

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche
Scientifique

Université A/Mira de Béjaia



جامعة بجاية
Tasdawit n' Bgayet
Université de Béjaïa

Faculté des Sciences Exactes

Département d'Informatique

MÉMOIRE DE FIN DE CYCLE

En vue de l'obtention du diplôme de Master en Informatique

Option : Administration et Sécurité des Réseaux

Thème

***Etude et évaluation des performances des
protocoles de routage pour les réseaux de
capteurs sans fil***

Réalisé par :

M^{elle} HAFIR Latifa

M^{elle} SLIMANI Radhia

Soutenu le 03 Juillet 2016 devant le jury composé de :

Président	M ^r	N. KADJOUH	M.A.B	U. A/Mira Béjaia.
Rapporteur	M ^r	A. CHENNA	M.A.B	U. A/Mira Béjaia.
Examineur	M ^r	K.MEHAOUAD	M.C.B	U. A/Mira Béjaia.

Promotion : 2015/2016

Remerciements

Tout d'abord, nous remercions Dieu le tout-puissant qui nous a donné le courage, la force et la volonté pour mener ce travail.

Un grand merci pour nos familles, surtout nos parents qui nous ont épaulés, soutenus et suivis tout au long de ce projet.

A nos chères amis qui ont toujours été présents et fidèles.

*A notre encadreur **Mr. CHENNA Abdelbassette** pour tout le temps qu'il nous a consacré, pour ces précieux conseils et pour toute son aide et son appui durant la réalisation de ce travail.*

Aussi à tous les enseignants et employés du département Informatique à qui on doit notre avancement.

Enfin, nous tenons aussi à remercier également tous les membres du jury pour avoir accepté d'évaluer notre travail.

Dédicaces

Je dédie ce modeste travail à la mémoire de mon père, que dieu l'accueille en son vaste paradis, père même si vous n'êtes plus là votre existence est éternelle dans mon cœur, je ne vous oublierai jamais.

À ma mère pour sa patience et bienveillance que « dieu la préserve » ,à mes très chères sœurs :Nouna et Biba et son marie Dedin et mon unique frère Adel sans oublié ma très cher nièce Mina .

À mes grands-parents que Dieu les protèges et à mes tantes et mes Oncles et à tous mes amis proches.

Latifa Hafir

Je dédie ce modeste travail à mes aimables et respectueux parents qui ont toujours viellé sur mes études.

À mes chers frères :Lamine et Hassan et à mes sœur Siham et Luiza et son marie Abdeslam .

À mes tantes, mes oncles et à tous mes amis et tous ceux qui me sont chers .

Radhia Slimani

Table des matières

Table des matières	i
Liste des figures	iv
Liste des tableaux	vi
Liste des abréviations	vii
Introduction générale	1
1 Généralités sur les réseaux de capteur	2
1.1 Introduction	2
1.2 Historique des réseaux de capteurs sans fil	3
1.3 Définitions	3
1.3.1 Un capteur	3
1.3.2 Un capteur intelligent	4
1.4 Architecture matérielle d'un capteur	4
1.5 Architecture d'un réseau de capteurs sans fil	6
1.5.1 Les réseaux de capteurs sans fil plats	7
1.5.2 Les réseaux de capteurs sans fil hiérarchiques	7
1.6 Réseaux de capteurs vs réseaux ad hoc	8
1.7 Caractéristique des réseaux de capteurs sans fil	9
1.8 Communication dans les réseaux de capteurs	10
1.9 Système d'exploitation	11
1.10 Domaines d'applications des réseaux de capteurs	12
1.11 Conclusion	16

2	Etat de l'art	17
2.1	Introduction	17
2.2	Définition de routage	17
2.3	Objectifs du routage dans les RCSFs	18
2.4	Les différents protocoles de routage	18
2.4.1	Les protocoles à plat et Data- centric	18
2.4.2	Les protocoles basés sur la localisation	20
2.4.3	Les protocoles hiérarchiques	22
2.4.4	Les protocoles non hiérarchiques	24
2.5	Outils de la simulation réseau	25
2.5.1	Network Simulator 2	26
2.5.2	NS-3 (Network Simulator version 3)	27
2.5.3	OMNet++	28
2.5.4	QualNet	30
2.5.5	OPNET (Optimum Network Performance)	30
2.5.6	GloMoSim (Global Mobile Information System Simulate)	31
2.5.7	JiST (Java in Simulation Time)	32
2.6	Tableau d'utilisation des outils de simulation	32
2.7	Conclusion	33
3	Simulation et évaluations des protocoles OLSR, AODV	34
3.1	Introduction	34
3.2	Etudes des Protocoles OLSR, AODV	34
3.2.1	Le protocole AODV	34
3.2.2	Protocol OLSR	37
3.3	Outils de simulation	39
3.3.1	NS-3 (Network Simulator version 3)	39
3.3.2	Installation de NS-3	39
3.4	Les critères d'évaluation	40
3.5	Interprétation et évaluation des résultats	41
3.6	Résultats de simulations	42
3.7	Conclusion	48
	Conclusion générale	49
	Bibliographie	50

A Annexe

54

LISTE DES FIGURES

1.1	Exemple de capteur sans fil.	4
1.2	L'architecture matérielle d'un capteur.	5
1.3	Architecture d'un réseau de capteurs sans fil.	7
1.4	Réseaux de capteurs sans fil plats.	7
1.5	Réseaux de capteurs sans fil hiérarchiques.	8
1.6	La Pile protocolaire d'un réseau de capteur.	11
1.7	Un service militaire utilisant les RCSF.	13
1.8	Application des RCSF dans le domaine sécurité	13
1.9	Application des RCSF dans le domaine environnementale.	14
1.10	Application des RCSF en médecine.	15
2.1	: le protocole SPIN.	19
2.2	Phases du protocole de diffusion dirigée.	20
2.3	Le routage hiérarchique	23
2.4	Algorithm de routage LEACH	23
2.5	La structure de NS-2	27
2.6	L'architecture de Ns-3	28
2.7	L'architecture d'OMNET++	29
3.1	Recherche de la destination par la requête RREQ et la réponse par la requête RREP	37
3.2	Diffusion pure et diffusion en utilisant les MPR s dans OLSR	38
3.3	Le débit moyen.	43
3.4	Le délai de bout en bout	43
3.5	Nombre de paquet perdu.	44
3.6	Taux de livraison des paquets.	45
3.7	Le débit moyen.	46

3.8	Le délai de bout en bout	47
3.9	Nombre de paquet perdus	47
3.10	Taux de livraison des paquets	48

LISTE DES TABLEAUX

1.1	Les trois générations des nœuds de capteurs [3].	3
1.2	Comparaison entre les réseaux de capteurs et les réseaux ad-hoc[5].	8
2.1	Composants de NS-2	27
2.2	Composants d'OMNET++ :	30
2.3	Cas d'utilisation des simulateurs	32
3.1	Les paramètres a utilisé dans les différents scénarios.	42

LISTE DES ABRÉVIATIONS

ACA	A utonomous C omponent A rchitecture
AODV	W ireless S ensors N etworks
ASIC	A pplication S pecific I ntegrated C ircuit
CAN	C onvertisseur A nalogique N umérique
CH	C luster H ead
CPU	C ontroler P rocessor U nit
DSP	D igital S ignal P rocessors
DSR	D ynamic S ource R outing
FPGA	F ield P rogrammable G ate A rray
FSR	F isheye S tate R outing
GAF	G eographic A daptive F idelity
GEAR	G eographic and E nergy- A ware R outing
GPS	G lobal P ositioning S ystem
IETF	I nternet E ngineering T ask F orce
IP	I nternet p rotocol
J-SIM	J ava S imulator
LEACH	L ow E nergy A daptive C lustering H ierarchy
MPR	M ultiPoint R elaying
MECN	M inimum E nergy C ommunication N etwork
NS	N etwork S imulator
ODMRP	O n D emand M ulticast R outing P rotocol
OLSR	O ptimized L ink S tate R outing)
OSI	O pen S ystem I nterconnection
PEGASIS	P ower-Efficient G athering in S ensor I nformation S ystems
RCSF	R éseau de C apteurs S ans F il
RF	R adio F réquence
RREP	R oute R eplay

RREQ	R oute R equest
SMECN	S mall M inimum E nergy C ommunication N etwork
SPIN	S ensor P rotocols I nformation N égociation
TCL	T ool C ommand L anguage
TCP	T ransmission C ontrol P rotocol
UDP	U ser D ata P rotocol
WRP	W ireless R outing P rotocols
WSN	W ireless S ensors N etworks

Introduction générale

Les réseaux de capteurs sans fil sont considérés comme un type spécial de réseaux ad hoc, composés d'un grand nombre de capteurs matériellement petits, et placés généralement près des objets auxquels ils s'intéressent dans les environnements où ils sont déployés. Ces capteurs sont capables de récolter, traiter et acheminer les données environnementales de la région surveillée d'une manière autonome, vers des stations de collecte appelées nœuds puits ou stations de base.

Notre travail entre dans le cadre de l'étude du problème de routage dans les réseaux de capteurs sans fil. Le routage est une méthode d'acheminement des informations vers la bonne destination à travers un réseau de connexion donnée. Il consiste à assurer une stratégie qui garantit, à tout moment, un établissement de routes qui soient correctes et efficaces entre n'importe quelle paire de nœuds appartenant au réseau. L'objectif du routage de l'information est d'assurer l'échange des messages d'une manière continue.

Vu les limitations des réseaux de capteurs, la construction des routes doit être faite avec un minimum de contrôle et de consommation de la bande passante. Il est important de minimiser à la fois les délais d'acheminement et les pertes des messages échangés entre les nœuds. Dans ce contexte, le réseau doit être robuste et avoir un temps d'attente très court et doit pouvoir transférer rapidement les messages.

L'objectif du projet est d'explorer un certain nombre de protocoles de routage dédiés aux réseaux ad hoc, afin d'étudier leur adaptabilité aux réseaux de capteur sans fil.

La suite de ce mémoire sera organisée comme suit : En premier lieu, on présente quelques généralités sur les réseaux de capteurs sans fil. Ensuite, on présentera un état de l'art sur les protocoles de routage dans les RCSFs ainsi que les outils de simulation. Pour finir par une simulation de ces protocoles afin d'évaluer leur performances.

1

Généralités sur les réseaux de capteur

1.1 Introduction

Dans le monde industriel, un capteur est un dispositif qui transforme une grandeur physique observée (température, pression, etc.) en une grandeur utilisable (intensité électrique, position d'un flotteur). Grace aux avancées technologiques de ces dernières années, principalement dans le domaine de la miniaturisation, les capteurs sont devenus des éléments de très petite taille. Ils sont ainsi dotés de moyens leur permettant : de stocker les résultats de leurs observations, d'effectuer un certain nombre de traitement sur ces résultats et de les transmettre au monde réel via une communication sans fil.

Toutes ces caractéristiques apportent au monde scientifique l'opportunité de déployer les capteurs à grande échelle dans des milieux, parfois difficiles voire, impossible d'y avoir l'accès, pour surveiller, collecter et transmettre les données collectées à un système capable de les analyser et prendre (si possible ou nécessaire) les décisions. Ce déploiement aboutit à un réseau de capteurs sans fils (RCSF), ou Wireless Sensor Network(WSN) en anglais.

Pour leurs caractéristiques très variées et leur faible coût de production, les RCSFs sont très utilisés dans plusieurs domaines, allant du domaine militaire au domaine médical,

en passant par l'industrie, l'écologie, la domotique, l'agriculture de précision, etc. Pour cela, les RCSFs forment un domaine de recherche très vaste et en pleine croissance. Ainsi, mener un travail de recherche dans ce domaine nécessite la connaissance d'un certain nombre de concepts généraux.

Dans la suite de ce chapitre, nous verrons ces concepts à travers un certains nombre de points et un plan méthodologique que nous avons adopté.

Nous commencerons par donner quelques définitions d'un capteur, l'architecture matérielle et voir comment ces capteurs sont déployés pour former des réseaux de capteurs sans fil, qui seront comparés aux réseaux ad hoc classiques. Ensuite, les caractéristique et la communication des RCSFs ainsi que les domaines d'applications des ces réseaux seront étudiés.

1.2 Historique des réseaux de capteurs sans fil

Les récents progrès des nouvelles techniques ont provoqué une énorme importance dans le domaine des réseaux sans fil. La technologie des réseaux de capteurs sans fil est devenue une des merveilleuses technologies dans le 21ème siècle , les réseaux de capteurs ont montré leur impact sur notre vie quotidienne. Le tableau suivant illustre l'évaluation des réseaux de capteurs.

Génération	Période	Taille	Poids	Batterie
1ère	Les années 80 et 90	Grande boîte à chaussures	Kilogrammes	Grosse
2ème	Entre 2000 et 2003	Boîte de cartes	Grammes	AA
3ème	2010	Particule de poussière	Négligéable	Solaire

TAB. 1.1 – Les trois générations des nœuds de capteurs [3].

1.3 Définitions

1.3.1 Un capteur

Un capteur est le dispositif qui transforme une grandeur physique observée (température, pression, humidité, etc.) en une grandeur utilisable (intensité électrique, position d'un flotteur) [1]. Pour cela, il possède au moins un transducteur dont le rôle est de convertir une grandeur physique en une autre.

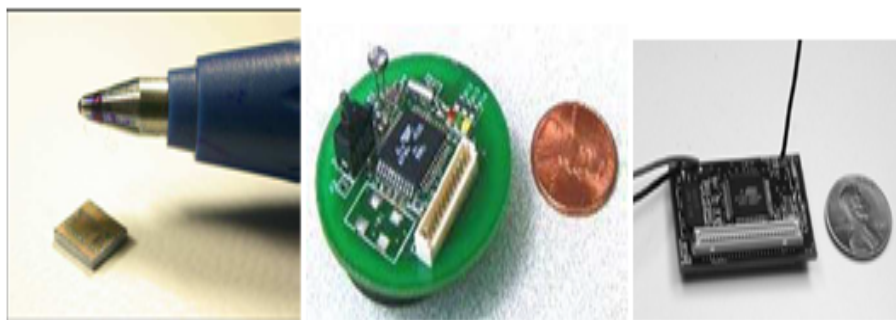


FIG. 1.1 – Exemple de capteur sans fil.

1.3.2 Un capteur intelligent

Le terme capteur intelligent (smart sensor ou intelligent sensor) a été utilisé dans l'industrie des capteurs pour désigner des capteurs qui ne fournissent pas seulement des mesures, mais aussi une fonctionnalité aux mesures spécifiques. Par rapport à un capteur classique, un capteur intelligent intègre de nombreux éléments électroniques additionnels, ainsi que des unités programmables et des aspects logiciels nécessaires au traitement des données, aux calculs, à la communication numérique. Il est donc caractérisé par sa capacité à effectuer une collecte des mesures, les traiter et à les communiquer au monde extérieur [2].

1.4 Architecture matérielle d'un capteur

Un capteur est composé principalement d'une unité de : captage, traitement, stockage, communication, et énergie. Des composants additionnels peuvent être ajoutés selon le domaine d'application, comme par exemple un système de localisation tels qu'un GPS (Global Positioning System), un générateur d'énergie (exemple : cellules solaires) ou un mobilisateur lui permettant de se déplacer. Ces éléments principaux et optionnels (représentés par les traits discontinus) sont visibles sur la figure 1.2.

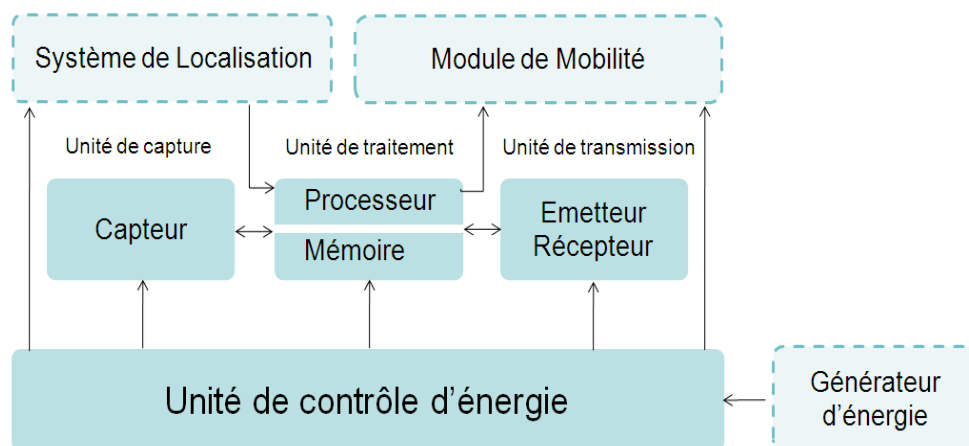


FIG. 1.2 – L'architecture matérielle d'un capteur.

1. Unité d'énergie

Un capteur est muni d'une source d'énergie, généralement une batterie, pour alimenter tous ses composants. Les batteries utilisées sont soit rechargeables ou non. Souvent, dans les environnements sensibles, il est impossible de recharger ou changer une batterie. Pour cela, l'énergie est la ressource la plus précieuse puisqu'elle influe directement sur la durée de vie des capteurs et donc d'un réseau de capteur [3].

