et ses périphériques hardwares. Le type du nœud peut être partagé entre plusieurs nœuds et peut aussi déterminer les propriétés communes entre ces nœuds. COOJA est capable d'exécuter des programmes Contiki de deux manières : soit en exécutant le programme tel qu'il est compilé dans un CPU, ou bien, en exécutant le programme dans un émulateur TI MSP430. COOJA peut également simuler des nœuds non-Contiki, tel que des nœuds implémentés dans Java ou même des nœuds qui exécutent d'autres systèmes d'exploitation.

Architecture du simulateur COOJA est constitué d'un ensemble de classes écrites en java, alors on peut changer les parties du simulateur en modifiant le code de ces classes. Les classes java qui interagissent avec les extensions (i.e. modèles ajoutés) sont brièvement décrites comme suite :

- Radio : permette de simuler le comportement de l'émetteur/récepteur radio.
- Mote : c'est l'abstraction d'un nœud capteur.
- Positioner : cette classe est utilisée pour déterminé la position du nœud capteur.
- RadioConnection : elle représente une connexion radio entre une source radio unique et n'importe quelle destination radio en communication.
- ConvertedRadioPacket : c'est la radio de transmission/réception la plus utilisée dans COOJA.
- Simulation : Cette classe gère l'exécution de chaque simulation.
- RadioMedium : c'est une classe abstraite où chaque implémentation d'une radio sous COOJA hérite de cette classe.

- AbstractRadioMedium : cette classe fournit les fonctions permettant d'implémenter une radio.
- UDGM : une radio abstraite permettant de transformer un rayon transmis sous forme d'un cercle.
- DirectedGraphMedium : La radio est implémentée à travers des structures en arcs.
- PacketAnalyzer : Cette classe permet d'analyser chaque paquet envoyé via la radio.
- IEEE802154Analyzer : Cette classe permet d'analyser des trames de la couche MAC IEEE 802.15.4.
- IPHCPacketAnalyzer : cette classe scanne chaque datagramme (paquet IPv6) envoyé à partir du client vers la racine en utilisant le protocole 6LowPAN
- ICMPv6Analyzer : Cette classe permet d'analyser le trafic au niveau routage.

Support de compilation Le simulateur est développé en Java pour le rendre facile aux utilisateurs, mais il permet d'écrire des softwares de nœuds capteur en C en utilisant l'interface Java Native. En plus, une application de nœud capteur peut être exécutée dans le simulateur COOJA, où elle peut être exécutée aussi dans un émulateur du nœud capteur.

Modèles de communication Chaque simulation dans COOJA utilise un modèle radio qui caractérise la propagation des ondes radio. On peut même aussi ajouter des modèles radio à l'environnement de la simulation, et le modèle radio peut être choisi à la création de la simulation. Ceci permet de simuler une solution sous différents modèles de communications. Généralement, un modèle radio fournit des extensions pour configurer et visualiser les conditions réseau simulées.

Outils de visualisation et d'analyse Il Fournit des extensions COOJA TimeLine permettant de visualiser, déboguer et analyser la simulation au moment de son exécution.

ns-3

Le simulateur réseau ns-3 [12] est un simulateur à événements discrets dans lequel le noyau et les modèles de simulation sont écrit en C++. Il est compilé en tant que bibliothèque qui peut être lié statiquement ou dynamiquement à un programme principale C++ qui définit la topologie du réseau et exécute la simulation. Presque tous les API du simulateur sont exportées au langage Python d'où il est possible d'importer un module ns-3 au langage Python de la même manière que le langage C++. Parmi les objectifs de conception du ns-3 c'est d'améliorer le réalisme des modèles de communication pour se rapprocher des implémentations logicielles réelles. Le simulateur ns-3 à inclus récemment un émulateur permettant d'exécuter des simulations dans des applications et périphériques réels. En fait, ns-3 n'est pas un nouveau simulateur, mais c'est une synthèse de plusieurs outils prédécesseurs, en incluant ns-2 lui-même ce qui à simplifier la réutilisation des modèles et des outils de ces prédécesseurs. Un autre avantage est le débogage simple offert par ns-3 qui le distingue de ces prédécesseurs [11].

Architecture du simulateur Le code source du simulateur est dans le dossier « src » de la collection 3.21. Le noyau du simulateur est implémenté dans le dossier « src/core » et les paquets sont les objets fondamentaux dans un simulateur réseau. L'implémentation des paquets est incluse dans le dossier « src/network ». Ces deux modules de simulation constituent un noyau générique de simulation qui peut être utilisé dans différents types de réseau et non seulement le réseau internet. Les modules constituant le noyau du simulateur ns-3 sont

indépendants du réseau et des modèles de périphériques. La figure 4.2 représente l'organisation des différents modules du simulateur ns-3.

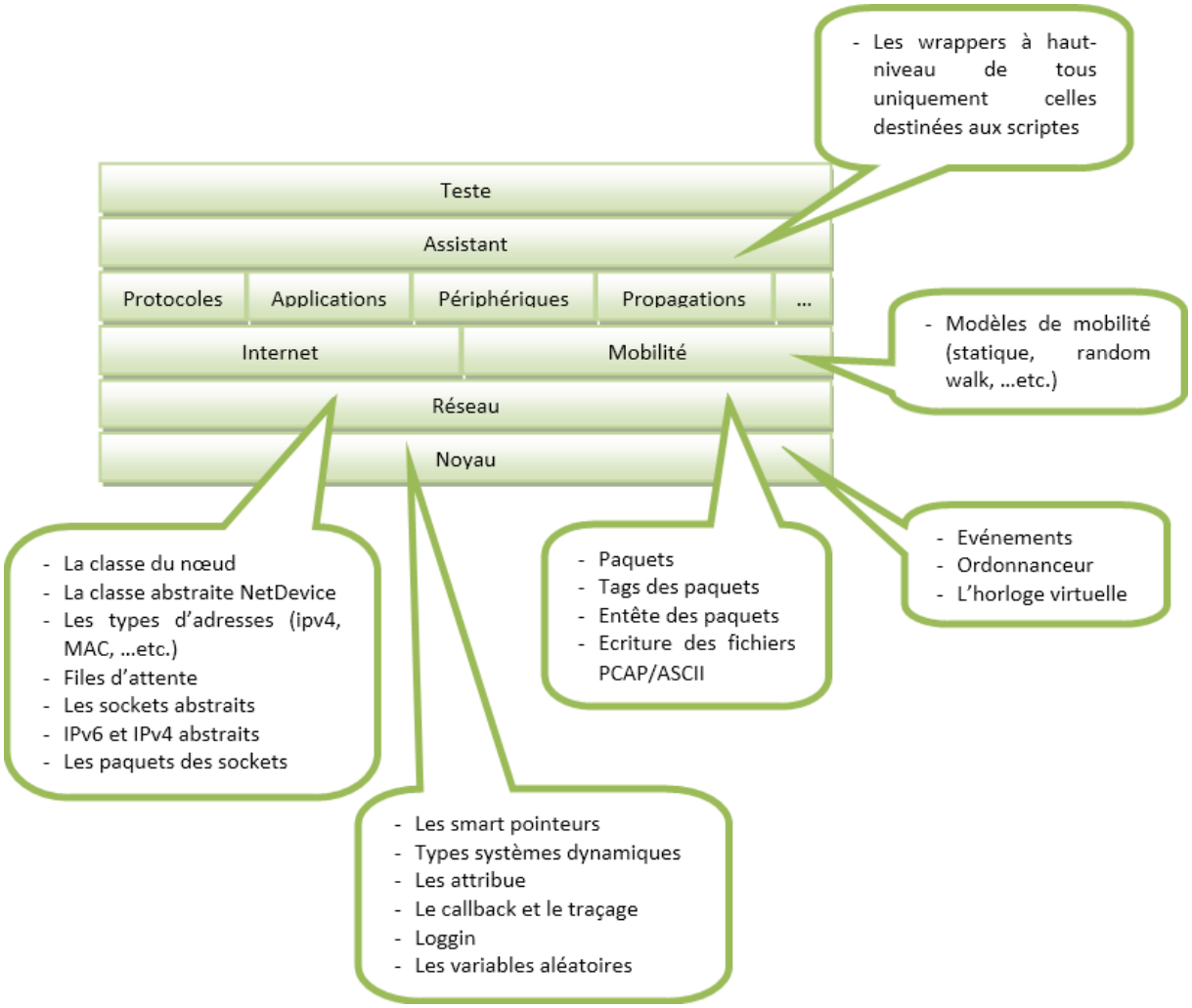

FIGURE 4.2 : Architecture du simulateur ns-3

Support de compilation Pour éviter la complexité de son prédécesseur ns-2 (l'utilisation de deux langages TCL et C++), ns-3 utilise uniquement le langage C++ pour décrire les modèles ainsi que la configuration du réseau permettant un débogage simple. Le simulateur ns-3 offre aussi des scripts à base d'API Python pour permettre d'intégrer ns-3 avec des environnements ou des modèles de programmation à base de Python.

Modèles de communication ns-3 utilise un module LTE qui fournit une implémentation basique des périphériques LTE, incluant un modèle de propagation et les couches MAC et physique. Les fonctionnalités importantes offertes par ce module sont une implémentation basique des équipements utilisateurs (EU) avec les périphériques du nœud (PN), un contrôle de ressources radio pour l'EU et le PN et une gestion de l'indicateur de la qualité du canal (CQI). Le modèle de la perte de propagation proposé est constitué de quatre composants : la perte du chemin, l'atténuation du signal, la fluctuation du signal et la pénétration de la perte sont implémentés respectivement dans les classes PathLossModel, JakesFadingLossModel, ShadowingLossModel et PenetrationLossModel. Toutes ces classes héritent de la classe DiscreteTimeLossModel qui fournit des fonctions et des variables pour les modèles de la perte de propagation [67]. Le simulateur ns-3 implémente aussi l'interface IEEE 802.15.4 qui permet de simuler un réseau de capteurs.

Outils de visualisation et d'analyse ns-3 inclus un composant de traçabilité plus performant que son prédécesseur ns-2. Ce composant est construit sous le concept de traçabilité indépendante pouvant être effectué au niveau source et au niveau puits.

Avantages et inconvénients

Le principal avantage de cette évaluation c'est qu'elle peut être effectuée sans créer un RCSF réel, donc le coût de l'évaluation est considérablement réduit. En plus, l'évaluation à base de simulation ne pose pas de problème de l'horloge distribué. Cependant, la crédibilité de l'évaluation est liée à la validité des modèles utilisés par le simulateur choisi.

4.3 Outils d'évaluation de performances utilisés dans CARTEE

Vue le coût élevé de l'évaluation d'un système réel, la solution CARTEE a été évaluée en conduisant des expérimentations à base de simulation. En plus, l'observation permet de préciser des événements fournis par la simulation à la microseconde prés ce qui a motivé le choix de l'approche à base de simulation, parce que le système réel manque de précision en termes d'observation des événements au moment de leurs apparitions. Cependant, comme il est mentionné dans les sections précédentes, le choix d'un simulateur est crucial dans cette évaluation. Parce que les modèles utilisés par le simulateur affectent la crédibilité de l'évaluation, surtout lorsque le simulateur repose sur des modèles de communication statistiques. Certains développeurs préfèrent évaluer un protocole sur plusieurs simulateurs. Cette manière d'évaluation semble plus crédible, mais en réalité elle permet de créer un doute dans la crédibilité des résultats lors de l'observation des écarts entre les résultats obtenus pour chaque évaluation.
La multitude de simulateurs réseau dans la littérature à rendue le choix du simulateur approprié un véritable défis pour réaliser des simulations [68]. L'identification et la structuration des critères de sélection se concentrent sur des décisions multi-attributs. La structuration des critères la plus utilisée est développée d'une manière hiérarchique en commençant par des objectifs généraux, qui sont raffinés par la suite à des sous-objectifs plus spécifiques. La tâche d'évaluation et de sélection d'un simulateur, qui est une décision multicritères, consomme plus de temps. Donc, il est nécessaire d'avoir des connaissances détaillées sur les fonctionnalités du simulateur pour ne pas perdre du temps dans cette décision de choix [69].

4.3.1 Classification des critères

Dans [69], les auteurs ont classifié les critères de sélection d'un simulateur en utilisant une structure hiérarchique. Le logiciel de simulation, le fournisseur du logiciel et l'utilisateur sont les éléments du haut niveau de la hiérarchie. Dans cette structuration, le logiciel de simulation couvre une large portée dans le défi de sélection du simulateur approprié. Les auteurs ont défini les sous-critères indiqués dans la figure 4.3.

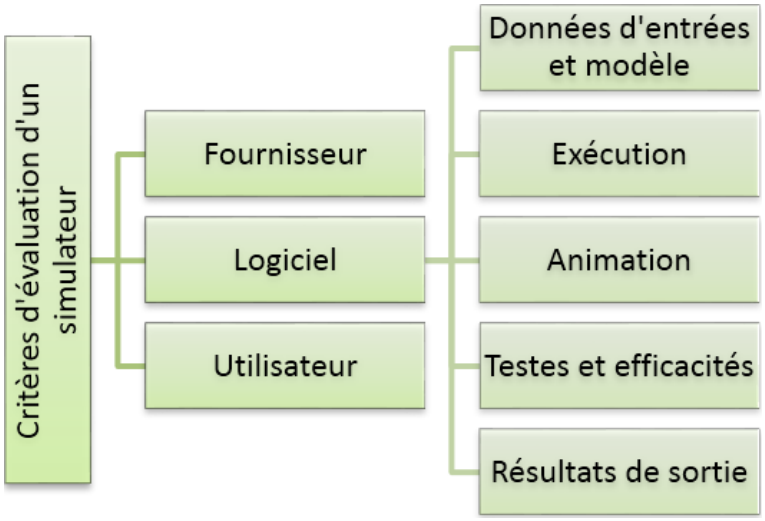

FIGURE 4.3 : Les principaux critères de la structuraion

Fournisseur de simulateur

Ce critère évalue la crédibilité du fournisseur du produit finale de simulation en évaluant les points suivants :

- Pédigrée : dans cette évaluation on s'intéresse à l'historique du simulateur et son fournisseur.
- Documentations : une bonne documentation permet à l'utilisateur de ne pas être dépendant du fournisseur pour des petits problèmes. La disponibilité d'un manuel avec des indexes, des références et des tutoriels qui assistent les utilisateurs sont des facteurs estimant la documentation.
- Support : un fournisseur sans support peut ne pas être une source de confiance. Tel que la disponibilité des cours d'entrainements permettant à l'utilisateur de découvrir les fonctionnalités du simulateur. La maintenance et les mises à jour de passage vers de nouvelles versions sont des facteurs qui mesurent le support du logiciel.
- Préachat : la disponibilité des démonstrations et des assistants avant l'achat du simulateur.

Spécification des données et du modèle

Avec ce critère on s'intéresse aux questions liées au développement du modèle et la spécification des données d'entrée. Donc on doit répondre aux questions suivantes :

- Construction du modèle : dans ce sous-critère on considère les facilités qui aident l'utilisateur pour développer un modèle.
- Données d'entrées : ici on s'intéresse comment spécifier les données d'entrées.
- Les distributions statistiques : ce sous-critère évalue les fonctions fournies pour la génération des variables aléatoires (i.e., les lois normale, exponentielle, gamma et les distributions rectangulaires).
- Les aspects du code source : Avec ces aspects on évalue la qualité du simulateur pour supporter de nouvelles extensions.

