

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Abderrahmane Mira de Béjaïa



Faculté des Sciences Exacte  
Département d'Informatique  
Mémoire de fin d'études  
En vue de l'obtention du diplôme de Master en Informatique  
Option :Génie logiciel

Thème

---

*Conception et réalisation d'un simulateur pour le  
routage opportuniste dans les RCSFs*

---

**Présenté par :**

Melle AGGAOUA Meriem

Melle AZZOUGUER Dalila

**Soutenu devant le jury composé de :**

|               |                     |                      |
|---------------|---------------------|----------------------|
| Président     | Dr.LAHLAH Souad     | Université de Béjaïa |
| Examineur     | Dr.OUADA Farah-Sara | Université de Béjaïa |
| Promotrice    | Dr.YESSAD Nawel     | Université de Béjaïa |
| Co-promotrice | Dr.TIAB Amal        | Université de Béjaïa |

Année Universitaire :2019-2020

# *Remerciement*

Avant tout développement sur cette expérience professionnelle, il apparaît opportun de commencer ce mémoire par un remerciement à dieu et à ceux qui nous ont conseillé et encouragé car ce mémoire n'aurait pas été possible sans l'intervention d'un grand nombre de personnes, nous souhaitons ici les en remercier..

Nous souhaitons adresser, en premier lieu, un grand remerciement à notre promotrice et co-promotrice **Mme.Yessad Nawel** et **Mme.Tiab Amal** de nous avoir honorés par leur encadrement pour le soutien qu'elles nous ont apporté ,leurs conseils,disponibilité, confiance et orientation durant toute la période de travail qui était vraiment difficile à cause de cette maladie dite Covid 19 (Corona).

Nous remercions aussi les membres de jurys pour leurs patiences et d'avoir accepter d'évaluer notre travail.

Nos remerciements s'étendent aussi à tous nos enseignants et les membres du département d'informatique de l'université **ABDERRAHMANE MIRA**.

Enfin,nous remercions nos familles et toutes les personnes qui nous ont aidé et soutenus toute au long de cette réalisation.

# *Dédicas*

Avec ma profonde gratitude et amour, je dédie ce travail

A mes chers parents, pour tous leurs sacrifices, confiance, soutien et leurs prières tout au long de mes études.

A mes adorables sœurs pour leurs encouragements et leur orientation tout au long de mon parcours, à mes nièces (Chaima, Héliane) et neveux (Alilou, Badis et Abdou) à qui je souhaite beaucoup de réussite et aussi à ma grand-mère qui ne cesse pas de prier pour ma réussite.

A Hafid, Mahmoud et Mounir pour leurs aide et soutient.

A mes chers amis(es) (Nawel, Sara, Dalia, Kahina, Fatima et karima) avec lesquels j'ai partagé mes moments de joie, de bonheur et mes moment difficile .

A toute la famille AGGAOUA et BOUMENIR.

A toute personne m'ayant aidé de près ou de loin.

A ma chère binôme Dalila.

Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infallible.

*A.Meriem*

# *Dédicas*

Avec ma profonde gratitude et amour, je dédie ce travail

A ma très chère mère : aucune dédicace ne saurait exprimer l'affectation et l'amour que j'éprouve pour elle. Que dieu la garde pour moi.

A mon père : pour tous tes nobles sacrifices afin d'assurer notre bien-être et notre éducation.

A ma tante : puisse ce travail être la récompense de ses soutiens moraux et sacrifices.

A mes frères et mes adorables sœurs pour leur encouragement permanents, sans oublier mes petits neveux Idris et Aris.

A tout la famille Azzouguer et Mehdi.

A mon fiancé kouceila qui est toujours là pour moi pour m'écouter et m'orienter au bon chemin et surtout pour sa compréhension et encouragement, sans oublier ma belle famille.

A mes amies (Rachida, Sara, Nawel, Kahina, Fatima...) et tous ceux qui m'ont aidé à réaliser ce travail.

A ma chère binôme Meriem.

A tous ce qui m'a orienté dans la bonne voie pour devenir ce que je suis aujourd'hui.

A tous ceux que j'aime et qui m'aiment.

*A. Dalila*

# Table des matières

|  |          |
|--|----------|
| Table des matières   | i        |
| Table des figures  | v        |
| Liste des tableaux   | vii      |
| Liste des abréviations                                     | viii     |
| Introduction générale                                      | 1        |
| <b>1 Généralités sur les réseaux de capteur sans fil</b>   | <b>3</b> |
| 1.1 Introduction . . . . .                                 | 3        |
| 1.2 Définitions . . . . .                                  | 3        |
| 1.2.1 Définition d'un nœud capteur . . . . .               | 3        |
| 1.2.2 Définition d'un réseau de capteur sans fil . . . . . | 4        |
| 1.3 Architecture . . . . .                                 | 4        |
| 1.3.1 Architecture d'un nœud capteur . . . . .             | 5        |
| 1.3.2 Architecture d'un RCSF . . . . .                     | 6        |
| 1.4 Caractéristiques d'un RCSF . . . . .                   | 6        |
| 1.5 Types des capteurs dans un RCSF . . . . .              | 7        |
| 1.6 Domaines d'application d'un RCSF . . . . .             | 8        |
| 1.6.1 Applications militaires . . . . .                    | 8        |
| 1.6.2 Applications médicales . . . . .                     | 8        |
| 1.6.3 Applications transportés . . . . .                   | 9        |
| 1.6.4 Applications agricoles . . . . .                     | 9        |
| 1.6.5 Applications de sécurité . . . . .                   | 9        |
| 1.6.6 Applications environnementales . . . . .             | 9        |
| 1.6.7 Applications commerciales . . . . .                  | 9        |

|         |   |    |
|---------|---|----|
| 1.7     | Contraintes de conception des RCSFs . . . . .                     | 10 |
| 1.7.1   | Tolérance aux fautes . . . . .                                    | 10 |
| 1.7.2   | Facteur d'échelle (Scalabilité) . . . . .                         | 10 |
| 1.7.3   | Coûts de production . . . . .                                     | 10 |
| 1.7.4   | Environnement . . . . .   | 10 |
| 1.7.5   | Topologie de réseau . . . . .                                     | 10 |
| 1.7.6   | Contraintes matérielles . . . . .                                 | 11 |
| 1.7.7   | Médias de transmission . . . . .                                  | 11 |
| 1.7.8   | Consommation d'énergie . . . . .                                  | 11 |
| 1.8     | Le Routage . . . . .  | 11 |
| 1.8.1   | Définition du routage . . . . .                                   | 11 |
| 1.8.2   | Le routage dans les RCSFs . . . . .                               | 11 |
| 1.8.3   | Classification des protocoles de routage dans les RCSFs . . . . . | 12 |
| 1.8.3.1 | Classification selon la structure du réseau . . . . .             | 13 |
| 1.8.3.2 | Classification selon les fonctions des protocoles . . . . .       | 16 |
| 1.8.3.3 | Classification selon l'établissement de la route . . . . .        | 18 |
| 1.8.3.4 | Classification selon l'initiateur de communication . . . . .      | 19 |
| 1.9     | Simulation dans les RCSFs . . . . .                               | 19 |
| 1.9.1   | Définitions . . . . .   | 19 |
| 1.9.2   | Etapas de simulation . . . . .                                    | 20 |
| 1.9.3   | Quelques simulateurs existants . . . . .                          | 21 |
| 1.10    | Conclusion . . . . .  | 23 |

**2 Le routage opportuniste dans les RCSFs 24**

|         |  |    |
|---------|--|----|
| 2.1     | Introduction . . . . .                       | 24 |
| 2.2     | Notion du routage opportuniste . . . . .     | 24 |
| 2.2.1   | Définition . . . . .                         | 24 |
| 2.2.2   | Avantages du routage opportuniste . . . . .  | 25 |
| 2.2.3   | Principe de fonctionnement . . . . .         | 26 |
| 2.2.4   | Composants du routage opportuniste . . . . . | 26 |
| 2.2.4.1 | Sélection des candidats . . . . .            | 27 |
| 2.2.4.2 | Coordination des candidats . . . . .         | 27 |
| 2.2.4.3 | Filtrage des candidats relais . . . . .      | 28 |
| 2.2.5   | Métriques de routage opportuniste . . . . .  | 28 |

|         |  |    |
|---------|--|----|
| 2.2.6   | Classification des protocoles de routage opportuniste . . . . .    | 29 |
| 2.2.6.1 | Protocoles basés sur la méthode de sélection de l'expéditeur . . . | 29 |
| 2.2.6.2 | Protocoles basé sur le codage réseau . . . . .                     | 30 |
| 2.2.6.3 | Protocoles de routage opportuniste géographiques . . . . .         | 31 |
| 2.2.6.4 | Protocoles de routage opportuniste conscients de l'énergie . . .   | 31 |
| 2.2.6.5 | Protocoles de routage opportuniste temps réel . . . . .            | 32 |
| 2.2.6.6 | Protocoles de routage opportuniste avec qualité de service (QoS)   | 32 |
| 2.3     | Evaluation des performances . . . . .                              | 33 |
| 2.3.1   | Métriques de performance . . . . .                                 | 34 |
| 2.4     | Conclusion . . . . .   | 34 |

**3 Spécification des besoins et Conception 35**

|       |   |    |
|-------|---|----|
| 3.1   | Introduction . . . . .  | 35 |
| 3.2   | Langage de modélisation unifié (UML) . . . . .  | 35 |
| 3.3   | Démarche UP . . . . .   | 35 |
| 3.4   | Analyse des besoins . . . . .   | 36 |
| 3.4.1 | Besoins fonctionnels . . . . .  | 36 |
| 3.4.2 | Besoins non fonctionnels . . . . .  | 36 |
| 3.5   | Diagramme de cas d'utilisation . . . . .  | 36 |
| 3.6   | Diagramme de séquence . . . . .   | 37 |
| 3.6.1 | Diagramme de séquence du cas d'utilisation «Créer une session» . . . . .                          | 38 |
| 3.6.2 | Diagramme de séquence du cas d'utilisation «Ouvrir une session» . . . . .                         | 38 |
| 3.6.3 | Diagramme de séquence du ca d'utilisation «Enregistrer une session» . . .                         | 39 |
| 3.6.4 | Diagramme de séquence du cas d'utilisation «Supprimer une session» . . .                          | 40 |
| 3.6.5 | Diagramme de séquence du cas d'utilisation «Introduire les paramètres<br>du réseau» . . . . .     | 42 |
| 3.6.6 | Diagramme de séquence du cas d'utilisation «Introduire le nombre de<br>nœud» . . . . .            | 43 |
| 3.6.7 | Diagramme de séquence du cas d'utilisation «Introduire le débit des nœuds»                        | 43 |
| 3.6.8 | Diagramme de séquence du cas d'utilisation «Introduire l'énergie initiale<br>des nœuds» . . . . . | 44 |
| 3.6.9 | Diagramme de séquence du cas d'utilisation «Introduire les métriques du<br>routage» . . . . .     | 45 |

|        |  |    |
|--------|--|----|
| 3.6.10 | Diagramme de séquence du cas d'utilisation «Introduire la politique du routage» . . . . .      | 45 |
| 3.6.11 | Diagramme de séquence du cas d'utilisation «Identifier la fonction objectif»                   | 46 |
| 3.6.12 | Diagramme de séquence du cas d'utilisation «Introduire les paramètres de simulation» . . . . . | 47 |
| 3.6.13 | Diagramme de séquence du cas d'utilisation «simuler» . . . . .                                 | 48 |
| 3.7    | Diagramme d'activité . . . . .   | 50 |
| 3.8    | Diagramme de classes . . . . .   | 51 |
| 3.8.1  | Dictionnaire des données . . . . .   | 53 |
| 3.8.2  | Modèle relationnel . . . . .   | 53 |
| 3.9    | Conclusion . . . . .   | 54 |

**4 Réalisation 55**

|       |  |    |
|-------|--|----|
| 4.1   | Introduction . . . . .                             | 55 |
| 4.2   | Environnement et outils de développement . . . . . | 55 |
| 4.2.1 | Java . . . . .                                     | 55 |
| 4.2.2 | IDE Eclipse . . . . .                              | 56 |
| 4.2.3 | WampServer . . . . .                               | 56 |
| 4.2.4 | Structured Query Language (MySQL) . . . . .        | 56 |
| 4.2.5 | Visual Paradigm for UML . . . . .                  | 56 |
| 4.2.6 | TEXstudio . . . . .                                | 57 |
| 4.3   | Présentation de «RoSim» . . . . .                  | 57 |
| 4.3.1 | Interface de gestion des sessions . . . . .        | 57 |
| 4.3.2 | Ouverture d'une session . . . . .                  | 60 |
| 4.3.3 | Gestion les paramètres du réseau . . . . .         | 61 |
| 4.3.4 | Identification des paramètres du routage . . . . . | 61 |
| 4.3.5 | Gérer les paramètres de la simulation . . . . .    | 62 |
| 4.3.6 | Résultat de simulation . . . . .                   | 63 |
| 4.4   | Conclusion . . . . .                               | 64 |

|  |           |
|--|-----------|
| <b>Conclusion générale et perspectives</b> | <b>65</b> |
|--|-----------|

|                     |           |
|---------------------|-----------|
| <b>Bbliographie</b> | <b>67</b> |
|---------------------|-----------|

|               |           |
|---------------|-----------|
| <b>Résumé</b> | <b>72</b> |
|---------------|-----------|



# Table des figures

|      |  |    |
|------|--|----|
| 1.1  | Exemple d'un nœud capteur. . . . .                                     | 4  |
| 1.2  | Réseau de capteurs sans fil. . . . .                                   | 4  |
| 1.3  | Architecture d'un nœud capteur. . . . .                                | 5  |
| 1.4  | Exemple d'une architecture d'un réseau de capteur sans fil. . . . .    | 6  |
| 1.5  | Quelques domaines d'application d'un RCSF. . . . .                     | 8  |
| 1.6  | Classification des protocoles de routage dans les RCSFs. . . . .       | 12 |
| 1.7  | Routage plat. . . . .  | 13 |
| 1.8  | Routage hiérarchique. . . . .  | 14 |
| 1.9  | Routage basé sur la localisation géographique. . . . .                 | 15 |
| 1.10 | Les étapes de simulation . . . . .                                     | 20 |
| 2.1  | Exemple de l'efficacité du routage opportuniste. . . . .               | 25 |
| 2.2  | Fonctionnement du routage . . . . .                                    | 26 |
| 3.1  | Diagramme de cas d'utilisation. . . . .                                | 37 |
| 3.2  | Diagramme de séquence créer une session. . . . .                       | 38 |
| 3.3  | Diagramme de séquence ouvrir une session. . . . .                      | 39 |
| 3.4  | Diagramme de séquence Enregistrer une session. . . . .                 | 40 |
| 3.5  | Diagramme de séquence Supprimer une session. . . . .                   | 41 |
| 3.6  | Diagramme de séquence introduire les paramètres du réseau. . . . .     | 42 |
| 3.7  | Diagramme de séquence introduire le nombre de nœuds. . . . .           | 43 |
| 3.8  | Diagramme de séquence introduire le débit des nœuds. . . . .           | 44 |
| 3.9  | Diagramme de séquence introduire l'énergie initiale des nœuds. . . . . | 44 |
| 3.10 | Diagramme de séquence introduire les métriques du routage. . . . .     | 45 |
| 3.11 | Diagramme de séquence introduire la politique du routage. . . . .      | 46 |
| 3.12 | Diagramme de séquence introduire la fonction objectif. . . . .         | 47 |
| 3.13 | Diagramme de séquence introduire les paramètres de simulation. . . . . | 48 |

|      |  |    |
|------|--|----|
| 3.14 | Diagramme de séquence simuler . . . . .                        | 49 |
| 3.15 | Diagramme d'activité simuler. . . . .                          | 51 |
| 3.16 | Diagramme de classes. . . . .                                  | 52 |
| 4.1  | Fenêtre de gestion des sessions. . . . .                       | 58 |
| 4.2  | Fenêtre menu fichier. . . . .                                  | 58 |
| 4.3  | Fenêtre menu édition. . . . .                                  | 59 |
| 4.4  | Page web. . . . .  | 60 |
| 4.5  | Fenêtre ouvrir un session. . . . .                             | 60 |
| 4.6  | Fenêtre gestion des paramètres du réseau . . . . .             | 61 |
| 4.7  | Fenêtre de gestion des paramètres du routage . . . . .         | 62 |
| 4.8  | Fenêtre de gestion des paramètres de simulation . . . . .      | 62 |
| 4.9  | Fenêtre de résultat de simulation du premier scénario. . . . . | 63 |
| 4.10 | Fenêtre de résultat de simulation du premier scénario. . . . . | 64 |