2. Unité de captage

L'unité de captage permet de capturer ou mesurer les données physiques à partir de l'objet cible. Elle est composée de 2 sous-unités : le récepteur (reconnaissant la grandeur physique à capter) et le transducteur (convertissant le signal du récepteur en signal électrique). Le capteur fournit des signaux analogiques au convertisseur Analogique/Numérique(CAN) qui transforme ces signaux en données numérique et les transmet à l'unité de traitement. Un capteur peut avoir un ou plusieurs unités de captage[4].

3. Unité de traitements (processeur)

Il recueille des données de l'unité de captage ou d'autres capteurs, effectue un traitement sur ces données et décide quand et où les envoyer. Les types de processeurs qui peuvent être utilisés dans un capteur incluent le microcontrôleur, les DSP (Digital Signal Processors), les FPGA (Field Programmable Gate Array) et les ASIC (Application Specific Integrated Circuit).[4].

4. Unité de transmission

Cette unité est responsable de toutes les émissions et réceptions de données via un support radiofréquence(RF), le laser et l'infrarouge [4].

1.5 Architecture d'un réseau de capteurs sans fil

Un réseau de capteur sans fil(RCSF), consiste en un ensemble de nœuds capteurs variantes de quelques dizaines d'éléments à plusieurs milliers, placés de manière, plus ou moins aléatoire, dans une zone géographique appelée zone de captage, ou zone d'intérêt, afin de surveiller un phénomène physique et de récolter leurs données d'une manière autonome.

Les nœuds capteurs utilisent une communication sans fil pour acheminer les données captées vers un nœud collecteurs appelé nœud puits (sink en anglais), ou par satellite à l'ordinateur central (Gestionnaire de tâches) pour analyser ces données et prendre des décisions. Ainsi, l'utilisateur peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises, puis récolter les données environnementales captées par le biais du nœud puits.

En plus des nœuds capteurs, le modèle peut introduire les super-nœuds, appelés des passerelles(Gatways) [6]. Ces derniers possèdent une source d'énergie importante, la capacité de traitement et de stockage plus élevées comparativement aux nœuds capteurs. Ils peuvent ainsi être utilisés pour exécuter les tâches plus complexes comme la fusion des données issues des capteurs d'une même zone.

Dans le cas le plus simple, les nœuds capteurs seront dans le voisinage direct du puits (communication à un a un saut). Cependant, dans le cas d'un réseau à grande échelle, ils ne sont pas tous dans le voisinage du puits et les données seront acheminées du nœud source vers le puits en transitant par plusieurs nœuds, selon un mode de communication multi-sauts comme l'illustre la figure 1.3 ci-après.

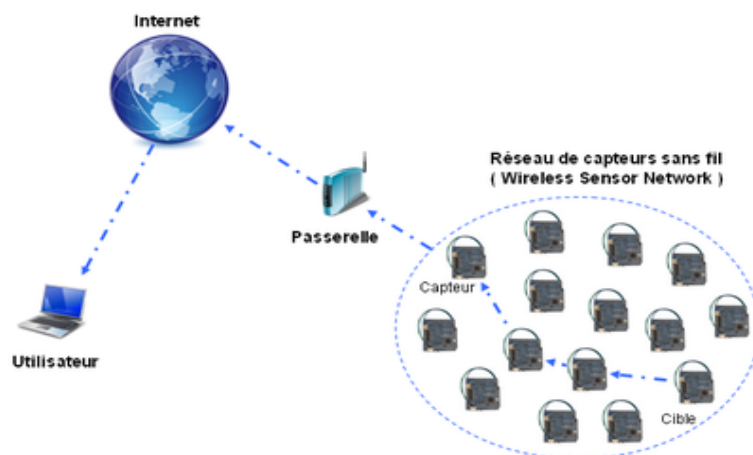


FIG. 1.3 – Architecture d'un réseau de capteurs sans fil.

1.5.1 Les réseaux de capteurs sans fil plats

Un réseau de capteur sans fil plats est un réseau homogène, où tous les nœuds disposent des mêmes capacités et fonctionnalités concernant le captage, la communication et la complexité du matériel, excepté le " Sink ". Le Sink joue le rôle d'une passerelle, il est responsable de la transmission de l'information collectée à l'utilisateur final.

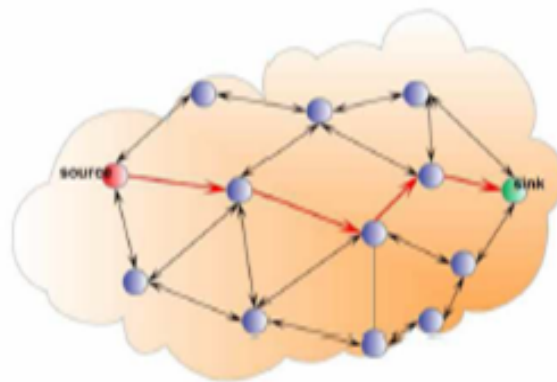


FIG. 1.4 – Réseaux de capteurs sans fil plats.

1.5.2 Les réseaux de capteurs sans fil hiérarchiques

Une architecture hiérarchique a été proposée pour réduire le coût et la complexité de la plupart des nœuds capteurs. Elle consiste à introduire un ensemble des nœuds plus coûteux et plus puissants, en créant une infrastructure qui décharge la majorité des nœuds simples à faible coût de plusieurs fonctions du réseau.

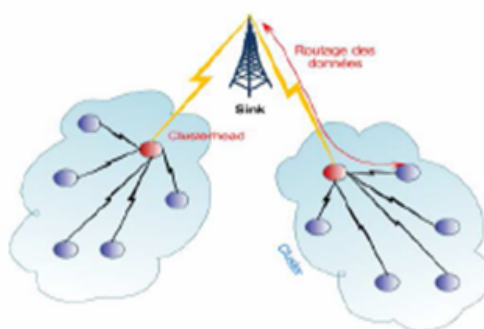


FIG. 1.5 – Réseaux de capteurs sans fil hiérarchiques.

1.6 Réseaux de capteurs vs réseaux ad hoc

Le tableau suivant résume les similitudes et les différences entre les réseaux de capteurs et les réseaux ad hoc ordinaires.

Réseau de capteurs sans fil	Réseau MANET ¹
Utilisation d'un médium sans fil.	Utilisation d'un médium sans fil.
Déploiement Ad Hoc.	Déploiement Ad Hoc.
Robuste aux pannes des nœuds (Auto-organisation).	Robuste aux pannes des nœuds (Auto-organisation)
Routage multi-saut.	Routage multi-saut.
La mobilité des nœuds est restreinte.	Mobilité des nœuds.
Des petits nœuds plus susceptibles aux pannes, avec moins capacité de traitements et de stockage.	Des nœuds ayant plus de capacité de traitements et de stockage.
Data-centric : souvent pas d'adresses uniques, les enquêtes sont envoyés à tous les nœuds.	Adress-centric : une adresse unique pour chaque nœud utilisée pour réaliser la communication entre les nœuds.
Flot de données " Many-to-one ".	Flot de données " Any-to-any ".
L'énergie est un facteur déterminant	Débit est majeur.
Nœuds collaborent pour remplir un objectif commun.	Chaque nœud a son propre objectif.
Les nœuds agrègent les données avant de les transmettre	Généralement pas d'agrégation de données.

TAB. 1.2 – Comparaison entre les réseaux de capteurs et les réseaux ad-hoc[5].

¹Mobile Ad hoc NETWORK

1.7 Caractéristique des réseaux de capteurs sans fil

Caractéristique des réseaux de capteurs sans fil se résumant en ce qui suit :

1. La densité des nœuds :

Les réseaux de capteurs se composent généralement d'un nombre très important de nœuds pour garantir une couverture totale de la zone à surveiller. Cette densité peut varier de 0,05 à 1 nœuds /m².

2. La durée de vie limitée :

Les nœuds capteurs sont très limités par la contrainte d'énergie, ils fonctionnent généralement sans surveillance dans les régions géographiques éloignées et par conséquent recharger et remplacer leur batterie est la plus part du temps impossible.

3. Communication multi-saut :

Les ressources physiques limitées des nœuds capteurs leur imposent des portées de transmission réduite. Ce qui oblige les nœuds à router les informations à travers les autres nœuds pour atteindre le nœud puits. En plus la haute densité de déploiement des capteurs favorise l'utilisation de cette communication multi-saut qui consomme moins d'énergie que la communication traditionnelle à un seul saut.

4. Ressource limitées :

Les nœuds capteurs ont une taille très petite. Ce petit facteur de forme limite la quantité de ressources qui peut être mises dans ces nœuds et par conséquent la capacité de traitement et de mémoire sont très limitées.

5. Bande passante limitée :

La bande passante réservée à chacun des nœuds capteurs est modeste en raison du partage du média de communication sans fil entre ces derniers. En conséquence, les nœuds capteurs ne peuvent pas supporter des débits élevés (10 Kbits/s/nœuds).

6. Absence d'adressage unique :

Les nœuds dans les réseaux sans fil classiques sont identifiés par des adresses IP. Cependant, cette notion n'existe pas dans les RWSN. Ces derniers utilisent un adressage basé sur l'attribut du phénomène capté, on parle donc de l'adressage basé-attribut. En effet, les requêtes des utilisateurs ne sont pas généralement destinées à un seul nœud, mais plutôt, à un ensemble de nœuds identifiés par un attribut. Par exemple, identifier un ensemble de nœuds par " les nœuds qui captent le volume de CO₂ dépassant 0,0375% dans l'atmosphère".

7. Topologie dynamique :

L'instabilité de la topologie des RCSF est le résultat des trois facteurs essentiels suivants : La mobilité des nœuds . La défaillance des nœuds. Rajout de nouveaux nœuds.

8. Auto-organisation :

Vu les différentes raisons qui provoquent une topologie instable du réseau de capteurs, ce dernier devra être capable de s'auto-organiser pour continuer ses applications. Cela voudrait dire que chaque nœud doit être capable de localiser ses voisins et établir des routes pour acheminer les paquets au sein du réseau.

9. Sécurité limitée :

Les principales caractéristiques des réseaux de capteurs rendent la sécurité très difficile à assurer .En effet, étant basés sur un déploiement hostiles, les réseaux de capteurs sont plus sensibles aux attaques physiques. En plus, la communication sans fil est plus vulnérable que les liaisons filaires .

10. La qualité de service (QoS) :

La QoS pour les RCSFs est que la quantité et la qualité d'informations extraites à partir des puits deviennent appropriées. Le niveau de QoS est défini par un ensemble d'attributs comme le temps d'attente, la largeur de bande, et la perte de paquets qu'on relie directement avec le type de service du réseau.

1.8 Communication dans les réseaux de capteurs

Dans les réseaux de capteurs, les nœuds doivent bien réagir avec l'environnement où ils sont placés. Ainsi, ils doivent permettre une communication multi-saut pour les données qui circulent dans la zone de capture. Pour cela, un modèle de communication est proposé par Ian et al dont le rôle principal est la standardisation de la communication entre les participants afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles. Ce combiné l'énergie et le routage, intègre les données avec les protocoles réseaux et permet une communication efficace entre les différents nœuds à travers un medium sans fil.

- **Pile protocolaire (modèle en couches)**

Par analogie au modèle OSI (Open Système Interconnexion) des réseaux filaires, le modèle de communication utilisé dans les réseaux de capteurs comprend cinq couches qui ont les mêmes fonctions que celles du modèle OSI. En plus de trois niveaux (plans) transverses qui sont :

- Le niveau de gestion d'énergie : chargé de contrôler la manière dont un nœud utilise son énergie.

- Le niveau de gestion des tâches : assure l'équilibrage de la distribution des tâches sur les différents nœuds pour accomplir un travail coopératif.

- Le niveau de gestion de la mobilité : détecte et enregistre tous les mouvements des nœuds capteurs.

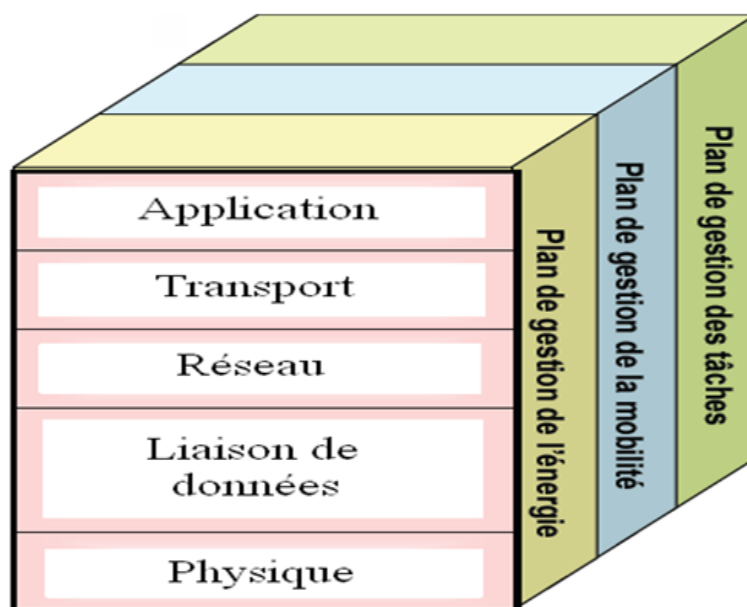


FIG. 1.6 – La Pile protocolaire d'un réseau de capteur.

1.9 Système d'exploitation

Le système d'exploitation (en anglais Operating System ou OS) est un ensemble de programmes responsables de la liaison entre les ressources matérielles d'un dispositif et les applications de l'utilisateur. Comme exemple d'OS pour les réseaux de capteurs nous pouvons citer :

1. **TinyOS (Tiny Operating System)** : est un système d'exploitation "open source" pour les réseaux de capteurs conçu par l'université américaine de BERKELEY. Sa

conception a été entièrement réalisée en NesC², langage orienté composant syntaxiquement proche du C. Il repose sur [8] :

- Une architecture basée composant.
- Un modèle de programmation basé évènement.
- Un modèle de concurrence basé sur des évènements et des tâches.

La bibliothèque de composants de TinyOS est particulièrement complète puisqu'on y retrouve des protocoles réseaux, des pilotes de capteurs et des outils d'acquisition de données. Un programme s'exécutant sur TinyOS est constitué d'une sélection de composants systèmes et de composants développés spécifiquement pour l'application à laquelle il sera destiné (mesure de température, du taux d'humidité...).

2. **Think** : C'est une implémentation du modèle Fractal en C [8]. Il est développé par l'INRIA³ et France Télécom pour :

- Créer des systèmes d'exploitation pour les systèmes embarqués.
- Créer les applications s'exécutant dessus.

Il repose sur une utilisation plus large de l'ingénierie logicielle basée composant (aspect dynamique) et propose une gestion des aspects non fonctionnels via des contrôleurs. Contrairement à TinyOS, il permet l'allocation dynamique de ressources.

1.10 Domaines d'applications des réseaux de capteurs

Le faible coût de production et l'usage de dispositifs miniaturisés et à faible consommation dans les réseaux de capteurs a révélé l'idée de leurs utilisations dans des domaines très variés. Les multiples applications envisagées dans ces réseaux sont souvent classés en cinq familles [7] :

1. Applications militaires

Comme dans le cas de plusieurs technologies, le domaine militaire a été un moteur initial pour le développement des réseaux de capteurs. Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui rendent ce type de réseaux un outil appréciable dans un tel domaine.

²Est un langage conçu pour incarner les concepts structurant et le modèle d'exécution de TinyOS. C'est une extension du langage C orientée composant

³Institut National de Recherche en Informatique et en Automatique



FIG. 1.7 – Un service militaire utilisant les RCSF.

Comme exemple d'application dans ce domaine, on peut penser à un réseau de capteurs déployé sur un endroit stratégique ou difficile d'accès, afin de surveiller toutes les activités des forces ennemies, ou d'analyser le terrain avant d'y envoyer des troupes (détection d'agents chimiques, biologiques ou de radiations). Des tests concluants ont déjà été réalisés dans ce domaine par l'armée américaine dans le désert de Californie.

2. Applications à la sécurité

Les altérations dans la structure d'un bâtiment, suite à un séisme ou au vieillissement, pourraient être détectées par des capteurs intégrés dans les murs ou dans le béton, sans alimentation électrique ou autres connexions filaires. Les capteurs doivent s'activer périodiquement et peuvent ainsi fonctionner durant des années, voire des décennies. Un réseau de capteurs de mouvements peut constituer un système d'alarme distribué qui servira à détecter les intrusions sur un large secteur. Déconnecter le système ne serait plus aussi simple, puisque il n'existe pas de point critique. La surveillance de voies ferrées pour prévenir des accidents avec des animaux et des êtres humains peut être une application intéressante des réseaux de capteurs.