Exécution

Ce critère répond aux questions des expérimentations effectuées dans le simulateur. Les sous-critères suivants sont évalués :

- La possibilité d'effectuer plusieurs exécutions.
- La possibilité d'atteindre l'état stationnaire de la simulation.
- L'exécution de la simulation à partir d'un état non-vide (ou non-initiale).
- Le contrôle de la vitesse de simulation.

Animation

Avec ce critère, on examine la création, l'exécution et la qualité de l'animation. Donc, on s'intéresse à l'évaluation des sous-critères suivants :

- Images : le nombre et la qualité des images sont importants dans l'animation.
- Disposition de l'affichage : ce critère examine les questions liées aux présentations graphiques qui apparaissent dans l'écran.
- Développement : c'est la même question discutée dans le développement du modèle.
- Exécution : on mesure l'impact de l'exécution de l'animation sur l'exécution du modèle.

Testes et efficacité

Ce critère est utilisé pour évaluer les tests, la puissance de débogage et l'efficacité du simulateur. Donc on répond aux questions liés à ces sous critères :

- Vérification et validation : Plusieurs éléments peuvent être fournit pour cette question (i.e., l'aide, les messages d'erreurs, et les tutoriels en lignes).
- Possibilité de faire un retour en arrière du temps (qui n'est pas offert par la majorité des simulateurs).
- Les générateurs des modèles conceptuelles : estimation de la capacité du simulateur pour générer une représentation graphique du modèle.
- Limitations : estimation des éléments imposant des limites à l'utilisateur (i.e., la taille du modèle, nombre d'icônes à afficher, . . . etc.).
- Les fonctionnalités d'affichage disponible dans le simulateur.
- Traçage : les fichiers traces contient des données collectés à partir des changements d'état du modèle durant son exécution.

Les résultats en sortie

Ce critère couvre quelques questions importantes, tel que :

- Rapport : généralement les simulateurs produisent quelques rapport standards (tel que délai de livraison de données, le nombre de paquets perdus, . . . etc.). Il est important que le simulateur offre à l'utilisateur des possibilités pour personnaliser ces rapports.
- Livraison : c'est la capacité du simulateur pour envoyer les résultats de sorties vers des fichiers, ainsi d'offrir un accès vers ces fichiers.
- La possibilité d'exporter les résultats vers d'autres formats de données (i.e., graphes, Excel, . . . etc.).
- Graphes : c'est la possibilité de présenter les résultats sous forme de graphes personnalisables par l'utilisateur.
- Analyse : l'analyse de la sortie est une question importante. Avec ce sous-critère on estime les analyses statistiques fournit par le simulateur (i.e., variance, intervalle de confiance, . . . etc.).

L'utilisateur

Ce critère examine les besoins des utilisateurs du simulateur, tel que :

- D'offrir la possibilité à l'utilisateur de choisir la simulation à événements discrets, continue ou les deux en même temps.
- D'estimer de la portabilité du modèle pour être exécuté sous plusieurs plateformes.
- Expérience exigée : c'est l'expérience nécessaire pour utiliser le simulateur.
- Aspect financier : c'est un sous-critère important permettant d'évaluer le coût du simulateur (i.e., le prix du simulateur, le coût de l'installation, le coût des exigences matérielles et le coût de la maintenance).
- La classe du logiciel : il existe trois types de logiciel de simulation : un langage général d'ordinateur (ex., Matlab, Scilab, . . .), un langage de simulation (ex., GPSS) ou un simulateur (ex., TOSSIM, COOJA, ns-3, . . .).

4.3.2 Simulateur utilisé dans l'évaluation

Le simulateur ns-3 a été choisi pour évaluer les performances du protocole CARTEE en se basant sur les critères cités dans la section précédentes. Le tableau 4.1 présente une évaluation des critères sur laquelle est basée la décision de choix du simulateur ns-3.

		TOSSIM	COOJA	ns-3
Fournisseur du logiciel de simulation	Pedigree	TOSSIM 1.0 : Créer en 2003 par Chad Metcalf de l'école des mines de Colorado.	Créer en 2003 par les développeurs du système d'exploitation Contiki	Network Simulator (ns) est le nom d'une série de simulateurs ns-1, ns-2, ns-3 et ns-4. La première version ns-1 a été développée par le laboratoire Lawrence Berkeley (LBNL) en 1995.
	Documentation	Disponible pour TinyOS et NesC	Disponible pour Contiki et le guide d'utilisation du simulateur	Le projet ns-3 fournit un manuel d'utilisation du simulateur, des tutoriaux et des références de L'API du simulateur dans le site officiel.
	Support	Des forums et des wikis disponibles pour répondre aux questions des développeurs.	Des forums et des wikis disponibles pour répondre aux questions des développeurs.	Le projet GSoC de Google a permet aux développeurs des possibilités de formations en ligne avec des experts de la simulation sous ns-3, ainsi que des wiki pour répondre aux questions liées à l'utilisation.
	Préachat	Gratuit avec certains exemples d'utilisation.	Gratuit avec certains exemples d'utilisation.	Gratuit avec une série d'exemples de modèles et de configurations réseau.
Spécific. des données et du modèle	Construction du modèle	Ecrit dans le langage NesC	Ecrit dans le langage C ou Java	Ecrit dans le langage C++. Offre des outils pour générer la structure de modèle
	Données d'entrée	La configuration du réseau est écrite en NesC	Offre une interface graphique pour spécifier la configuration du réseau	La configuration du réseau est construite à base d'un programme écrit en C++
	Distributions statistiques	Plusieurs	Plusieurs	Plusieurs (très riche)
	Aspects du code source	Code source libre et gratuit	Code source libre et gratuit	Code source libre et gratuit
Exécution	Plusieurs exécutions	Possible	Possible	Possible
	Les périodes stationnaires	Les modèles de communications utilisés influent sur le simulateur pour atteindre l'état stationnaire	Les modèles de communications utilisés influent sur le simulateur pour atteindre l'état stationnaire	Atteint facilement les périodes stationnaires grâce aux modèles de communications réalistes (LTE)
	Démarrage à l'état non initial	Non disponible	Non disponible	Disponible
	Contrôle de la vitesse de simulation	Compliqué	Plus compliqué	Offre des paramètres permettant d'ajuster l'intervalle d'exécution de la simulation

		TOSSIM	COOJA	ns-3
Animation	Icônes	Non disponible	Disponible	Disponible
	Disposition de l'affichage	Langage de commande	Interface graphique conviviale	Langage Python
	Développement	Faible	Faible	Faible
	Exécution	Exécution rapide, et faiblement dépendante du modèle	Exécution moins rapide et totalement indépendante du modèle	Exécution très rapide et indépendante du modèle
Tests et efficacités	Vérification et validation	Outils Indisponibles	Outils Indisponibles	Outils Disponibles
	Retour en arrière	Impossible	Impossible	Possible
	Génération du modèle conceptuel	Non disponible	Non disponible	Possible
	Limites	Evolutif	Evolutif	Evolutif
	Fonctionnalités d'affichage	Offre des outils de visualisation de la simulation	Offre des outils de visualisation de la simulation	Offre des outils de visualisation de la simulation
	Traçage	Traçabilité des événements qui occurrent dans le réseau.	Traçabilité des événements qui occurrent dans le réseau.	Traçabilité des activités et des événements qui occurrent dans le réseau.
Résultats de sortie	Rapports	Disponible	Disponible	Disponible
	Livraison des résultats	Possibilité d'exportation et de visualisation en plusieurs formats	Possibilité d'exportation et de visualisation en plusieurs formats	Possibilité d'exportation et de visualisation en plusieurs formats
	Intégration	Extensible	Extensible	Extensible
	Graphes	Conviviale	Conviviale	Conviviale
	Analyse	Disponible avec certaines limites.	Disponible avec certaines limites.	Très disponible surtout avec l'analyse en utilisant les fichiers trace PCAP.
Utilisateur	Type de simulation	Simulation à événements discrets	Simulation hybride	Simulation à événements discrets
	Système d'exploitation	Unix	Unix ou Windows	Unix
	Expérience exigé	– Maitrise du langage NesC. – Maitrise du système d'exploitation TinyOS. – Maitrise de la simulation à événements discrets.	– Maitrise du langage C. – Maitrise du langage Java. – Maitrise du système d'exploitation COOJA. – Maitrise de la simulation à événements discrets et continue.	– Maitrise du langage C++. – Maitrise de la simulation à événements discrets.
	Aspect financière	Pas d'exigence matériels ou logiciels.	Pas d'exigence matériels ou logiciels.	Pas d'exigence matériels ou logiciels.
	Classe du logiciel	Simulateur	Simulateur	Simulateur

TABLE 4.1 : Evaluation des critères de selection d'un simulateur

4.4 Implémentation du protocole CARTEE sous ns-3

Comme il est mentionné dans le tableau d'évaluation (présenté dans la section précédente), ns-3 offre plusieurs outils permettant d'assister à la création et la structuration, ainsi que la génération des modèles de simulation. Parmi ces outils, ns-3 fournit le programme Python « create-module.py » permettant de créer la structure d'un nouveau modèle avec lequel a été créée la structure du modèle CARTEE. L'implémentation du protocole CARTEE sous ns-3 est réalisée avec quatre classes : `CARTEEL4Protocol`, `CARTEESocket`, `CARTEESourcesList` et `CARTEESourceEntry`. Au niveau de chaque nœud du réseau une instance de la classe CARTEEL4Protocol est initialement créée. Cette instance est toujours existante tout au long de la durée de vie du nœud. Cette classe permet de gérer le cache des segments reçus au niveau d'un nœud et la file d'attente de transmission. Elle implémente les quatre principaux mécanismes de CARTEE : de transmission de la fenêtre glissante, d'acquittement, de détection de congestion et d'adaptation du taux de transmission. Lors de l'initiation d'une communication, chacun des nœuds source et puits instancie un objet de la classe `CARTEESocket` et un autre de la classe `CARTEESourcesList`. La classe `CARTEESourcesList` est exploitée uniquement au niveau puits.

4.4.1 Diagramme de classe

La figure 4.4 décrit le diagramme de classe du modèle CARTEE. Il représente les classes, les attributs et les méthodes implémentés dans le simulateur ns-3.

La classe CARTEEL4Protocol

Cette classe implémente les méthodes des principaux mécanismes du protocole CARTEE.

Attributs

- `transmissionQueue` : représente la file d'attente de transmission.
- `estimatedTransmissionRate` : représente le taux de transmission estimé au niveau de la couche MAC (utiliser pour la détection de la sous-estimation de la bande passante).
- `estimatedReceptionRate` : représente le taux de réception estimé au niveau de la couche MAC (utiliser pour la détection de la congestion).
- `transmissionFrequency` : c'est le taux de transmission au niveau d'un nœud source ou intermédiaire.

Méthodes

- `Send` : permet d'envoyer des segments vers leurs destinations. En fait, cette méthode dépose seulement les segments au niveau de la file d'attente pour transmission ultérieure. Elle est invoquée par un objet `CARTEESocket`.
- `Receive` : déclenchée lors de la réception d'un segment à partir de la couche réseau.
- `SendFragments` : gère la transmission des segments inclus dans la file d'attente de transmission. Après transmission de sept segments, elle invoque automatiquement la méthode Overhears.

FIGURE 4.4 : Diagramme de classe du modèle CARTEE implémenté sous ns-3

- `Overhears` : lance l'écoute du canal au niveau de la couche MAC pour détecter une éventuelle transmission à partir du nœud suivant.
- `OverheardCallBack` : invoquée par la couche MAC lors de l'écoute d'un segment déjà transmis par le nœud.

- `MACDataResponseCallBack` : déclenchée lors de l'acquittement d'une trame au niveau de la couche MAC. Donc, elle estime le taux de transmission du nœud (ou bien la bande passante maximale du nœud).
- `MACDataIndicationCallBack` : déclenchée lors de la réception d'une trame au niveau de la couche MAC. Elle estime le taux de réception du nœud.
- `SendEACK` : permet d'envoyer un acquittement explicite vers le nœud précédent.

La classe CARTEESocket

Cette classe implémente les méthodes invoquées au niveau du nœud source et puits (i.e., segmentation, réassemblage).

Attributs

- `address` : adresse IP locale du nœud.
- `port` : numéro de port utilisé par l'application
- `l4Protocol` : une instance de la classe `CARTEEL4Protocol` pour transmettre les segments vers la couche transport.
- `sourcesList` : cet attribut est une liste constituée d'instances de la classe `CARTEESourceEntry`.

Méthodes

- `Send` : permet à la couche application d'envoyer un flux de données vers la couche transport, et spécialement vers le protocole CARTEE.
- `Receive` : invoquée lors de la réception d'un segment au niveau du nœud puits.
- `CreateFragments` : invoquée par la méthode Send lors de la réception d'un flux de donnée. Donc, elle renvoie une série de segments du protocole CARTEE.
- `SendEACK` : réalise la même action que celle de la classe `CARTEEL4Protocol`.

La classe CARTEESourcesList

Cette classe représente la liste des sources qui transmettent leurs flux vers le puits. Cette classe est utilisée uniquement au niveau du nœud puits, pour lui permettre de gérer les flux de données et leurs sources.

Attributs

- `entries` : une liste d'instances de la classe `CARTEESourceEntry`.

Méthodes

- `Add` : ajoute une nouvelle entrée `CARTEESourceEntry` dans la liste entries.
- `Remove` : supprime une entrée existante dans la liste entries.

La classe CARTEESourceEntry

Chaque instance de cette classe représente une source de la liste CARTEESourcesList.

Attributs

- `sourceAddress` : l'adresse IP du nœud source.
- `destinationAddress` : l'adresse IP du nœud puits.
- `streamID` : l'identifiant du flux transmit.
- `receivedDataCache` : inclut les données des segments dont la série ne présente pas de trous.
- `inorderedFragments` : segments reçus en désordre où leurs présences causent des trous dans le cache.

Méthodes

- `ReceiveFragment` : invoquée lors de la réception d'un nouveau segment du flux au niveau du nœud puits. Son objectif est de mettre à jour le cache de données.
- `DataStreamIsCompleted` : renvoie une valeur booléenne indiquant la complétude du flux de données.

4.4.2 Diagramme de sequence

Cette section présente des diagrammes de séquence permettant de décrire les interactions entre les objets du modèle CARTEE implémenté sous ns-3.

Transmission au niveau du nœud source

Le démarrage d'une transmission au niveau d'un nœud source instancie un objet de la classe `CARTEESocket`. Lors de sa création, l'instance de la classe `CARTEESocket` est responsable de la communication de bout-en-bout entre la source et le puits. Cette instance reçoit le flux de données transmis par la couche application au niveau du nœud source. A la réception d'un flux de données, l'instance `CARTEESocket` est responsable de la segmentation du flux en segments de taille 50 octets. Pour chaque segment, et comme unité de contrôle du protocole (PCU), l'instance doit inclure un numéro de séquence, un identifiant du flux et le nombre de segments dans le flux. Après la création de ces segments, l'objet instancié de `CARTEESocket` transmet les segments vers l'instance `CARTEEL4Protocol` qui à son tour exécute le mécanisme de transmission à base de fenêtre glissante pendant un intervalle de transmission. Après cet intervalle, cette instance déclenche une attente active pour écouter d'éventuels acquittements implicites ou explicites. La figure 4.5 représente un diagramme de séquences du processus de transmission au niveau du nœud source.