# Liste des tableaux

- 1.1 Tableau de Comparaison de quelques simulateurs. . . . . 23
- 3.1 Tableau de dictionnaire de données. . . . . 53

# Liste des abréviations

## A

|      |                                  |
|------|----------------------------------|
| ACK  | ACKnowledgment (Acquittement)    |
| ADC  | Analog to Digital Converter      |
| ADV  | ADVertissing                     |
| AODV | Ad hoc On-demand Distance Vector |

## C

|        |  |
|--------|--|
| CodeOR | Opportunistic Routing in Wireless Networks with Segmented Network Coding |
| CORE   | Coding-Aware Opportunistic Routing Mechanism                             |

## D

|    |                    |
|----|--------------------|
| DD | Directed Diffusion |
|----|--------------------|

## E

|        |  |
|--------|--|
| EAR    | Energy Aware Routing for Low Energy Ad Hoc Sensor Networks                     |
| EAX    | Expected Any-path Transmission   |
| EAOR   | Energy Aware Opportunistic Routing protocol                                    |
| EARTOR | Energy-Aware Real-Time Opportunistic Routing protocol                          |
| EFFORT | Energye eFFicient Opportunistic Routing Technology in wireless sensor networks |
| EOT    | Expected One-hop Throughput  |
| ETT    | Expected Transmission Time   |
| ETX    | Expected Transmission Count  |
| ExOR   | Extremely Opportunistic Routing  |

## G

|       |                              |
|-------|------------------------------|
| GPS   | Global Position System       |
| GAF   | Geographic Adaptive Fidelity |
| GeRaF | Geographic Random Forwarding |

|          |  |
|----------|--|
| GOR      | Geographically Opportunistic Routing               |
| GCF      | Geographic Collaborative Forwarding                |
| <b>J</b> |  |
| J-Sim    | Java-SIMultiator                                   |
| <b>L</b> |  |
| LEACH    | Low -Energy Adaptive Clustering Hierarchy          |
| <b>M</b> |  |
| MORE     | Mac independent Opportunistic and Routing Encoding |
| MTS      | Minimum Transmission Selection                     |
| <b>N</b> |  |
| NC       | Nœud Collecteur                                    |
| NS       | Nœud Source  |
| NR       | Nœud Relais  |
| NS-2     | Network Simulator 2                                |
| <b>O</b> |  |
| OEC      | Opportunistic End to end Cost                      |
| OLSR     | Optimized Link State Routing                       |
| ORTR     | Opportunistic Real-time Routing                    |
| <b>P</b> |  |
| PDR      | Paquet Delivery Ratio                              |
| PHP      | Hypertext Preprocessor                             |
| <b>Q</b> |  |
| QoS      | Quality of Service                                 |
| OMNET ++ | Objective Modular NETwork Test-bet in C++          |
| QOR      | QoS oriented Opportunistic Routing Protocol        |
| <b>R</b> |  |
| RCSF     | Réseaux de Capteurs Sans Fil                       |
| REQ      | REQuest  |
| RO       | Routage Opportuniste                               |

## **S**

- SAR Sequential Assignement Routing
- SOAR Simple Opportunistic Adaptive Routing Protocol
- SPIN Sensors Protocols for Information via Négociation

## **T**

- TOSSIM TinyOS SIMulator

## **U**

- UML Unified Modeling Language
- UP Unified Prosses (processus unifié)

## **W**

- WAMP Windows, Apache, MySQL, and PHP

# Introduction générale

Actuellement, les Réseaux de Capteurs Sans Fil (RCSFs) suscitent un intérêt croissant pour les technologies de l'information et de communication (TIC). Ces réseaux ont des atouts manifestes : ils peuvent être déployés très rapidement, couvrir de très grands espaces géographiques et fonctionner sans intervention humaine avec une grande tolérance aux défaillances. En raison de leur flexibilité, de leur faible coût et de la facilité de déploiement, les RCSFs promettent de révolutionner notre vie à travers plusieurs domaines d'applications tels que la détection et la surveillance des désastres, le contrôle de l'environnement et la cartographie de la biodiversité, le bâtiment intelligent, l'agriculture de précision, la surveillance et la maintenance préventive des machines, la médecine et la santé, la logistique et les transports intelligents.

L'acheminement des données dans un RCSF utilise la notion de routage. Ce dernier est le processus qui permet de sélectionner des chemins dans un réseau pour transmettre des données depuis un expéditeur jusqu'à un ou plusieurs destinataires. Ces capteurs sont parfois déployés dans des zones hostiles. Il est donc nécessaire d'avoir une stratégie efficace qui prend en considération l'énergie du réseau pour augmenter sa durée de vie en réduisant la perte d'énergie tout en étant réactive aux changements de l'environnement.

Plusieurs études antérieures dans le domaine des RCSFs ont mis l'accent sur des mécanismes traditionnels de routage, les protocoles dont ils disposent sont généralement caractérisés par un faible débit de données, un faible coût et une faible consommation énergétique pour transmettre ces données d'une source vers une destination. Afin de remédier à tous ces problèmes, un nouveau concept de routage appelé le routage opportuniste est apparu pour bénéficier de tous les avantages offerts.

Le routage opportuniste utilise le concept de diffusion qui caractérise les RCSFs, tel que chaque nœud qui reçoit un paquet et soit proche de la destination peut participer dans l'envoi des paquets reçus, ce qui donne naissance à plusieurs avantages à savoir l'augmentation de la fiabilité et amélioration du débit ainsi qu'une bonne optimisation d'énergie qui est le résultat de détermination efficace des nœuds relais.

Pour étudier ou implémenter de nouvelles idées tel que le routage opportuniste, la mise en place de toute une infrastructure est une procédure lente, les chercheurs ont donc recours aux simulateurs de RCSF. En effet, après avoir vu quelques simulateurs et environnements de simulation ainsi que leurs critères de classification et de qualification, nous pouvons constater que la plupart de ces simulateurs sont destinés à la recherche et à l'émulation de réseaux, leur utilisation nécessite donc un certain savoir-faire et des connaissances assez avancées en modélisation et simulation des RCSFs. De plus, on a constaté l'absence de tout module dédié au routage opportuniste

C'est dans cette optique que s'inscrit notre projet de fin d'étude qui consiste en la conception et la réalisation d'un nouveau simulateur que nous nomme RoSim (Routage Opportuniste-Simulateur) principalement pour le routage opportuniste dans les RCSFs. Pour réaliser cet objectif nous avons opté pour le langage UML pour la spécification des besoins et des différents diagrammes de conception et le langage de programmation orientée objet Java pour l'implémentation et développement des modules de notre simulateur.

Notre simulateur RoSim permet à l'utilisateur de simuler un routage opportuniste en tenant compte de délai de transmission ainsi que l'énergie comme métriques du routage et de visualiser les résultats sous formes graphiques.

Le reste de ce mémoire est structuré comme suit :

**Premier chapitre :** s'intitule généralité sur les réseaux de capteurs sans fil, il a pour objectif de présenter les RCSFs (caractéristiques, Architecture, domaines d'application...) Ainsi que le défi du routage dans les RCSFs et présenter et certains simulateurs présentés dans la littérature.

**Deuxième chapitre :** intitulé le routage opportuniste dans les RCSFs, c'est une présentation de la notion du routage, son fonctionnement, ses composants, ces métriques ainsi qu'une la classification de quelques protocoles.

**Troisième chapitre :** nommé Spécification des besoins et conception, consacré à l'analyse des besoins et la conception de notre simulateur, et dans lequel nous présenterons les différents cas d'utilisation, les diagrammes de séquence respectifs et le diagramme de classe.

**Quatrième chapitre :** s'intitule Réalisation présente les différents outils utilisés pour la réalisation ainsi qu'une présentation de notre simulateur en décrivant ses différentes interfaces.

Nous clôturons le mémoire par une conclusion générale dans laquelle nous exposons quelques perspectives.



# Chapitre 1

## *Généralités sur les réseaux de capteur sans fil*

### 1.1 Introduction

Un réseau de capteurs est une collection de capteurs déployés dans une zone de capture pour prélever des mesures physiques. Dans ce chapitre, nous présentons les concepts de base liés aux réseaux de capteur sans fil. Ainsi, nous avons organisé ce chapitre comme suit. Dans la section 1.2, nous présentons la définition d'un nœud capteur et d'un réseau de capteurs sans fil (RCSF). La section 1.3 présente l'architecture d'un nœud capteur et d'un réseau de capteurs sans fil. La section 1.4 et 1.5 introduisent les caractéristiques d'un RCSF, et les types des capteurs dans un RCSF de ses domaines d'application en section 1.6. Les contraintes de conception et la problématique de routage dans un RCSF sont présentées en sections 1.7 et 1.8 et pour la simulation elle est présentée dans la section 1.9. Le chapitre s'achève par une conclusion dans la section 1.10.

### 1.2 Définitions

Dans cette section nous présentons la définition d'un nœud capteur suivit de la définition d'un réseau de capteurs sans fil.

#### 1.2.1 Définition d'un nœud capteur

Un capteur sans fil est un petit dispositif électronique capable de mesurer une valeur physique environnementale et de la communiquer à un centre de contrôle via une station de base

[1]. Les capteurs sont capables de surveiller une grande variété de phénomènes ambiants notamment : la température, l'humidité, le mouvement des véhicules, la pression, le taux de bruits, la présence ou l'absence de certains types d'objets. En plus, des autres caractéristiques, tels que la vitesse, la direction et le volume d'un objet donné. L'ensemble de ces capteurs déployés pour une application forme un réseau de capteurs [2]. La figure 1.1 montre un exemple de nœud capteur [2].

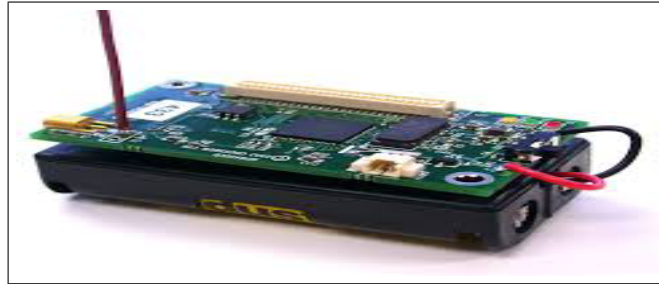


FIGURE 1.1 – Exemple d'un nœud capteur.

### 1.2.2 Définition d'un réseau de capteur sans fil

Un réseau de capteurs sans fil (RCSF) ou Wireless Sensor Network en anglais se définit comme un ensemble de capteurs connectés entre eux où chaque capteur étant muni d'un émetteur-récepteur [3]. En effet, les RCSFs sont considérés comme un type spécial des réseaux Ad Hoc où l'infrastructure fixe de communication et l'administration centralisée sont absentes et les nœuds jouent, à la fois, le rôle des hôtes et des routeurs [4]. La figure 1.2 montre une vue globale sur les réseaux de capteurs sans fil [4].

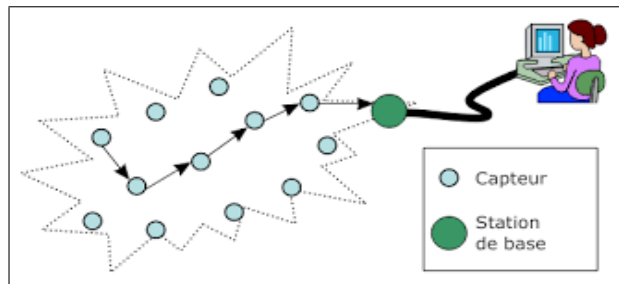


FIGURE 1.2 – Réseau de capteurs sans fil.

## 1.3 Architecture

Dans cette section nous présentons l'architecture d'un nœud capteur suivit de l'architecture d'un réseau de capteurs sans fil.

### 1.3.1 Architecture d'un nœud capteur

Un capteur est composé de quatre éléments ou modules de base qui sont : l'unité de captage (Sensing Unit), l'unité de traitement (Processing Unit), l'unité de transmission (Transceiver Unit) et l'unité de contrôle d'énergie (Power Unit) [5]. La figure 1.3 ci-dessous illustre l'architecture d'un nœud capteur [5].

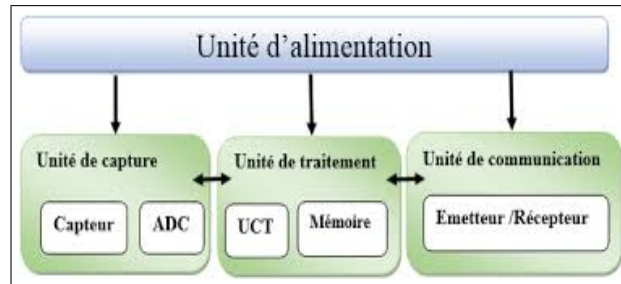


FIGURE 1.3 – Architecture d'un nœud capteur.

#### ★ Unité de capture

Elle est composée de deux sous unités, un dispositif de capture physique qui prélève l'information de l'environnement local et un convertisseur analogique/numérique appelé ADC (Analog to Digital Converter). Ce dernier transforme ces signaux en un signal numérique compréhensible par l'unité de traitement [5].

#### ★ Unité de traitement

Elle comprend un processeur associé généralement à une petite unité de stockage et fonctionne à l'aide d'un système d'exploitation spécialement conçu pour les micro-capteurs. Cette unité est chargée d'exécuter les protocoles de communication qui permettent de faire collaborer le nœud avec les autres nœuds du réseau. Elle peut aussi analyser les données captées pour alléger la tâche du nœud puits [6].

#### ★ Unité de transmission (communication)

Elle est composée d'un émetteur/récepteur (module radio) permettant la communication entre les différents nœuds du réseau [5].

#### ★ Unité de contrôle d'énergie (alimentation)

C'est la batterie qui, n'est généralement ni rechargeable ni remplaçable. La capacité d'énergie limitée au niveau des capteurs représente la contrainte principale lors de la conception de protocoles pour les réseaux de capteurs. Les unités d'énergie peuvent être supportées par des photopiles qui permettent de convertir l'énergie lumineuse en courant électrique [5].

### 1.3.2 Architecture d'un RCSF

Les nœuds capteurs sont habituellement dispersés dans une zone de capture, ces nœuds sont organisés en champs appelé Zone d'intérêt en anglais dite « Sensor Fields ». Chaque nœud a la possibilité de collecter les données et de les router (transférer) vers un point de collecte appelé station de base ou puits dit « SINK en Anglais » du réseau de communication par l'intermédiaire d'une architecture multi-sauts. Le puits transmet ensuite les données capturées à une machine puissante via Internet ou par satellite. En outre, l'utilisateur peut à son tour utiliser la station de base comme passerelle, afin de transmettre ses requêtes au réseau de capteurs [6] [7]. L'architecture d'un réseau de capteurs sans fil est illustrée dans la figure suivante [6] :

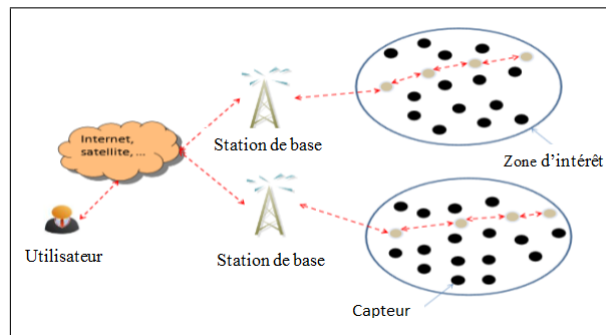


FIGURE 1.4 – Exemple d'une architecture d'un réseau de capteur sans fil.