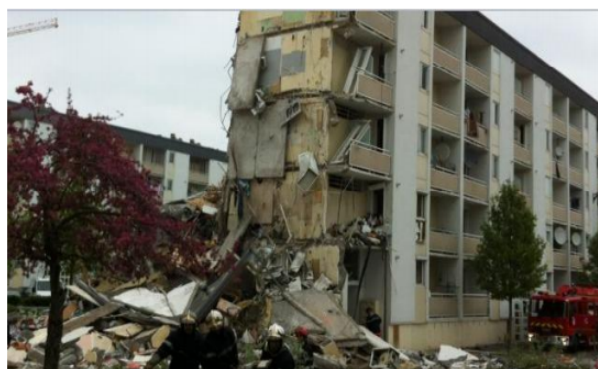


FIG. 1.8 – Application des RCSF dans le domaine sécurité .

La protection des barrages pourrait être accomplie en y introduisant des capteurs.

La détection de fuites d'eau permettrait d'éviter des dégâts. Les êtres humains sont conscients des risques et attaques qui les menacent. Du coup, ils mettent à disposition toutes les ressources humaines et financières nécessaires pour leur sécurité. Cependant, des failles sont toujours présentes dans les mécanismes de sécurisation appliqués aujourd'hui, sans oublier leur coût très élevé. L'application des réseaux de capteurs dans le domaine de la sécurité pourrait diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et à la protection des êtres humains tout en garantissant des meilleurs résultats.

3. Applications environnementales

Des thermo-capteurs dispersés à partir d'un avion sur une forêt peuvent signaler un éventuel début d'incendie dans le champ de captage, ce qui permettra une meilleure efficacité pour la lutte contre les feux de forêt. Dans les champs agricoles, les capteurs peuvent être semés avec les graines. Ainsi, les zones sèches seront facilement identifiées et l'irrigation sera donc plus efficace. Sur les sites industriels, les centrales nucléaires ou dans les pétroliers, des capteurs peuvent être déployés pour détecter des fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.) et alerter les utilisateurs dans un délai suffisamment court pour permettre une intervention efficace.



FIG. 1.9 – Application des RCSF dans le domaine environnementale.

Une grande quantité de capteurs peut être déployée en forêt ou dans un environnement de conservation de la faune afin de recueillir des informations diverses sur l'état du milieu naturel et sur les comportements de déplacement. Par exemple, l'université de Pise en Italie a réalisé des réseaux de capteurs pour le contrôle des parcs naturels (feux, animaux,..). Il est ainsi possible "d'observer", sans déranger, des espèces animales difficiles à étudier dans leur environnement naturel et de proposer des solutions plus efficaces pour la conservation de la faune. Les éventuelles conséquences de la dispersion en masse des micro-capteurs dans l'environnement ont soulevé plusieurs inquiétudes.

En effet, chaque micro-capteur est doté d'une batterie qui contient des métaux nocifs. Néanmoins, le déploiement d'un million de capteurs de 1 millimètre cube chacun ne représente qu'un volume total d'un litre. Même si tout ce volume était constitué de batteries, cela n'aurait pas des répercussions désastreuses sur l'environnement.

4. Applications médicales

On pourrait imaginer que dans le futur, la surveillance des fonctions vitales de l'être humain serait possible grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau. Actuellement, des micro-caméras qui peuvent être avalées existent. Elles sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain avec une autonomie de 24 heures. Les auteurs d'une récente étude, présentent des capteurs qui fonctionnent à l'intérieur du corps humain pour traiter certains types de maladies. Leur projet actuel est de créer une rétine artificielle composée de 100 micro-capteurs pour corriger la vue.

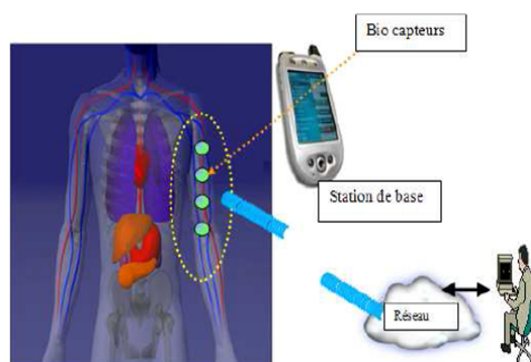


FIG. 1.10 – Application des RSCF en médecine.

D'autres ambitieuses applications biomédicales sont aussi présentées, tel que : la surveillance du niveau de glucose, le monitoring des organes vitaux ou la détection de cancers. L'utilisation des réseaux de capteurs dans le domaine de la médecine pourrait apporter une surveillance permanente des patients et une possibilité de collecter des informations physiologiques de meilleure qualité, facilitant ainsi le diagnostic de quelques maladies.

5. Applications commerciales

Il est possible d'intégrer des nœuds capteurs au processus de stockage et de livraison. Le réseau ainsi formé, pourra être utilisé pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Il devient alors possible pour un client qui attend la réception d'un paquet, d'avoir un avis de livraison en temps réel et de connaître la position actuelle du paquet. Pour les entreprises manufacturières, les ré-

seaux de capteurs permettront de suivre le procédé de production à partir des matières premières jusqu'au produit final livré. Grâce aux réseaux de capteurs, les entreprises pourraient offrir une meilleure qualité de service tout en réduisant leurs coûts. Dans les immeubles, le système de climatisation peut être conçu en intégrant plusieurs micro-capteurs dans les tuiles du plancher et les meubles. Ainsi, La climatisation pourra être déclenchée seulement aux endroits où il y a des personnes présentes et seulement si c'est nécessaire. Le système distribué pourra aussi maintenir une température homogène dans les pièces. Utilisée à grande échelle, une telle application permettrait de réduire la demande mondiale en énergie réduisant du même coup les gaz à effet de serre. Rien que pour les Etats-Unis, on estime cette économie à 55 milliards de dollars par an avec une diminution de 35 million de tonnes des émissions de carbone dans l'air. Ainsi, dans un contexte mondial où le réchauffement de la planète devient une préoccupation grandissante, une telle conséquence environnementale serait un pas dans la bonne direction.

1.11 Conclusion

Un réseau de capteurs est une collection de capteurs déployés dans une zone de capture pour prélever des mesures physiques. Dans ce chapitre, nous avons introduit ce type de réseau, en particulier, nous avons présenté l'architecture, les caractéristiques, le système d'exploitation, et la communication des réseaux de capteurs, ainsi que les domaines d'applications de ce genre de réseau. Dans ce qui suit nous présenterons l'état de l'art sur les réseaux de capteurs.

2

Etat de l'art

2.1 Introduction

Dans le cadre des réseaux de capteurs, le routage doit être efficace en énergie. Pour cela, il faut bien sûr être capable de trouver une route qui ne coûte pas trop d'énergie, une route pas trop longue. Mais il faut aussi être capable de trouver ou de maintenir les routes sans dépenser trop d'énergie. Nous définirons tout d'abord ce qu'est le routage dans un réseau de capteur et les différents protocoles de routage ainsi que les outils de simulation.

2.2 Définition de routage

Le routage est une méthode d'acheminement des informations à la bonne destination à travers un réseau de connexion donné. Le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau au sens d'un certains critères de performance. Le problème consiste à trouver l'investissement de moindre cout en capacité est nominales et de réserves qui assure le routage du trac nominal et garantit sa survie en cas de n'importe quelle panne d'arc ou de nœud. Toute conception de protocole de routage doive étudier les problèmes suivants :

1. La minimisation de la charge du réseau.
2. Offrir un support pour pouvoir effectuer des communications multipoints fiables.
3. Assurer un routage optimal.
4. Offrir une bonne qualité concernant le temps de latence suivant la manière de création et de maintenance de routes lors de l'acheminement des données.

2.3 Objectifs du routage dans les RCSFs

Les soucis majeurs du routage est la conservation d'énergie qui peut étendre considérablement la durée de vie d'un RCSF. Un bon routage doit être capable d'acheminer les informations avec les critères suivantes :

- Une latence faible.
- En utilisant peu d'énergie, donc en utilisant les meilleures liaisons de données.
- En minimisant la charge de réseau, afin de maximiser la qualité total d'informations pouvant être transmises sur le réseau dans son ensemble.
- En étant faible, et en particulier, en étant tolérant aux pannes de nœuds du réseau.

2.4 Les différents protocoles de routage

Les protocoles de routage sont en fait découpés en quatre familles : les protocoles de routage à plat et hiérarchiques et non hiérarchiques ,localisation.

2.4.1 Les protocoles à plat et Data- centric

Le routage à plat et données centrales [9, 10, 11, 12, 13, 14, 15], est le modèle le plus simple où chaque nœud dans le réseau transmet à la station de base. Chaque nœud joue typiquement le même rôle et les nœuds capteurs collaborent pour accomplir la tâche de capteur. La station de base envoie des requêtes à certaines régions et se met en attente des données des capteurs situés dans les régions choisies. On peut citer comme exemple :

2.4.1.1 Propagation et discussion (flooding and gossiping)

Dans la propagation, chaque capteur recevant un paquet de données le renvoi à tous ses voisins. Ce processus continue jusqu'à ce que le paquet arrive à destination ou jusqu'à ce que le nombre maximum de sauts pour le paquet soit atteint. D'autre part, le gossiping est une version légèrement améliorée du flooding où le nœud récepteur envoie le paquet à un voisin choisi aléatoirement[10].

2.4.1.2 Protocoles de capteur pour l'information par négociation (SPIN)

L'idée derrière le SPIN est de nommer les données sans utilisant des descripteurs de haut niveau ou des méta données. Avant la transmission, les méta- données sont échangées entre les capteurs par un mécanisme de publicité de données. Chaque nœud recevant de nouvelles données, l'annonce à ses voisins et ceux intéressés récupèrent les données en envoyant une requête (figure 2.1)[10].

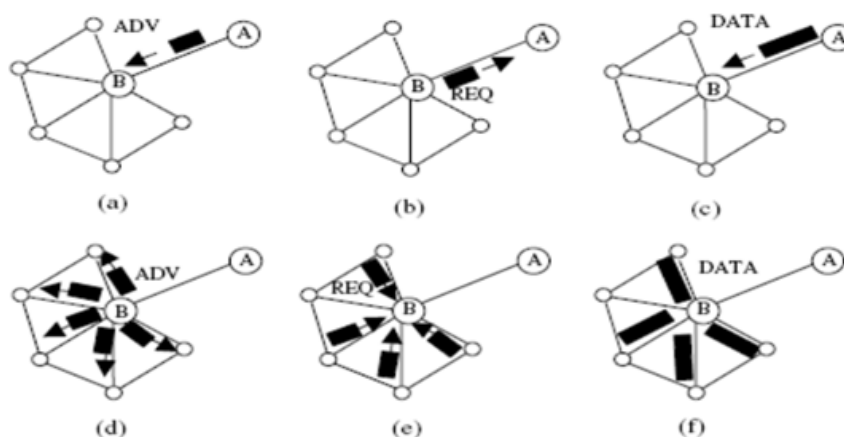


FIG. 2.1 – : le protocole SPIN.

2.4.1.3 Routage par rumeur

L'idée est de transmettre les requêtes aux nœuds qui ont observé un évènement particulier. Quand un nœud détecte un évènement, il l'ajoute à sa table locale et génère un agent. L'agent parcourt le réseau afin de propager des informations sur les évènements locaux aux nœuds distants. Quand un nœud génère une requête pour un évènement, les nœuds qui connaissent l'itinéraire peuvent répondre en référant la table d'évènements[11].

2.4.1.4 Diffusion dirigée

Une requête intérêt est définie par une liste de couples attribut-valeur et est diffusée par le sink vers ses voisins. L'intérêt contient aussi plusieurs champs gradient. Un gradient est un lien de réponse de la part du voisin recevant l'intérêt. En utilisant les intérêts et les gradients [16], plusieurs chemins peuvent être établis entre le sink et la source. L'un de ces chemins est sélectionné par renforcement (figure 2). Si ce chemin échoue, un nouveau ou un alternatif doit être identifié, puisque les données sont demandées par des requêtes. En outre, cette méthode est coûteuse en termes de consommation d'énergie et ne représente pas un bon modèle pour WSN.

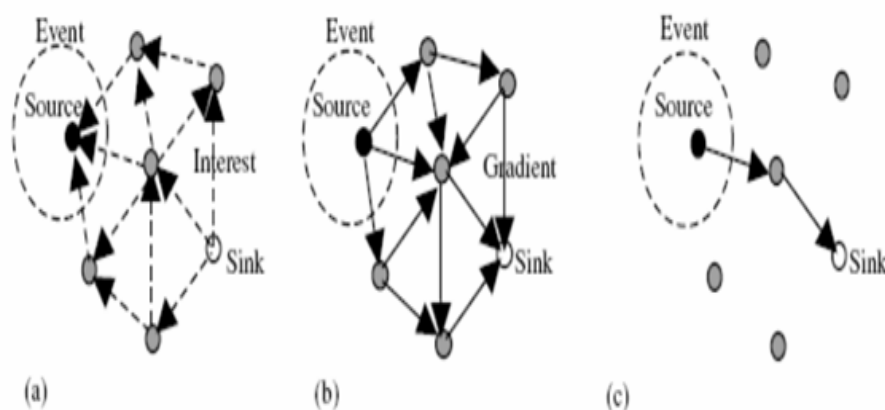


FIG. 2.2 – Phases du protocole de diffusion dirigée.

Les avantages et les inconvénients

Avantage :

- L'agrégation des données utilisées par les protocoles de ce type conserve une quantité considérable d'énergie.
- Le fait que tous les nœuds d'un réseau plat ont le même rôle et les mêmes propriétés ,et les nœuds ont besoin de connaître seulement leurs voisins, rend ce protocole passable à l'échelle .
- La possibilité de réaliser un routage optimal réside de l'une des caractéristiques de routage plat celle que tous les nœuds peuvent communiquer entre eux sans avoir appel à un intermédiaire .
- Les réseaux à plat permettent aux protocoles de routage d'être simples, puisque les nœuds communiquent entre eux sans avoir appel à un intermédiaire.

Inconvénients :

- Les nœuds qui entourent un nœud ,car le trafic le plus élevé dans un réseau se trouve autour des nœuds puits.

2.4.2 Les protocoles basés sur la localisation

Dans ce type de routage [9, 10, 11, 12, 13, 14, 15], les nœuds capteurs sont adressés en fonction de leurs localisations. La distance entre les nœuds voisins peut être estimée sur la base des forces entrantes de signal. Des coordonnées relatives des nœuds voisins peuvent être obtenues en échangeant une telle information entre les voisins. Alternativement, la location des nœuds peut être disponible directement en communiquant avec un satellite en utilisant GPS (système de positionnement global). Dans la plupart des protocoles de

routage, l'information sur la localisation des nœuds est nécessaire afin de calculer la distance entre deux nœuds particuliers de sorte que la consommation d'énergie puisse être estimée. On peut mentionner parmi ces protocoles :

2.4.2.1 MECN et SMECN

MECN (Minimum energy communication network) utilise le GPS à basse puissance. L'idée principale est de trouver un sous réseau, qui aura moins de nœuds et qui exige moins de puissance pour la transmission entre deux nœuds particuliers quelconques. SMECN (Small MECN) est une extension de MECN. Le sous- réseau construit par SMECN est probablement plus petit (en termes de nombre d'arcs) que celui construit par MECN[12].

2.4.2.2 GAF

GAF (Geographic adaptive fidelity) conserve l'énergie par la mise en veille des nœuds inutiles dans le réseau sans affecter le niveau de fidélité du routage. Il forme une grille virtuelle pour le domaine couvert. Chaque nœud emploie sa position indiquée par le GPS pour s'associer à un point dans la grille virtuelle. Des nœuds liés au même point sur la grille sont considérés équivalents en terme de coût de routage. Une telle équivalence est exploitée en maintenant quelques nœuds, situés dans un secteur particulier de la grille, dans l'état de sommeil afin d'économiser de l'énergie[12].

2.4.2.3 GEAR (geographic and energy-aware routing)

L'idée est de restreindre le nombre d'intérêts dans la diffusion dirigée en considérant seulement certaines régions plutôt que d'envoyer les intérêts au réseau tout entier. Le routage géographique suppose que tous les nœuds connaissent leur position. Une solution basée sur le GPS peut être trop coûteuse, d'autant plus que le nombre de nœuds à équiper est très grand[13].

Les avantages et les inconvénients

Avantage :

- L'utilisation des systèmes de positionnement, tels que le GPS, fournit un avantage assez important. Ces systèmes ont pour but l'amélioration de la connaissance de positionnement des capteurs au centimètre près, ainsi de faciliter le contrôle de la topologie et celui de la puissance de transmission des capteurs.
- Avec les informations de positionnement, le réseau minimise l'énergie grâce à la facilité du calcul des routes énergétiquement optimales.
- Dans le routage basé sur la localisation géographique, la région de captage est connue et les requêtes peuvent être donc dirigées uniquement vers cette région, ce qui éliminera le nombre de transmission de manière significative.
- Le routage basé localisation est bien adapté à un déploiement aléatoire des nœuds, vu le traitement sophistiqué qui permet d'améliorer la précision de positionnement.

Inconvénients :

- La première contrainte se situe dans les nœuds devront être équipés d'un système de localisation par satellite.
- Les récepteurs GPS sont coûteux en énergie et relativement peu précis.
- Les nœuds doivent être équipés d'un système de localisation par satellite.