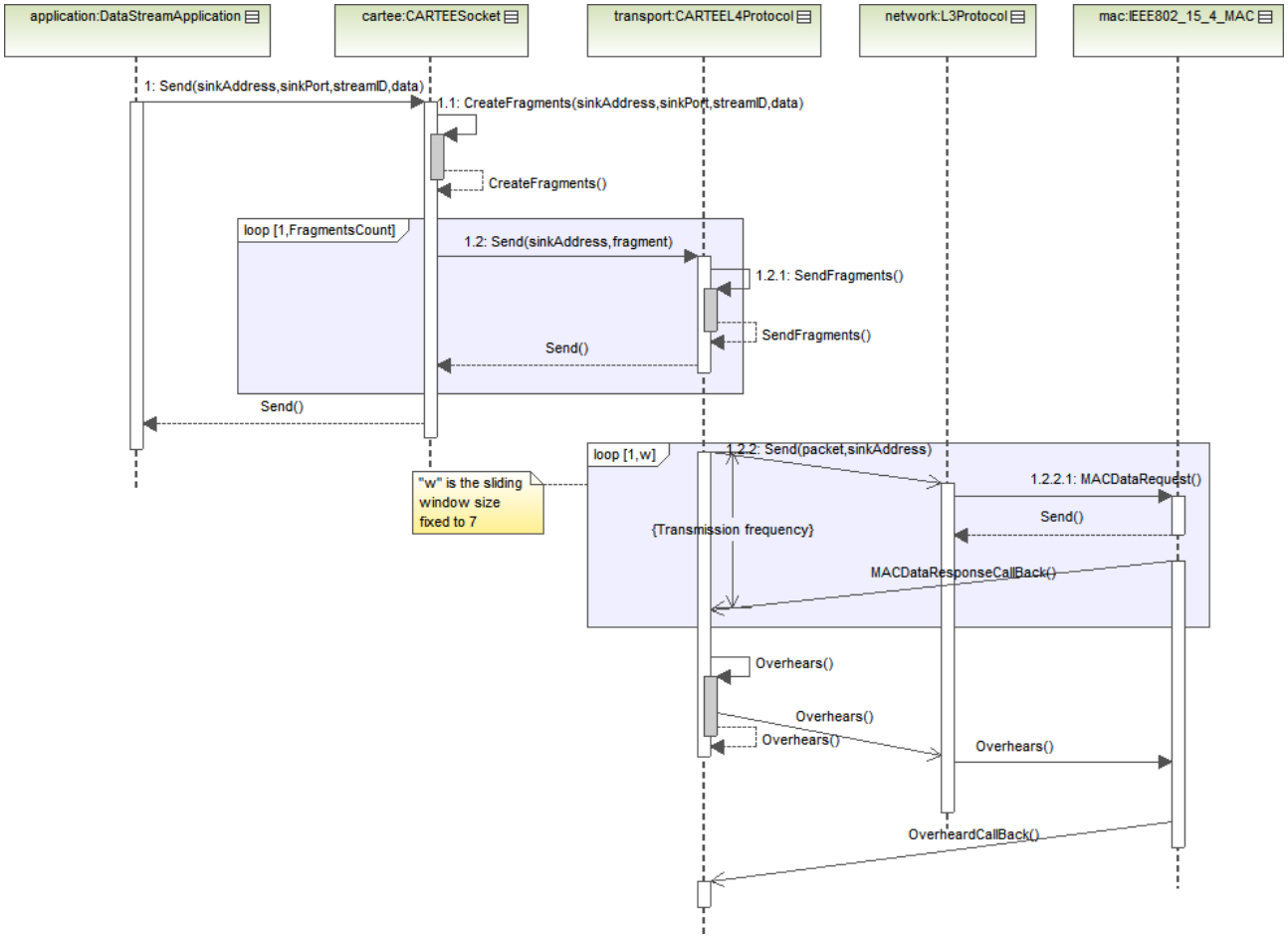

FIGURE 4.5 : Diagramme de séquences de transmission

Acheminement au niveau des nœuds intermédiaires

Chaque nœud dans le réseau possède un objet de la classe `CARTEEL4Protocol` qui permet de gérer les mécanismes de transmission et d'acquittement. En plus, l'instance `CARTEEL4Protocol` estime les taux de transmission et de réception afin de détecter d'éventuelles congestions. Cette instance peut recevoir des paquets à partir de l'agent de routage. Dans ce cas-ci, l'instance `CARTEEL4Protocol` ordonne le paquet reçu dans la file de transmission pour l'acheminer vers la destination finale (puits). La figure 4.6 schématise les interactions entre l'objet `CARTEEL4Protocol` et l'agent de routage. Cette figure illustre aussi les interactions de la couche MAC.

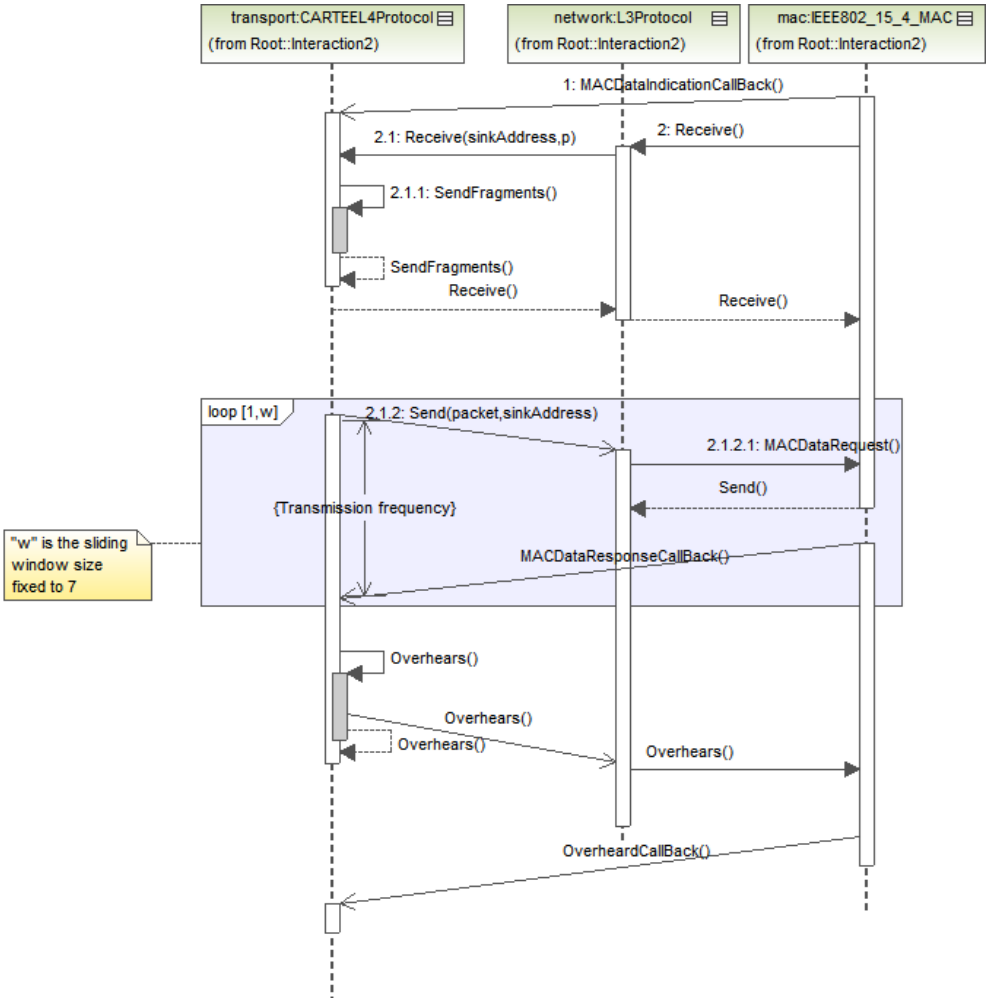

FIGURE 4.6 : Diagramme de séquence d'acheminnement

Réception au niveau du nœud puits

Le nœud puits instancie des objets de la même manière que le nœud source. Donc il inclut une instance de la classe `CARTEESocket` et évidement une autre de la classe `CARTEEL4Protocol`. Les interactions entre l'objet et l'agent de routage sont identiques que ceux du nœud intermédiaire à la seule différence est que le nœud puits crée une instance de la classe `CARTEESourceEntry` pour chaque source de données. Cette instance gère les séquences de segments reçus au niveau du puits pour transmettre un flux complet vers la couche application. La figure 4.7 décrit le diagramme de séquences illustrant les interactions des objets au niveau du nœud puits.

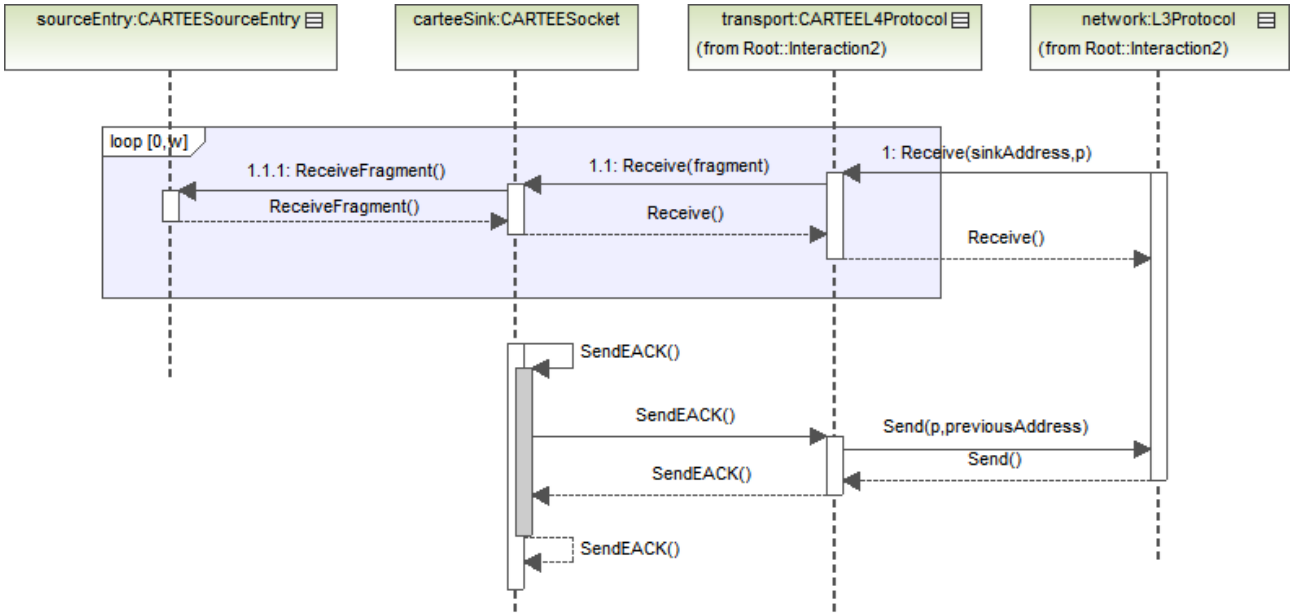

FIGURE 4.7 : Diagramme de séquences de réception

4.5 Conclusion

Ce chapitre a fait l'objet des démarches d'évaluation des performances du protocole CARTEE. La variété des approches et des outils d'évaluation compliquent l'évaluation des performances d'une solution. Les deux approches largement utilisées dans l'évaluation de performances d'une solution sont discutées. Vu le coût considérable de l'approche d'évaluation à base d'un réseau réel et vue les possibilités offertes par son homologue (évaluation à base de simulation), le protocole CARTEE a été évalué en conduisant une évaluation à base de simulation. Dans ce chapitre l'importance de la décision du choix du logiciel de simulation utilisé dans l'évaluation fut soulevée car elle influe sur la crédibilité des résultats. En se basant sur des travaux réalisés dans la littérature qui définissent des critères pour évaluer les logiciels de simulation, une évaluation de ces logiciels (les simulateurs les plus utilisés dans la littérature) est aussi présentée. A partir de cette évaluation le simulateur ns-3 a été choisi pour simuler le protocole CARTEE. Ce chapitre a présenté également une description détaillée de l'implémentation du protocole CARTEE dans le simulateur ns-3.
Donc, dans ce chapitre on a décrit les initiatives suivis pour évaluer les performances du protocole CARTEE, ainsi que les décisions qui ont été prises pour choisir les outils appropriés pour conduire cette évaluation. Sur la base des résultats de simulation obtenus, le prochain chapitre sera consacré à la valorisation des performances de CARTEE par rapports aux solutions de transport fiable pour les RCSF dans la littérature.

CHAPITRE 5

EVALUATION DES PERFORMANCES DU PROTOCOLE CARTEE

5.1 Introduction

Le coût et la complexité de la validation des protocoles conçus pour les réseaux sans fil ont encouragés les chercheurs à recourir aux approches de validation à base de simulation. Cependant, l'exactitude de cette validation reste toujours indéterminée due à l'absence des pratiques d'évaluation de résultats. Les méthodes d'analyse mathématique fournissent cette assertion, mais la complexité des modèles mathématiques utilisés dans la description des protocoles et des communications sans fil soulève des difficultés d'analyses. La simulation est utilisée pour réduire cette complexité en évaluant les performances des réseaux à travers un ensemble de modèles qui imitent le comportement de ces protocoles ce qui accélère considérablement l'exploration des conceptions. Malgré les avantages apportés par les outils de simulation, il est nécessaire de donner une explication convaincante des résultats de simulation. Donc, la validation des résultats obtenus permet d'assurer que le model simulé est capable de fournir des réponses significatives aux questions posées.
Dans le cadre de validation du protocole CARTEE, ce chapitre met l'accent sur une expérimentation conduite à base de simulation pour évaluer ses performances. Les hypothèses qui doivent être considérées pour valider les résultats de simulation et également quelques directives pour réussir cette validation sont exposées en premier suivi de la discussion de l'évaluation des performances de la solution CARTEE [10] dans différentes situations en utilisant le simulateur ns-3.