## 1.4 Caractéristiques d'un RCSF

Parmi les caractéristiques les plus importantes d'un réseau de capteurs, nous citons :

- **Durée de vie limitée**

Les capteurs sont très limités par la contrainte d'énergie, ils fonctionnent habituellement sans surveillance dans des régions géographiques éloignées. De ce fait, il est impossible de remplacer ou recharger leurs batteries [8].

- **Ressources limitées**

Habituellement les capteurs ont une taille très petite, ce facteur de forme limite la quantité de ressources qui peuvent être intégrées dans les capteurs. Par conséquent, la capacité de traitement et de stockage devient très limitée [8].

- **Topologie dynamique**

Comme les capteurs et la station de base peuvent être mobiles, et qu'ils peuvent être déployés dans des environnements hostiles, où la défaillance d'un capteur est très probable, alors la topologie du réseau est fréquemment changeable [8].

- **Auto-organisation**

Pour remédier au problème de changement non prédictive de topologie, une auto organisation du réseau s'avère nécessaire, c'est-à-dire que les capteurs doivent être capables de localiser leurs voisins et d'établir des routes pour que l'information puisse circuler sur le réseau [9].

- **Absence d'adressage fixe**

Les nœuds dans les réseaux sans fil classiques sont identifiés par des adresses IP. Cependant, cette notion n'existe pas dans les RCSFs. En effet, les requêtes des utilisateurs ne sont pas généralement destinées à un seul nœud, mais plutôt, à un ensemble de nœuds identifiés [9].

## 1.5 Types des capteurs dans un RCSF

Selon l'application et la structure choisie, un RCSF peut contenir différents types de nœuds, pour optimiser certains paramètres comme la durée de vie du réseau ou le délai de livraison des données. Certains travaux se sont focalisés sur l'architecture (plat, hiérarchique, multi-niveaux) des RCSF. Ces architectures définissent souvent les rôles joués par les nœuds dans un RCSF. Nous distinguons principalement 3 rôles à savoir :

- ✓ **Nœud Source (NS)**

Le rôle principal du nœud source est de détecter les phénomènes physiques ou physiologiques qui se produisent dans son environnement afin de les transmettre, directement ou via multiples sauts, à un utilisateur final. C'est en fait un nœud capteur [10].

- ✓ **Nœud Relais (NR)**

Les nœuds relais ont pour rôle d'agréger et de retransmettre les mesures provenant des NS afin que celles-ci parviennent à un utilisateur final. Dans une architecture à plat, les NS sont considérés comme NR. Dans une architecture à 2 niveaux, un nœud passerelle joue le rôle de NR pour un ou plusieurs nœuds sources. Dans ce type de configuration réseau, la capacité de transmission du NR est supposée généralement plus grande que celle du NS [10].

- ✓ **Nœud Collecteur de données (NC)**

Les nœuds collecteurs ont pour rôle de collecter les mesures provenant des nœuds sources et éventuellement de les agréger. Généralement, un "Cluster-Head " ou chef de cluster est utilisé comme NC dans une architecture hiérarchique où les NS sont partitionnés en plusieurs groupes [10].

## 1.6 Domaines d'application d'un RCSF

Les RCSFs peuvent être appliqués à la plupart des environnements, en particulier ceux dans lesquels les systèmes conventionnels de capteurs câblés sont irréalisables ou indisponibles, comme dans les terrains inhospitaliers, les champs de bataille, l'espace ou les profondeurs des océans. La figure suivante représente quelques domaines [9] :

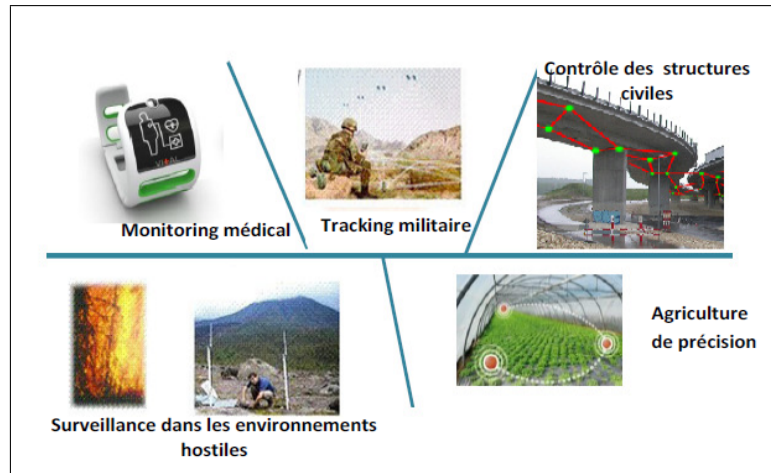


FIGURE 1.5 – Quelques domaines d'application d'un RCSF.

### 1.6.1 Applications militaires

Comme pour de nombreuses autres technologies, le domaine militaire a été le 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 font de ce type de réseaux un outil appréciable dans un tel domaine. Actuellement, les RCSFs peuvent être une partie intégrante dans le commandement, le contrôle, la communication, la surveillance, la reconnaissance, etc. [4].

### 1.6.2 Applications médicales

Les réseaux de capteurs sont également largement répandus dans le domaine médical. Cette classe inclut des applications comme : fournir une interface d'aide pour les handicapés, collecter des informations physiologiques humaines de meilleure qualité, facilitant ainsi le diagnostic de certaines maladies, surveiller en permanence les malades et les médecins à l'intérieur de l'hôpital [9].

### 1.6.3 Applications transportés

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 [9].

### 1.6.4 Applications agricoles

Dans le domaine de l'agriculture, les capteurs peuvent être utilisés pour réagir convenablement aux changements climatiques par exemple le processus d'irrigation lors de la détection de zones sèches dans un champ agricole [9].

### 1.6.5 Applications de sécurité

L'application des réseaux de capteurs dans le domaine de la sécurité peut diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et des êtres humains. Ainsi, l'intégration des capteurs dans de grandes structures telles que les ponts ou les bâtiments aidera à détecter les fissures et les altérations dans la structure suite à un séisme ou au vieillissement de la structure [4].

### 1.6.6 Applications environnementales

Dans ce domaine, les capteurs peuvent être exploités pour détecter les catastrophes naturelles (feux de forêts, tremblements de terre, etc.), des produits toxiques (gaz, produits chimiques, pétrole, etc.) dans des sites industriels tels que les centrales nucléaires et pétrolières [9].

### 1.6.7 Applications commerciales

Des nœuds capteurs peuvent être utilisés pour améliorer les processus de stockage et de livraison. Le réseau peut ainsi être utilisé pour connaître la position, l'état et la direction d'une marchandise. Un client attendant une marchandise peut alors avoir un avis de livraison en temps réel et connaître la position des marchandises qu'il a commandées [2].

## 1.7 Contraintes de conception des RCSFs

Les principaux facteurs et contraintes influençant l'architecture des réseaux de capteurs peuvent être résumés comme suit :

### 1.7.1 Tolérance aux fautes

La tolérance aux fautes est la capacité de maintenir les fonctionnalités du réseau en présence de fautes. La fiabilité des réseaux de capteurs sans fil est affectée par des défauts qui se produisent à cause de diverses raisons, telles que le mauvais fonctionnement du matériel ou cause d'un manque d'énergie. Ces problèmes n'affectent pas le reste du réseau [11].

### 1.7.2 Facteur d'échelle (Scalabilité)

Le nombre de nœuds de capteurs augmente sur un réseau sans fil et ce nombre peut atteindre un million. Un nombre aussi important de nœuds engendre beaucoup de transmissions entre les nœuds et peut imposer des difficultés pour le transfert de données [11].

### 1.7.3 Coûts de production

Souvent les réseaux de capteurs sont composés d'un très grand nombre de nœuds. Le prix d'un nœud est critique afin de pouvoir concurrencer un réseau de surveillance traditionnel [11].

### 1.7.4 Environnement

Les capteurs sont souvent déployés en masse dans des endroits tels que des champs de bataille, à l'intérieur de grandes machines, au fond d'un océan, dans des champs biologiquement ou chimiquement souillés. Par conséquent, ils doivent pouvoir fonctionner sans surveillance dans des régions géographiques éloignées [11].

### 1.7.5 Topologie de réseau

Le déploiement d'un grand nombre de nœuds nécessite une maintenance de la topologie. Cette maintenance consiste en trois phases : déploiement, post-déploiement (les capteurs peuvent bouger, ne plus fonctionner,...) et redéploiement de nœuds additionnels [12].



### 1.7.6 Contraintes matérielles

La principale contrainte matérielle est la taille du capteur. Les autres contraintes sont la consommation d'énergie qui doit être moindre pour que le réseau survive le plus longtemps possible, qu'il s'adapte aux différents environnements (Fortes chaleurs, eau,...), qu'il soit autonome et très résistant vu qu'il est souvent déployé dans des environnements hostiles [12].

### 1.7.7 Médias de transmission

Dans un réseau de capteurs, les nœuds sont reliés par une architecture sans fil. Pour permettre des opérations sur ces réseaux dans le monde entier, le média de transmission doit être standardisé. On utilise le plus souvent l'infrarouge, le Bluetooth et les communications radio Zig Bee [11].

### 1.7.8 Consommation d'énergie

Un capteur, de par sa taille, est limité en énergie ( $<1.2$  joule). Dans la plupart des cas le remplacement de la batterie est impossible. Ce qui veut dire que la durée de vie d'un capteur dépend grandement de la durée de vie de la batterie. Dans un réseau de capteurs (multi-sauts) chaque nœuds collecte des données et envoie/transmet des valeurs. Le dysfonctionnement de quelques nœuds nécessite un changement de la topologie du réseau et un re-routage des paquets. Toutes ces opérations sont gourmandes en énergie, c'est pour cette raison que les recherches actuelles se concentrent principalement sur les moyens de réduire cette consommation [11].

## 1.8 Le Routage

### 1.8.1 Définition du routage

Le routage est un processus qui permet de sélectionner des chemins dans un réseau pour transmettre des données depuis un expéditeur jusqu'à un ou plusieurs destinataires. On distingue généralement deux entités : L'algorithme de routage et le protocole de routage [13].

### 1.8.2 Le routage dans les RCSFs

Dans les réseaux de capteurs sans fil, le routage constitue un service important. Il permet le transfert des paquets de données à partir des nœuds capteurs vers le nœud puits. Ces nœuds sont déployés dans la région à surveiller d'une manière très dense, ce qui les rend très proches les uns

des autres. Pour ce genre de situations, plusieurs techniques de communication sont utilisées. De plus, ces nœuds doivent consommer moins d'énergie en transmettant les données vers le nœud puits. Par conséquent, l'utilisation des protocoles de routage conçus pour les réseaux ad hoc traditionnels est inappropriée pour ces réseaux de capteurs. Ceci, est dû principalement aux caractéristiques qui distinguent ces deux types de réseaux. Il a donc fallu concevoir de nouveaux protocoles de routage spécifiques aux réseaux de capteurs. Pour cela, de nombreux travaux ont été effectués, et plusieurs protocoles ont été proposés pour permettre un routage à basse consommation d'énergie. Par ailleurs, ces protocoles sont influencés par de multiples contraintes dans leur conception. Il est à noter que plusieurs classifications ont été dédiées à ces protocoles [14].

### 1.8.3 Classification des protocoles de routage dans les RCSFs

Dans un réseau de capteurs, les nœuds sont déployés d'une manière dense dans un champ de captage. Pour permettre la communication dans le réseau déployé, des protocoles de routage spéciaux basés sur la communication multi-sauts sont nécessaires entre les nœuds capteurs et le nœud puits du réseau. Le principe de fonctionnement de chaque protocole diffère suivant la philosophie de l'approche à laquelle il appartient. Ces approches peuvent être distinguées suivant : la structure du réseau, les fonctions des protocoles, l'établissement des routes et l'initiateur de la communication [6]. La figure suivante montre la classification des différentes approches de routage dans les RCSFs selon les critères cités précédemment :

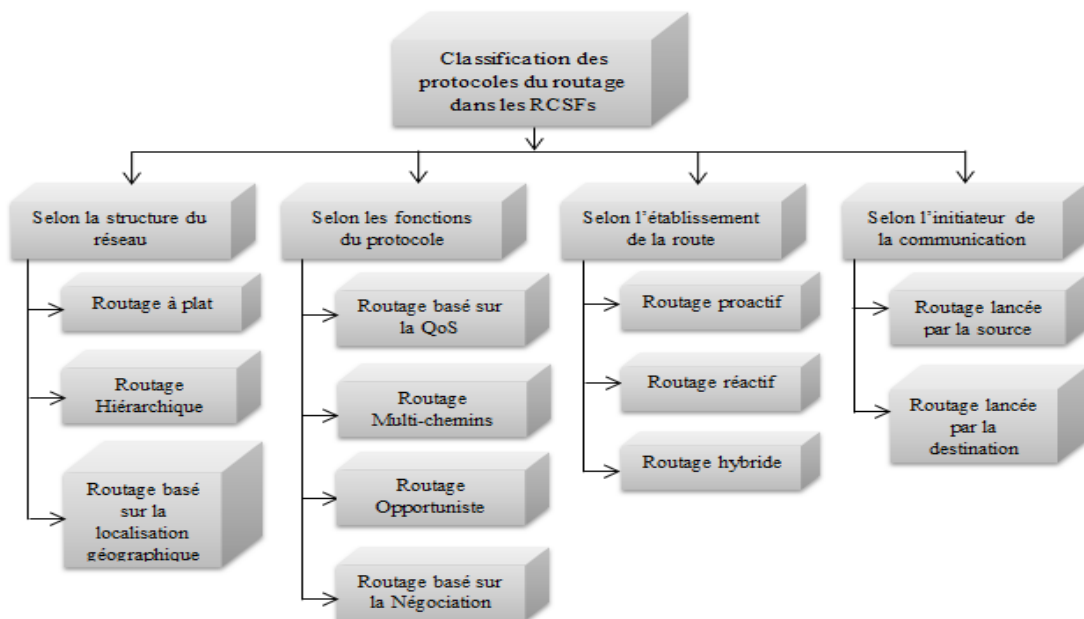


FIGURE 1.6 – Classification des protocoles de routage dans les RCSFs.

### 1.8.3.1 Classification selon la structure du réseau

La topologie détermine l'organisation logique adaptée par les protocoles de routage afin d'exécuter les différentes opérations de découverte de routes et de transmission de données.

Elle joue un rôle significatif dans le fonctionnement d'un protocole. Les protocoles de cette classe peuvent être classifiés en trois catégories : routage à plat, routage hiérarchique et routage basé sur la localisation géographique [15].

#### 1. Routage à plat

La première catégorie des protocoles de routage est celle des protocoles de routage à plat, ces protocoles considèrent que tous les nœuds sont identiques, c'est à dire ont les mêmes fonctions à exécuter sauf le nœud de contrôle (sink) qui est chargé de collecter toutes les informations issues des différents nœuds capteurs pour les transmettre vers l'utilisateur final. La décision d'un nœud de router des paquets vers un autre dépendra de sa position et pourra être remise en cause au cours du temps [15]. Un exemple est donné dans la figure suivante [15] :

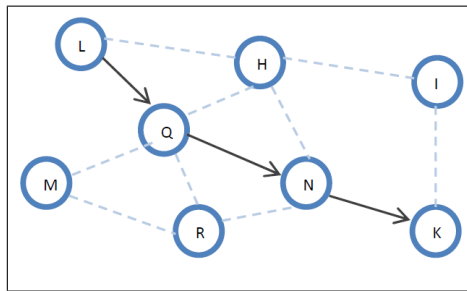


FIGURE 1.7 – Routage plat.