2.4.3 Les protocoles hiérarchiques

Le routage hiérarchique [9, 10, 11] est considéré comme étant l'approche la plus favorable en terme d'efficacité énergétique. Il se base sur le concept (nœud standard - nœud maître) où les nœuds standard acheminent leurs messages à leur maître, lequel les achemine ensuite dans le réseau tout entier via d'autres nœuds maîtres jusqu'à la station de base (sink).

Le point fort de ce type de protocoles est l'agrégation et la fusion des données afin de diminuer le nombre de messages transmis au sink, ce qui implique une meilleure économie d'énergie. En fait, deux grandes approches sont dérivées de ce type de protocoles à savoir : chaîne-based approach (approche chaînée) comme PEGASIS et cluster-based approach (approche à grappe) comme LEACH[9].

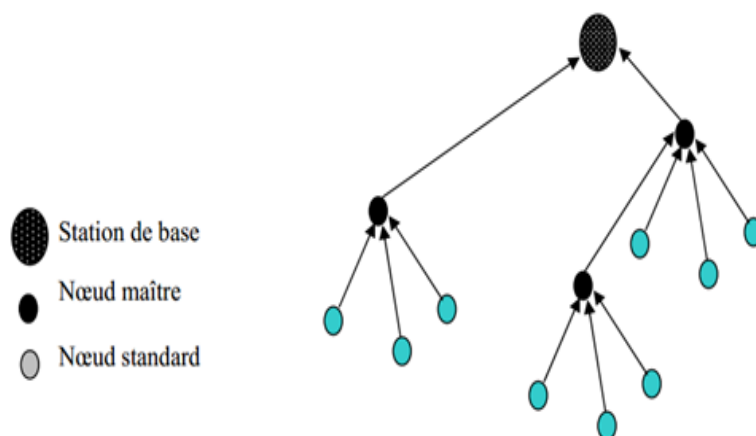


FIG. 2.3 – Le routage hiérarchique

2.4.3.1 LEACH (Low Energy Adaptive Clustering Hierarchy)

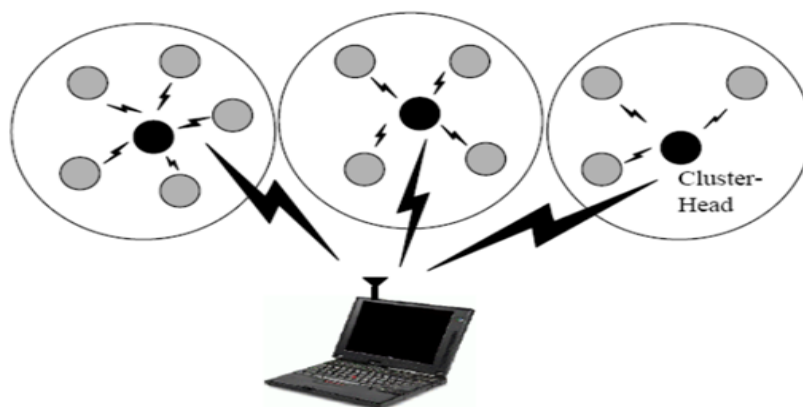


FIG. 2.4 – Algorithm de routage LEACH

LEACH (Low Energy Adaptive Clustering Hierarchy) [17, 18] est un protocole auto-organisateur basé sur le clustering adaptatif, qui utilise la rotation randomisée des têtes de grappe pour distribuer équitablement la charge d'énergie entre les nœuds capteurs dans le réseau. Il est considéré comme étant l'une des premières approches de routage hiérarchique basées sur le clustering. LEACH est fondé sur deux hypothèses de base :

1. La station de base est fixe et est placée loin des capteurs.
2. Tous les nœuds du réseau sont homogènes et limités en énergie.

L'idée derrière LEACH est de former des clusters des nœuds capteurs selon la force reçue du signal et d'utiliser les têtes locales de grappe (cluster head) comme des routeurs pour conduire des données à la station de base. Les dispositifs principaux de LEACH sont [17] :

1. Coordination et contrôle localisés pour l'initialisation et le traitement de grappe.
2. Rotation randomisée de la grappe (effectuée par «les stations de base» ou «les têtes de cluster»).
3. Compression locale (agrégation) les nœuds CH compressent les données arrivant des nœuds appartenant à leurs grappes respectives, et envoient un paquet d'agrégation à la station de base afin de réduire la quantité d'information qui doit être transmise à la station de base.

2.4.3.2 PEGASIS (Power-Efficient Gathering in Sensor Information Systems)

Power-Efficient Gathering in Sensor Information Systems (PEGASIS) est une version améliorée du protocole LEACH. PEGASIS forme des chaînes plutôt que des clusters de nœuds de capteurs afin que chaque nœud transmette et reçoive uniquement des données d'un voisin. Un seul nœud est sélectionné à partir de cette chaîne pour transmettre à la station de base. L'idée de PEGASIS est qu'il utilise tous les nœuds pour transmettre ou recevoir des données avec ses plus proches voisins. Il déplace les données reçues de nœud à nœud, puis les données seront agrégées jusqu'à ce qu'elles atteignent tous la station de base. Donc, chaque nœud du réseau est tour à tour un chef de file de la chaîne, ainsi que responsable pour transmettre l'ensemble des données recueillies et fusionnées par la chaîne de nœuds au niveau de la station de base[18].

Les avantages et inconvénients

Avantage

- Agrégation des données : l'avantage réside dans le fait que les données du cluster entier peuvent être combinées par le cluster-head et envoyées vers la destination.

Inconvénients

- Les cluster-head consomment plus d'énergies que les autres nœuds, s'ils restent les mêmes pour longtemps, le réseau risquera de se découper, ainsi les nœuds capteurs.
- Ne pourront pas communiquer avec la station de base, et les informations de capture n'y seront pas transmises.
- Agrégation des données : l'avantage réside dans le fait que les données du cluster entier peuvent être combinées par le cluster-head et envoyées vers la destination.

2.4.4 Les protocoles non hiérarchiques

Sont des protocoles de routage ad hoc adaptés aux RCSF. Ces protocoles peuvent être classés en deux catégories : les protocoles réactifs et proactifs .

2.4.4.1 Les protocoles réactifs

Contrairement aux protocoles proactifs, les protocoles réactifs ne calculent la route que sur demande. Ils tiennent compte de l'évolution des trafics et beaucoup d'autres paramètres dans le réseau afin de déduire le chemin optimal pour transmettre l'information d'un nœud à un autre. Ces protocoles sont surtout utilisés dans le cas des nœuds mobiles et quand les liens entre les nœuds varient progressivement. Si un nœud source a besoin d'envoyer un message à un nœud destination, alors il envoie une requête à tous les membres de réseau. Après la réception de la requête, le nœud destination envoie un message réponse qui remonte vers la source (méthode Backward Learning). Cependant, le routage à la demande génère une lenteur à cause de la recherche des routes. Cela peut entraîner une dégradation des performances des applications[30]. Ce type de protocole présente l'inconvénient d'être très coûteux en transmission de paquets lors de la détermination des routes. Par conséquent, il n'y a pas une route prédéfinie, on ne maintient la route que si nécessaire. Ces protocoles ont alors l'avantage de ne pas avoir à maintenir des informations inutilisées dans les tables de routage. Exemple : AODV (Ad Hoc On-Demand Distance Vector Routing), DSR (Dynamic Source Routing), etc.

2.4.4.2 Les protocoles de routages proactifs

Les protocoles de routage proactifs sont basés sur la même philosophie que les protocoles de routage utilisés dans les réseaux filaires conventionnels. Les deux principales méthodes utilisées dans cette classe de protocoles proactifs sont[31] :

- La méthode Etat de Lien ("Link State").
- La méthode du Vecteur de Distance ("Distance Vector").

Ces méthodes sont utilisées aussi dans les réseaux sans fil. Parmi les protocoles de routage proactifs les plus connus on citera le DSDV, FSR, OLSR.

2.5 Outils de la simulation réseau

Les simulateurs fournissent des structures virtuelles, créent une configuration du réseau désiré et de produire le trafic virtuel dans ce dernier. Après la simulation les résultats sont analysés. Cela est réalisable en utilisant des outils permettant de visualiser la simulation. L'analyse se fera à l'aide des fichiers trace.

Plusieurs dimensions sur la façon de répartir les outils, d'inclure la convivialité, le niveau d'extension, mécanisme de personnalisation, la vitesse d'exécution, évolutivité de la capacité avec la taille, support de l'émulation, la diversité du modèle, la fidélité, et le niveau de

soutien.

Les facteurs le plus importants qui influent sur le choix d'utilisation d'un simulateur sont le cout, la facilité d'utilisation, la disponibilité des modèles de simulation pour des applications souhaitées et la précision avant de procéder à une quelconque comparaison entre certains Simulateurs, il est bon de les présenter. Nous pourrons ensuite dresser la liste des points de comparaison retenus.

2.5.1 Network Simulator 2

Le simulateur réseau (NS-2) est un outil de simulation à événements discrets. Il permet la simulation des réseaux IP filaires et sans fil. Il a débuté en 1989, on l'a amélioré au fil des versions. Depuis 1995 NS est soutenu par la contribution de DARPA (Defense Advanced Research Project Agency). En 1996, la version 2 de ns a été introduite avec, bien sûre, des améliorations architecturales. En 1997, le ns-2 à été étendu pour soutenir les réseaux sans fil. De nos jours, le NS-2 est l'outil de simulation le plus utilisée pour les réseaux typiques ainsi que pour l'attribution des ressources [19,20].

NS-2 est un outil de simulation de réseaux de données. Il est bâti autour d'un langage de programmation appelé Tcl dont il est une extension. Du point de vue de l'utilisateur, la mise en œuvre de ce simulateur se fait via une étape de programmation qui décrit la topologie du réseau et le comportement de ses composants, puis vient l'étape de simulation proprement dite et enfin l'interprétation des résultats. Cette dernière étape peut être prise en charge par un outil annexe, appelé Nam qui permet une visualisation et une analyse des éléments simulés.

NS2 est en réalité un programme relativement complexe écrit en C++ et interfacé via Tcl. Pour modifier le comportement d'objets existants, il est donc nécessaire de modifier le code C++ qui en réalise l'implantation.

Il existe dans le NS-2 une extension, nam (Network Animator) qui permet la visualisation de la topologie du réseau et peut afficher les flux de paquets et des files d'attente. On a aussi Gnuplot, Tracegraph qui aide lors de l'interprétation des résultats. NS-2 a la structure :

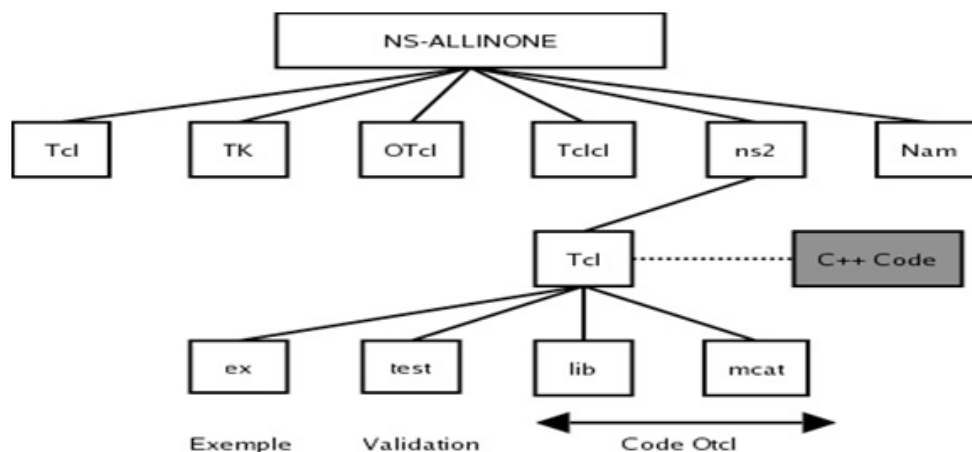


FIG. 2.5 – La structure de NS-2

A.1.1 Composants de NS-2 :

Application	Web, FTP, Telnet, générateur de trafic .
Transport	TCP, UDP, RTP, SRM
Routage	Statique, dynamique (vecteur distance) et routage multipoint (DVMRP, PIM)
Gestion des files d'attente	RED, Drop Tail, Token bucket
Discipline de service	CBQ, SFQ, DRR, Fair Queuing
Système de transmission	CSMA/CD, CSMA/CA, lien point à point

TAB. 2.1 – Composants de NS-2

2.5.2 NS-3 (Network Simulator version 3)

C'est un simulateur à événements discrets de troisième génération, il a été annoncé le 02/07/2006 la première version 3.1 est sortie en juin 2008. NS-3 est composé de modules. NS-3 est écrit en C++, mais certaines parties optionnelles de simulation peuvent être introduites en d'autres codes comme le Python. NS-3 n'est pas rétro compatible avec NS-2, il est aussi continuellement amélioré. En ce qui concerne l'architecture de NS-3 intègre également un langage évolué GTNets qui est un simulateur de caractéristiques évoluées. Cette conception est faite précisément en raison de compatibilité. Dans ce qui suit vous pouvez voir l'architecture de NS-3 :

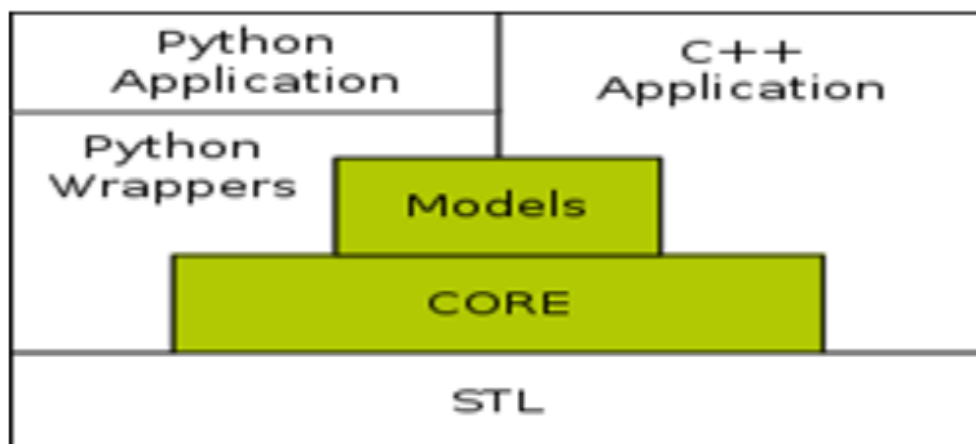


FIG. 2.6 – L'architecture de Ns-3

2.5.3 OMNet++

OMNeT++ est un environnement de simulation open source possédant une interface graphique solide intégrant un noyau de simulation. C'est un outil de simulation à événements discrets orientés objets, basé sur C++, gratuit pour les utilisations académiques. Tandis qu'il existe une version commerciale qui est OMNEST. Il a été conçu pour simuler[21,22,23] :

- Les systèmes réseaux de communication.
- Les systèmes multi processeurs.
- D'autres systèmes distribués.
- Actuellement il est utilisé dans les universités pour la validation de nouveaux matériels et logiciels, ainsi que pour l'analyse et l'évaluation des protocoles de communication.
- Les réseaux de fil d'attente.

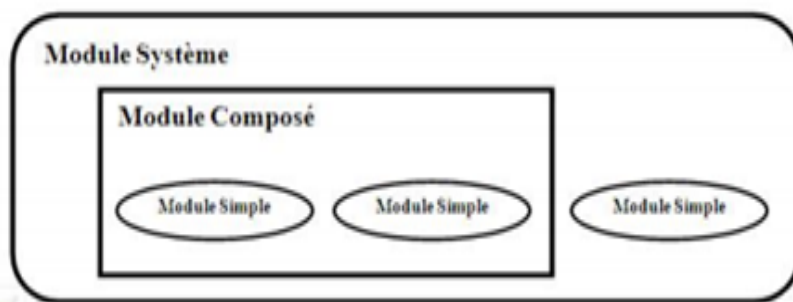
OMNet++ est réalisé grâce au projet d'étude de Technical University of Budapest en 1992 et on l'améliore au fil des années. Cependant, la simulation des réseaux d'ordinateurs est le principal domaine d'application, pour cela il existe des cadres qui fournissent des modules pour la simulation filaire et le sans fil, ou dans le cadre de la mobilité. OMNet++ est connu par la facilité d'apprentissage, d'intégration de nouveaux modules et la modification de ceux déjà implémentés.

2.5.3.1 Architecture d'OMNeT++ :

L'architecture du modèle OMNeT++ se compose de plusieurs modules hiérarchiquement (visualisé dans la figure ci-dessous) imbriqués qui sont

- Le module système.

- Modules simples (les feuilles) : programmés en C++ encapsulant le comportement réel d'un système. Pour chaque module simple correspond un fichier .cc et un fichier .h.
- Modules composés : constitués d'un ou de plusieurs modules simples ou des modules composés reliés entre eux. Les paramètres, les ports et les modules de chaque module sont spécifiés dans un fichier .ned. l'architecture d'OMNET++ est visualisée dans la figure qui suit :



Architecture modulaire du simulateur Omnet++.