5.2 La validation à base de simulation

Dans tous les processus de développement, la validation est une phase d'affirmation de la similitude des résultats obtenus. Donc, la validation des résultats de simulation consiste à assurer que les modèles conçus fournissent des réponses significatives et convaincantes au champ de question de l'étude envisagé. D'une manière générale, on peut imaginer que la validation est une comparaison des résultats de simulation par rapport aux résultats du monde réel. Malheureusement, cette vision est particulièrement appropriée pour les petits réseaux et elle devient plus compliquée pour les grandes spécifications. Les modèles conçus pour la simulation décrivent d'une manière approximative des phénomènes du monde réel, ce qui implique que la validation offre une confiance que ces modèles imitent approximativement le comportement de la réalité. Parfois il est nécessaire d'effectuer plusieurs niveau de validation pour s'assurer de l'exactitude du modèle, parce qu'on peut obtenir des résultats de simulation qui semble valide pour certains aspects alors qu'ils deviennent invalide pour d'autres aspects du modèle simulé. Pa exemple, dans le cas d'un réseau filaire, on peut décrire la couche

physique et la couche MAC en utilisant un modèle d'erreur simple qui repose sur le délai de la bande passante. Ceci représente parfaitement un réseau filaire de haut débit présentant un faible taux d'erreur de bit, mais par contre, un environnement sans fil qui souffre des effets d'atténuation, d'interférence et de mobilité nécessite des modèles plus complexes pour imiter ces communications [70]. Alors, la validation soulève des enjeux considérables dans la simulation pour mieux comprendre et prédire le comportement des protocoles dans les réseaux.
Pour mettre en évidence les défis de validation dans la simulation des réseaux et discuter les approches utilisées, l'institut national de technologie et des standards (NIST) et l'agence de défense des projets de recherche avancés (DARPA) ont sponsorisés un workshop dans le mois de mai 1999. Les participants du workshop ont soumis des articles adressant les défis qui concernent la validation de la simulation dont certaines discutent des approches de validation suivis par des chercheurs et des pratiquants. A partir d'un résumé des discussions du workshop afin de tirer profit de ces travaux, cette section énumère les conseils suivant :

- Généralement on fait recourt au comportement du monde réel pour valider un modèle. Une telle approche qui semble évidente repose sur la comparaison des résultats de simulation par rapport aux résultats obtenus à partir d'un réseau réel. La comparaison directe peut être réalisée pour des petits réseaux, mais par contre, dans un réseau à grande échelle (i.e., RCSF de 1000 nœuds) cette comparaison peut s'avérer difficile.

- Lors de la spécification d'un protocole, les concepteurs se concentrent sur les performances du protocole dans un niveau spécifique de la pile protocolaire. Ceci engendre des décisions qui vont être prise par les programmeurs du protocole. En fait, ces décisions génèrent plusieurs implémentations du protocole toute en gardant sa spécification originale. Par exemple, dans la spécification du protocole TCP, les détails du temps d'acquittement sont cédés aux décisions de l'implémentation telle que la détermination du temps est empirique. En conséquence, on peut obtenir plusieurs implémentations pour une même spécification avec des performances différentes.

- On ne peut pas compter sur la comparaison avec une implémentation spécifique pour valider un protocole, parce qu'un simulateur crédible peut être dépassé avec l'évolution du trafic et du protocole. Dans cette situation il est nécessaire de valider la simulation avec les futures implémentations du protocole au lieu d'utiliser les implémentations actuelles.

- Lors de l'évolution de la conception du protocole, les implémentations actuelles du protocole vieillissent par rapport aux versions actuelles de recherche. Donc, la simulation aussi devient dépassée. Par exemple, l'implémentation de TCP Reno a subit une dégradation des performances exactement lors de l'occurrence de plusieurs pertes de paquets dans une seul période de temps d'aller-retour (RTT). Pendant la standardisation de l'option d'acquittement sélective du TCP, les performances actuelles de TCP Reno sont devenues invalides.

- L'internet a reconnu un changement dramatique dans le trafic qui s'écoule dans le réseau. Donc, la validation par rapport au trafic précédent peut ignorer les situations actuelles, alors que la validation par rapport au trafic actuel sous-estime les futures modèles.

5.2.1 Métriques de validation

Pour réaliser une validation, on doit toujours définir un ensemble de métriques qui mesurent les performances du protocole. Par exemple le protocole TCP est constitué de plusieurs algorithmes (i.e., la transmission à base de fenêtre glissante, démarrage lent et retransmission rapide). Le test de ces algorithmes dans un simulateur est

similaire au test de l'implémentation dans un réseau réel. Donc, il est nécessaire d'utiliser un ensemble d'outils d'analyse pour réaliser cette évaluation (i.e., graphes événement/temps, animation des paquets, génération des traces). Pour garantir que le protocole simulé s'exécute similairement à la spécification, on doit tester son comportement d'une manière correcte.
Généralement les concepteurs de modèles préfèrent les comparaisons visuelles dans le processus d'évaluation. Malheureusement, ces comparaisons visuelles sont limitées en termes d'efficacité due aux difficultés de quantifications du temps et du comportement. L'agrégation des mesures statistiques (i.e., paquets envoyé, débit et délai de livraison) peut fournir une alternative de la comparaison visuelle, mais il faut tenir compte que cette agrégation doit être choisie d'une manière appropriée pour éviter la mal-interprétation de la comparaison. Par exemple la comparaison du taux de transmission de données sur une période de temps par rapport aux différences de sporadicité du protocole.

5.2.2 Configuration du réseau

La simulation sous différentes configurations réseau par l'utilisation d'un ensemble de conditions renforce la crédibilité des résultats. Par exemple dans le cas de TCP, one ne peut pas valider le mécanisme de retransmission uniquement sous un environnement de faible taux d'erreurs, parce que dans un environnement à grande échelle on peut rencontrer d'autres défis (i.e., augmentation des communications et des traitements) non prisent en compte dans l'analyse sensitive. Donc, il est nécessaire de varier la topologie du réseau, le nombre de nœud et le trafic écoulé.

5.2.3 Coût et largeur de validation

Pour valider un protocole, il est nécessaire de considérer le cout et la largeur de validation par rapport au bénéfice apporté par cette dernière. Il est évident qu'une validation couteuse et détaillée est la plus appropriée, mais il faut prendre en compte que dans certaines situations, on ne peut jamais atteindre le niveau de détail désiré quel que soit les dépenses. En plus, dans certains cas, cette validation pourrait être non-nécessaire même si on atteint le niveau de détail souhaité. La validation d'une simulation est destinée à prouver à un client que le produit satisfait ses spécifications toute en diminuant les coûts et les exigences des différents scénarios.

5.2.4 Le passage à l'échelle

La simulation doit prendre en considération le passage à une grande échelle pour s'assurer de l'évolutivité du protocole. Si on considère le réseau internet aujourd'hui, la question clé pour comprendre le comportement d'un protocole est d'exécuter le protocole dans un grand nombre de nœuds, avec un trafic varié et avec plus de détails. Dans ce passage il existe deux approches complémentaires largement utilisées :

Exécution parallèle : aujourd'hui les simulateurs offre une exécution parallèle dans des ordinateurs multiprocesseurs. Cette approche peut élargir les simulations de 10 jusqu'à 100 fois que l'exécution séquentielle [71, 72].

Modèles abstraits : une approche complémentaire est l'utilisation de l'abstraction pour factoriser les détails non important à la simulation manuelle [73]. L'abstraction a été utilisée pour fournir une augmentation de 100 fois jusqu'à 1000 fois dans les tailles possibles de la simulation pour des questions d'une recherche particulière. Ceci veut dire que l'abstraction doit être appliquée soigneusement parce que dans l'absence d'une dérivation mathématique explicite, un modèle abstrait doit être encore validé par rapport à un modèle plus détaillé fonctionnant à une vitesse plus lente ou par rapport à des expériences réelles

à une échelle suffisamment grande. En outre, des nouveaux phénomènes peuvent s'émerger à partir des interactions avec l'augmentation de la taille du réseau.

On peut réaliser des simulations à grande-échelle sur la base de sous-modèles à petite-échelle en utilisant deux approches :

Composition récursive : dans cette approche on commence à partir des composants bien validés et une composition bien validée vers la génération des modèles large en utilisant la composition hiérarchique [72].

Comparaison des simulations : Dans cette approche on compare les simulations détaillées et abstraites à petite-échelle pour générer une large abstraction de scénarios [73].

5.3 Directives de validation

Le plus important du workshop organisé par INST et DARPA c'est la bonne compréhension des pratiques utilisées par la communauté pour valider les simulations des réseaux. Ce qui est intéressant aussi dans ce workshop, c'est que parmi les participants de l'industrie, un conférencier a donné un résumé concis de pratique et de recommandation pour faire réussir la validation [74, 75]. Les directives clés discutées dans ce résumé sont :

- La validation est plus simple lorsqu'elle se focalise sur une étude comparative au lieu d'une étude absolue. Ceci est évident lorsqu'on on compare une nouvelle proposition par rapport à une solution existante.
- L'utilisation des outils de conception et de représentation visuelle pour examiner la simulation permet de reconnaitre rapidement les comportements invalides. Malheureusement, les approches d'examinassions et de visualisation des modèles larges représentent un grand défis de recherche parce que ces modèles demandent des instruments intégrés avec plusieurs étapes de filtrage et de classification de données.
- Une variété de formes d'implémentations et de modèles met l'accent sur plusieurs aspects d'un système de communication. Donc, les concepteurs doivent comparer les résultats de simulation avec plusieurs représentations alternatives. L'augmentation du nombre de représentations alternatives par rapport auquel le modèle est comparé augmente la probabilité de découvrir les erreurs, les incohérences et les hypothèses non valides.
- Lorsque le modèle nécessite des interactions au cours du temps entre des entités indépendantes, on doit s'assurer d'introduire l'asynchronisme pour imiter le fonctionnement des systèmes réels.
- Les résultats de simulations doivent être reproductibles parce que plusieurs facteurs jouent un rôle important pour approuver la reproductibilité, incluant des algorithmes déterministes pour générer les numéros de séquences pseudo-aléatoire et l'atténuation des erreurs d'arrondissement des représentations en virgule flottante. Les erreurs d'arrondissement peuvent affecter la concurrence des événements, spécialement où la synchronisation optimiste est utilisée quand la simulation est exécutée dans les systèmes d'exploitation parallèles. En générale, des soins doivent être pris en compte pour assurer et veiller à ce que le temps et la cause de l'événement sont modélisés avec précision quand les systèmes de traitements parallèles sont utilisés pour exécuter des simulations.
- Lorsqu'il est nécessaire de réduire la taille de la simulation afin d'être exécutés dans des ressources mémoires et CPU limitées, il faut être prudent pour éviter d'introduire des limites artificielles dans le modèle. Par exemple, les effets transitoires de démarrage ou une topologie physique artificielle peuvent introduire des erreurs.

Bien que, ces recommandations sont plus importantes pour les développeurs des simulations réseau, ils sont applicables aux utilisateurs de simulation qui doivent évaluer la validité de leurs conclusions. Tout comme un développeur, l'utilisateur doit sélectionner une technique d'analyse, s'assurer que l'approche n'introduit pas des erreurs additionnelles, que les résultats sont interprétés de manière appropriée, même à des échelles différents.

5.4 Validation du protocole CARTEE

Les performances du protocole CARTEE ont été évaluées en utilisant le simulateur ns-3. Pour valider le comportement des principales caractéristiques de cette solution, l'évaluation a été conduite en utilisant deux modèles de simulation : multi-sauts et multi-sources. Le premier modèle a été utilisé pour valider la fiabilité (le mécanisme de transmission à fenêtre glissante et le mécanisme d'acquittement) en utilisant un déploiement linéaire : 3, 6 et 9 nœuds linéaires. Dans chacun de ces déploiements, la communication est initiée entre les deux nœuds d'extrémités. L'intérêt derrière ces simulations est pour exposer l'impact du multi-saut sur la consommation énergétique, le délai de livraison de données et la fiabilité. Le deuxième modèle est utilisé pour évaluer l'évolutivité de la solution ainsi que les mécanismes de contrôle de congestion. Pour faire ceci, 121 nœuds ont été déployés dans une grille 11×11. Concernant le nœud puits, il a été placé dans le centre de la zone de déploiement. Le nombre de sources de données dans le réseau varie entre 30, 60 et 90. L'objectif de cette simulation est pour illustrer l'influence du nombre de sources sur la consommation énergétique, le délai de livraison de données, la fiabilité et spécialement la congestion et le taux d'occupation du cache.

5.4.1 Paramètres de simulation

Le tableau 5.1 résume les paramètres de simulations communes dans les deux modèles. Le protocole CARTEE

Paramètre	**Valeur sélectionné**
Taille du flux de données	1024 octets de données au niveau application fragmentées en segments de 50 octets.
Fréquence de transmission du flux	1 flux de données par minute
Courant de transmission de l'antenne sans fil	17.4 mA
Courant de réception de l'antenne sans fil	18.8 mA
Courant d'estimation du canal libre	15 mA
Courant d'inactivité du canal	426 µA
Voltage supplémentaire	3 V
Protocole de routage	AODV [76]
Interface réseau	IEEE 802.15.4

TABLE 5.1 : Paramètres communs de simulation

a été conçu pour être exploité par n'importe quelle classe de protocole de routage d'une manière autonome. Le routage dynamique (i.e., centré sur les données, hiérarchique, géo localisé et multi-chemin), où la table de routage change fréquemment pendant la communication, sont largement utilisés dans les RCSF due aux caractéristiques et contraintes des RCSF [77, 78, 79]. Vue leurs dynamiques, ces protocole présentent certaines défis (i.e., le défis du réseau dynamique et le réseau et le chemin à multi-saut). Pour valider le comportement de protocole CARTEE [10], cette évaluation a été conduite pour démontrer l'impact de ces défis sur les performances de la solution. Le protocole de routage AODV [76] est utilisé pour créer des chemins multi-saut

à la demande dans un réseau dynamique [80]. En plus, les paquets de contrôle AODV présentent une faible surcharge dans le réseau, ce qui est appropriée pour valider l'efficacité de l'énergie du protocole CARTEE.
Les applications multimédia [50] dans les RCSF nécessitent la transmission d'une quantité importante de données. Cependant, les interfaces réseau des nœuds capteurs permettent uniquement la transmission des trames courtes (127 octets d'unité maximale de transmission dans l'interface IEEE 802.15.4 [17]). Comme il est mentionné dans le chapitre 3, les données applications multimédia dans les RCSF nécessitent un mécanisme de fragmentation avant la transmission de données et un mécanisme de réassemblage lors de la réception de données. Pour évaluer ce comportement, on a utilisé une détection d'une voix d'un (01) kilo octets. Pour transmettre les données détectées à travers l'interface réseau IEEE 802.15.4, ils doivent êtres fragmentées en de petits segments de 50 octets, sans oubliés les entêtes (PCU) qui vont être rajoutés au niveau de la couche réseau et la couche MAC.

5.4.2 Impact du multi saut sur les performances de la solution

Comme il est discuté en haut, ce scénario montre l'impact d'un chemin multi-saut sur les performances du protocole CARTEE. Dans ce modèle on déploie linéairement des nœuds en utilisant une distance de $70m$ entre les nœuds. La simulation commence par un réseau constitué de 3 nœuds, puis à 6 et par la suite à 9 nœud (cf. figure 5.1). Pour évaluer la fiabilité de la solution CARTEE, la couche réseau utilise un modèle d'erreur qui suit la loi binomiale avec un taux de bruit (SNR) et une distribution de nombre de bits. Ce modèle d'erreur est utilisé pour produire un nombre maximum d'erreur de bit en utilisant différentes qualités des liens entre les nœuds.
Les métriques suivantes ont été sélectionnées pour évaluer les performances de la solution CARTEE et les

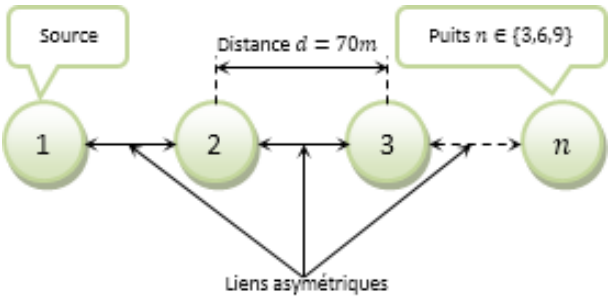

FIGURE 5.1 : Topologie multi-sauts

comparer avec les autres approches :

- Taux de fiabilité : estimé à partir du ratio des segments correctement reçus par rapport aux segments réellement transmis. Cette métrique mesure la fiabilité de livraison de bout-en-bout.
- Délai de livraison du flux : Estime le temps nécessaire pour délivrer un flux par succès vers le nœud puits.
- Moyenne de consommation énergétique : Représente le coût des communications effectuées par les nœuds (le mode de transmission, de réception, d'écoute et d'inactivité). A partir de cette métrique on peut observer le coût des retransmissions et des paquets de contrôle. Pour calculer la consommation énergétique, on a implémenté le modèle LRWPAN pour ns-3, qui représente le modèle d'énergie de la radio pour l'interface réseau IEEE 802.15.4. Dans ce modèle, la consommation énergétique est calculée à chaque transmission, réception, teste du canal libre et inactivité de l'interface.