Quelques exemples de protocoles pour ce type de routage :

#### \* Flooding

La technique du flooding (Inondation) est un mécanisme classique pour transmettre par relais des données dans les réseaux de capteurs. Dans cette approche, chaque nœud recevant un paquet de données le diffuse à tous ses nœuds voisins, jusqu'à ce que le nombre maximum de sauts pour ce paquet soit atteint (inondation de tout le réseau). Le flooding est une technique réactive qui ne nécessite pas une maintenance coûteuse de la topologie du réseau, ni des algorithmes complexes pour la découverte des routes, mais elle présente plusieurs inconvénients tel que : l'implosion, le chevauchement et l'ignorance des ressources d'énergie [6].

#### \* Gossiping

Le Gossiping est une version légèrement améliorée du Flooding où un nœud recevant un message ne le diffuse pas à tous ses voisins, mais il le transmet à un seul sélectionné aléatoirement, jusqu'à ce que les données atteignent la station de base. Bien que cette approche évite

le problème d'implosion en ayant une seule copie du message au niveau de chaque nœud, elle prend beaucoup plus de temps pour propager les messages dans tout le réseau [6].

\* Directed Diffusion (DD )

Directed diffusion est considérée comme un protocole de référence dans le domaine du routage centré données. Il comprend quatre opérations principales :

– Dissémination d'intérêt : dans cette première étape, la station de base réclame les données requises par l'envoi de son intérêt sous forme de requête vers les nœuds du réseau.

– Propagation des données : les nœuds capteurs sources intéressés par une requête envoient leurs premiers paquets de l'information, avec un taux de transmission faible, pour explorer les chemins possibles entre les nœuds source et la station de base.

– Renforcement des routes : quand la station de base reçoit les premiers paquets envoyés par les nœuds sources, elle envoie le message original d'intérêt à travers le chemin sélectionné avec un taux de transfert plus élevé.

– Maintenance des routes : quand un chemin employé devient défaillant, une nouvelle route alternative doit être identifiée. Pour cela, le protocole réinitialise la phase de renforcement par une recherche parmi les autres routes qui envoient les données avec un taux d'émission faible [6].

2. Routage hiérarchique

Ces protocoles fonctionnent en confiant des rôles différents aux nœuds du réseau. Certains nœuds sont sélectionnés pour exécuter des fonctions particulières. Un nœud peut être, par exemple, une passerelle pour un ensemble de nœuds. Dans ce cas, le routage devient plus simple, puisqu'il s'agit de passer par les passerelles pour atteindre le nœud destination qui lui est directement attaché. Un exemple est donné par la figure 1.7 : Pour que les paquets générés par le nœud F atteignent le nœud L, ils doivent passer par les passerelles P, S et R [15].

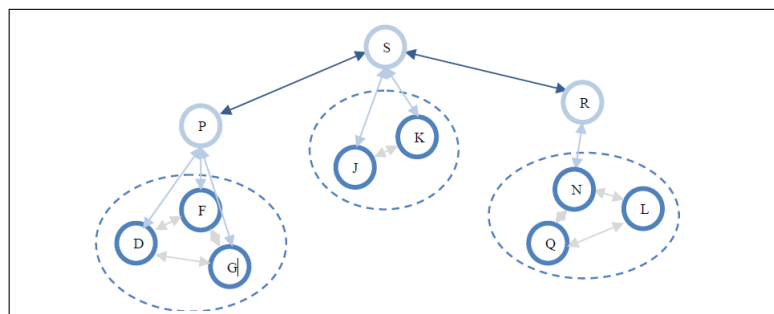


FIGURE 1.8 – Routage hiérarchique.

Quelques exemples de protocoles pour ce type de routage :

\* LEACH(*Low -Energy Adaptive Clustering Hierarchy*)

C'est l'un des Protocoles les plus populaires pour les réseaux de capteurs sans fil. Son principe est de former des zones communes de calcul et de traitement en se basant sur la puissance du signal et le niveau d'énergie des nœuds capteurs. Chaque zone est dirigée par un chef de zone, jouant le rôle d'agrégateur et de routeur, en effectuant des traitements sur les données reçues de son cluster et leur expédition vers la prochaine destination. Ce rôle de chef de zone est échangé entre les nœuds d'un cluster afin de répartir équitablement la consommation d'énergie entre eux [11].

\* Power-Efficient GATHERing in Sensor Informations System (PEGASIS)

C'est un autre protocole de routage conçu pour les réseaux de capteurs, il a été proposé comme amélioration du protocole LEACH. Dans ce protocole, un nœud peut seulement communiquer avec son voisin le plus proche, et doit attendre son tour pour pouvoir transmettre à la station de base.

La motivation principale de ce protocole, consiste à prolonger la durée de vie des nœuds capteurs et réduire la bande passante consommée en utilisant la collaboration locale entre les nœuds et en tolérant la défaillance des nœuds capteurs. Initialement, le nœud capteur doit trouver son voisin le plus proche par l'envoi d'un signal et l'atténuation graduelle de ce dernier, jusqu'à ce qu'il soit reçu par un seul nœud. Les données captées sont transmises à la station de base en utilisant la fusion qui combine deux paquets de données ou plus et les envoie dans un seul paquet, ce qui est similaire au protocole LEACH [6].

### 3. Routage basé sur la localisation géographique

Un routage est dit géographique lorsque les décisions du routage sont basées sur la position des nœuds. Ce type est illustré dans la figure suivante [15] :

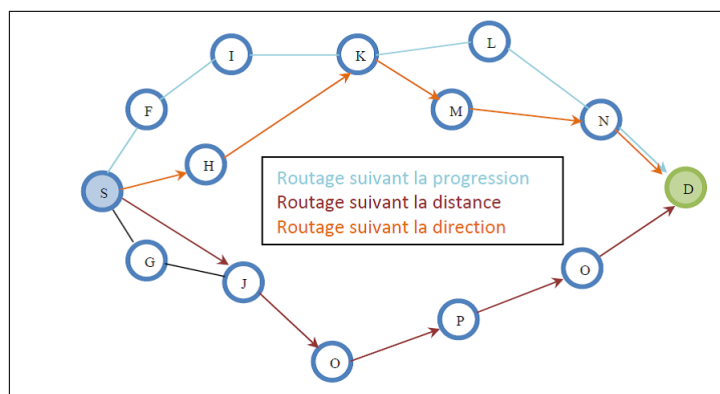


FIGURE 1.9 – Routage basé sur la localisation géographique.

Les prés-requis pour effectuer un routage géographique dans un réseau ad hoc sont [15] :

- Tous les nœuds possèdent un moyen de localisation, soit un système natif comme le GPS (Global Position System), soit un système logiciel comme un protocole de localisation.

- Un nœud source connaît toujours la position du nœud destinataire. Pour ce faire, soit tous les nœuds connaissent les positions initiales de tous les nœuds, soit un service de localisation doit être utilisé. Exemple de protocoles pour ce type de routage :

- \* Geographic Adaptive Fidelity (GAF)

GAF est un protocole de routage, basé sur la localisation géographique, efficace en consommation d'énergie. Ce protocole a été initialement destiné aux réseaux ad hoc, néanmoins il peut être appliqué aux réseaux de capteurs. L'idée principale de GAF consiste à éteindre l'ensemble des nœuds inutiles dans le réseau sans affecter le niveau de fidélité de l'opération du routage. Pour cela, le protocole forme une grille virtuelle à travers le champ de captage, où chaque nœud utilise le service GPS pour pouvoir se positionner, et s'associer à un point particulier de la grille construite. Les nœuds se trouvant dans la même zone de la grille sont considérés équivalents en terme de coût lié au routage des paquets. Cette équivalence est par la suite exploitée pour pouvoir mettre en veille certains nœuds associés au même point de la grille, afin d'optimiser l'énergie consommée. Ceci dit, GAF peut substantiellement augmenter la durée de vie du réseau au fur et à mesure que le nombre de nœuds augmente [6].

### 1.8.3.2 Classification selon les fonctions des protocoles

Le mode de fonctionnement définit la manière avec laquelle les données sont propagées dans le réseau. Selon ce critère, les réseaux de capteurs sans fils peuvent être regroupés en quatre catégories : Routage basé sur la Qualité de service QoS (Quality of Service), routage multi-chemins (Multipath), routage opportuniste et le routage basé sur la négociation (Négociation) [16].

#### 1. Routage basé sur QoS

Ce type de protocoles tend à satisfaire certaines métriques, pendant la transmission des données vers la destination finale. Parmi ces métriques, nous citons : le délai de bout en bout, la gigue, l'énergie consommée [15].

Exemple de protocoles pour ce type de routage :

- \* Sequential Assignment Routing (SAR)

SAR est une approche multi-chemins qui s'efforce à réaliser l'efficacité énergétique et la tolérance aux fautes. SAR crée des arbres en prenant en compte les métriques QoS, la ressource

énergétique sur chaque chemin et le niveau de priorité de chaque paquet. En utilisant ces arbres, des routes multiples du sink aux capteurs sont formés. Une ou plusieurs routes peuvent, alors, être emprunter [15].

## 2. Routage multi-chemins

Il se base sur l'adoption de plus qu'un chemin menant vers la destination, et ce, pour avoir des chemins de secours si jamais le chemin principal serait rompu [15].

Exemple de protocoles pour ce type de routage :

- \* Energy Aware Routing for Low Energy Ad Hoc Sensor Networks (EAR)

L'utilisation répétée des routes à consommation d'énergie optimale, épuiserait rapidement l'énergie des nœuds faisant partie de ces routes. Pour remédier à cette lacune, un chemin alternatif est utilisé avec une certaine probabilité. EAR adopte ce principe en maintenant un ensemble de routes entre la source et la destination, mais n'utilise qu'une seule route à chaque fois. A chaque route est assignée une probabilité dépendante d'une métrique d'énergie. Les nœuds du réseau relaient les paquets reçus en choisissant une route selon les probabilités affectées. EAR s'exécute sur trois phases : initialisation, communication de données et maintenance de route [14].

## 3. Routage opportuniste

Le routage est un processus qui permet de sélectionner des chemins dans un réseau pour transmettre des données depuis un expéditeur jusqu'à un ou plusieurs destinataires [14].

Exemple de protocoles pour ce type de routage :

- \* Coding-Aware Opportunistic Routing Mechanism (CORE)

Le protocole CORE est basé sur le codage réseau, mais l'ordre des relais candidats dans la liste est établi de façon dynamique en fonction des opportunités de codage [9].

- \* Opportunistic Routing in Dynamic Ad Hoc Networks (OPRAH)

OPRAH construit un ensemble de chemin sous forme de tresse multi-chemins depuis la source vers la destination par l'intermédiaire du cheminement sur demande pour expédier les données à travers le routage opportuniste [9].

Ce type de routage sera détaillé dans le chapitre suivant.

## 4. Routage basé sur la négociation

Quand le réseau est inondé par des paquets de même données, on utilise le protocole de négociation qui met fin à la redondance de données, en effet avant de transmettre, les nœuds capteurs négocient entre eux les données en échangeant les paquets de signalisation spéciales, appelés métadonnées, ces paquets permettent de vérifier si le paquet est déjà transmis ou pas, et ainsi garantir que seul les informations utiles seront transmises [6].

Exemple de protocoles pour ce type de routage :

- \* Sensors Protocols for Information via Négociation (SPIN)

Ce protocole est considéré comme le premier protocole centré données ayant adopté un mécanisme de négociation pour éliminer la redondance des données reportées, et par conséquent, préserver l'énergie des nœuds. Le protocole SPIN utilise trois types de messages :

- ADV (ADVertissing) : quand un nœud a une donnée à envoyer, il avertit ses voisins en utilisant ce message.
- REQ (REQuest) : un nœud envoie ce message s'il désire recevoir une donnée.
- DATA (Information) : ce message contient la donnée avec un en-tête contenant la méta-donnée [6].

### 1.8.3.3 Classification selon l'établissement de la route

Suivant la manière de création et de maintien des chemins pendant le routage nous distinguons trois catégories de protocoles du routages [16] :

#### 1. Routage proactif

Les protocoles de routage proactifs essaient de maintenir les meilleurs chemins existants vers toutes les destinations possibles au niveau de chaque nœud du réseau. Pour ce faire, les nœuds du réseau maintiennent des tables de routage pour toutes les destinations indépendamment de l'utilité des routes. Ainsi, les routes sont établies à priori, ce qui facilite l'acheminement des données [6].

Exemple de protocoles pour ce type de routage :

- \* Optimized Link State Routing (OLSR)

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 [15].

#### 2. Routage réactif

Les protocoles de routage réactifs (dits aussi, les protocoles de routage à la demande) créent et maintiennent des routes selon les besoins. Lorsque le réseau a besoin d'une route, une procédure de découverte de route est lancée [6].

Exemple de protocoles pour ce type de routage :

- \* Ad Hoc On-Demand Distance Vector (AODV)

Ce protocole est spécialement conçu pour les réseaux mobiles pour créer et découvrir les



liaisons entre la source et la destination. AODV offre plusieurs routes de la source vers la destination et il choisit celle qui est la plus courte, et si un tour de routage échoue, le nœud source relance la requête PREQ avec un temps T plus important, et si plusieurs séries de route Request échouent, alors aucune route ne sera trouvée [11].

### 3. Routage hybride

Ce protocole est utilisé pour apprendre le proche voisinage (par exemple le voisinage à deux ou à trois sauts), ainsi, ils disposent des routes immédiatement dans le voisinage. Au-delà de la zone de voisinage, le protocole hybride fait appel à un protocole réactif pour chercher des routes [6].

#### 1.8.3.4 Classification selon l'initiateur de communication

Le paradigme de communication est déterminé par les contraintes sous lesquelles les nœuds du réseau sont interrogés ceci peut être lancée par les nœuds sources ou par les nœuds destinataires [17].

##### 1. Routage lancé par la source

Les nœuds envoient des données à la destination une fois capturées. Ces protocoles utilisent les données rapportées avec le temps (*time-driven*) ou avec l'évènement (*event-driven*). Ceci signifie que les données sont envoyées à certains intervalles ou quand les nœuds capturent certains évènements [6].

##### 2. Routage lancé par la destination

Ce protocole utilise un modèle de données basé sur les requêtes, dans ce cas, les nœuds répondent aux requêtes envoyées par la destination. Autrement dit, la station de base décrit aux nœuds capteurs le type de données qu'ils doivent capter et ainsi réduire les transmissions inutiles, mais en contrepartie, les requêtes de grandes tailles affaiblissent la batterie [6].

## 1.9 Simulation dans les RCSFs

### 1.9.1 Définitions

La simulation est une méthode de modélisation utilisée dans l'évaluation de performances des systèmes informatiques et réseaux de communication. Il s'agit d'implanter un modèle simplifié du système à l'aide d'un programme de simulation approprié, ce qui permet d'évaluer le comportement d'un système complexe dont la formalisation à l'aide de méthodes analytiques est difficile [18].

Nous appelons simulateur un programme qui met en œuvre un modèle de simulation par événements discrets afin d'assurer que la chronologie des événements soit respectée [12].

### 1.9.2 Etapes de simulation

Le processus de réalisation d'une simulation est résumé par les étapes suivantes [12] :

1. Définition du problème en délimitant les frontières du système et en identifiant ses composants
2. Définition d'objectif et différentes variables (données) du système.
3. Construction du modèle de simulation : représentation simplifiée du système qu'on veut simuler.
4. Codage ou implémentation du modèle en utilisant des outils de simulations.
5. Vérification et validation du modèle : C'est la partie la plus importante dans une modélisation, elle se fait généralement en effectuant des simulations sous des conditions connues et en comparant par la suite les résultats de système réels.
6. Documentation des résultats : C'est la visualisation des résultats après les avoir analyser.