FIG. 2.7 – L'architecture d'OMNET++

Les modèles OMNet++ sont écrits en langage C++. La communication entre deux modules communiquant ce fait par échange de messages qui sont soit des paquets, des trames d'un réseau informatique, des clients dans une fil d'attente ou d'autres types d'entités. Les messages sont échangés à travers des interfaces d'entrées/ sorties des modules. Dans la construction d'un nouveau programme de simulation, il y aura à chaque fois des informations chargées dynamiquement tel que la topologie du réseau à partir des fichiers .ned, et les configurations sont disponibles dans les fichiers .ini. Lors de la simulation de différents fichiers trace seront remplies. On a aussi le plove qui est un outil pouvant visualiser les données enregistrées. On aura aussi les deux fichiers omnet.vec et omnet.sca seront utiles lors de la tracé de la courbe et calculer des statistiques.

2.5.3.2 Composants d'OMNET++ :

Application	FTP, Telnet, générateur de trafic (IPTrfGen..), Ethernet, Ping App, UDP App, TCP App.
Transport	TCP, UDP, RTP
Réseau	IPv4, IPv6, ARP, OSPF, LDP, MPLS, ICMP, TED
Gestion des files d'attente	Mgmt, MAC, Radio
Node	CBQ, SFQ, DRR, Ad Hoc, Wireless, MPLS

TAB. 2.2 – Composants d'OMNET++ :

2.5.4 QualNet

C'est un programme qui gère le (le filaire) ainsi que le sans fil. Son architecture est répartie en 3 couches :

- Le noyau de simulation : qui est un planificateur d'évènements discrets comme une couche de base.
- Les bibliothèques de modèles.
- La couche supérieure est définie par QualNet Developer GUI.

Qualnet permet de créer et de visualiser des scénarios de réseau, et d'analyser les résultats de la simulation à l'aide de GUI. Lors de la création du scénario du réseau les informations basics seront stockés dans un fichier de description du scénario (XML). L'environnement sans fil, la pile, le protocole, la collection statique ou le paquet de traçage sont configurés ainsi que l'emplacement des nœuds et la définition de la topologie utilisé. QualNet prend en charge plusieurs protocoles de la couche application (CBR, FTP ou VoIP). Vous pouvez avoir la simulation dans l'outil "Animator". Comme d'autres simulateurs, la vitesse de simulation peut-être accélérée ou ralentie. Mais il n'y a pas d'analyse durant la simulation les sorties de la simulation sont différentes par rapport à celles enregistrées dans les fichiers de sorties. Le fichier de sortie .stat contient les informations statistiques. Il y a même le fichier trace qui englobe les paquets enregistrés.

2.5.5 OPNET (Optimum Network Performance)

L'outil de simulation OPNET permet de simuler les infrastructures des réseaux puissants, codé en C++ et conçu en orienté objet. Il permet aussi de concevoir et d'étudier des réseaux de communications, on se base sur ce simulateur surtout pour le réseau des capteurs sans fil, les nouvelles technologies. . .etc. Mais il est long lors de sont apprentissage.

Il est composé de plusieurs éditeurs : le module nœuds, le module process et le module simulation (exécution de la simulation qui apportera des changements aux nœuds. OPNET englobe plusieurs fonctions propriétaires. Il gère aussi certains types d'objets tels que les liens et les formats de paquets OPNET est un simulateur assez complexe ce qui résulte les difficultés d'apprentissage. Il est basé sur deux méthodes qui sont l'utilisation des nœuds programmés et définition des modèles[26,27]. OPNET peut-être utilisé dans :

- Modélisation de trafic des réseaux de télécommunication.
- Modélisation des protocoles.
- Modélisation des queues réseaux.
- Validation des architectures hardware.
- Evaluation de l'aspect de la performance des systèmes logiciel complexes.

L'architecture d'OPNET est hiérarchique imbriquée, dont le niveau le plus haut le domaine réseau qui permet de définir la topologie de ce dernier. On a aussi un niveau plus qui est le node domaine permettant de définir la constitution des nœuds. Et enfin le process domaine où s'effectue la définition de chaque module programmable. La simulation sous OPNET est basée sur deux méthodes, soit par utilisation des nœuds programmés fournis par OPNET, ou bien par la définition d'un modèle de lien, des protocoles et des modèles processus.

2.5.6 GloMoSim (Global Mobile Information System Simulate)

C'est un outil de simulation créé par UCLA. Il permet la simulation des réseaux filaires et sans fils à grandes échelles et en précisant les modèles. Ils utilisent dans cet outil de simulation les capacités du langage Parsec. Il offre la possibilité de simuler un réseau WLAN avec tous les protocoles qu'ils incluent (AODV, DSR, algorithme de Bellman Ford (routage par vecteur de distance), ODMRP (On-Demand multicast Routing qui est un protocole réactif), WRP (Wireless Routing Protocols qui est un protocole réactif), FSR (Fisheye State Routing qui est le routage à état du lien), ... etc. La majorité des systèmes de réseaux de Glomosim sont construites en se basant sur les sept couches OSI. La simulation sous GloMoSim utilise la technique de parallélisme. GloMoSim est un outil de simulation connu par un passage à l'échelle, la précision des modèles. En outre le GloMoSim est connu par la complexité de son apprentissage[28].

2.5.7 JiST (Java in Simulation Time)

JiST est une nouvelle version de simulation, comme son nom l'indique ce simulateur utilise comme langage de programmation le JAVA et est utilisé généralement avec un autre simulateur de réseaux mobiles ad hoc qui est le SWANS. La simulation en JiST est constituée en entités qui représentent les éléments constituant le réseau (les nœuds,...), avancent le temps de simulation cela en tenant compte des interactions seulement entre entités différentes ; c'est-à-dire que le temps de simulation correspond aux points de synchronisation entre des entités dans les quelles des programmes sont entrain de s'exécuter simultanément et cela permet d'obtenir une amélioration en performance. Pour exécuter une application en temps de la simulation, Jist utilise un chargeur JAVA personnalisé qui permet de réécrire dynamiquement le bit code de l'application. Le développement de JIST n'a pas été amélioré davantage, depuis que son créateur Rimon Barr a abandonné ce projet mais l'université Ulm a mis quelques points sur des améliorations récentes concernant le simulateur.

2.6 Tableau d'utilisation des outils de simulation

Simulateurs	NS-2	OMNet++	NS-3	OPNET
Cas d'utilisation	Réseau adhoc	Réseau ad hoc et réseaux Complexe (filaire et sans fil).	Simulation de protocoles (TCP, UDP, IPv4, IPv6...) Simulation de médias (Ethernet, WIFI, WIMAX...) Définition de la topologie du réseau (statistique ou dynamique).	Réseaux de capteurs sans fils.
Simulateurs	QualNet	J-SIM (Java Simulator)	GloMoSim	JiST
Cas d'utilisation	Réseau filaire et sans fil	Réseau de capteurs.	Réseau ad hoc. Réseau de capteurs.	Utilisé essentiellement dans les réseaux étendus sans fil ad hoc

TAB. 2.3 – Cas d'utilisation des simulateurs

2.7 Conclusion

Dans ce chapitre, nous avons mis l'accent sur les principaux protocoles de routage dans les réseaux de capteurs. Ces réseaux se classent selon quatre catégories principales : les protocoles hiérarchiques, les protocoles non hiérarchiques, les protocoles basés sur la localisation et les protocoles data-centric. Enfin, Nous nous sommes concentrés sur les Outils de la simulation réseau qui fournissent des structures virtuelles d'un réseau. Dans le prochain chapitre on va étudier les protocoles AODV et OLSR ainsi que les résultats de simulation.

3

Simulation et évaluations des protocoles OLSR, AODV

3.1 Introduction

Les réseaux de capteurs sans fil nécessitent de techniques de communication efficace afin assurer une bonne connectivité et rapidité des données. Plusieurs protocoles de routage ad hoc tels qu'AODV et OLSR ont été proposés pour ce genre de réseau. Dans ce chapitre, nous définissons ces protocoles et leurs principes de fonctionnement, nous exposant aussi comment effectuer la simulation de ces deux protocoles afin de calculer des critères d'évaluations. Enfin l'interprétation et l'évaluation des résultats de simulation.

3.2 Etudes des Protocoles OLSR, AODV

3.2.1 Le protocole AODV

Le protocole AODV (Ad-hoc On-Demand Distance Vector) est l'un des protocoles les plus étudiés par les chercheurs, ce qui se traduit par une maturité dans les implémentations disponibles ainsi que dans les différents comparatifs l'opposant à de nombreuses autres

solutions. Ce protocole a donné naissance à une RFC actuellement au stade expérimental (RFC3561) et gérée par le groupe de travail de l'IETF spécialisé dans le domaine des réseaux ad hoc.

Lorsqu'un nœud désire envoyer un message à une destination pour laquelle il ne possède pas de route, la procédure de construction est initiée par l'envoi d'un message RREQ à tous ses voisins physiques par diffusion (broadcast). A la réception d'un tel message, les nœuds insèrent dans leurs tables de routage, une route vers le nœud à l'origine de la requête passant par le nœud ayant envoyé le message qui devient le prochain pas (next hop) vers le nœud à l'origine de la requête.

Cette démarche permet de créer une route de retour (reverse path) pour les futurs messages transitant de la destination à la source. En insérant cette route, les nœuds font l'hypothèse que les liens physiques sont symétriques et qu'ils pourront atteindre le nœud ayant envoyé le message. Chaque nœud recevant ce message va le retransmettre de la même manière ce qui à terme va inonder le réseau. Ces messages sont identifiables par chacun des nœuds ce qui leur permet de les traiter qu'une seule fois et empêche qu'une requête transite indéfiniment dans le réseau. Les principales informations de la table de routage du nœud 7 sont représentées en guise d'exemple, tel que le prochain nœud vers la destination, le numéro de séquence de la destination ainsi qu'une métrique (nombre de sauts jusqu'à la destination).

Dans ce protocole, les nœuds intermédiaires ainsi que le nœud recherché peuvent répondre à ces requêtes par l'envoi d'un message RREP qui va transiter jusqu'au nœud à l'origine de la requête.

Ce nouveau type de message n'inonde pas le réseau mais est envoyé en empruntant la route créée lors du passage de la requête. Les nœuds intermédiaires peuvent répondre uniquement s'ils possèdent une route assez fraîche vers la destination recherchée. Cette caractéristique permet de construire plusieurs routes vers la destination. Ainsi, le nœud à l'origine de la construction va potentiellement recevoir plusieurs messages RREP. Ce dernier va sélectionner la route la plus courte en fonction du nombre de pas physiques. Le niveau de fraîcheur des routes est établi sur la base d'un numéro de séquence maintenu par chaque nœud et est comparé avec la fraîcheur minimale désirée par le nœud à l'origine de la requête dans le message RREQ.

Ce numéro de séquence permet de garantir la construction de routes dépourvues de boucles et d'imposer la reconstruction d'une route. A la réception d'un message RREP, les nœuds insèrent dans leurs tables de routage la route directe (forward route) vers le nœud recherché. Ainsi, l'algorithme a créé la route bidirectionnelle reliant le nœud à l'origine de la requête et le nœud recherché.

Lorsqu'une route est créée, une procédure de maintenance est initiée, visant à garantir une route exploitable et consistante à détecter les liens brisés. Dans la version originale du protocole, cette procédure est mise en œuvre par l'échange périodique de messages entre les voisins physiques des nœuds constituant la route. Un nœud ne recevant pas de message de maintenance durant un certain nombre de périodes va considérer la route comme étant brisée. Le processus de gestion des erreurs offre deux possibilités aux nœuds détectant une route brisée. La première consiste à réparer localement la route en initiant une nouvelle requête de construction et à stocker dans un tampon les éventuels messages nécessitant la route brisée au lieu de les retransmettre plus tard. La deuxième solution consiste à avertir tous les nœuds jusqu'au nœud source de l'état de la route. Pour chaque destination contenue dans la table de routage d'un nœud, une liste des prédécesseurs est créée. Cette liste contient les nœuds susceptibles d'utiliser ce nœud pour transférer de l'information vers la destination.

Le nœud détectant une route brisée se base sur ces listes pour envoyer des messages RRER aux nœuds ciblés, contenant la destination du nœud qui n'est plus atteignable. Les nœuds recevant un tel message se comportent de la même manière ce qui va faire transiter l'information jusqu'au nœud source. Le nœud source va reconstruire la route uniquement lorsqu'il en aura à nouveau besoin. Les routes ont une durée de vie limitée qui est mise à jours après chaque utilisation. Une route inutilisée durant une longue période (3 secondes par défaut) n'est plus valide et le processus de maintien de la route est stoppé ce qui permet d'éviter des échanges de messages de maintenance inutiles.

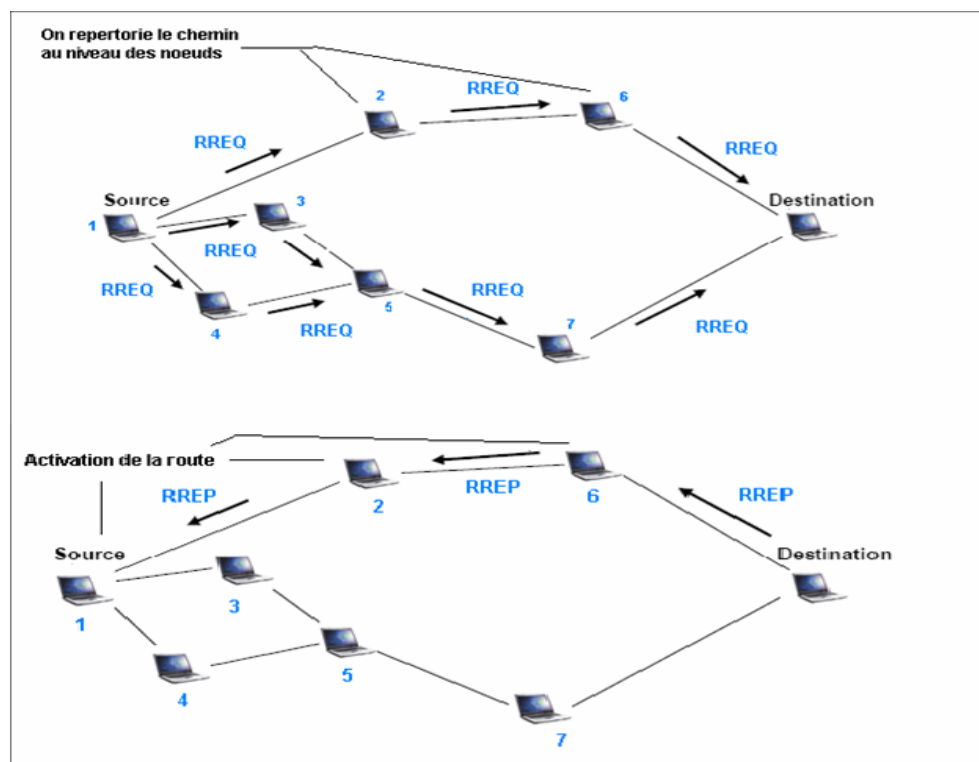


FIG. 3.1 – Recherche de la destination par la requête RREQ et la réponse par la requête RREP

3.2.2 Protocol OLSR

Comme son nom l'indique, OLSR (Optimized Link State Routing) est un protocole proactif à état des liens optimisé, il permet d'obtenir aussi des routes de plus court chemin. Alors que dans un protocole à état des liens, chaque nœud déclare ses liens directs avec ses voisins à tout le réseau, dans le cas d'OLSR, les nœuds ne déclarent qu'une sous-partie de leur voisinage grâce à la technique des relais multipoints (MultiPoint Relaying, MPR) décrite par la suite.

a) Relais multipoints

Cette technique consiste essentiellement, pour un nœud donné, à ignorer un ensemble de liens et de voisins directs, qui sont redondants pour le calcul des routes de plus court chemin. Plus précisément, dans l'ensemble des voisins d'un nœud, seul un sous-ensemble de ses voisins est considéré comme pertinent. Il est choisi de façon à pouvoir atteindre tout le voisinage à deux sauts (tous les voisins des voisins), cet ensemble est appelé l'ensemble des relais multipoints.

Ces relais multipoints sont utilisés de deux façons : pour diminuer le trafic dû à la diffusion des messages de contrôle dans le réseau, et aussi pour diminuer la taille du sous-

ensemble des liens diffusés à tout le réseau puisque les routes sont construites à base des relais multipoint.