Fiabilité

Pour évaluer la fiabilité du protocole CARTEE ainsi que les autres implémentations, on estime le nombre de segments correctement reçus au niveau du puits par rapport au nombre de segments envoyés par le nœud source. La figure 5.2 présente le taux de fiabilité de chaque solution simulée, y compris le protocole CARTEE. On remarque que les protocoles CARTEE, PSFQ [6] et HDRTP [35] sont fiables à 100%. Mais en considérant le protocole ERTP [1] et ESRT [8], tel qu'il est présenté dans la figure 5.2, ils sont témoins d'un taux d'erreur respectivement dans le rang $[0.01, 0.02]$, $[0.03, 0.14]$ selon le nombre de saut dans le chemin.

Le taux d'erreur causé par le protocole ERTP est dû au mécanisme d'acquittement implicite qui n'est pas

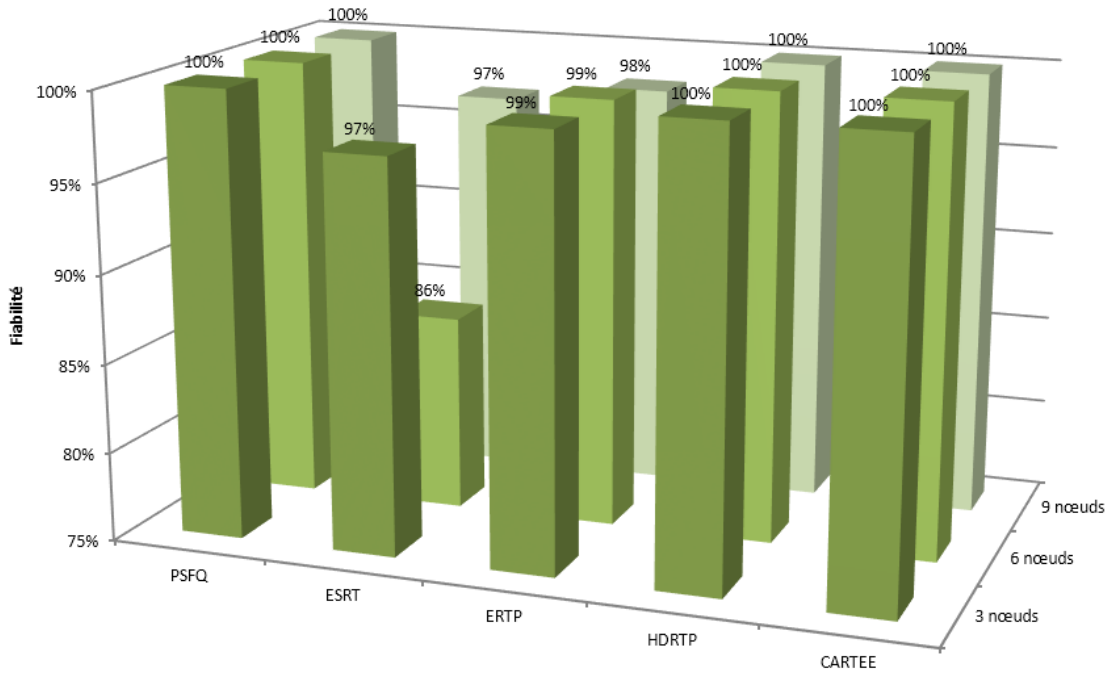

FIGURE 5.2 : Taux de fiabilité dans un model multi-saut

suffisant pour atteindre une fiabilité de 100%. D'autre part, la combinaison de l'acquittement implicite et explicite dans le protocole CARTEE a fait ses preuves en termes d'efficacité et de fiabilité. La dégradation de la fiabilité dans le protocole ESRT est due à sa dépendance de la fiabilité assurée au niveau du routage. Cependant, la majorité des protocoles de routage n'assurent pas la fiabilité (tel que AODV), et dont l'absence de la fiabilité, le nœud puits ne pourrait pas atteindre la région d'exploitation optimale (OOR).

Délai de livraison du flux

Le délai de livraison représente la durée entre le temps d'émission du premier segment du flux et le temps de réception du flux en entier au niveau du nœud puits. Les figures 5.3, 5.4 et 5.3 schématisent le délai de livraison de 120 flux de données à partir de la source vers le puits dans le cas de 3, 6 et 9 nœuds linéaires. Dans le cas de 3 nœuds, tel qu'il est représenté dans la figure 5.3, CARTEE effectue un court délai de livraison par rapport à HDRTP, ERTP et PSFQ. Tel qu'il est présenté dans les résultats de simulations, ESRT effectue un court délai de livraison dans la région OOR, mais l'absence du mécanisme de fiabilité empêche le protocole pour atteindre cette région.

Le mécanisme de transmission à base de fenêtre glissante du protocole CARTEE réduit le délai de livraison

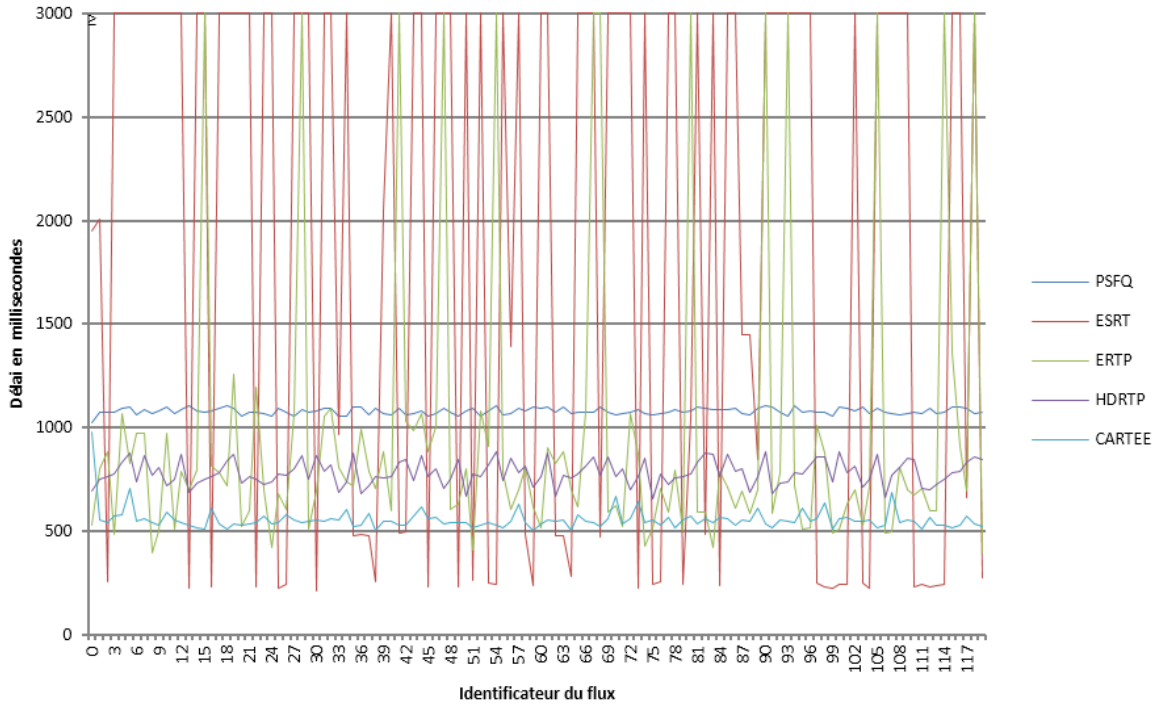

FIGURE 5.3 : Délai de livraison du flux dans un chemin constitué de 3 nœuds

par rapport au mécanisme send-and-wait utilisé dans le protocole ERTP. Le temps de transmission aléatoire implémenté dans le protocole PSFQ évite la congestion mais il agrandit le délai de livraison. Cependant, l'assignement dynamique des paramètres rajouté dans le protocole HDRTP réduit le délai de livraison par rapport à PSFQ, mais le protocole CARTEE surpasse HDRTP à cet égard.
Dans un chemin constitué de 6 nœuds, tel qu'il est présenté dans la figure 5.4, il est très clair que le protocole CARTEE surpasse les protocoles PSFQ, ESRT, ERTP et HDRTP en termes de délai de livraison.

La cause de dégradation du délai de livraison dans le protocole ESRT est due à l'ajustement de la fréquence de transmission. La non-existence d'un mécanisme de fiabilité empêche le protocole ESRT pour atteindre la région OOR.

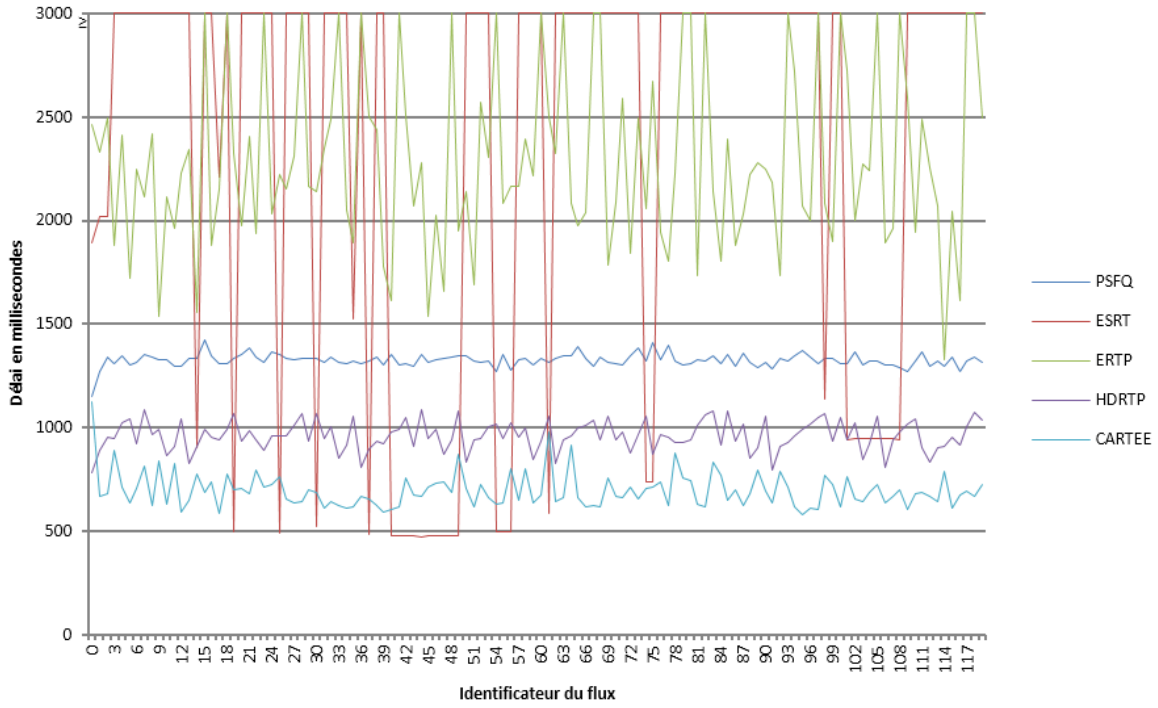

FIGURE 5.4 : Délai de livraison du flux dans un chemin constitué de 6 nœuds

Dans le cas où le chemin est composé de 9 nœuds (cf. figure 5.5), le délai de livraison du protocole CARTEE reste toujours le même que dans le cas de 3 et 6 nœuds mais le délai augmente avec les protocoles PSFQ, HDRTP et ERTP. Dans ce cas-ci, on observe que le protocole ESRT atteint la région OOR et opère d'une manière optimale. Evidemment, le protocole ESRT réalise un court délai dans la région OOR.

A partir des résultats de simulation obtenus, on peut conclure que l'ajustement de la fréquence du protocole ESRT possède les meilleures performances lorsqu'il est renforcé avec un mécanisme de fiabilité. On conclut également que dans la plupart des cas, le protocole CARTEE se rapproche au délai de livraison optimal tout en assurant une fiabilité à 100%.

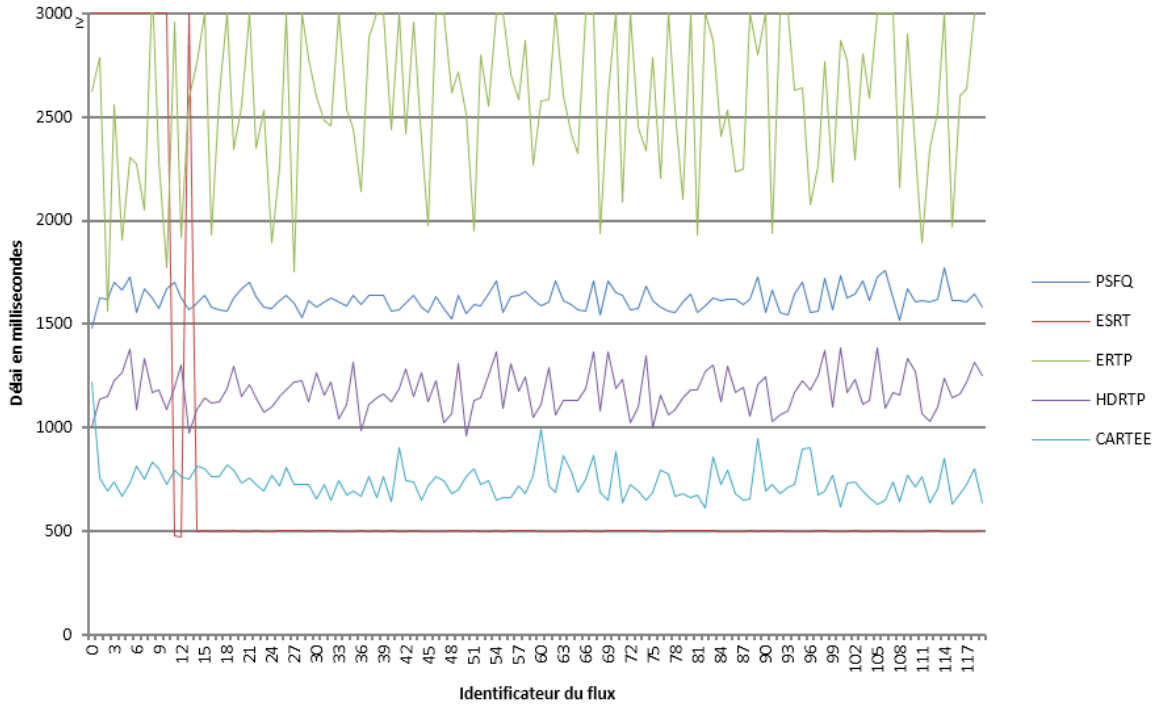

FIGURE 5.5 : Délai de livraison du flux dans un chemin constitué de 9 nœuds

Consommation énergétique

Pour évaluer cette métrique, on a implémenté le modèle radio LRWPAN sous ns-3. L'énergie consommée est calculée lors de l'état d'émission, de réception, d'estimation du canal libre et d'inactivité. La consommation énergétique est de 18.8 mA/second (respectivement de 17.4 mA/second, 15mA/second, 426 µA/second) pour l'état de transmission (respectivement pour l'état de réception, de test du canal et d'inactivité)

La figure 5.6 présente la moyenne de consommation de l'énergie dans le réseau. En réalité, le protocole CARTEE consomme plus d'énergie par rapport au protocole PSFQ, ERTP et HDRTP. Cependant, ESRT minimise la consommation énergétique par rapport à la solution CARTEE dans un chemin de 6 et 9 nœuds, parce qu'il n'effectue pas de retransmission. Le gaspillage de l'énergie présenté par le protocole ERTP est causé par les retransmissions non nécessaires, par contre, dans le protocole PSFQ, le gaspillage est dû aux acquittements négatifs et le temps de transmission aléatoire. HDRTP génère plus de communications par rapport à PSFQ à cause des messages d'acquittement et d'information utilisés pour initier la communication, ce qui mène à une consommation énergétique supplémentaire. La dégradation des performances du protocole ESRT dans le cas de 3 nœuds est provoquée par la diffusion de la fréquence de transmission.