-Ces étapes sont illustré dans la figure suivante [12] :

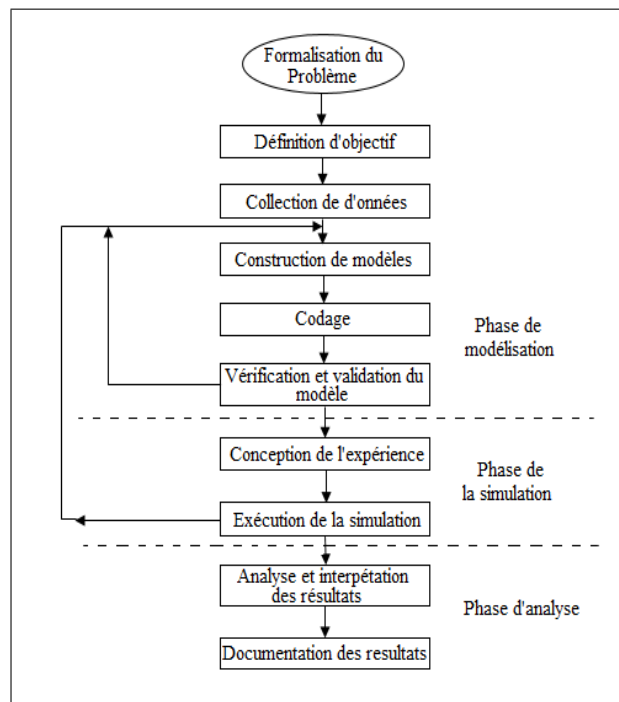


FIGURE 1.10 – Les étapes de simulation

### 1.9.3 Quelques simulateurs existants

Dans ce qui suit, nous allons présenter un ensemble de simulateurs de RCSFs qui sont :

★ Network Simulator 2 (NS-2)

Le simulateur du réseau NS-2 est un outil de simulation de réseaux informatiques, utilisé pour la simulation du routage et de protocoles d'émission/réception et surtout pour la recherche dans les réseaux ad-hoc, il supporte plusieurs protocoles réseaux et permet la simulation des réseaux câblés et sans fils aussi [12].

★ OMNeT++

Le simulateur OMNeT++ est un environnement open source qui fournit des outils pour la création et les configurations des modèles de réseaux et des outils pour l'exécution d'un lot de programme ainsi que pour l'analyse des résultats de simulation. OMNeT++ a fait son succès dans d'autres domaines grâce son architecture générique et flexible comme le domaine de systèmes informatiques, les réseaux de files d'attente, des architectures matérielles [12].

★ J-Sim

J-Sim est un environnement open source de simulation de réseaux à base de composants entièrement développé en Java. Il permet de simuler des réseaux de l'ordre de 1000 nœuds.

L'architecture et le code sont suffisamment bien structurés pour permettre une prise en main relativement rapide. De plus il permet d'utiliser n'importe quelle application Java comme générateur de trafic [18].

★ TOSSIM

TOSSIM est un simulateur/émulateur, permet de simuler le comportement d'un capteur (envoi/réception de messages via les ondes radios, traitement de l'information, etc.) au sein d'un réseau de capteurs [18]. Son grand avantage par rapport à beaucoup d'autres simulateurs de réseaux de capteurs est qu'il supporte différents types de nœuds de RCSFs. De plus, il peut simuler un nombre très grand simultanément [12]. Le plus intéressant dans ce simulateur est son extension PowerTOSSIM qui permet de simuler la consommation dans chaque périphérique au niveau de chaque nœud du réseau.

★ SENS

SENS est un simulateur à base de composants personnalisables pour les applications de RCSF. Il se compose d'éléments interchangeable et extensibles pour les applications, la communication réseau, et l'environnement physique [18].

Dans SENS, chaque nœud est divisé en quatre composantes principales :

- **Application** : simule l'application logicielle du nœud de détection.
- **Network** : traite les paquets entrants et sortants.
- **Physique** : lit les informations détectées.
- **Environnement** : c'est les caractéristiques de propagation du réseau.

SENS définit trois modèles de réseau qui peuvent être utilisés. Le premier transmet des paquets avec succès à tous les voisins, le second livre avec un risque de perte basé sur une probabilité fixe et le troisième considère le risque de collision au niveau de chaque nœud.

#### ★ Prowler

Prowler est un simulateur de réseaux sans fil conçu pour fonctionner sous l'environnement Matlab. Il est extensible également pour des plates-formes plus générales. Prowler est implémenté et mis en application sous le langage de Matlab, ce qui rend le code de simulation direct, par exemple protocole de routage. Prowler Est un simulateur capable de simuler la transmission par radio, la propagation, la réception comprenant des collisions dans les réseaux de radio ad hoc, et l'opération de la couche MAC [12].

Dans le tableau ci-dessous, on trouve des avantages et des inconvénients de chaque simulateur cité.

| Simulateurs | Avantages   | Inconvénients  |
|-------------|---|--|
| NS-2        | -Un logiciel de simulation multi-couches. -Un outil complètement libre pour plusieurs plateformes.<br>-Possibilité d'ajouter des composants à la demande.   | -Modélisation dans NS-2 reste une tâche complexe : il n'y a pas d'interface graphique.<br>-Une forte technicité est requise pour utiliser ce simulateur.                   |
| OMNeT++     | -Architecture modulaire permettant l'intégration de nouveaux modèles.<br>-Les classes de base du simulateur peuvent être étendues et personnalisées.  | -Peu de modèles pour les RCSF.<br>-Il y a un manque cruel de protocoles disponibles dans la bibliothèque comparé à d'autres simulateurs.                                   |
| J-Sim       | -Facile à installer et à simuler.<br>-Peut simuler les chaînes de radio et de consommation d'énergie dans les RCSFs ,simuler un grand nombre de nœuds capteurs et peut économiser beaucoup de taille mémoire. | -Le temps d'exécution est beaucoup plus long.<br>-La conception inhérente de J-Sim rend l'ajout de nouveaux protocoles et composants plus difficile pour les utilisateurs. |

|         |  |  |
|---------|--|--|
| TOSSIM  | <ul style="list-style-type: none"> <li>-Permet de simuler fidèlement le comportement d'un réseau de capteurs.</li> <li>-L'affichage des évènements et des messages de débogage pour chaque capteur mais aussi simultanément pour l'ensemble des capteurs.</li> </ul>             | <ul style="list-style-type: none"> <li>- Deux capteurs ne peuvent pas exécuter deux applications différentes.</li> <li>- Il ne peut pas simuler correctement les questions de la consommation d'énergie dans les RCSFs .</li> </ul>  |
| SENS    | <ul style="list-style-type: none"> <li>-Le code source de SENS peut être porté directement dans les nœuds capteurs réels, ce qui permet la portabilité des applications.</li> <li>-Elle fournit un module de puissance pour le développement fiable des applications.</li> </ul> | <ul style="list-style-type: none"> <li>- SENS est moins personnalisable que beaucoup d'autres simulateurs, ne fournit aucune chance de modifier le protocole MAC, ainsi que d'autres protocoles de réseau de bas niveau.</li> </ul>  |
| Prowler | <ul style="list-style-type: none"> <li>-Il peut être utilisé pour analyser l'énergie des protocoles de routage dans les RCSFs, le taux de livraison et l'analyse du retard du paquet.</li> </ul>   | <ul style="list-style-type: none"> <li>-Prowler ne prend pas en compte la modélisation de l'énergie des nœuds capteur.</li> <li>-Seul un guide utilisateur est disponible sur le simulateur Prowler, par contre la documentation est actuellement indisponible.</li> </ul> |

TABLE 1.1 – Tableau de Comparaison de quelques simulateurs.

## 1.10 Conclusion

A travers ce chapitre nous avons présenté les réseaux de capteurs sans fil : leurs architecture, leurs caractéristiques, les principaux types de capteur et leurs domaines d'applications. Nous avons également définis les contraintes de conception et le routage dans ce ce type de réseau puis nous avons présenté une classification des protocoles de routage. Nous avons défini la simulation, cité quelques simulateurs et aussi donnés leurs avantages et inconvénients.

Dans le chapitre suivant, nous présenterons le routage opportuniste dans les réseaux de capteurs sans fil.

# Chapitre 2

## *Le routage opportuniste dans les RCSFs*

### 2.1 Introduction

Le routage Opportuniste (RO) est un sujet d'actualité, il a reçu beaucoup d'attention et de recherches, car il est considéré comme une direction prometteuse pour améliorer l'exécution des réseaux de capteurs sans fil.

Dans ce chapitre nous définissons le routage opportuniste, ses avantages et le principe de son fonctionnement, ensuite nous listons les composants et les métriques du routage opportuniste, ainsi qu'une classification de quelques protocoles de ce dernier et enfin nous discutons l'évaluation des performances de ce type de routage.

### 2.2 Notion du routage opportuniste

#### 2.2.1 Définition

Le routage opportuniste est un nouveau mécanisme de routage qui utilise le concept de diffusion qui caractérise les RCSFs, tel que chaque nœud qui reçoit un paquet et soit proche de la destination est apte à participer dans l'envoi des paquets reçus [18].

Autrement dit, Le routage opportuniste est une approche permettant de prendre en compte l'environnement immédiat d'un nœud tel qu'il est et non pas tel qu'il est perçu par ce nœud. Cette approche fonctionne en envoyant d'abord la trame de donnée, et en sélectionnant ensuite un relais parmi l'ensemble de voisins ayant reçu cette trame [19].

La Figure 2.1 montre un exemple sur l'efficacité du routage opportuniste par rapport au routage traditionnel [19]. Cette figure est composée d'une source, une destination séparée par une chaîne de nœuds intermédiaires.

Une approche traditionnelle choisirait la séquence SRC-B-D-DST qui offre le meilleur compromis en termes de nombre de sauts et taux de réception. Si une transmission depuis la source va moins loin que prévu et n'atteint que « A » sans atteindre « B », cette transmission est gâchée et la source va retransmettre la trame. Si à l'inverse la transmission va plus loin que prévu et atteint « C », le routage traditionnel ne peut pas exploiter cette transmission et « C » détruira la trame.

Dans le routage opportuniste, « A » aurait pu retransmettre la trame et permettre à celle-ci de progresser vers la destination un peu plus que si cette dernière était restée à la source.

Le routage opportuniste exploite la diversité des chemins et peut faire des progrès vers la destination de façon plus efficace qu'en routage traditionnel [19].

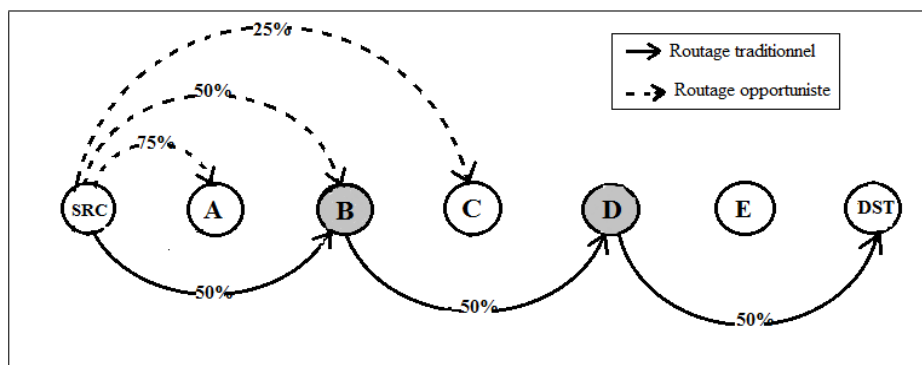


FIGURE 2.1 – Exemple de l'efficacité du routage opportuniste.

### 2.2.2 Avantages du routage opportuniste

Le routage opportuniste offre plusieurs avantages dont :

- ✓ **Augmenter la fiabilité** : Le RO transmet les paquets à travers n'importe quel lien possible plutôt qu'à travers un seul lien spécifique, en d'autres termes le RO a des liens de secours additionnels ce qui permet de réduire la probabilité d'échec de transmission [20].
- ✓ **Augmenter la qualité de la transmission** : Le RO considère toutes les liaisons pour transmettre un paquet, y compris les liaisons de bonne qualité à courte portée et les liaisons de basse qualité à longue portée, ce qui fait qu'avec une seule transmission, un paquet peut être délivré avec succès grâce à une transmission longue portée au lieu de passer par tous les nœuds intermédiaires [20].
- ✓ **Optimiser l'énergie** : Le RO peut être appliqué pour réduire la consommation d'énergie provoquée par les retransmissions et le détour dynamique des nœuds qui ont moins d'énergie [20].

- ✓ **Déterminer efficacement les nœuds d'expédition** : le prochain relais est choisi dynamiquement à travers plusieurs relais, ce qui permet de diminuer le risque de perte des paquets [21].
- ✓ **Améliorer le débit** : Après la transmission d'un paquet, le routage opportuniste choisi chaque saut du chemin afin de savoir quel relais à effectivement reçu ce paquet. Ce qui permet d'offrir un débit plus élevé que le routage traditionnel, puisque chaque transmission peut avoir plus de chance d'être reçue et transmise [22].

### 2.2.3 Principe de fonctionnement

Le routage opportuniste se compose conceptuellement de trois étapes [23] :

- ◆ Diffuser un paquet vers des relais candidats : Au lieu de choisir un relais spécifique à chaque transmission, le routage opportuniste diffuse le paquet à un ensemble de relais candidats.
- ◆ Choisir le meilleur relais et : les candidats qui ont reçu le paquet avec succès, exécute un protocole de coordination pour choisir le meilleur chemin pour retransmettre le paquet.
- ◆ Retransmettre le paquet : Le paquet sera retransmis avec le relais qui a été choisi après l'exécution de protocole de coordination.

La figure 2.2 montre un exemple sur le fonctionnement du routage opportuniste. La source S envoie un paquet à la destination D à travers les relais R1, R2 et R3, et le nombre dans chaque arc indique la séquence d'événements. La source S diffuse un paquet, R2 et R3 l'ont reçu avec succès mais R1 a échoué. Par la suite R2 et R3 exécute un protocole de coordination et décident que R2 transmet le paquet vers la destination D [23].

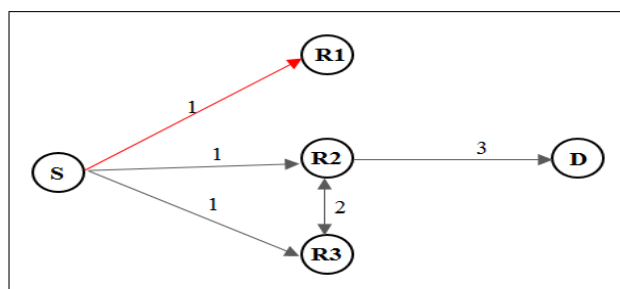


FIGURE 2.2 – Fonctionnement du routage .

### 2.2.4 Composants du routage opportuniste

Chaque protocole utilisé dans le routage opportuniste prend en compte certaines notions de bases résumées ci-dessous :



### 2.2.4.1 Sélection des candidats

Après avoir envoyé une trame en diffusion, cette dernière va être reçue potentiellement par tous les nœuds du voisinage. Parmi les récepteurs, on appelle relais potentiels les récepteurs pouvant faire progresser la trame vers la destination. La première difficulté est donc d'avoir un moyen permettant de sélectionner le sous-ensemble de nœuds parmi les récepteurs pouvant participer au relayage [19].

Pour cela, tous les nœuds dans le réseau doivent utiliser un algorithme de sélection pour choisir l'ensemble des nœuds voisins (candidats) qui peuvent aider de la meilleure façon qui soit dans le processus d'expédition vers une destination donnée. Ces algorithmes du RO utilisent une ou plusieurs métriques présentées dans la suite de ce chapitre.

### 2.2.4.2 Coordination des candidats

La coordination est le mécanisme utilisé par les nœuds candidats pour découvrir lequel a la priorité la plus élevée expédier le paquet [24].