L'idée de MPR est de minimiser l'inondation du trafic de contrôle dans un réseau en réduisant les retransmissions dupliquées dans la même région. Chaque nœud dans le réseau sélectionne un ensemble de nœuds de son voisinage auxquels ses messages seront transmis. Un nœud sélectionne ses MPRs parmi ses voisins à un saut avec un lien symétrique. Cet ensemble est choisi de manière à couvrir tous les nœuds qui sont à deux sauts. Les nœuds sélectionnés comme MPRs annoncent régulièrement leur condition de MPR dans les messages de contrôle envoyés à son voisinage. De cette façon, un nœud annonce au réseau qu'il est capable d'atteindre les nœuds qui l'ont élu comme MPR. Dans le calcul de la route, les MPRs sont utilisés pour la mise en place des routes vers toutes les destinations du réseau. Ainsi, en sélectionnant la route par l'intermédiaire des MPRs, on évite les problèmes liés à la transmission de paquets sur des liens unidirectionnels. Chaque nœud maintient l'information sur ses voisins qui ont été sélectionnés comme MPR. Un nœud obtient cette information par les messages de contrôle reçus périodiquement de ses voisins.

La Figure montre la différence entre une diffusion pure et la diffusion en utilisant les MPRs. Par exemple, afin d'atteindre tous les nœuds à 3 sauts, la diffusion pure a besoin de 24 retransmissions du paquet envoyé par la source (voir la Figure , côté gauche). En utilisant les MPRs ou les relais multipoints (nœuds en gras dans le côté droit de la Figure), il suffit de retransmettre le paquet de la source 11 fois pour atteindre les nœuds à 3 sauts.

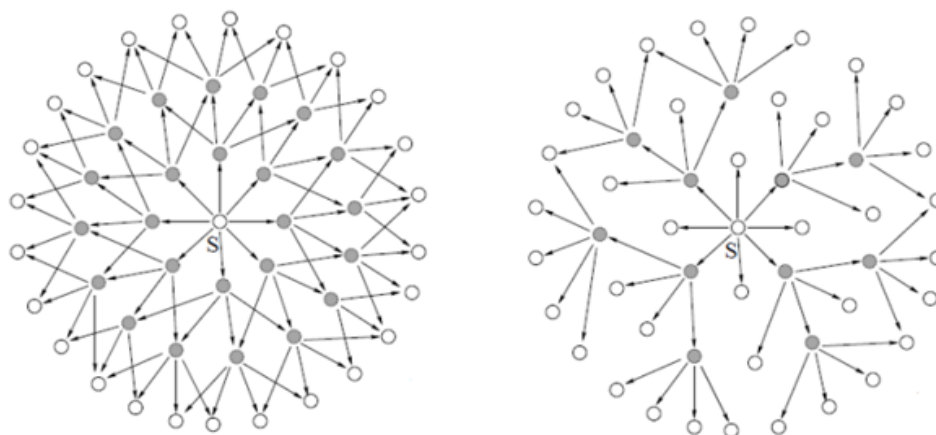


FIG. 3.2 – Diffusion pure et diffusion en utilisant les MPR s dans OLSR

b) Fonctionnement du protocole

OLSR est un protocole proactif et très bien adapté aux réseaux larges et denses. Tous

les nœuds du réseau serviront de routeur et OLSR maintient sur chaque nœud une table de routage complète (comprenant une entrée pour tous les autres nœuds du réseau). Le protocole est complètement distribué, il n'y a pas d'entité centrale. Chaque nœud choisit la route la plus adaptée en fonction des informations qu'il a reçues. Pour maintenir à jour toutes les informations nécessaires au choix des relais multipoints (MPRs) et au calcul de la table de routage, les nœuds OLSR ont besoin de s'échanger des informations périodiquement. Pour s'informer du proche voisinage, les nœuds OLSR envoient périodiquement des messages dits HELLO contenant la liste de leurs voisins. Ces messages permettent à chacun de choisir son ensemble de relais multipoints. Le deuxième type de message d'OLSR est le message TC (Topology Control). Par ce message les sous-ensembles de voisinage que constituent les relais multipoints sont déclarés périodiquement dans le réseau. Ils sont diffusés en utilisant une diffusion optimisée par relais multipoints. Ces informations offrent une carte du réseau contenant tous les nœuds et un ensemble partiel de liens, mais suffisant pour la construction de la table de routage.

La table de routage est calculée par chacun des nœuds et le routage des données s'effectue saut par saut sans l'intervention d'OLSR dont le rôle s'arrête à la mise à jour de la table de routage.

3.3 Outils de simulation

3.3.1 NS-3 (Network Simulator version 3)

Un simulateur de réseau se compose d'une large gamme de technologies de réseau et de protocoles. Il est conçu pour aider les utilisateurs à créer des réseaux complexes à partir des classes représentant les modules de base pour construire un réseau. Nous avons choisi de faire notre simulation avec le simulateur de réseau NS-3 (Network Simulator 3). Les raisons de ce choix sont justifiées par la robustesse du simulateur NS-3. En effet, ce simulateur offre de meilleures performances en termes de rapidité de calcul et en temps de réponse. De plus, il garantit un passage à l'échelle jusqu'à 3000 nœuds. Enfin, c'est un simulateur destiné à la recherche.

3.3.2 Installation de NS-3

NS-3 est conçu pour fonctionner sur les plates-formes Linux. Toutefois il peut être installé sur MAC OS et Ms Windows en utilisant Cywing. L'installation comprend les étapes suivantes :

1. Télécharger la dernière version du NS-3 dans le site [32].
2. Décompresser l'archive : manuellement où dans un terminal exécute la commande suivant `tar -jxf ns-3.19.tar.bz2`.
3. Importer les package nécessaires : dans un terminale exécute la commande ci-dessous :
 - `sudo apt-get upgrade`
 - `sudo apt-get installs gcc g++ python python-dev mercurial bzip2 gdb valgrind gsl-bin libgsl0-dev libgsl0ldbl flex bison tcpdump sqlite sqlite3 libsqlite3-dev libxml2 libxml2-dev libgtk2.0-0 libgtk2.0-dev uncrustify doxygen graphviz imagemagick texlive texlive-latex-extra texlive-generic-extra texlive-generic-recommended texinfo dia texlive texlive-latex-extra texlive-extra-utils texlive-generic-recommended texi2html python-pygraphviz python-kiwi python-pygoocanvas libgoocanvas-dev python-pygccxml`
4. Après l'importation des packages nécessaires exécuter les commande suivant :
 - . `/configure`.
 - . `/waf`.
 - . `/waf buil`

3.4 Les critères d'évaluation

Notre objectif dans ce projet, est d'effectuer une évaluation des performances des protocoles AODV et OLSR a travers les métriques suivante :

1. Taux de livraison de paquets

Ce taux noté PDR pour Packet Delivery Ratio [33] est le nombre de paquets de données reçus avec succès par la destination par rapport au nombre de paquets de données émis par la source. Il nous permet de vérifier si l'extension du protocole à un impact sur le transfert de paquets de données avec succès.

$$\text{PDR} = \frac{\sum \text{nbr de paquets recus par la destination}}{\sum \text{nbr depaquets envoyes par tous les noeuds source}} \times 100$$

2. Le délai de bout en bout

Il s'agit du temps moyen qu'un paquet de données envoyé avec succès prend pour atteindre la destination. Il est noté RAL pour Route Acquisition Latency ou encore End-to-End Delay [33]. Ce temps inclut le délai de traitement ainsi que le délai d'at-

tente dans les files d'attente dans chaque nœud intermédiaire. Il est calculé selon la formule suivante

$$D = \frac{1}{n} \sum_{k=0}^n (T_{ri} - T_{si}) \times 1000 [ms]$$

avec

T_{ri} : Instant où le paquet " i " quitte la source.

T_{si} : Instant où le " i " arrive à la destination.

3. Nombre de Paquets perdus

Ce sont les paquets qui n'ont pas pu atteindre leur destination .Cela est traduits mathématiquement par l'équation suivante :

$$D = \frac{\text{nbr paquets envoyes} - \text{nbr paquets recus}}{\text{nbr paquet envoyes}}$$

4. Le débit moyen

Parfois appelé bande passante, il définit le volume maximal d'information (bits) par unité de temps (b/s).

$$\text{debit moyen} = \sum_{k=1}^n \frac{\text{taille du paquet reçu}}{(\text{temps reception} - \text{temps emission})} \times \frac{8}{1000}$$

3.5 Interprétation et évaluation des résultats

Pour notre simulation, les protocoles que nous avons simulés sont OLSR et AODV. Pour évalués ces derniers, on revient sur les métriques de performances. Ensuite, on applique les deux différents scénarios que nous avons définis, suivi des paramètres d'évaluations. Enfin, on interprète les graphes tracé à partir des résultats collectés lors de la simulation.

Avant de commencer les tests, nous allons présenter les différents paramètres que nous avons utilisés sont illustrés dans le tableau suivant :

Paramètres	Scénario 1	Scénario 2
Protocole	AODV, OLSR	AODV, OLSR
Nombre de nœuds	20, 40,60	60
Temps de simulation	600 sec	600 sec
Temps de pause	1 sec	2, 5,7sec
Taille d'un paque	1024 ko	1024 ko
Nombre de paquet par nœud	100	100
Plat forme	Ubuntu 14.04	Ubuntu 14.04
Simulateur	NS3	NS3

TAB. 3.1 – Les paramètres a utilisé dans les différents scénarios.

3.6 Résultats de simulations

L'exécution des protocoles de routage est évaluée sur NS3 simulateur. Nous avons examiné quatre paramètres pour évaluation : le débit, délais, nombre de paquet perdus, taux de livraison des paquets.

Premier scénario

Dans ce scénario le nombre de nœud dans le réseau est varié.

1. Le débit moyen

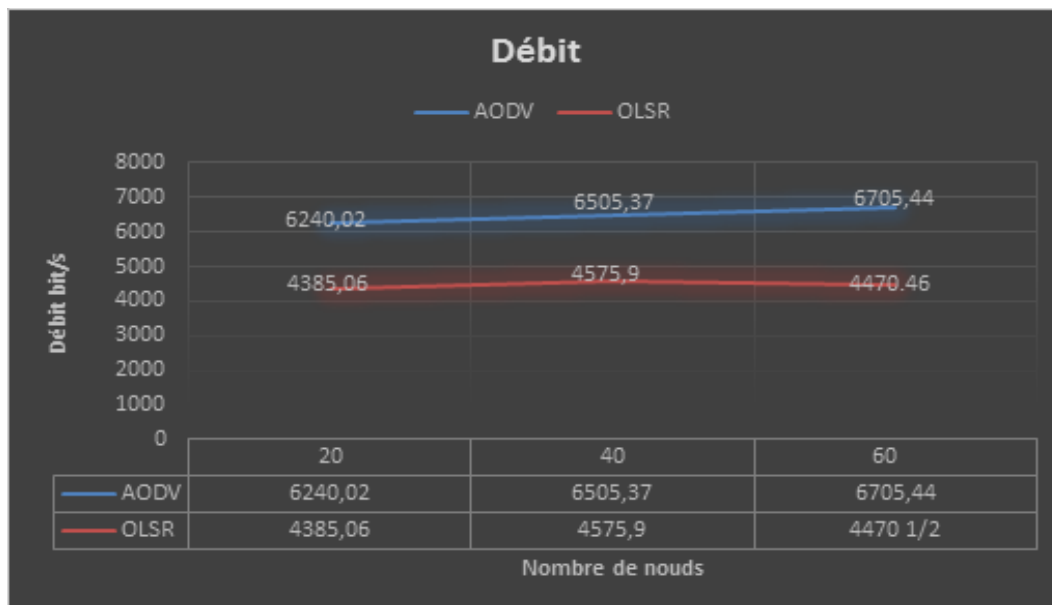


FIG. 3.3 – Le débit moyen.

Dans ce graphe qui représente le débit en fonction du nombre de nœuds, on constate que le débit global d'AODV est mieux qu'OLSR.

2. Le délai de bout en bout

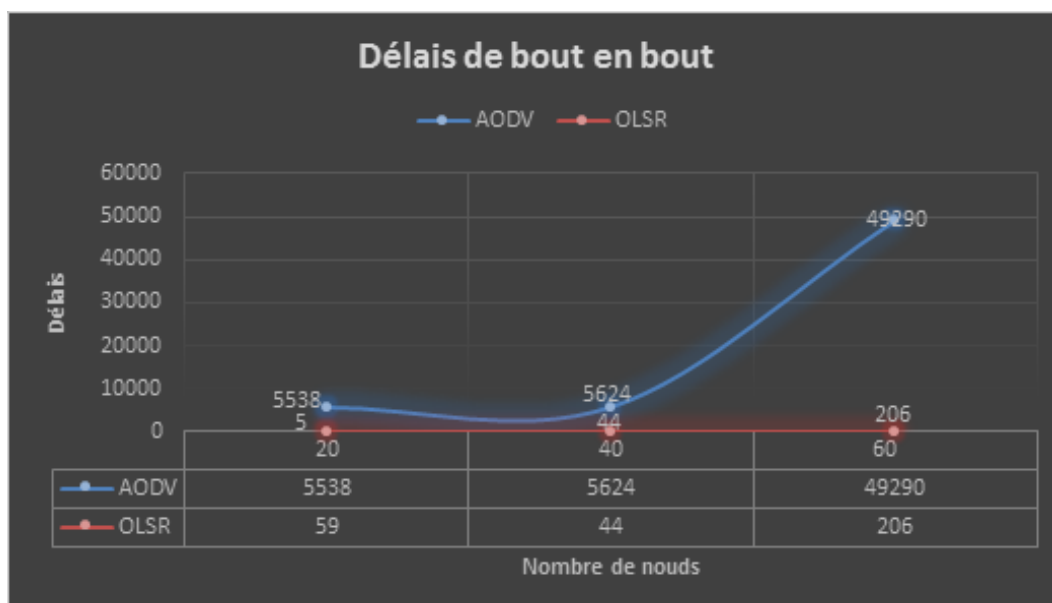


FIG. 3.4 – Le délai de bout en bout .

Les résultats obtenus confirment que les performances du protocole OLSR reste plus ou moins stable comme le montre sa courbe en augmentant le nombre de nœuds les performances diminuées. Tandis que les performances AODV diminuées en augmentant le nombre de nœuds. Cela est expliqué par le fait que le protocole AODV fait appel à une procédure de découverte de route avant la transmission des paquets et le protocole OLSR entame son processus de sélection des nœuds MPRs.

3. Nombre de paquet perdu

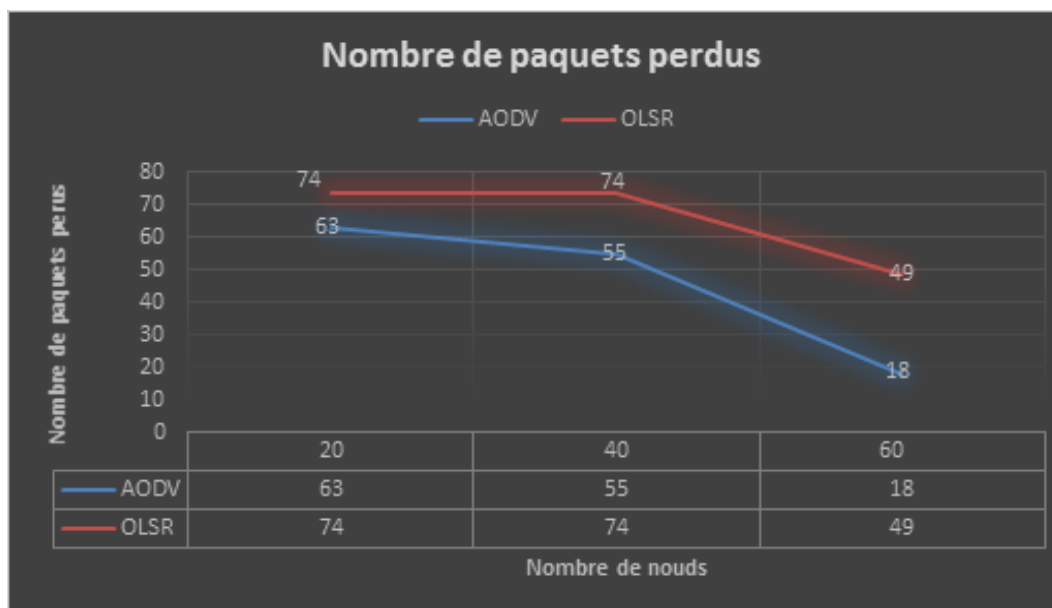


FIG. 3.5 – Nombre de paquet perdu.

En augmentant le nombre de nœuds, on constate qu'AODV présente moins de perte de paquets que le protocole OLSR.