Les résultats obtenus (taux de fiabilité, délai de livraison et consommation énergétique) valide la fiabilité et l'efficacité de la solution CARTEE, ces résultats présente aussi que le nombre de nœud dans le chemin n'affecte pas les performances de cette solution. D'une manière plus précise, ces résultats démontrent que le protocole CARTEE atteint une fiabilité de 100% en réduisant significativement le délai de livraison et la consommation énergétique.

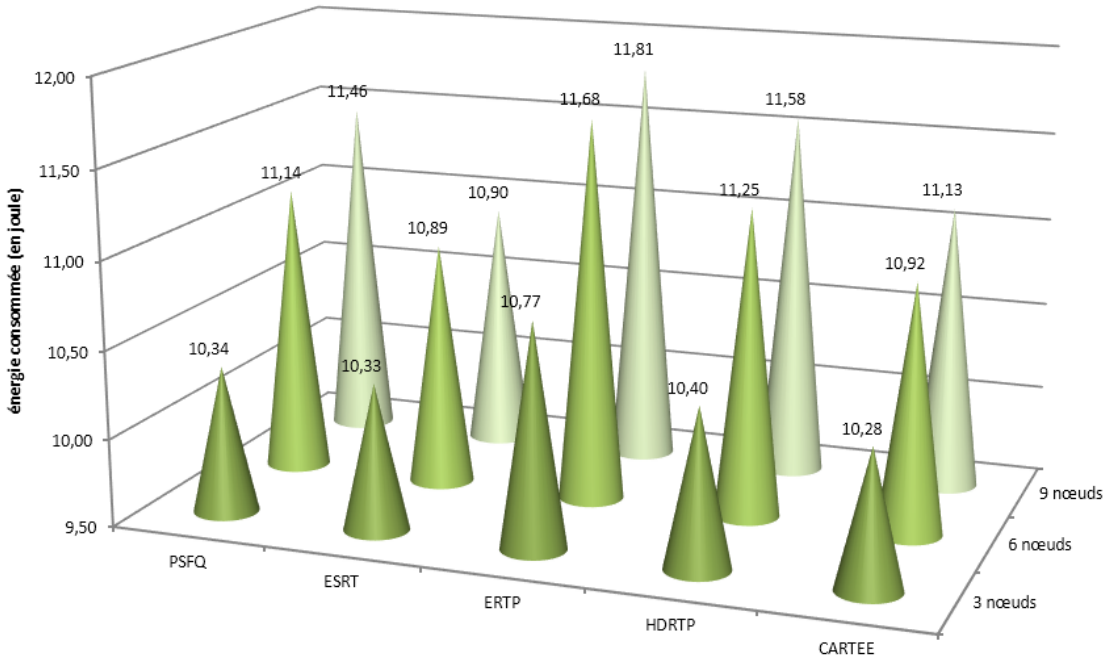

FIGURE 5.6 : Moyenne de consommation de l'énergie par les nœud du réseau

5.4.3 Impact des multi-sources sur les performances de la solution

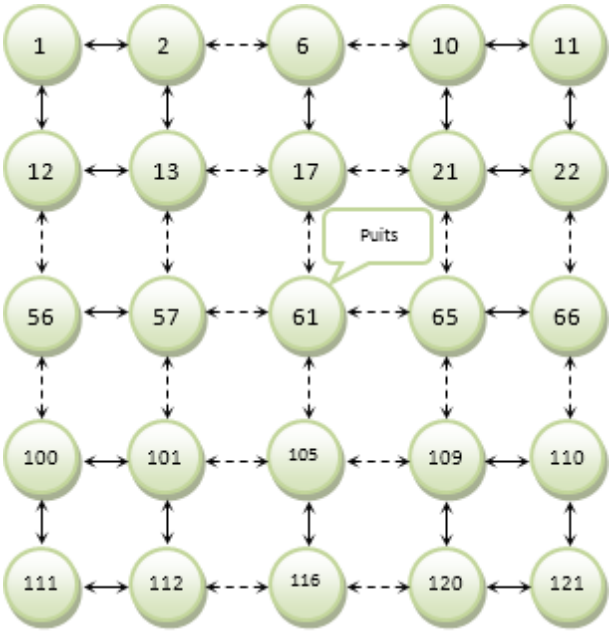

FIGURE 5.7 : Topologie multi-sources

Comme il est présenté dans la figure 5.7, ce modèle montre l'impact de plusieurs sources de données sur les performances du protocole CARTEE. L'intérêt ici est de faire apparaitre la capacité de protocole CARTEE pour éviter la congestion en utilisant le mécanisme d'adaptation de taux de transmission. Dans ce modèle on a déployé 121 nœud dans une grille 11×11 sur une zone de $700m \times 700m$. La distance entre chaque deux nœuds est $70m$. Le nœud puits est placé au centre de la grille. Le trafic est initié à partir de n source de données

vers le puits, tel que n varie entre 30, 60 et 90 sources aléatoirement choisi dans le réseau.
Vu que le protocole PSFQ assure la fiabilité dans une communication point-à-multipoints (à partir du puits vers les nœuds sources), ce dernier est modifié pour supporter la communication multipoints-à-point. En considérant les métriques évaluées précédemment (fiabilité, délai de livraison du flux et consommation énergétique), deux autres métriques sont considérées :

- Le nombre de congestions causées : pour compter le nombre de congestions qui s'est produites dans le réseau.
- Cache nécessaire : pour calculer le maximum et le minimum du nombre de segments inclus au niveau du cache des nœuds pendant la communication.

Fiabilité

Le taux de fiabilité de chaque source a été estimé comme il est discuté dans le modèle multi-saut. La figure 5.8 décrit la moyenne des taux de fiabilité au niveau de chaque source dans le réseau. Selon les valeurs indiquées dans cette figure, CARTEE et HDRTP sont les seuls solutions qui assurent 100% de fiabilité. Mais on observe que les autres solutions présentent différents taux d'erreur (ERTP : $[0.05, 0.05]$, ESRT : $[0.85, 0.9]$ et PSFQ : $[0.65, 0.71]$).
L'insuffisance de l'acquittement implicite est apparue clairement dans le modèle multi-sources. La combinaison de l'acquittement implicite et explicite a fait ses preuves dans le protocole CARTEE (100% de fiabilité peut être atteinte quelques soit le nombre de sources).

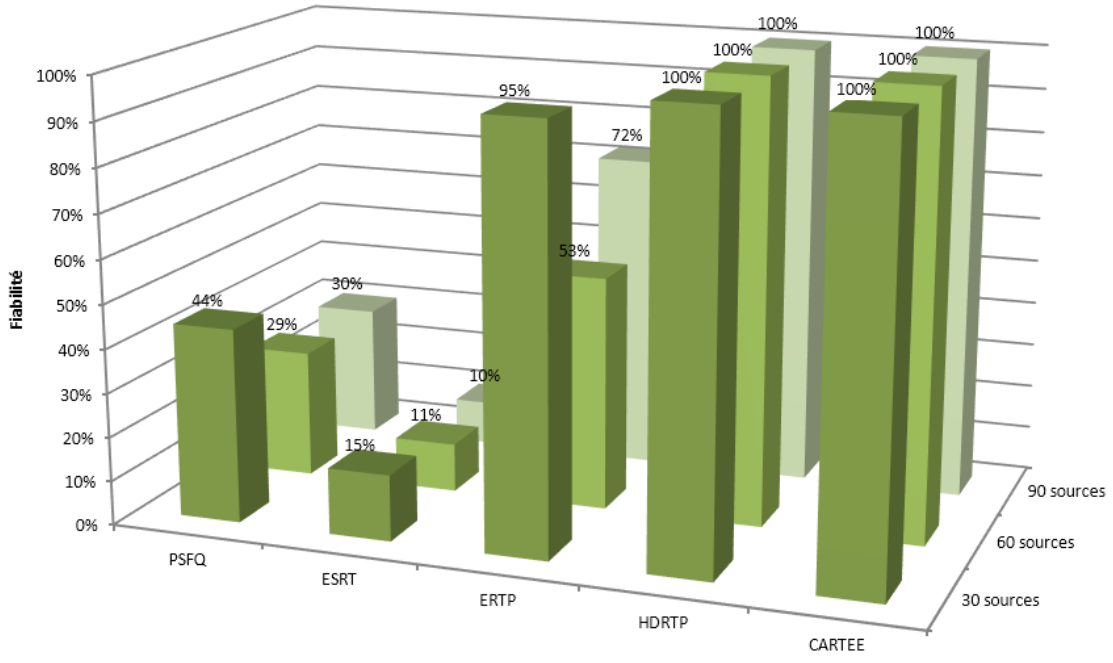

FIGURE 5.8 : Taux de fiabilité dans un model multi-sources

Délai de livraison du flux

Les résultats présentés dans les figures 5.9, 5.10 et 5.11 représentent le délai de livraison moyen de tous les nœuds sources. Le protocole CARTTEE montre un court délai de livraison par rapport aux protocoles ERTP, ESRT et PSFQ avec 30 sources de données. L'utilisation de la fenêtre glissante est derrière ce court délai de livraison. Par contre, le mécanisme send-and-wait du protocole ERTP et le temps de transmission aléatoire du PSFQ retardent la livraison de données. En assignant dynamiquement les paramètres t_{min}, t_{max} et α, le protocole HDRTP effectue un court délai par rapport à PSFQ, mais les congestions provoquées dans ce protocole ont augmenté le délai de livraison par rapport au protocole CARTEE.

La figure 5.10 (cas de 60 sources de données) montre que le protocole CARTEE surpasse les protocoles ESRT

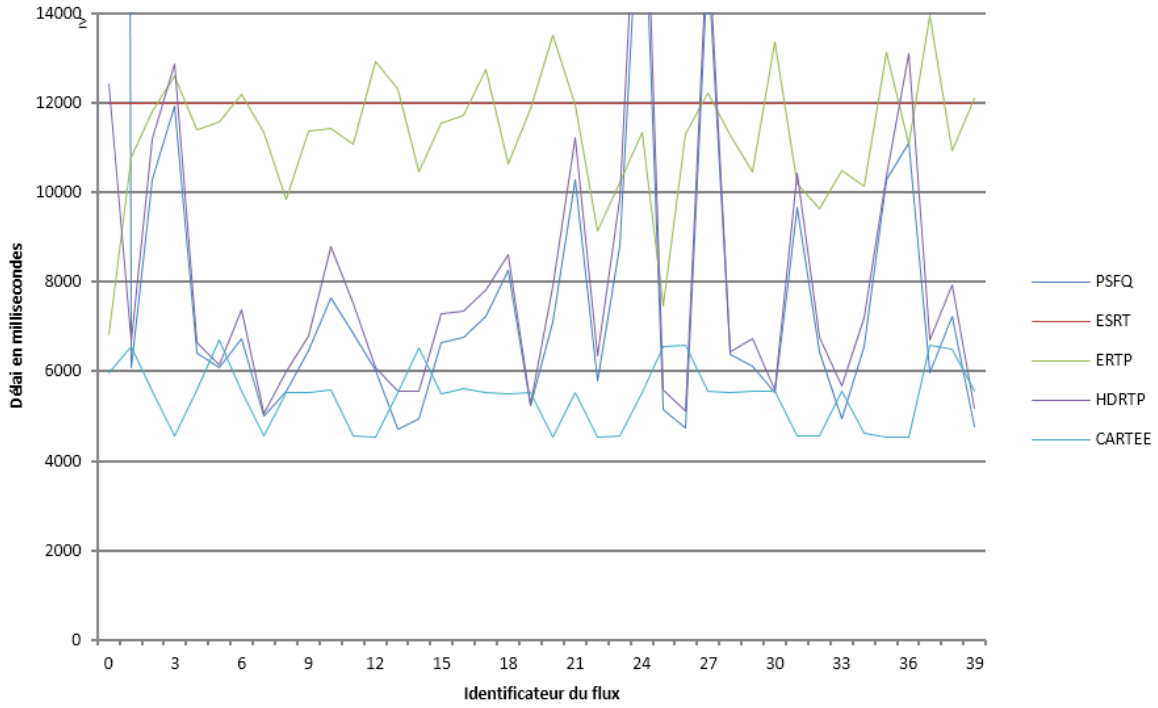

FIGURE 5.9 : Délai de livraison du flux avec 30 sources de données

et ERTP en terme de délai de livraison. On souligne qu'à cause de la dégradation de la fiabilité, le protocole PSFQ apporte un court délai de livraison. HDRTP atteint un court délai de livraison dans le cas de 60 sources de données.

Comme il est discuté dans le modèle précédent (modèle multi-saut), la dégradation du délai de livraison dans le protocole ESRT est causée par l'ajustement de la fréquence de transmission effectuée au niveau du nœud puits.

Finalement, la figure 5.11 (cas de 90 sources) montre que le protocole CARTEE est évolutifs et plus efficace que le protocole ESRT et ERTP.