Une bonne méthode de coordination ne devrait pas dupliquer les transmissions et ne devrait pas engendrer un grand coût en termes de temps et de frais généraux de contrôle [23]. Il existe quatre méthodes de coordination classées comme suit :

- Coordination basé sur un acquittement (*acknowledgment-based coordination*) :C'est l'une des premières méthodes qui ont été proposées pour la coordination des relais candidats. A la réception d'un paquet de données, les candidats envoient un accusé de réception ACK par ordre décroissant de priorité des candidats [23].La coordination est ainsi réalisée, les relais candidats récepteurs du paquet de données renvoient un ACK à la source. Sur la base de l'ACK, la source envoie un ordre de retransmission au meilleur relais candidat qui à son tour transmettra un ACK à la source [23].

- Coordination temporisée (*timer based coordination*) :Dans cette méthode de coordination, pour sélectionner le meilleur relais d'une liste de relais candidats admissibles, les relais candidats attendent avant de retransmettre un paquet pendant un délai basé sur un ordre prédéfini. Après qu'un paquet de données soit diffusé, les relais candidats répondront dans l'ordre, c.-à-d. le i ème relais candidat prioritaire répondra au i ème slot de temps. Une fois qu'un relais candidat répond, il est choisi en tant que prochain relais et empêchera les autres de répondre. [23].

- Coordination à base de jeton :Dans cette méthode de coordination, seulement le détenteur du jeton peut transmettre les paquets et la transmission dupliquée est totalement empêchée mais au coût de paquets de contrôle supplémentaires [23].

- Coordination basé sur le codage réseau (*Network Coding coordination*) : Ahlswede et al ont introduit le concept du codage réseau au routage. Une fois combiné avec le routage opportuniste, des transmissions dupliquées peuvent être évitées sans coordination en codant les paquets [23].

### 2.2.4.3 Filtrage des candidats relais

Le risque de provoquer plusieurs retransmissions de la même trame par des relais différents est augmenté à cause de l'augmentation des nombre de relais potentiels, ce qui nécessite l'utilisation d'un mécanisme permettant de coordonner les nœuds à la fois pour sélectionner un seul relais parmi l'ensemble des relais potentiels et supprimer les autres pour les empêcher de propager inutilement la trame [18].

### 2.2.5 Métriques de routage opportuniste

L'objectif du routage opportuniste est de réduire le nombre attendu de transmissions requis pour un paquet de la source à la destination. La sélection des nœuds qui ont reçus des paquets est faite à travers des contraintes puis le meilleur nœud est choisi comme candidat de relais pour expédier les paquets [24].

Par conséquent, utiliser une bonne métrique pour sélectionner et classer les candidats est un facteur clé dans la conception du protocole de RO. Ces métriques peuvent être classées comme suit :

#### ✂ Métrique de bout-en-bout

Cette métrique est constituée des propriétés des nœuds ou des liens dans le chemin vers la destination [23]. Autrement dit, Elle peut être définie comme le nombre de périphériques réseaux entre le nœud source et le nœud destinataire ou le nombre de liaisons point à point dans un chemin de transmission.

#### ✂ Métrique Expected Transmission Count (ETX)

La métrique ETX calcule le nombre moyen de transmission pour envoyer de manière fiable un paquet sur une liaison ou une route le long du chemin de la source vers la destination. Elle étudie les pertes de liaisons radio et les contraintes d'interférences pour trouver le meilleur chemin dans le réseau sans fil avec le moins de perte de transmission radio possible et le minimum d'interférences [24].

#### ✂ Métrique Expected Any-path Transmission (EAX)

EAX est une extension d'ETX, elle capture le nombre attendu de transmissions en tenant compte de tous les chemins qui peuvent être utilisés dans un routage opportuniste [23].

**✕ Métrique Expected transmission time (ETT)**

La métrique ETT estime le temps requis pour transmettre un paquet sur une liaison. ETT peut être calculée en ajustant la métrique ETX selon la taille du paquet et la capacité de transmission de la liaison [18].

**✕ Métrique Expected One-hop Throughput (EOT)**

EOT est une nouvelle métrique locale, elle correspond au débit prévu à un saut pour caractériser le comportement local du protocole géographique GOR en termes de l'avancement bits-mètres par seconde [24].

**✕ Métrique d'avancement de paquet**

Cette métrique est géographique, elle mesure la distance à partir d'un nœud candidat vers la destination finale et compare le résultat à la distance à partir de la source vers la destination [23].

## 2.2.6 Classification des protocoles de routage opportuniste

Une des caractéristiques principales des protocoles de routage opportunistes est comment déterminer l'ensemble des nœuds expéditeur et comment assigner la priorité à ces nœuds dans cet ensemble. Pour déterminer sa priorité, une métrique est employée et selon le type de la métrique et en se basant sur la classification présentée dans [23] nous avons classifié les protocoles opportunistes dans les classes suivantes :

### 2.2.6.1 Protocoles basés sur la méthode de sélection de l'expéditeur

On s'intéresse dans cette classe de protocoles à la sélection d'une bonne métrique pour assigner des priorités à l'ensemble d'expéditeurs choisis.

**★ Extremely Opportunistic Routing (ExOR)**

ExOR est le premier protocole de routage opportuniste proposé par Biswas et Morris en 2004. Ce protocole intègre la couche réseau et MAC. Il suppose que tous les nœuds connaissent le taux de perte sur chaque lien sans fil et peuvent employer un routage à état des liens. La liste des relais candidats est construite en se basant sur la métrique ETX, et ordonnée en fonction du nombre de sauts, cette liste est incluse dans l'en-tête du paquet de données. Quand un relais candidat reçoit le paquet, il doit répondre avec une trame d'acquiescement ACK. L'émission de l'ACK est également ordonnée de sorte que le relais candidat de priorité plus élevée répond le premier. Pour cela, les relais candidats peuvent être au courant de la réception des paquets par les autres nœuds qui ne sont pas leurs voisins.

Toutefois, la retransmission du paquet de données est retardée, puisque les nœuds doivent attendre que la séquence complète des slots affectés pour chaque relais candidat pour émettre son acquittement. Une fois cette période terminée, les nœuds décident de retransmettre le message en se basant sur les informations incluses dans les trames ACK [6].

#### ★ Simple Opportunistic Adaptive Routing Protocol (SAOR)

SOAR est un protocole de routage opportuniste, il a été proposé après ExOR afin de tirer parti de la diversité des chemins tout en évitant les transmissions en double. Il sélectionne le chemin le plus court entre la source et la destination en utilisant la métrique ETX et en employant une coordination temporisée. SOAR permet aux nœuds non sélectionnés et qui soient proches de la route établie par défaut de participer dans la transmission afin de minimiser des transmissions inutiles [18] [24].

#### ★ Minimum Transmission Selection (MTS)

MTS est un autre algorithme de sélection de transmission minimum et peut être intégré aux protocoles de routage opportuniste existants pour sélectionner les relais candidats optimaux en utilisant la métrique EAX. Il a pour objectifs de minimiser le temps de transmission prévu ou la consommation d'énergie dans le routage opportuniste [18].

### 2.2.6.2 Protocoles basé sur le codage réseau

Dans cette classe de protocoles, les paquets sont codés à l'émission. L'intégration du RO avec une stratégie appropriée de codage peut considérablement l'améliorer.

#### ★ Mac independent Opportunistic and Routing Encoding (MORE)

MORE utilise l'approche de codage réseau, les paquets de données sont toujours codés à leur émission. La source émet une combinaison linéaire des paquets et continue à envoyer de tels paquets codés dehors jusqu'à ce que le groupe entier soit reconnu par la destination. Lors de la réception, un nœud dans la liste des expéditeurs (relais candidats) vérifie d'abord l'innovation du paquet (c.-à-d., s'il contient de nouvelles informations non incluses dans les paquets de données précédemment reçus). Si c'est le cas, le relais candidat retransmet un nouveau paquet codé généré à partir des paquets de données reçus [23].

#### ★ Coding-Aware Opportunistic Routing Mechanism (CORE)

Le protocole CORE est basé sur le codage réseau, mais l'ordre des relais candidats dans la liste est établi de façon dynamique en fonction des opportunités de codage. Un nœud possède plus de possibilités de codage quand il détient plus de paquets à coder. Pour être un relais

candidat, le nœud doit être un nœud voisin de la source et il doit être géographiquement plus proche de la destination que la source [23].

#### ★ **Opportunistic Routing in Wireless Networks with Segmented Network Coding (CodeOR)**

CodeOR est proposé par Yunfeng Lin et al, il effectue le codage segmenté de réseau, celui-ci partitionne les données en de multiples segments et code seulement les paquets dans le même segment. Le protocole CodeOR emploie le codage réseau afin d'améliorer le débit de transmission [18].

#### 2.2.6.3 Protocoles de routage opportuniste géographiques

Cette classe de protocoles se base sur l'idée que chaque nœud a une certaine connaissance sur sa position et les positions des autres nœuds.

#### ★ **Geographic Random Forwarding (GeRaF)**

Ce protocole de routage opportuniste sélectionne un ensemble de relais candidats et les ordonne en utilisant des informations de localisations géographiques. Seuls les nœuds voisins les plus proches de la destination que la source peuvent être candidats [23]. Il soutient la mobilité de nœud également.

Le nœud relais n'est pas connu antérieurement par l'expéditeur mais on le connaît seulement après que le paquet a été transmis. Si les nœuds relais sont en mode de sommeil, ils ne pourront pas recevoir le paquet [18].

#### ★ **Geographically Opportunistic Routing (GOR)**

GOR est utilisé dans les scénarios de routage géographiques et adopte la coordination temporisée avec un ordre de priorité local des relais candidats. Les auteurs de ce protocole ont montré que sectionner rapprocher les relais les plus près de la destination ne permet pas toujours d'obtenir le meilleur débit. Ils ont proposés une nouvelle métrique locale nommée Expected One-hop Throughput (EOT) et sur la base de cette dernière qui considère les frais généraux de coordination, les auteurs ont proposé un nouveau mécanisme de sélection de relais candidats [24].

#### 2.2.6.4 Protocoles de routage opportuniste conscients de l'énergie

Dans cette classe, les auteurs proposent des protocoles qui se basent sur la métrique de l'énergie.

**★ Geographic Collaborative Forwarding (GCF)**

C'est le premier protocole qui a tenté d'adopter l'idée du routage opportuniste dans les RCSFs. Le but été de réduire le coût énergétique de bout-en-bout en sélectionnant les relais candidats qui ont les plus courtes distances géographiques à la destination. Cependant, il aborde le problème de la réduction du coût de l'énergie totale plutôt que de considérer l'énergie résiduelle de chaque nœud capteur, ce qui entraîne le dysfonctionnement de certains capteurs rapidement [24].

**★ Energy Aware Opportunistic Routing protocol (EAOR)**

Est un protocole opportuniste conscient de l'énergie conçu pour les RCSFs. Ce protocole essaie de transmettre les paquets sur les relais qui sont proches de la destination et qui ont aussi un haut niveau d'énergie. De cette façon, il peut découvrir plus de chemins de routage par rapport aux autres protocoles de routage opportuniste et peut réduire la consommation d'énergie totale et prolonger la durée de vie du réseau [23].

**2.2.6.5 Protocoles de routage opportuniste temps réel**

Cette classe de protocoles est conçue dans le but de satisfaire les exigences temporelles des applications temps réel.

**★ Opportunistic Real-time Routing (ORTR)**

ORTR est une méthode heuristique qui intègre les fonctionnalités de la couche réseau et de la couche MAC pour la transmission des données temps réel dans les RCSFs. ORTR calcule une région optimale où les données doivent être envoyées pour garantir la livraison en temps réel, en utilisant une puissance de transmission efficace [23].

**★ Energy-Aware Real-Time Opportunistic Routing protocol (EARTOR)**

EARTOR est un protocole de routage opportuniste temps réel, il a été proposé afin de remédier aux problèmes du protocole ORTR. EARTOR est efficace en énergie pour les applications avec des contraintes de latence de bout-en-bout spécifiées. Son but est d'équilibrer entre la consommation d'énergie et la latence de bout-en-bout et vise à maximiser le nombre de transmissions réalisées [23].

**2.2.6.6 Protocoles de routage opportuniste avec qualité de service (QoS)**

Cette classe de protocoles est conçue dans le but de satisfaire les exigences de la QoS dans les RCSFs.

**★ QoS oriented Opportunistic Routing Protocol (QOR)**

QOR est un protocole de routage opportuniste orienté QoS pour la collecte de données dans les RCSFs. Il construit un système d'adressage basé sur un préfixe commun et une structure de routage afin d'optimiser l'utilisation des liens sans fil opportunistes et de faire face aux problèmes de réplication. QOR réduit le délai de bout-en-bout et la consommation d'énergie tout en offrant un taux de livraison élevé [23].

**★ Energy eFFicient Opportunistic Routing Technology in wireless sensor networks (EFFORT)**

EFFORT est un protocole de routage opportuniste distribué qui introduit une nouvelle métrique, permettant de calculer le coût opportuniste de bout-en-bout, appelé Opportunistic End to end Cost (OEC). EFFORT étend efficacement la durée de vie du réseau par rapport aux autres protocoles de routage opportuniste comme GCF [23].

## 2.3 Evaluation des performances

Après avoir vu et étudié en chapitre 1 quelques simulateurs existants (NS-2, OMNeT++, Prowler, SENSE, TOSSIM,) et environnements de simulation ainsi que leurs critères de classification et de qualification, nous pouvons constater que la plupart de ces simulateurs sont destinés à la recherche et à l'émulation de réseaux, leur utilisation nécessite donc un certain savoir-faire et des connaissances assez avancées en modélisation et simulation des RCSFs et peu adaptés à nos besoins, Nous pouvons citer quelques critiques :

- J-Sim : le nombre de nœuds est limité à 1000 nœuds.
- NS-2, pour lequel beaucoup de modifications étaient requises pour aboutir au résultat voulu, le temps qui aurait été nécessaire pour réaliser ces modifications était bien supérieur à celui requis pour réaliser un simulateur minimal adapté à nos problèmes.
- OMNeT++ nécessite la configuration du modèle du réseau.
- OMNeT++, J-Sim sont spécifiés pour les réseaux Ad hoc sauf que OMNeT++ a un package spécifique pour les réseaux de capteur.
- En plus tous ces simulateurs n'ont pas de paramètres du routage opportuniste.

### 2.3.1 Métriques de performance

Ces métriques sont évaluées selon les facteurs suivants :

\* **Energie consommée**

Cette métrique représente la moyenne de l'énergie consommée par tous les nœuds participants au transfert des paquets à partir du nœud source jusqu'au nœud destination, et c'est le paramètre le plus étudié lors de la validation.

\* **Durée de vie du réseau**

La durée de vie du réseau peut être définie comme étant la durée d'épuisement des nœuds capteurs par unité de temps jusqu'à la mort du premier nœud lors d'une transmission de paquets. Elle peut être également définie comme le nombre de nœuds capteurs vivants au fil du temps.

\* **Taux de Livraison de Paquets de donnée**

Le taux de livraison de paquets est défini comme le rapport entre le nombre de paquets de données reçus avec succès par le sink et le nombre de paquets de données générés par la source. Une grande valeur de taux de livraison de paquets de donnée signifie une meilleure performance du protocole du routage.

\* **Délai Moyen de Bout-en-Bout**

Cette métrique représente le temps nécessaire aux paquets de données pour atteindre la destination, divisée par le nombre de paquets reçus. Plus que la valeur de délai de bout-en-bout est petite, meilleure sera la performance du protocole.

## 2.4 Conclusion

Ce chapitre a été axé sur le routage opportuniste dans les RCSFs. Nous avons présenté la définition, les avantages et le principe de fonctionnement du routage opportuniste, ainsi que les composants et les métriques et une classification des protocoles de ce dernier. Le chapitre suivant sera consacré à la spécification des besoins et la conception de notre nouveau simulateur du routage opportuniste dans les réseaux de capteur sans fils.