4. Taux de livraison des paquets

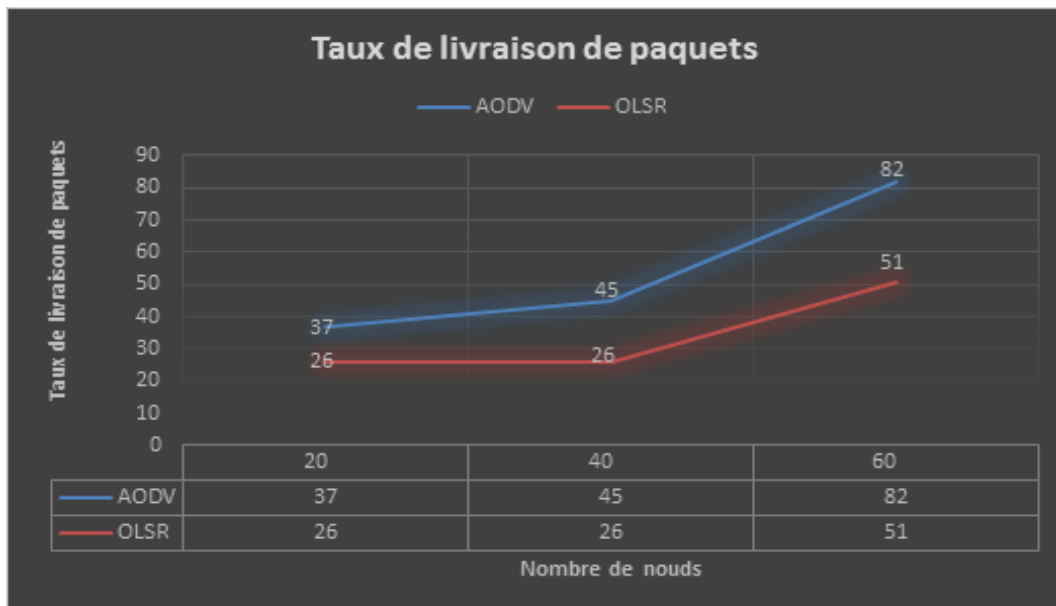


FIG. 3.6 – Taux de livraison des paquets.

Pour AODV en augmentant nombre de nœuds la livraison de paquets augment par contre OLSR entre 20 et 40 nœuds la livraison des paquets est stable et partir de 40 nœuds en remarque que le taux augment. Donc les résultats obtenus confirment que le taux de paquets délivrés par le protocole AODV est plus élevé par rapport au protocole OLSR.

Deuxième scénario

Dans ce scénario le nombre total de nœud dans le réseau est fixé et le temps de pause varie. Le temps de pause est l'intervalle de temps qui sépare deux transmissions de paquets successives. Nous avons obtenu les graphes suivants :

1. Le débit moyen

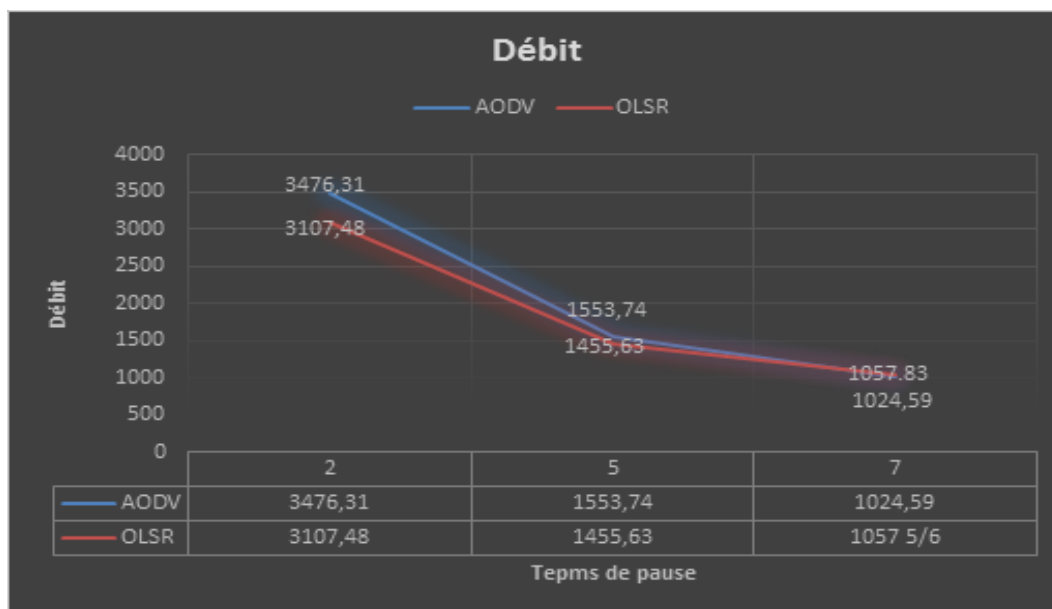


FIG. 3.7 – Le débit moyen.

Ce graphique représente le débit en fonction de temps de pause, on remarque que pour le temps de pause entre 2 et 5 second AODV présente de meilleurs résultats que ceux de protocole OLSR. A partir du temps de pause égale à 5 seconds, on remarque que les courbes des deux protocoles sont inversement proportionnelles au temps de pause, ceci est dû au processus de maintien de route dans AODV.

2. Le délai de bout en bout

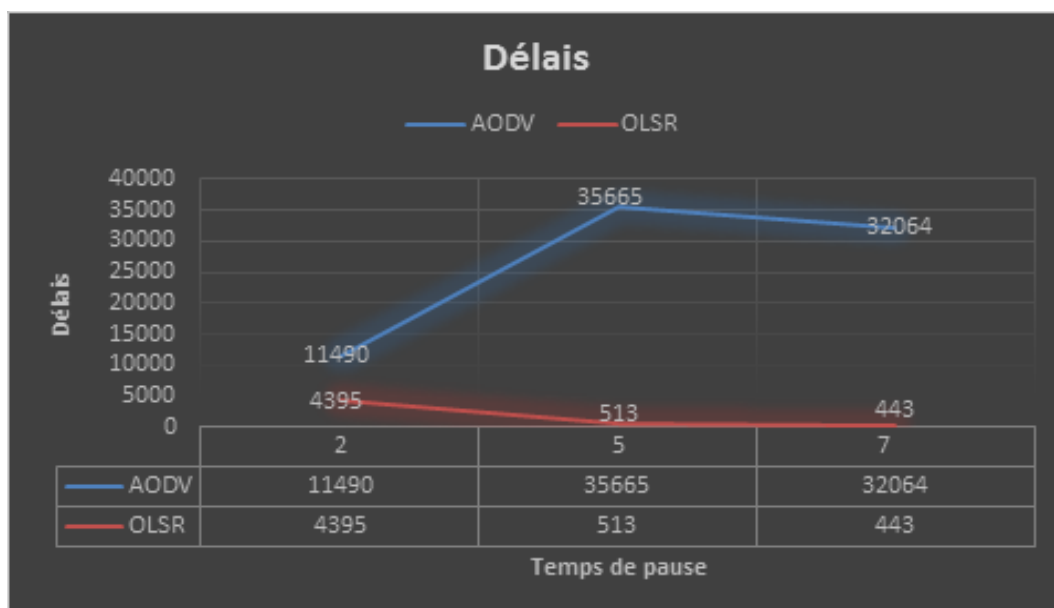


FIG. 3.8 – Le délai de bout en bout .

D’après le graphe, le délai de bout en bout de AODV est élevés au début de la simulation mais s’améliore en augmentant le temps .OLSR présente un délai de bout en bout proportionnel à l’augmentation du temps de pause, donc les performances de OLSR sont meilleur que AODV.

3. Nombre de paquet perdus

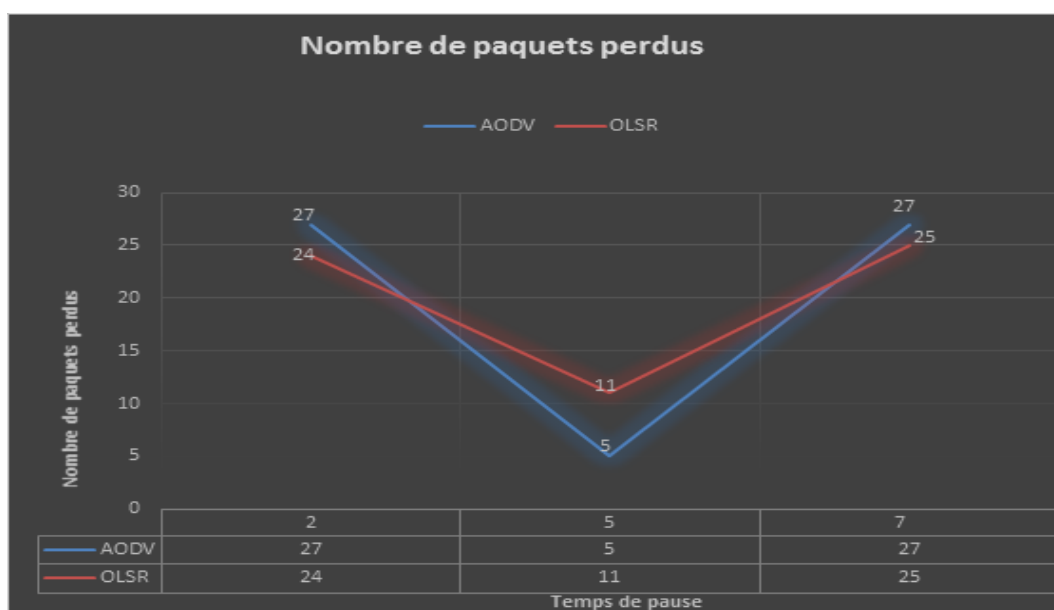


FIG. 3.9 – Nombre de paquet perdus .

D'après le graphe on remarque que pour le temps de pause 2 et 7 la perte des paquets de AODV est plus élevée que OLSR par contre pour le temps de pause 5 le AODV présente moins de perte que OLSR donc les résultats obtenus confirment que le OLSR présente moins de perte de paquet que le AODV.

4. Taux de livraison des paquets

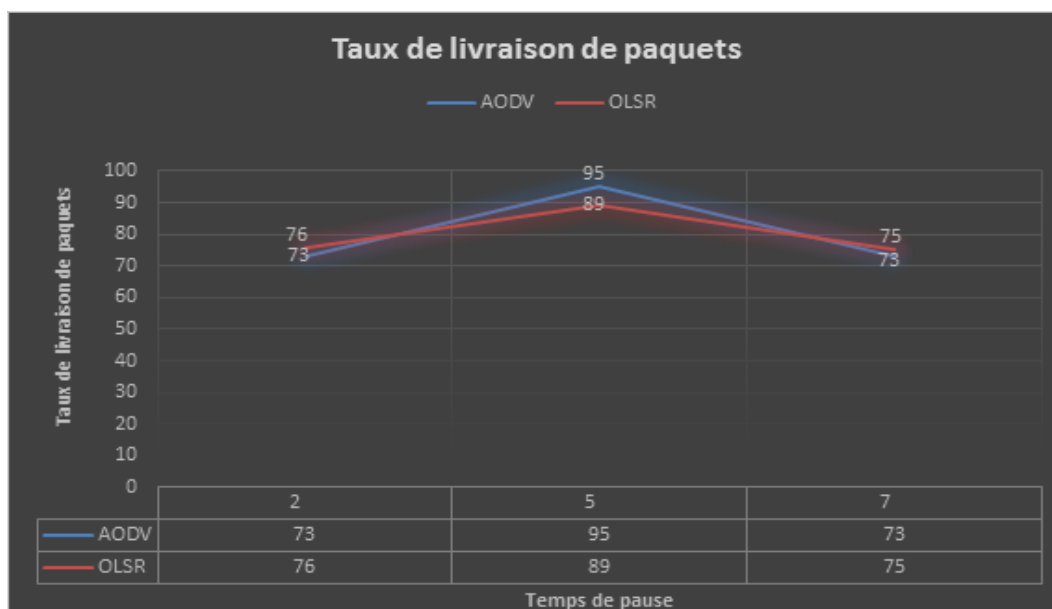


FIG. 3.10 – Taux de livraison des paquets .

Les résultats obtenus confirment que les résultats d'AODV sont élevés par rapport à OLSR pour le temps de pause 5 par contre pour le temps 2 et 7 on constate que les résultats d'OLSR sont élevés par rapport à l'AODV.

3.7 Conclusion

Suite à notre étude comparative entre les deux protocoles AODV qui est un protocole réactif (topologique a vecteur de distance) et OLSR qui est un protocole proactif. Nous avons constaté que le protocole AODV offre de meilleurs résultats.

Les résultats présentés dans notre étude sont conformes à la réalité vu la simulation que nous avons réalisée. De ce fait, on peut conclure que parmi les deux protocoles étudié AODV répond le mieux au besoin des réseaux de capteur sans fil et particulièrement au environnement urbain qui sont à la base de nos simulations.

Conclusion générale

Les réseaux de capteurs sans fil sont une nouvelle technologie qui a surgi après les grands progrès technologiques concernant le développement des capteurs, des processeurs puissants et des protocoles de communication sans fil. Ce type de réseau composé de certains ou de milliers d'éléments, a pour but la collecte de données de l'environnement, leur traitement et leur dissémination vers le monde extérieur. Les applications des réseaux de capteurs sont nombreuses. Elle comprend différents domaines : médicale, agricole, militaire, etc.

Tout au long de notre projet, nous avons constaté que la réalisation d'un RSCF pose de grands défis auxquels il faut répondre ; la fiabilité de ces réseaux est l'un des défis les plus importants à considérer.

Certains travaux de recherche sur ce type de réseaux, sont consacrés à la conception des protocoles de routage ad hoc adaptés à ces réseaux et visant à améliorer la qualité des données.

Dans ce projet nous avons basé sur la simulation des deux protocoles AODV et OLSR avec le simulateur ns-3, nous nous sommes intéressés à l'analyse de leurs performances selon le : le débit, le délai de bout en bout, le taux de paquets délivrés et le taux de paquets perdus.

Les deux protocoles ont été étudiés, simulés et comparés dans des scénarios qui ont donné une vision assez claire sur les contraintes. En effet, les comparaisons entre la simulation prouvent que l'AODV offre un meilleur débit et livraison du paquet. Pour les délais de bout en bout l'OLSR est meilleur, AODV donne moins de perte de paquets que l'OLSR. On constate que globalement l'AODV est le meilleur protocole convenable pour les réseaux de capteurs afin de garantir la rapidité et la transmission sûre et moins de perte des données.

Bibliographie

- [1] F. Brissaud, D. Charpentier, A. Barros et C.Bérenguer, " capteurs intelligents : nouvelles technologies et nouvelles problématiques pour la sureté de fonctionnements ", Maîtrise des risques et de sureté de fonctionnement, Lambda-Mu 16, Avignon : France(2008).

- [2] Vernon S. Somerset, Intelligent and Biosensors, Edited by Vernon S. Somerset Intech,January2010 .

- [3] A. Montoya, D. C. Restrepo et D.A.Ovalle, Artificial Intelligence for Wireless Sensor Networks Enhancement", InTech, 2010.

- [4] Qinghue Wang et Ilangko Balasingham, "Wireless Sensor Networks-An Introduction" Wireless Sensor Network : Application-Centric Design, Geoff V Merrett and Yen Kheng Tan(Ed), ISBN : 978-952-307-321-7, InTech, 2010.

- [5] B. Krishnamachari. Networking Wireless sensors. Cambridge University Press, 2006.

- [6] Gauray Jolly, Mustafa C. Kusçu, Pallavi Kokate et Mohamed Younis, "A Low-Energy Key Management Protocol for Wireless Sensor Network", Proceedinge of the Eighth IEEE International Symposium on Computers and Communication, (ISCC'03), IEEE COMPUTER SOCIETY ? 2003.

- [7] S. Sentilles, " Architecture logicielle pour capteurs sans fils en réseau ", rapport de recherche, Université de Pau et des pays de l'Adour, juin 2006.

- [8] Jean-Philippe Fassino. THINK : vers une architecture de systèmes flexibles. laboratoire Architecture des Systèmes Répartis de la Direction des Techniques Logicielles de France

Télécom Recherche & Développement 2001.

- [9] F. Akyildiz, Weilian Su, Sankarasubramaniam, E. Cayirci, "A survey on sensor networks", IEEE Communications, Aug 2002.
- [10] C. Schurgers and M.B. Srivastava, "Energy efficient routing in wireless sensor networks", MILCOM Proceedings on Communications for Network-Centric Operations, 2001.
- [11] Callaway, Edgar H., 'Wireless Sensor Networks : Architectures and Protocols', Auerbach Publications (2003).
- [12] Al-Karaki, J. N. and Kamal, A. E., "Routing Techniques in Wireless Sensor Networks : A Survey," IEEE Wireless Communications, Vol. 11, pp. 6-28 (2004).
- [13] E. Royer, and C. Toh, "A Review of Current Routing Protocols for ad hoc Mobile Wireless Networks," IEEE Personal Communication, Vol. 6, pp. 46-55 (1999).
- [14] Ioan Raicu "Routing Algorithms for Wireless Sensor Networks", ACM MOBICOM 2002.
- [15] K. Akkaya and M. Younis, "A survey of routing protocols in wireless sensor networks", Elsevier Ad Hoc Network Journal, vol. 3, no. 3, pp. 325-349, May 2005.
- [16] Intanagonwiwat, C., Govindan, R., Estrin, D., "Directed Diffusion : a scalable and robust communication paradigm for sensor networks", ACM Press.
- [17] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Micro sensor Networks", IEEE Transactions on the wireless communications, Vol. 1, No. 4, pp. 660-670, Oct. 2002.
- [18] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan "Energy-efficient communication protocol for wireless micro sensor networks", IEEE Hawaii International Conference on System Sciences, 2000.