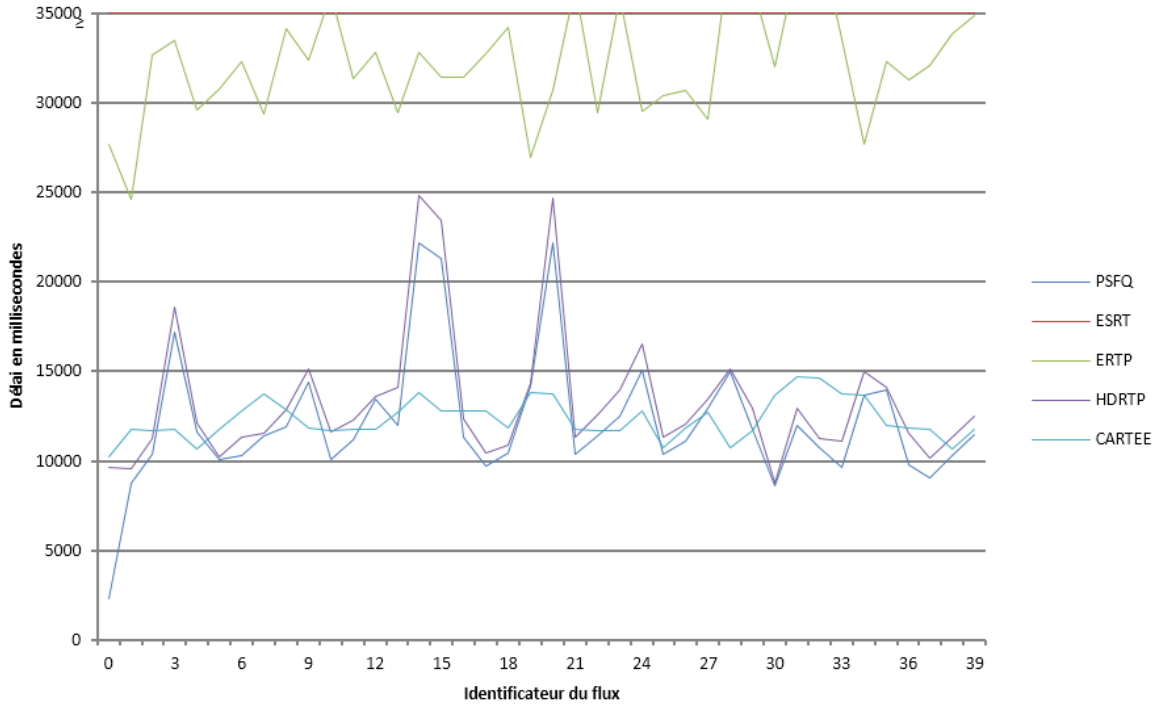

FIGURE 5.10 : Délai de livraison du flux avec 60 sources de données

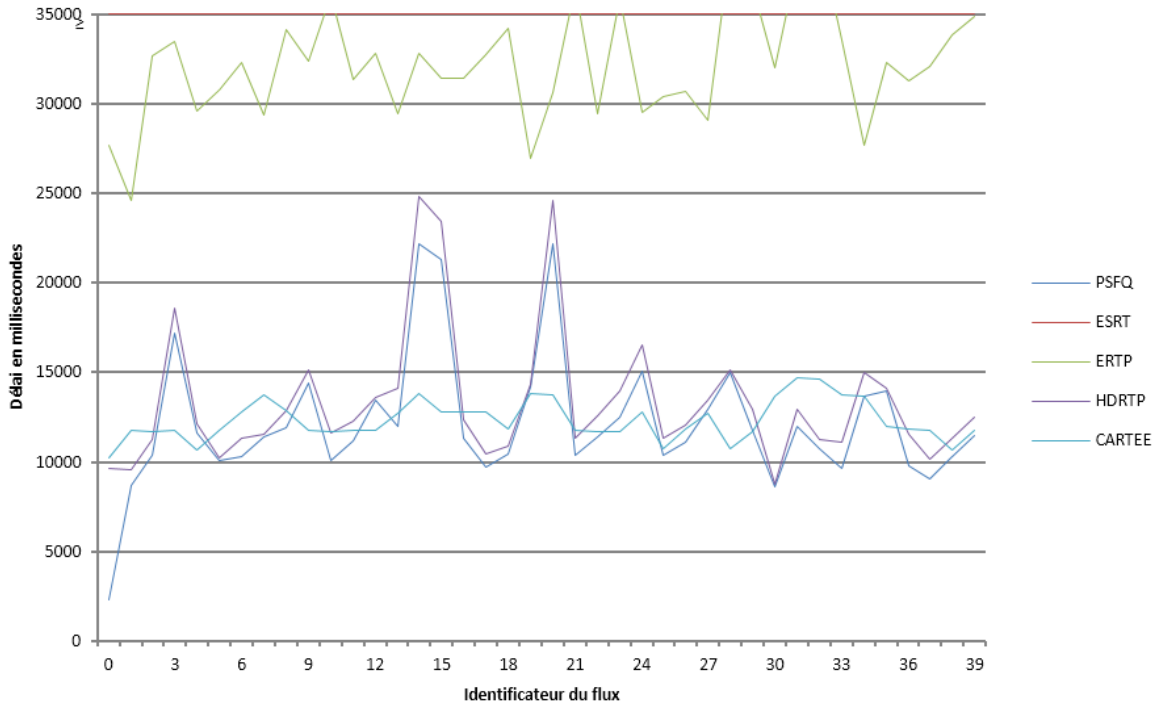

FIGURE 5.11 : Délai de livraison du flux avec 90 sources de données

Consommation énergétique

La figure 5.12 présente la moyenne de la consommation énergétique du réseau. A partir de cette figure on observe que la solution CARTEE consomme moins d'énergie par rapport aux autres solutions, ceci est évident parce que cette solution réduit les transmissions. Les retransmissions non nécessaires après une transmission

correcte du protocole ERTP engendrent une consommation supplémentaire des ressources énergétiques. En ce qui concerne le protocole PSFQ et HDRTP, les acquittements négatifs et le temps de transmission aléatoire, ainsi les messages d'information et d'acquittement spécifiques au protocole HDRTP nécessitent une consommation énergétique significative. En réalité, est spécialement dans le protocole **PSFQ**, lorsque le nombre de nœuds augmente, la perte de segments augmente (c'est les cas des flux irrécupérables), ce qui signifie que les communications sont réduites et évidement la consommation de l'énergie est ainsi. Comme il est démontré dans la figure 5.12, la diffusion de la fréquence du protocole ESRT le conduit rapidement pour épuiser l'énergie du nœud. Vraiment, la différence entre le protocole CARTEE et ESRT est en générale de 67% en terme de consommation de l'énergie.

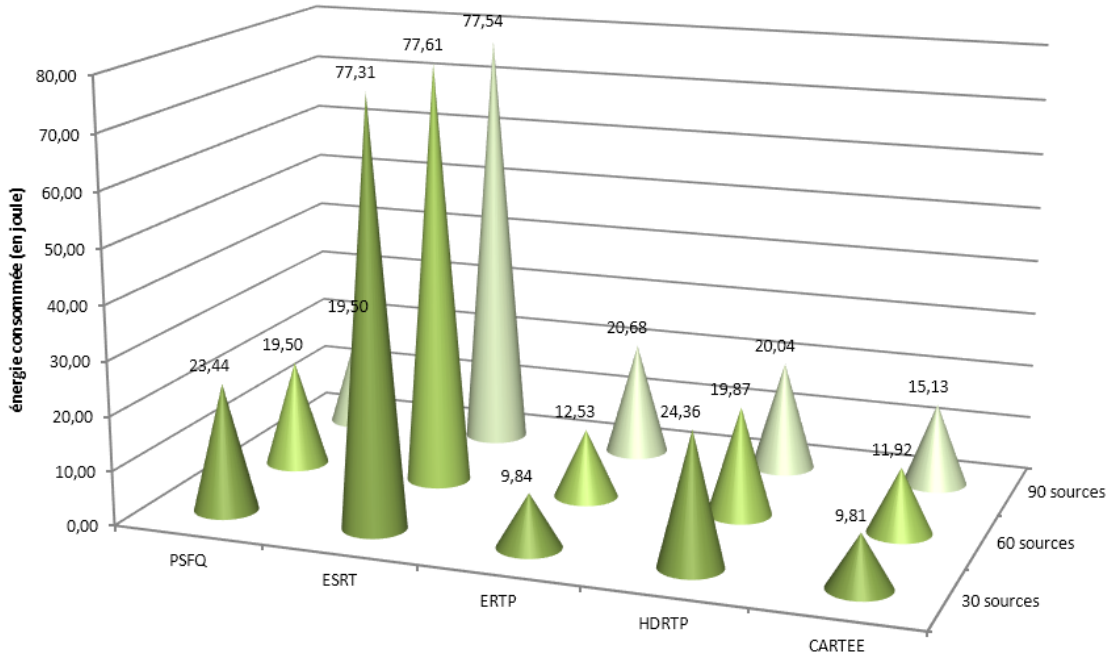

FIGURE 5.12 : Moyenne de consommation de l'énergie par les nœud du réseau

Congestion et cache de données

Le tableau 5.2 récapitule le nombre de congestions produites dans le réseau entier ainsi que l'occupation du cache de données pendant la communication. L'état du cache est donné en deux nombres (bornes) ; le nombre maximum et minimum de segments misent dans le cache en estimant le cache de donnée au niveau de chaque nœud. Comme il est démontré dans le tableau 5.2, CARTEE et ERTP évitent la congestion. Cependant, CARTEE réduit l'occupation du cache par rapport au protocole ERTP (40% de réduction de l'occupation du cache).

Nom du protocole	Nombre de sources	Nombre de congestions produites dans le réseau	Taille du cache de données (nombre segments par nœud)
PSFQ	30 sources de données	116116 situations	Entre 21 et 252 segments dans le cache.
	60 sources de données	99350 situations	Entre 21 et 336 segments dans le cache.
	90 sources de données	31542 situations	Entre 21 et 420 segments dans le cache.
ESRT	30 sources de données	99350 situations	Aucun cache de données.
	60 sources de données	33511 situations	Aucun cache de données.
	90 sources de données	74800 situations	Aucun cache de données.
ERTP	30 sources de données	Aucune congestion	Entre 1 et 65 segments dans le cache.
	60 sources de données	Aucune congestion	Entre 1 et 105 segments dans le cache.
	90 sources de données	Aucune congestion	Entre 1 et 197 segments dans le cache.
HDRTP	30 sources de données	82116 situations	Entre 21 et 252 segments dans le cache.
	60 sources de données	100253 situations	Entre 21 et 336 segments dans le cache.
	90 sources de données	138348 situations	Entre 21 et 420 segments dans le cache.
CARTEE	30 sources de données	Aucune congestion	Entre 1 et 35 segments dans le cache.
	60 sources de données	Aucune congestion	Entre 1 et 77 segments dans le cache.
	90 sources de données	Aucune congestion	Entre 1 et 135 segments dans le cache.

TABLE 5.2 : Récapitulation de la congestion et la taille du cache de données

5.5 Conclusion

Dans ce chapitre, une étude de performance du protocole CARTEE à base de simulation est présentée. L'absence des méthodes et techniques conduisant à une simulation réussie ont orienté les recherches à sélectionner arbitrairement des approches de validation sans se préoccuper de l'exactitude de la simulation. Pour remédier à cette lacune, des hypothèses permettant de valider la simulation ont étés présentées. Ce chapitre présente aussi un ensemble de directives pour réussir cette validation. Ces directives ont été prises en considération dans la validation du protocole CARTEE. Pour établir un lien entre l'étude théorique et la réalité, ce chapitre a présenté une évaluation des performances de cette solution sous deux conditions : cas des chemins multi-saut et le cas de multi-sources de données. Les performances du protocole CARTEE ont été comparées par rapports aux propositions existantes dans la littérature (i.e., PSFQ, ESRT, ERTP et HDRTP). Sous ces conditions, CARTEE a fait ses preuves en termes de fiabilité, de délai de livraison, de consommation énergétique, de congestion et d'occupation mémoire par rapport aux solutions existantes.
D'un point de vue pratique, on peut conclure que notre solution garantie les hypothèses établies dans la description et l'analyse formelle de cette solution décrites dans le chapitre 3.

CONCLUSION GÉNÉRALE

Dans cette thèse, on a présenté une solution de transport fiable de données dans les RCSF baptisée CARTEE. Le transport fiable de données est un des défis parmi plusieurs qu'on rencontre souvent dans les RCSF. Le besoin croissant en applications multimédia dans les RCSF durant ces dernières années, a encouragé la recherche de nouvelles solutions pour faire face à ces défis. Cependant, les solutions proposées dans la littérature ne remplissent pas totalement les exigences de ces applications dans les RCSF, ce qui nous a conduits à proposer le protocole CARTEE pour combler ces lacunes.

Nous avons commencé par l'étude d'une manière générale à ce type de réseaux avec une description détaillée de leurs caractéristiques et contraintes. Nous nous sommes intéressés aussi aux architectures processeurs proposées pour améliorer les performances des dispositifs de détection. Le défi de transport dans les RCSF et ses principaux problèmes nous ont poussés à chercher les mécanismes exploités dans la résolution de ce problème. Le choix entre ces mécanismes est crucial dans la conception d'un protocole de transport pour les RCSF. Plusieurs travaux de recherche ont été étudiés pour faire un bilan sur les protocoles de transport proposés pour les RCSF. A partir d'une étude préliminaire, on a constaté que chaque solution proposée traite certains aspects de ce problèmes et négligent d'autres qui sont d'une importance capitale.

Le protocole CARTEE a été proposé pour assurer un transport fiable dans les RCSF. Les motivations de conception du protocole ainsi que l'étude théorique derrière les choix des mécanismes utilisés dans ce protocole ont été présentés. Le protocole CARTEE repose sur quatre principaux mécanismes : (1) transmission à base de fenêtre glissante fixe, (2) acquittement implicite/explicite alternative, (3) une technique de détection de congestion originale et (4) un ajustement de taux de transmission distribué. Ces quatre mécanismes qui représentent la clé du succès du protocole CARTEE ont été aussi discutés. A partir d'une évaluation analytique, le protocole CARTEE montre de meilleures performances par rapport aux solutions existantes. Les performances de CARTEE, ont été évaluées à base de simulation. Pour justifier le choix du simulateur utilisé dans l'évaluation, on a présenté une proposition de classification des critères de sélection du simulateur approprié pour faire cette évaluation. Dans cette classification, les critères d'évaluations du simulateur sont structurés d'une manière hiérarchique en partant des objectifs généraux vers l'exploration des objectifs spécifiques. Sur la base de cette classification, l'évaluation des simulateurs utilisés dans les RCSF confirme la crédibilité du simulateur ns-3 par rapport aux autres simulateurs. Après l'évaluation de ces simulateurs, ns-3 a été utilisé pour mesurer les performances du protocole CARTEE, ce qui a mené à la description de l'implémentation du protocole CARTEE sous ns-3. A partir d'une étude comparative des résultats de simulation, il a été observé que CARTEE surpasse les solutions de transport citées dans la littérature en termes de fiabilité, délai de livraison de données, consommation énergétique et occupation de la mémoire.

En conclusion, le nouveau protocole présenté dans cette thèse a fait ses preuves pour répondre aux exigences

des applications multimédia dans les RCSF. Il assure une certaine autonomie au niveau de la couche transport indépendamment des couches inférieures de la pile protocolaire. Donc, cette solution promet d'être largement utilisée par les applications multimédia pour l'échange des grandes quantités de flux de données (i.e., image, voix) dans les RCSF.