# Chapitre 3

## *Spécification des besoins et Conception*

### 3.1 Introduction

Dans ce chapitre, nous définissons le langage UML (*Unified Modeling Language*) et le processus de développement UP (*Unified Process*). Ensuite, nous identifions les fonctionnalités de l'utilisateur en spécifiant les besoins fonctionnels et non fonctionnels, suivi par l'élaboration des diagrammes de cas d'utilisations, de séquences, d'activité et de classe.

### 3.2 Langage de modélisation unifié (UML)

**Définition :** L'UML est un langage de modélisation graphique, permet d'expliquer les modèles objets à travers un ensemble de diagrammes. En effet, un diagramme d'UML est une représentation graphique, qui offre 13 types de diagrammes et chaque type de diagramme offre une vue d'un système [28].

Nous avons opté pour l'UML pour sa grande flexibilité qui s'exprime par l'utilisation des diagrammes, la spécification des besoins. De plus il apporte une compréhension rapide du programme à d'autres développeurs externes en cas de reprise du logiciel et facilite sa maintenance.

### 3.3 Démarche UP

Le processus unifié est un processus de développement logiciel construit sur UML. Il est itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. Il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel [29].

## 3.4 Analyse des besoins

Notre travail consiste à réaliser un outil de simulation des protocoles de routage opportuniste dans les réseaux de capteurs qui a comme objectif de bien mener la phase d'évaluation de performances des protocoles conçus et de les comparer suivant quelques critères à savoir l'énergie, le temps de transmission et de faciliter les tâches de l'utilisateur afin d'arriver à des résultats.

Dans cette partie nous identifions les fonctionnalités de notre système.

### 3.4.1 Besoins fonctionnels

Notre système doit comporter principalement les exigences fonctionnelles suivantes :

- ★ Visualiser les résultats de simulation (courbe 2D).
- ★ Créer des scénarios de simulation.
- ★ Gérer les paramètres du réseau.
- ★ Introduire les paramètres de routage.
- ★ Gérer les paramètres de simulation.

### 3.4.2 Besoins non fonctionnels

Les principales exigences non fonctionnelles de notre simulateur se résument dans les points suivants :

- ★ Le simulateur doit offrir une interface ergonomique et conviviale.
- ★ Le simulateur doit permettre aux utilisateurs de suivre et sauvegarder leur travail.
- ★ Introduire la notion de sécurité (Contrôle d'accès ).
- ★ Le simulateur doit répondre à toutes les exigences.

## 3.5 Diagramme de cas d'utilisation

Un diagramme de Cas d'Utilisation montre l'interaction entre le système et les entités externes au système. Ces entités externes sont désignés comme acteurs. Les acteurs représentent des rôles qui peuvent inclure des utilisateurs humains, matériel externe ou d'autres systèmes. Un acteur est d'habitude dessiné comme un chiffre de bâton nommé, ou encore comme un rectangle de classe avec le mot-clé «acteur» [30]. Le diagramme de cas d'utilisation de notre système est illustré sur la Figure 3.1 suivante :

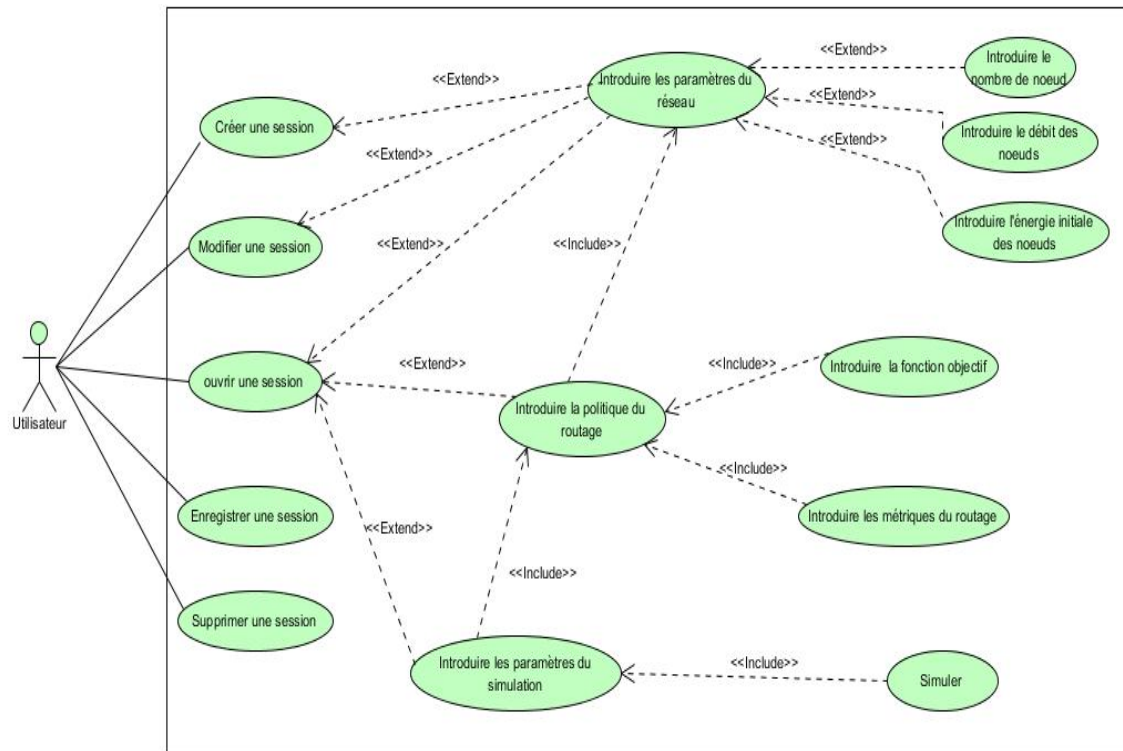


FIGURE 3.1 – Diagramme de cas d'utilisation.

Les paramètres du routage représentent les variables selon lesquelles s'effectue le choix du chemin de routage de l'information dans le réseau (Le paramètre énergie du nœud se base sur le choix du nœud ayant le niveau d'énergie le plus élevé, le paramètre délai d'un saut est le temps que prennent les données entre un nœud vers son voisin).

La fonction objectif a pour but de minimiser/maximiser des résultats de la simulation (Taux d'erreur du paquet, Taux de transmission).

Les paramètres de simulation indiquent les variables selon lesquelles s'effectue le choix de simulation, à savoir :

- La consommation d'énergie : quantité d'énergie consommée dans le réseau.
- Le temps de transmission : la durée de la transmission d'un message.

### 3.6 Diagramme de séquence

Le diagramme de séquence est une représentation graphique qui décrit le déroulement et les interactions entre un groupe d'objets (les éléments du système et ses acteurs) en montrant de façon séquentiel, les envois des messages qui interviennent entre les objets dans le cadre d'un scénario d'un diagramme de cas d'utilisation [25].

### 3.6.1 Diagramme de séquence du cas d'utilisation «Créer une session»

Il s'agit du premier cas d'utilisation, consiste à créer une nouvelle session pour préparer son environnement de simulation, ce cas est illustré dans la Figure 3.2.

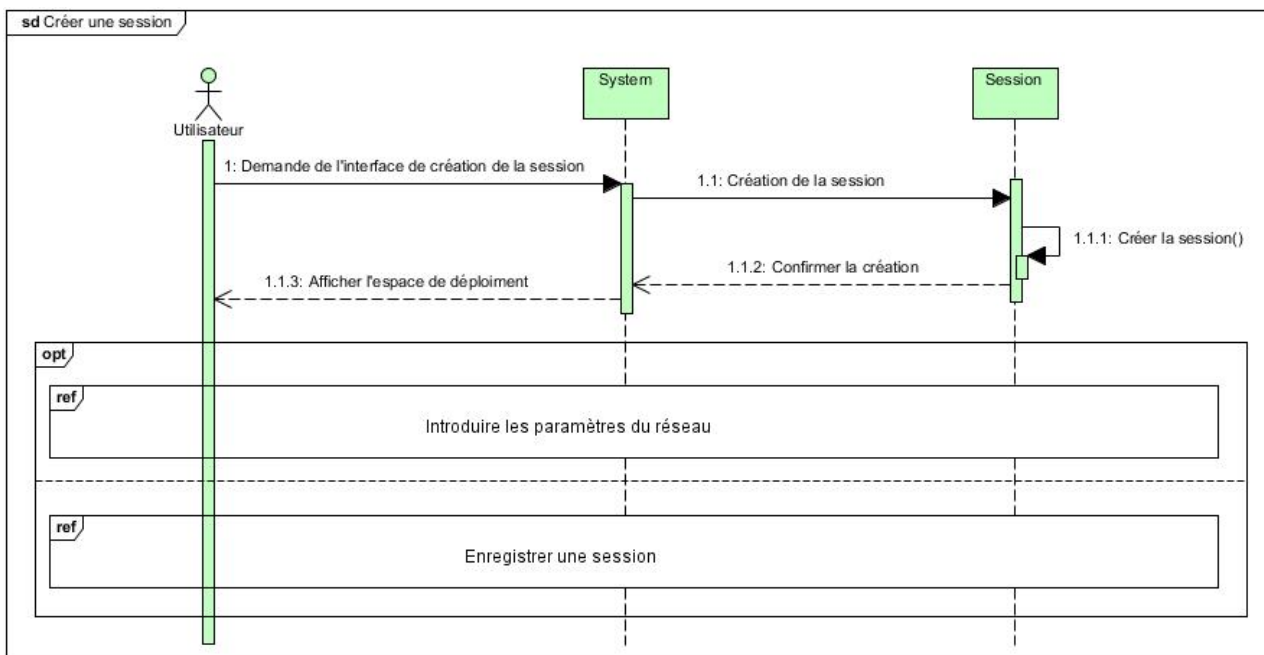


FIGURE 3.2 – Diagramme de séquence créer une session.

### 3.6.2 Diagramme de séquence du cas d'utilisation «Ouvrir une session»

Ce cas est lorsqu'un utilisateur ouvre une session déjà créée et enregistrée dans ce simulateur, ce dernier est illustré dans la Figure 3.3.

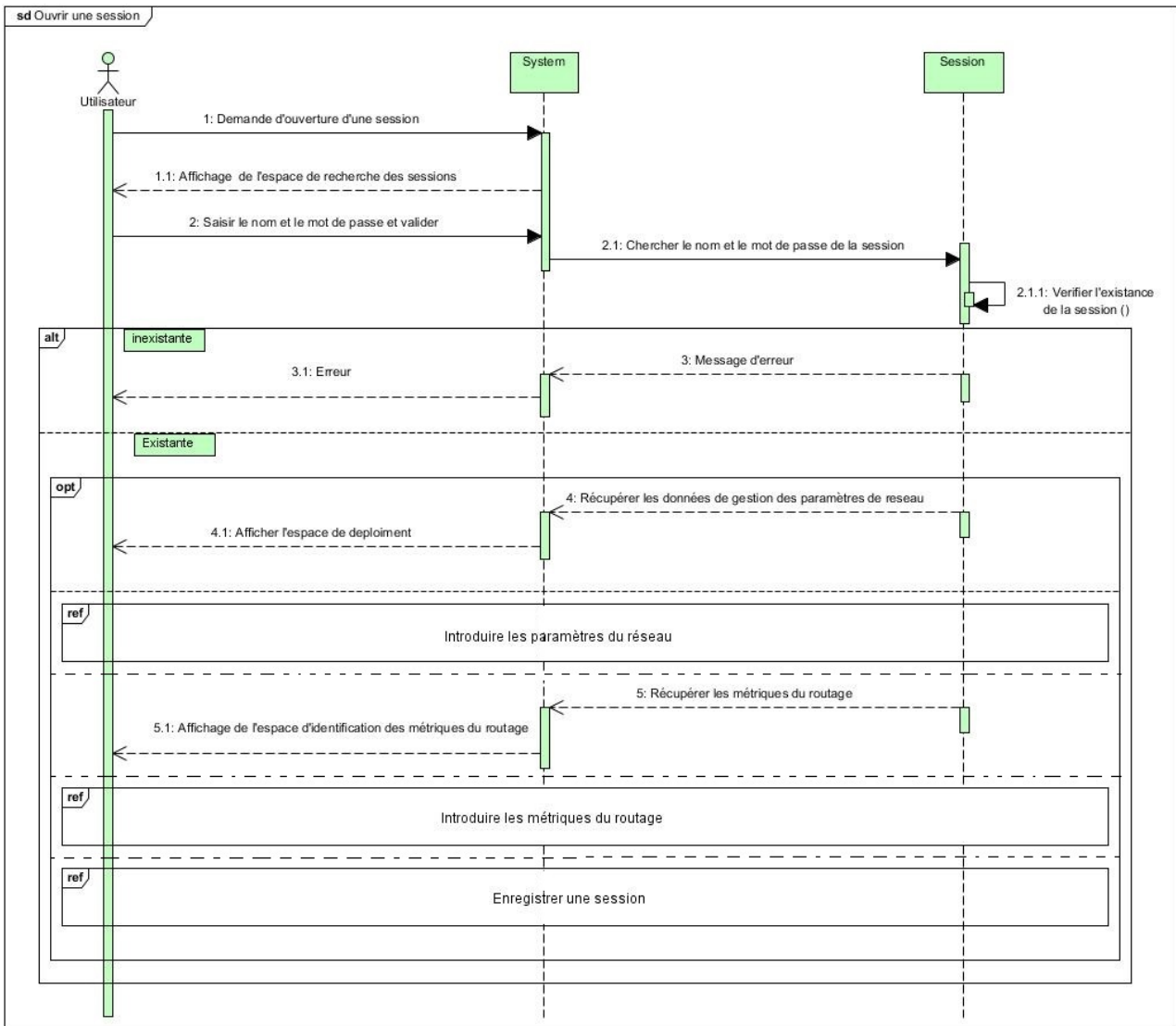


FIGURE 3.3 – Diagramme de séquence ouvrir une session.

### 3.6.3 Diagramme de séquence du ca d'utilisation «Enregistrer une session»

Le cas illustré dans la figure 3.4 représente le cas où l'utilisateur enregistre la session qu'il a déjà créé ou des modifications apporté à une session.