- [19] The network simulator ns-2. <http://www.isi.edu/nsnam/ns/>, 2001.
- [20] Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, John Heidemann, Ahmed Helmy, Polly Huang, Steven McCanne, Kannan Varadhan, YaXu, and Haobo Yu. Advances in network simulation. *Computer*, 33(5) :5967, May 2000.ns-2.
- [21] Mobility framework for omnet++. <Http://mobility-fw.sourceforge.net>.
- [22] Omnet++ discrete event simulator. <Http://www.omnetpp.org>.
- [23] A. Vargas. OMNET++ Discrete Event Simulation System, version 2.3 edition, 2003.
- [24] Ahmed Sobeih, Wei-Peng Chen, Jennifer C. Hou, Lu-Chuan Kung, Ning Li, Hyuk Lim, Hung-Ying Tyan, and Honghai Zhang. J-sim : A simulation environment for wireless sensor networks.2005.
- [25] A. Sobeih and J. C. Hou. A simulation framework for sensor networks in J-Sim. Technical Report UIUCDCS-R-2003-2386, Department of Computer Science, University of Illinois at Urbana-Champaign, November 2003.
- [26] "OPNET Modeler". Software. <http://www.opnet.com/products/modeler/home.html>, 2006.
- [27] OPNET Technologies, Inc. OPNET Modeler.
- [28] MAKHOUL Abdellah. (2008). " Réseaux de capteurs : localisation, couverture et fusion de données ".Thèse de doctorat de l'Université de Franche-Comté(LIFC).
- [29] BABOURI Karima, KHELLASI Linda, DJEBROUNI Karima, BESSES Malika. (2013). Les simulateurs réseaux (Technologie réseau). Université des Sciences et de la Technologie Houari Boumediene USTHB. 2013. 19 pages.
- [30] C. E. PERKINS, E. M. ROYER, AND S. R. DAS. Ad hoc on demand distance vector (aodv) routing. In IETF, Internet Draft, draft-ietf-manet-aodv-05.txt, 2000. .

[31] Yaser Yousef. Routage pour la gestion de l'énergie dans les réseaux de capteurs sans fil. Computer Science. Université de Haute Alsace - Mulhouse, 2010. French.

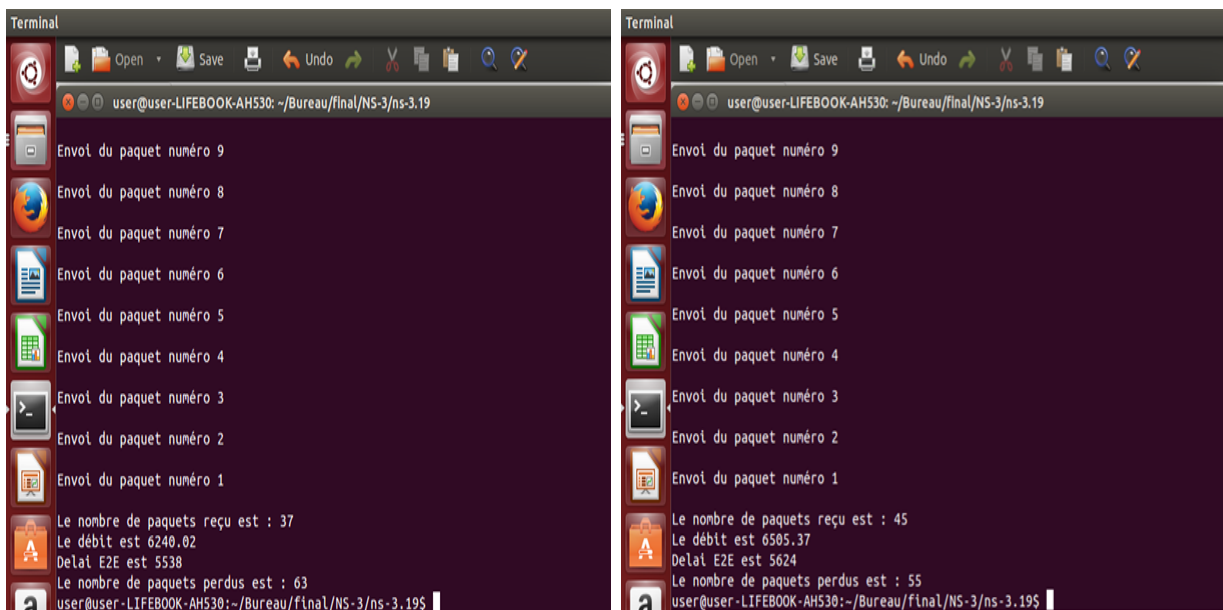
[32] <http://www.nsnam.org/releases/>.

[33] M. Frikha. Planification et simulation des réseaux. Collection performance des réseaux. André-Luc Beylot, 1er Edition, Septembre 2007.

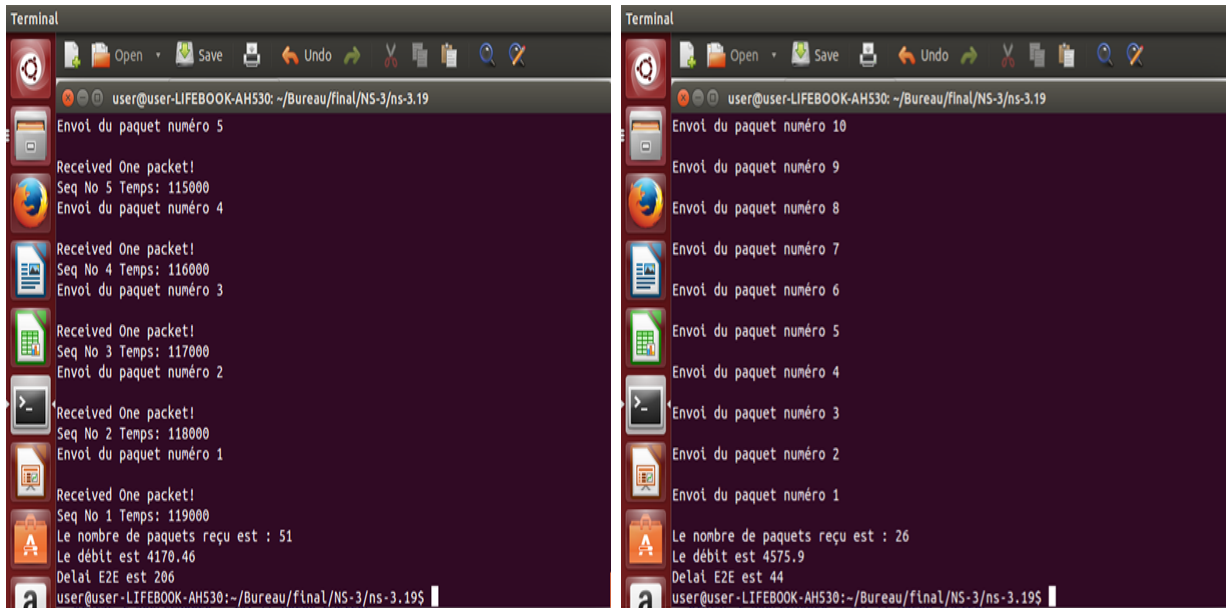


Annexe

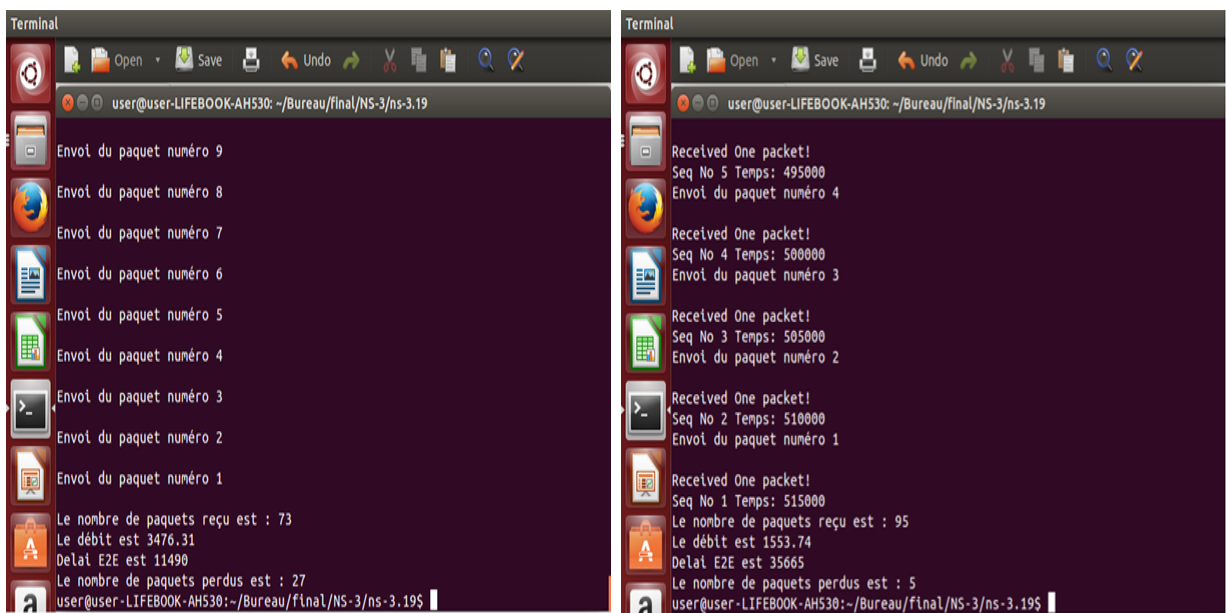
Résultats de simulations



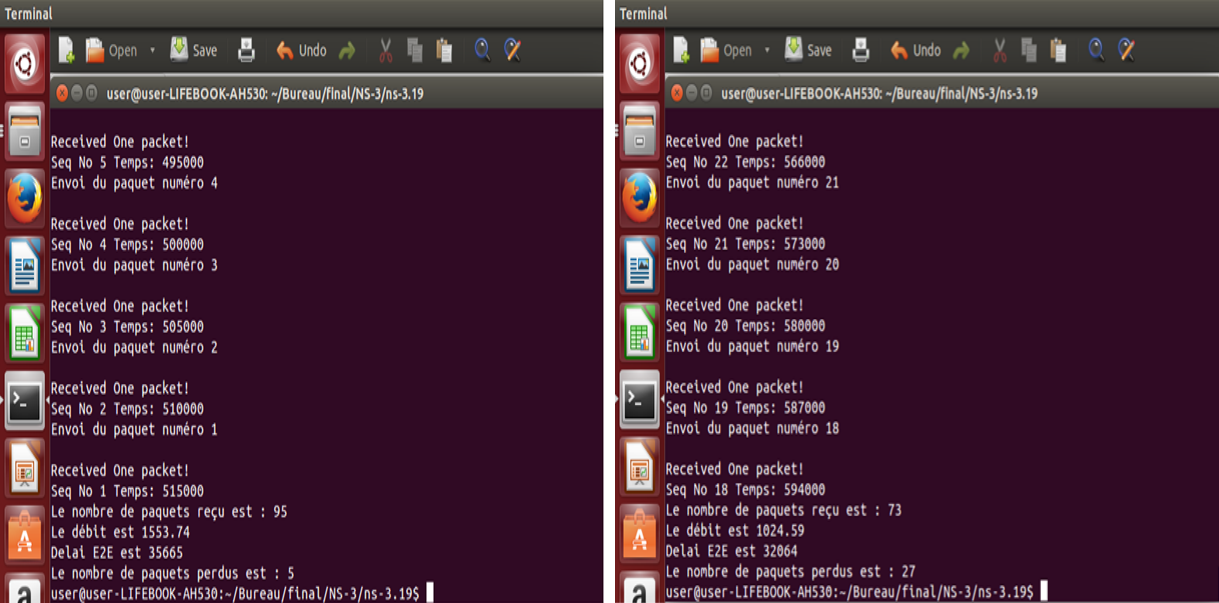
Scénario1(AODV)



Scénario1(OLSR)



Scénario2(AODV)



```
Terminal
user@user-LIFEB00K-AH530: ~/Bureau/final/NS-3/ns-3.19$
Received One packet!
Seq No 5 Temps: 495000
Envoi du paquet numéro 4
Received One packet!
Seq No 4 Temps: 500000
Envoi du paquet numéro 3
Received One packet!
Seq No 3 Temps: 505000
Envoi du paquet numéro 2
Received One packet!
Seq No 2 Temps: 510000
Envoi du paquet numéro 1
Received One packet!
Seq No 1 Temps: 515000
Le nombre de paquets reçu est : 95
Le débit est 1553.74
Delai E2E est 35665
Le nombre de paquets perdus est : 5
user@user-LIFEB00K-AH530:~/Bureau/final/NS-3/ns-3.19$

Terminal
user@user-LIFEB00K-AH530: ~/Bureau/final/NS-3/ns-3.19$
Received One packet!
Seq No 22 Temps: 566000
Envoi du paquet numéro 21
Received One packet!
Seq No 21 Temps: 573000
Envoi du paquet numéro 20
Received One packet!
Seq No 20 Temps: 580000
Envoi du paquet numéro 19
Received One packet!
Seq No 19 Temps: 587000
Envoi du paquet numéro 18
Received One packet!
Seq No 18 Temps: 594000
Le nombre de paquets reçu est : 73
Le débit est 1024.59
Delai E2E est 32064
Le nombre de paquets perdus est : 27
user@user-LIFEB00K-AH530:~/Bureau/final/NS-3/ns-3.19$
```

Scénario2(OLSR)

```

// La fonction received paquet -> retourne les métriques d'évaluation.
void ReceivePacket (Ptr<Socket> socket)
{
    NS_LOG_UNCOND ("Received One packet!");

    Ptr<Packet> packet ;
    while(packet = socket->Recv())//
    {
        SeqTsxHeader seqTsx;
        packet->RemoveHeader (seqTsx); // séparer l'entete
SeqTsxHeader du paquet
        currentSequenceNumber = seqTsx.GetSeq (); // retourne le
numéro de séquence

        bytesTotal += packet->GetSize (); //byteTotal <- taille
des donnée du paquet (sans entête)
        packetsReceived += 1;//pour calculer le nombre de
paquets reçus

        rcv = Simulator::Now().GetMilliSeconds(); // rcv <- temps
de reception du paquet en cours
        sqhd = seqTsx.GetTs().GetMilliSeconds(); // sqhd <- le
temps d'envoi du paquet

        std::cout << "Seq No " << currentSequenceNumber << "
Temps: " << sqhd << "\n";//Juste pour vérifier le numéro de séquence
et le temps d'envoi du paquet

        delay = delay + (rcv - sqhd); //calcul du delai de bout
en bout (le temps de reception du paquet - le temps d'envoi du paquet)
    }

    xx = rcv - senttime; // temps de reception du dernier
paquet - temps de reception du premier paquet

    throughput = (bytesTotal*1000*8) / (xx); //débit d'envoi
calculé en bit/s
}

```

Les métriques d'évaluation

Script et résultats de simulation

```
//ecrire les résultats de la simulation
std::cout << "Le nombre de paquets reçu est : " << packetsReceived
<< "\n"; // les paquets reçus
std::cout << "Le débit est " << throughput << "\n"; // le débit
std::cout << "Delai E2E est " << delay << "\n"; // delais de
propagation point-à-point
std::cout << "Le nombre de paquets perdus est : "
<< numPackets - packetsReceived << "\n";

.
.

//*****le script de simulation*****
int main (int argc, char *argv[])
{
    double duration=600.0; //temps de la simulation
    std::string phyMode ("OfdmRate6MbpsBW10MHz");// type de modulation
    OFDM 6Mbps BP=10MHZ (802.11p)
    uint32_t packetSize = 1024; // en byte
    uint32_t numPackets = 100; // nombre de paquet
    uint32_t numNodes = 60;//nombre de noeud
    uint32_t sourceNode = 1; // noeud emeteur
```

RÉSUMÉ

Les réseaux de capteurs sans fil est un système embarqué autonome, auto configurable, auto organisable et déployé dans divers applications. Il existe de différents protocoles de routage qui peuvent fournir des avantages importants pour les réseaux de capteurs sans fil en termes à la fois de la performance et de la fiabilité. Beaucoup de protocoles de routage ont été conçus pour les réseaux capteurs sans fil. Mais on a choisi de travailler sur OLSR et AODV qui sont des protocoles conçus à la base pour les réseaux ad hoc afin d'étudier leur adaptabilité au RCSFs. Le simulateur NS-3 a été utilisé pour comparer les performances de ces 2 protocoles en visant les métriques débits, délai de bout en bout, nombre de paquets perdus, et taux de livraison de paquets.

Mots clés : AODV, OLSR, RCSFs, NS3.

ABSTRACT

wireless sensor networks is an embedded system autonomous, self configurable, self organizable and deployed in various applications. There are different routing protocols that can provide significant advantages for wireless sensor networks in terms of both performance and reliability. Many routing protocols have been designed for wireless sensor networks. But we chose to work on OLSR AODV are protocols designed the basis for ad hoc networks to study their adaptability to WSNs. Le NS-3 simulator was used to compare the performance of these two protocols aimed metric flows, end to end delay, packet loss, and packet delivery rate.

Key words : AODV, OLSR, WSNs, NS3.