Perspectives : Le protocole proposé dans cette thèse ouvre des perspectives et tendances pour le transport fiable des applications multimédia dans les RCSF. Ces applications bénéficient de plusieurs avantages offerts par cette solution, surtout dans le fusionnement des données multimédia. Par exemple, les applications de télésurveillances, où les cameras détectent plusieurs vues dans une région de surveillance, fusionnent les images détectées, on peut établir une scène en 3D de la région surveillée. Ces applications exigent un transport fiable avec un court délai de livraison des données détectées.
Un autre défi qui peut être adressé réside dans le contrôle de contention en conjonction avec la congestion pour réduire la consommation énergétique dans les RCSF. En réalité, ce travail a été déjà abordé et les résultats préliminaires obtenus sont assez prometteurs. Pour améliorer les performances du protocole CARTEE, l'optimisation multi-objectif joue un rôle important dans l'étude théorique de ce protocole. Donc le problème de transport peut être éventuellement abordé comme étant un problème d'optimisation multi-objectif où chaque facteur de performance représente une fonction objective. L'ensemble des fonctions objectives influe sur l'optimisation globale, ce qui démontre l'impact des facteurs de performances sur la fonction objective globale.

BIBLIOGRAPHIE

[1] T. Le, W. Hu, P. Corke, and S. Jha, "Ertp : Energy-efficient and reliable transport protocol for data streaming in wireless sensor networks," *Computer Communications*, vol. 32, no. 7–10, pp. 1154 – 1171, 2009.

[2] F. Stann and J. Heidemann, "Rmst : reliable data transport in sensor networks," in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, pp. 102–112, May 2003.

[3] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, pp. 51–58, May 2000.

[4] I. F. Akyildiz and M. C. Vuran, *Wireless Sensor Networks*. John Wiley & Sons, 2010.

[5] W. Dargie and C. Poellabauer, *Fundamentals of Wireless Sensor Networks : Theory and Practice*. Wiley Publishing, 2010.

[6] C.-Y. Wan, A. T. Campbell, and L. Krishnamurthy, "Psfq : A reliable transport protocol for wireless sensor networks," in *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA '02, (New York, NY, USA), pp. 1–11, ACM, 2002.

[7] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "Coda : Congestion detection and avoidance in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, (New York, NY, USA), pp. 266–279, ACM, 2003.

[8] S. Yogesh, A. Özgür B., and A. I. F., "Esrt : Event-to-sink reliable transport in wireless sensor networks," in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, MobiHoc '03, (New York, NY, USA), pp. 177–188, ACM, 2003.

[9] J. Paek and R. Govindan, "Rcrt : Rate-controlled reliable transport for wireless sensor networks," in *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, (New York, NY, USA), pp. 305–319, ACM, 2007.

[10] A. Benyahia, A. Bilami, and M. Sedrati, "Cartee : Congestion avoidance with reliable transport and energy efficiency for multimedia applications in wireless sensor networks," *Wireless Networks*, 2018.

[11] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*, pp. 15–34. Berlin, Heidelberg : Springer Berlin Heidelberg, 2010.

[12] *ns-3 Manual Release ns-3.29*. ns-3 Project, 2018.

[13] "Unmanned aerial vehicles." `http://www.fas.org/irp/program/collect/`, 2018.

[14] "Camalievineyards." `http://www.camalie.com`, 2018.

[15] L. Yong-Min, W. Shu-Ci, and N. Xiao-Hong, "The architecture and characteristics of wireless sensor network," in *2009 International Conference on Computer Technology and Development*, vol. 1, pp. 561–565, Nov 2009.

[16] G. Liljana, K. Srdjan, M. Veljko, S. Ivan, and T. Roman, *Application and Multidisciplinary Aspects of Wireless Sensor Networks*. New York, NY, USA : Springer, 2011.

[17] "Ieee standard for information technology– local and metropolitan area networks– specific requirements– part 15.4 : Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (wpans)," *IEEE Std 802.15.4-2006 (Revision of IEEE Std 802.15.4-2003)*, pp. 1–320, Sept 2006.

[18] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks : Technology, Protocols, and Applications*. New York, NY, USA : Wiley-Interscience, 2007.

[19] B. Krishnamachari, D. Estrin, and S. B. Wicker, "The impact of data aggregation in wireless sensor networks," in *Proceedings of the 22Nd International Conference on Distributed Computing Systems*, ICDCSW '02, (Washington, DC, USA), pp. 575–578, IEEE Computer Society, 2002.

[20] E. F. Nakamura, H. A. B. F. de Oliveira, L. F. Pontello, and A. A. F. Loureiro, "On demand role assignment for event-detection in sensor networks," in *11th IEEE Symposium on Computers and Communications (ISCC'06)*, pp. 941–947, June 2006.

[21] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8 - Volume 8*, HICSS '00, (Washington, DC, USA), pp. 8020–, IEEE Computer Society, 2000.

[22] S. Lindsey, C. Raghavendra, and K. M. Sivalingam, "Data gathering algorithms in sensor networks using energy metrics," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, pp. 924–935, Sep 2002.

[23] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion : A scalable and robust communication paradigm for sensor networks," in *Proceedings of the 6th Annual International Conference on Mobile Annual International Conference on Mobile Computing and Networking*, Mobicom 2000, pp. 56–67, ACM, 2000.

[24] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," in *Proceedings of the 2Nd ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing*, MobiHoc '01, (New York, NY, USA), pp. 251–254, ACM, 2001.

[25] A. Boukerche, *Algorithms and Protocols for Wireless Sensor Networks*. Wiley-IEEE Press, 2008.

[26] S. Ci, H. Sharif, and K. Nuli, "A ukf-based link adaptation scheme to enhance energy efficiency in wireless sensor networks," in *2004 IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications (IEEE Cat. No.04TH8754)*, vol. 4, pp. 2483–2488 Vol.4, Sept 2004.

[27] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems*, SenSys '04, (New York, NY, USA), pp. 95–107, ACM, 2004.

[28] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies*, Infocom 2002, pp. 1567–1576, June, 2002.

[29] E. Shih, S.-H. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks," in *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, MobiCom '01, (New York, NY, USA), pp. 272–287, ACM, 2001.

[30] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, pp. 16–27, Oct 2000.

[31] K. L. Mills, "A brief survey of self-organization in wireless sensor networks : Research articles," *Wirel. Commun. Mob. Comput.*, vol. 7, pp. 823–834, Sept. 2007.

[32] E. Shih, P. Bahl, and M. J. Sinclair, "Wake on wireless : An event driven energy saving strategy for battery operated devices," in *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*, MobiCom '02, (New York, NY, USA), pp. 160–171, ACM, 2002.

[33] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 12, pp. 493–506, June 2004.

[34] A. Bourmada and A. Bilami, "Cross-layer energy efficient protocol for qos provisioning in wireless sensor network," *International Journal of Systems, Control and Communications (IJSCC)*, vol. 8, no. 3, pp. 230–249, 2017.

[35] B. Sharma and T. C. Aseri, "A hybrid and dynamic reliable transport protocol for wireless sensor networks," *Computers & Electrical Engineering*, vol. 48, pp. 298 – 311, 2015.

[36] M. A. Mahmood, W. K. Seah, and I. Welch, "Reliability in wireless sensor networks : A survey and challenges ahead," *Computer Networks*, vol. 79, pp. 166 – 187, 2015.

[37] S. A. Shah, B. Nazir, and I. A. Khan, "Congestion control algorithms in wireless sensor networks : Trends and opportunities," *Journal of King Saud University - Computer and Information Sciences*, vol. 29, no. 3, pp. 236 – 245, 2017.

[38] L. M. Borges, F. J. Velez, and A. S. Lebres, "Survey on the characterization and classification of wireless sensor network applications," *IEEE Communications Surveys Tutorials*, vol. 16, pp. 1860–1890, Fourthquarter 2014.

[39] D. A. Maltz, *On-Demand Routing in Multi-Hop Wireless Mobile Ad Hoc Networks*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, United States, 2001.

[40] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proceedings of the Seventh Annual International Conference on Mobile Computing and Networking*, pp. 61–69, Jully, 2001.

[41] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building efficient wireless sensor networks with low-level naming," *SIGOPS Oper. Syst. Rev.*, vol. 35, pp. 146–159, Oct. 2001.

[42] A. Woo, T. Tong, and D. Culler, "Taming the underlying challenges of reliable multihop routing in sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, (New York, NY, USA), pp. 14–27, ACM, 2003.

[43] D.-M. Chiu and R. Jain, "Analysis of the increase and decrease algorithms for congestion avoidance in computer networks," *Computer Networks and ISDN Systems*, vol. 17, no. 1, pp. 1 – 14, 1989.

[44] J. hyoung Lee and I. bum Jung, "Reliable asynchronous image transfer protocol in wireless multimedia sensor networks," 2010.

[45] H. Wu and A. A. Abouzeid, "Error robust image transport in wireless sensor networks," in *5th Workshop on Applications and Services in Wireless Networks (ASWN)*, 2005.

[46] H. Wu and A. A. Abouzeid, "Error resilient image transport in wireless sensor networks," *Comput. Netw.*, vol. 50, pp. 2873–2887, Oct. 2006.

[47] S. M. Aziz and D. M. Pham, "Energy efficient image transmission in wireless multimedia sensor networks," *IEEE Communications Letters*, vol. 17, pp. 1084–1087, June 2013.

[48] J. Long, M. Dong, K. Ota, A. Liu, and S. Hai, "Reliability guaranteed efficient data gathering in wireless sensor networks," *IEEE Access*, vol. 3, pp. 430–444, 2015.

[49] F. Yunus, N.-S. Ismail, S. Ariffin, A. Shahidan, N. Fisal, and S. Syed-Yusof, "Proposed transport protocol for reliable data transfer in wireless sensor network (wsn)," in *Modeling, Simulation and Applied Optimization (ICMSAO), 2011 4th International Conference on*, pp. 1–7, April 2011.

[50] I. Lee, W. Shaw, and X. Fan, *Wireless Multimedia Sensor Networks*, pp. 561–582. London : Springer London, 2009.

[51] T. Bonald and J. W. Roberts, "Internet and the erlang formula," *SIGCOMM Comput. Commun. Rev.*, vol. 42, pp. 23–30, Jan. 2012.

[52] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, (New York, NY, USA), pp. 171–180, ACM, 2003.

[53] Q.-M. He, *From the Birth-and-Death Process to Structured Markov Chains*, pp. 155–234. New York, NY : Springer New York, 2014.

[54] E. Zuazua, "Switching control," *Journal of the European Mathematical Society*, vol. 013, no. 1, pp. 85–117, 2011.

[55] M. Guizani, A. Rayes, B. Khan, and A. Al-Fuqaha, *Network Modeling and Simulation : A pratical perspective*. John Wiley & Sons, 2010.

[56] G. Fiche and G. Hébuterne, *Network Modeling and Simulation : A pratical perspective*. Kogan Page Science, 2004.

[57] A. Nath, *Packet Analysis with Wireshark*. Packt Publishing, 2015.

[58] F. Fuentes and D. C. Kar, "Ethereal vs. tcpdump : A comparative study on packet sniffing tools for educational purpose," *J. Comput. Sci. Coll.*, vol. 20, pp. 169–176, Apr. 2005.

[59] M. O. Farooq and T. Kunz, "Operating systems for wireless sensor networks : A survey," *Sensors*, vol. 11, no. 6, pp. 5900–5930, 2011.

[60] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, and D. Culler, *TinyOS : An Operating System for Sensor Networks*, pp. 115–148. Berlin, Heidelberg : Springer Berlin Heidelberg, 2005.

[61] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *29th Annual IEEE International Conference on Local Computer Networks*, pp. 455–462, Nov 2004.

[62] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han, "Mantis os : An embedded multithreaded operating system for wireless micro sensor platforms," *Mobile Networks and Applications*, vol. 10, pp. 563–579, Aug 2005.

[63] A. Eswaran, A. Rowe, and R. Rajkumar, "Nano-rk : an energy-aware resource-centric rtos for sensor networks," in *26th IEEE International Real-Time Systems Symposium (RTSS'05)*, pp. 10 pp.–265, Dec 2005.

[64] Q. Cao, T. Abdelzaher, J. Stankovic, and T. He, "The liteos operating system : Towards unix-like abstractions for wireless sensor networks," in *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pp. 233–244, April 2008.

[65] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim : Accurate and scalable simulation of entire tinyos applications," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, SenSys '03, (New York, NY, USA), pp. 126–137, ACM, 2003.

[66] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, pp. 641–648, Nov 2006.

[67] G. Piro, N. Baldo, and M. Miozzo, "An lte module for the ns-3 network simulator," in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, SIMUTools '11, (ICST, Brussels, Belgium, Belgium), pp. 415–422, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011.

[68] R. J. P. Jalal Nikoukaran, "Simulation software selection "whys and hows"," *The Yugoslav Journal of Operations Research*, vol. 8, no. 15, pp. 93–102, 1998.

[69] J. Nikoukaran, V. Hlupic, and R. J. Paul, "Criteria for simulation software evaluation," in *1998 Winter Simulation Conference. Proceedings (Cat. No.98CH36274)*, vol. 1, pp. 399–406 vol.1, Dec 1998.

[70] J. Heidemann, K. Mills, and S. Kumar, "Expanding confidence in network simulations," *Netwrk. Mag. of Global Internetwkg.*, vol. 15, pp. 58–63, Sept. 2001.

[71] R. Bagrodia, R. Meyer, M. Takai, and J. Martin, "Parsec : a parallel simulation environment for complex systems," *Computer*, vol. 31, pp. 77–85, Oct 1998.

[72] J. H. Cowie, D. M. Nicol, and A. T. Ogielski, “Modeling the global internet,” *Computing in Science and Engg.*, vol. 1, pp. 42–50, Jan. 1999.

[73] P. Huang, D. Estrin, and J. Heidemann, “Enabling large-scale simulations : selective abstraction approach to the study of multicast protocols,” in *Proceedings. Sixth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (Cat. No.98TB100247)*, pp. 241–248, July 1998.

[74] B. D. Lubachevsky, “Recipes for validation,” in *Proceedings of the NIST/DARPA Workshop on Validation of Network Simulations*, WSNA 2002, May 1999.

[75] A. Benyahia and A. Bilami, “Adaptation du protocole tcp pour les réseaux de capteurs sans fil,” tech. rep., Université El Hadj Lakhdar Batna, 2012.

[76] C. Perkins, E. Belding-Royer, and S. Das, “Ad hoc on-demand distance vector (aodv) routing,” 2003.

[77] K. Akkaya and M. Younis, “A survey on routing protocols for wireless sensor networks,” *Ad Hoc Networks*, vol. 3, no. 3, pp. 325 – 349, 2005.

[78] D. Goyal and M. R. Tripathy, “Routing protocols in wireless sensor networks : A survey,” in *2012 Second International Conference on Advanced Computing Communication Technologies*, pp. 474–480, Jan 2012.

[79] D. E. Boubiche and A. Bilami, “Heep (hybrid energy efficiency protocol) based on chain clustering,” *International Journal of Sensor Networks (IJSNet)*, vol. 10, no. 1/2, pp. 25–35, 2011.

[80] I. Minakov, R. Passerone, A. Rizzardi, and S. Sicari, “Routing behavior across wsn simulators : The aodv case study,” in *2016 IEEE World Conference on Factory Communication Systems (WFCS)*, pp. 1–8, May 2016.