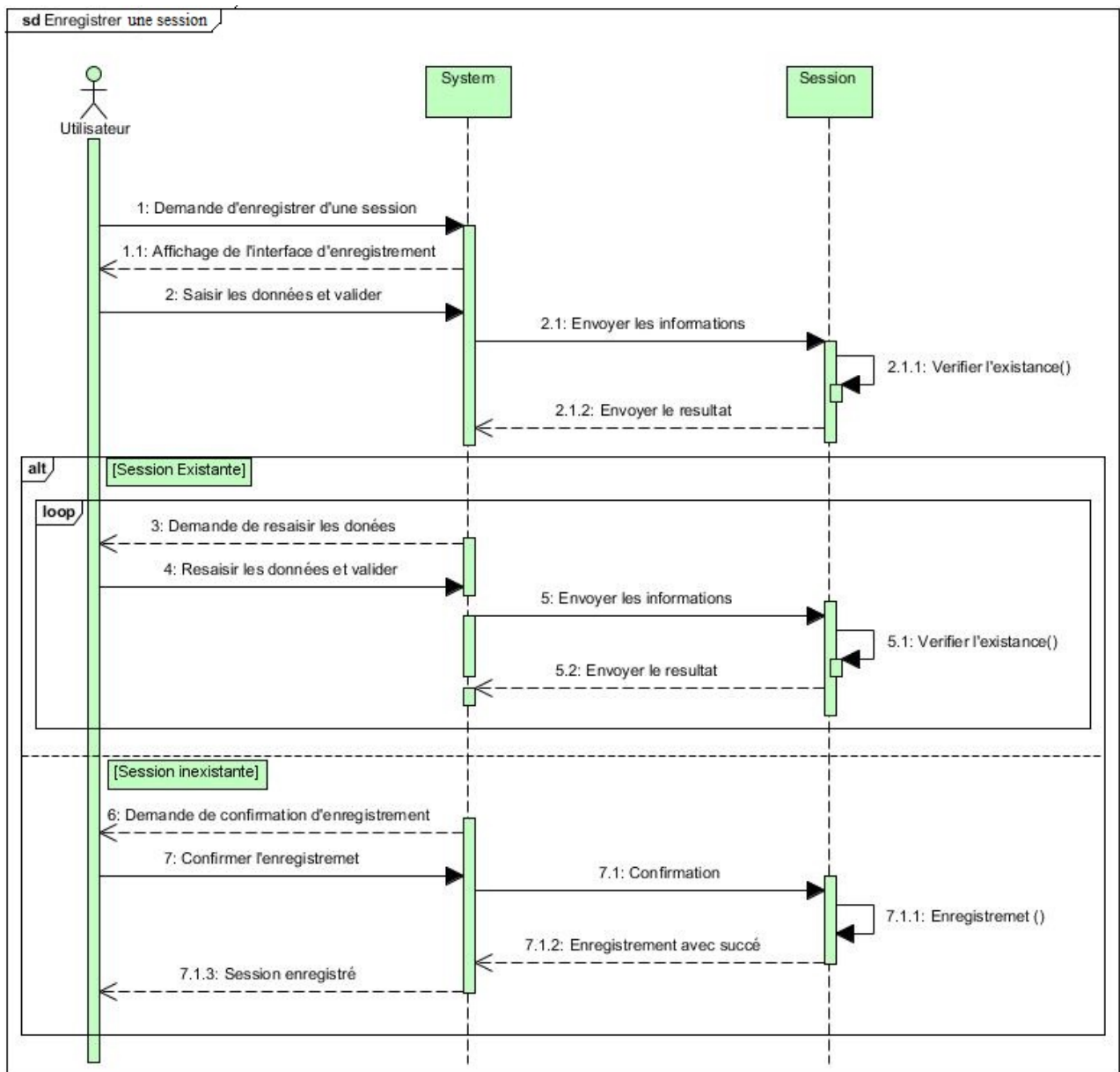


FIGURE 3.4 – Diagramme de séquence Enregistrer une session.

### 3.6.4 Diagramme de séquence du cas d'utilisation «Supprimer une session»

Il s'agit de supprimer une session existante. Ce cas est illustré dans la figure 3.5.

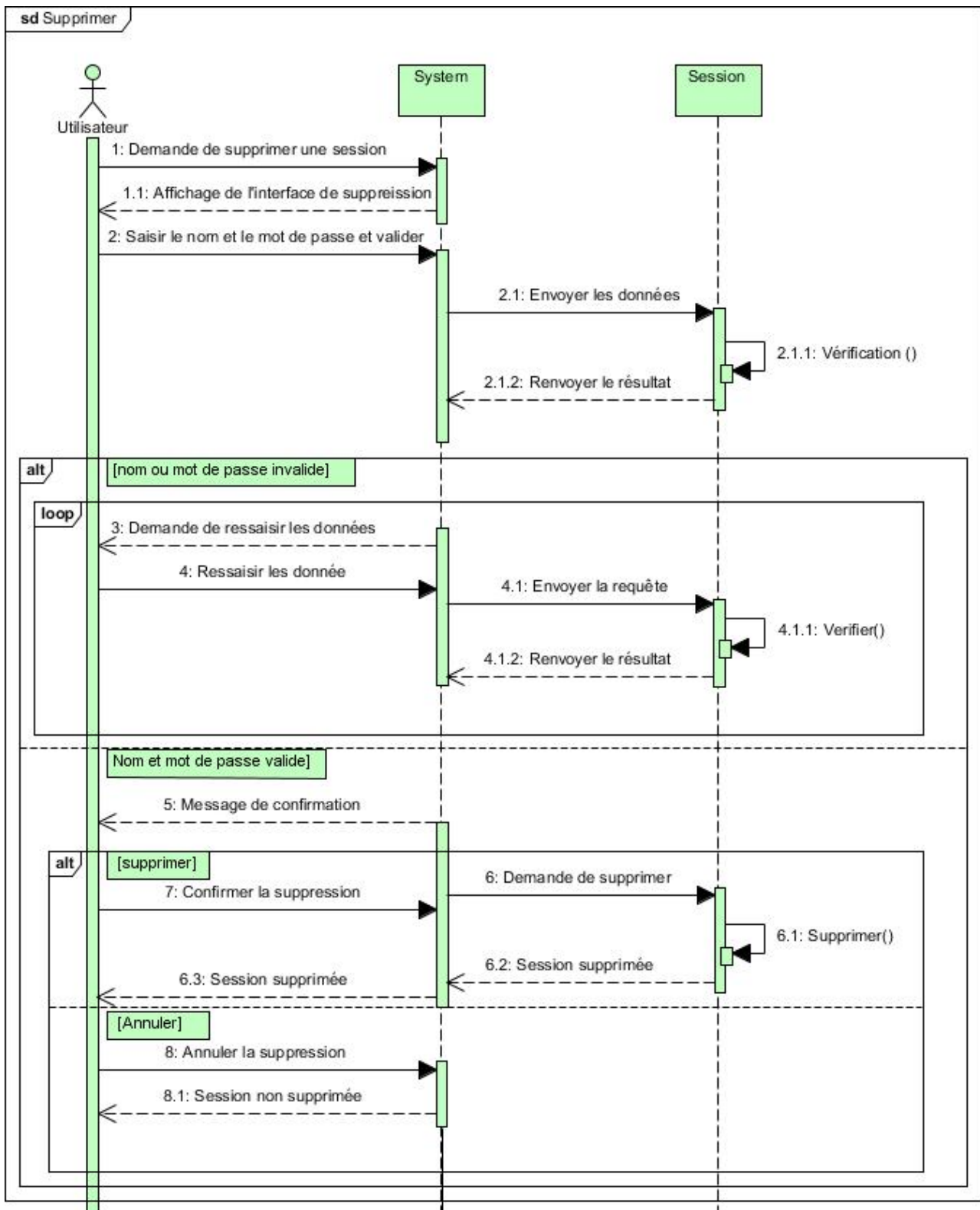


FIGURE 3.5 – Diagramme de séquence Supprimer une session.

### 3.6.5 Diagramme de séquence du cas d'utilisation «Introduire les paramètres du réseau»

Dans ce cas l'utilisateur peut effectuer plusieurs actions tel que préciser le nombre de nœuds et aussi leur énergie initial.

La Figure 3.6 suivante représente ce cas.

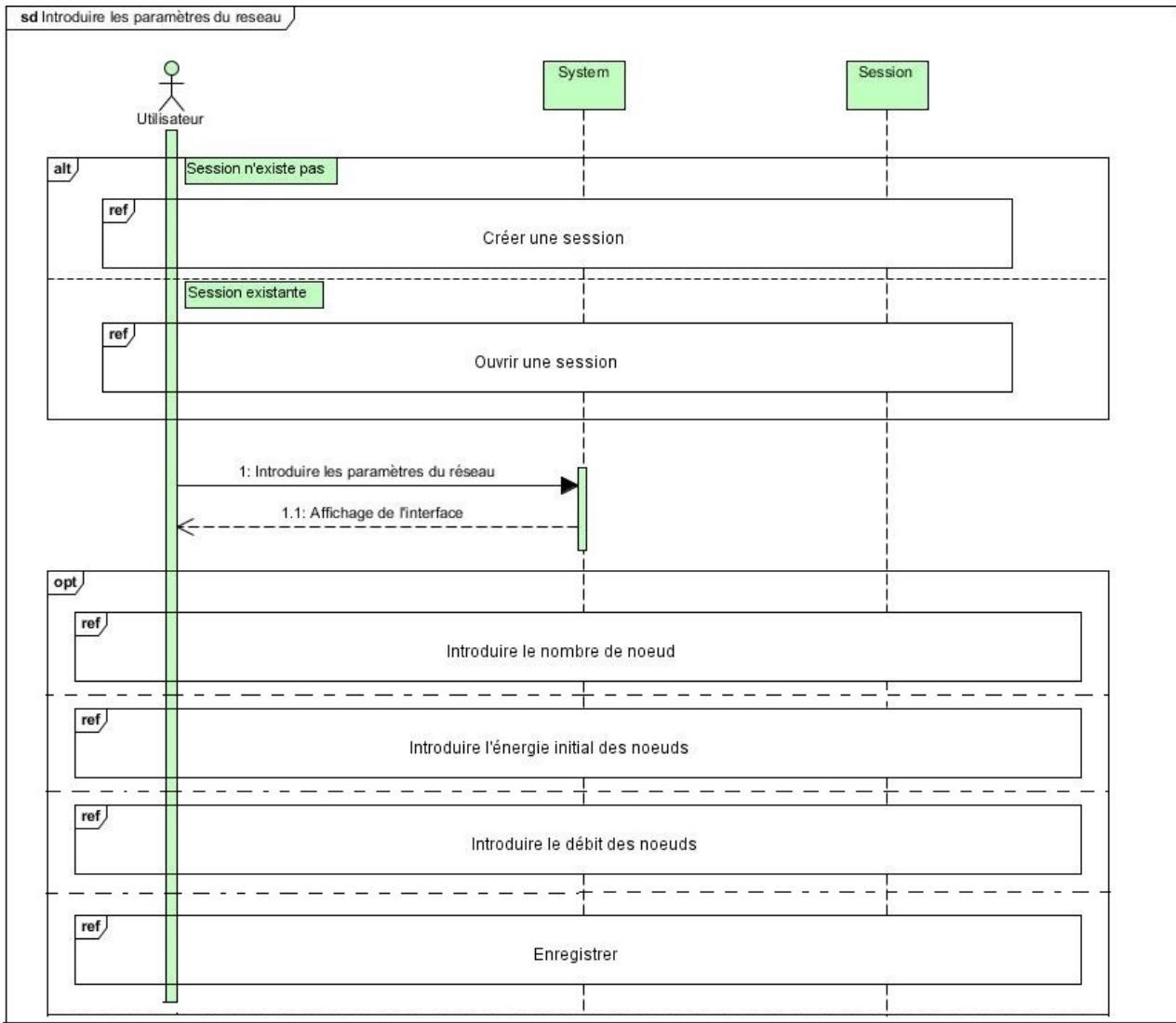


FIGURE 3.6 – Diagramme de séquence introduire les paramètres du réseau.



### 3.6.6 Diagramme de séquence du cas d'utilisation «Introduire le nombre de nœud»

Dans ce cas d'utilisation (voir Figure 3.8), l'utilisateur peut choisir le nombre de nœuds à déployer pour un scénario de simulation.

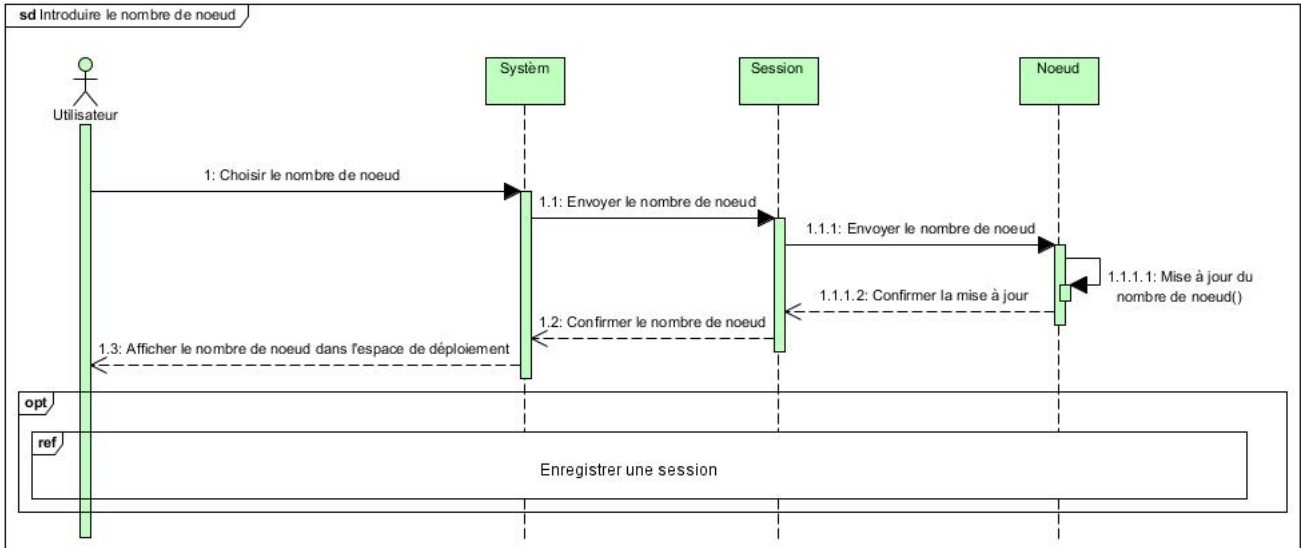


FIGURE 3.7 – Diagramme de séquence introduire le nombre de nœuds.

### 3.6.7 Diagramme de séquence du cas d'utilisation «Introduire le débit des nœuds»

Dans ce cas d'utilisation (voir Figure 3.9), l'utilisateur peut introduire le débit de transmission des données qui sera utilisé lors de la simulation.

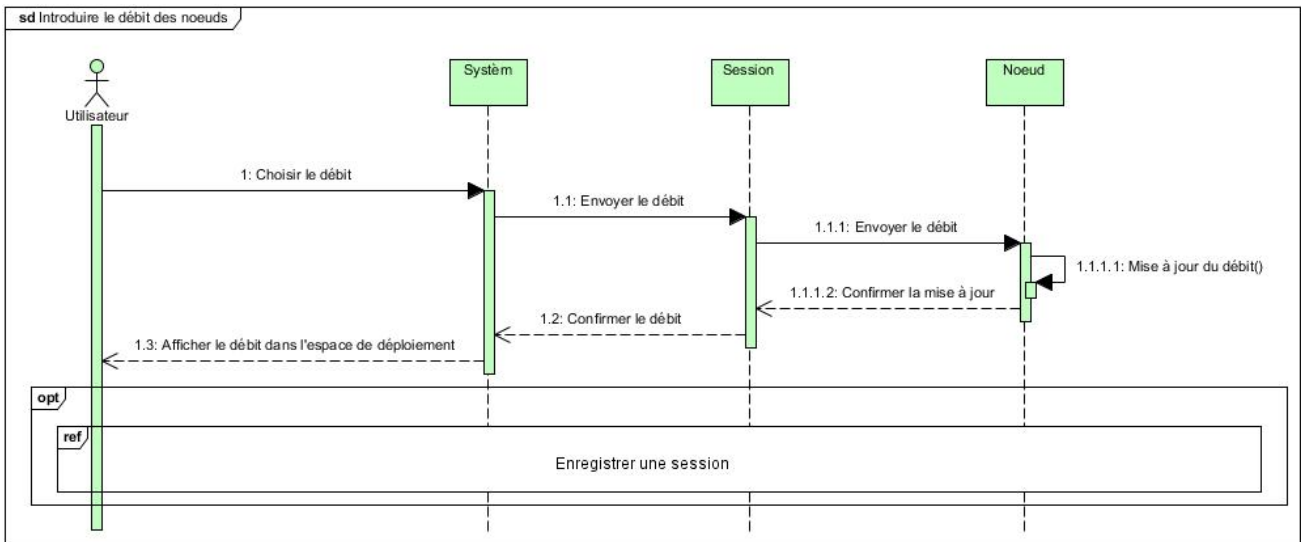


FIGURE 3.8 – Diagramme de séquence introduire le débit des nœuds.

### 3.6.8 Diagramme de séquence du cas d'utilisation «Introduire l'énergie initiale des nœuds»

Dans ce cas d'utilisation (voir Figure 3.10), l'utilisateur peut introduire l'énergie initiale des nœuds qui sera utilisé lors de la simulation.

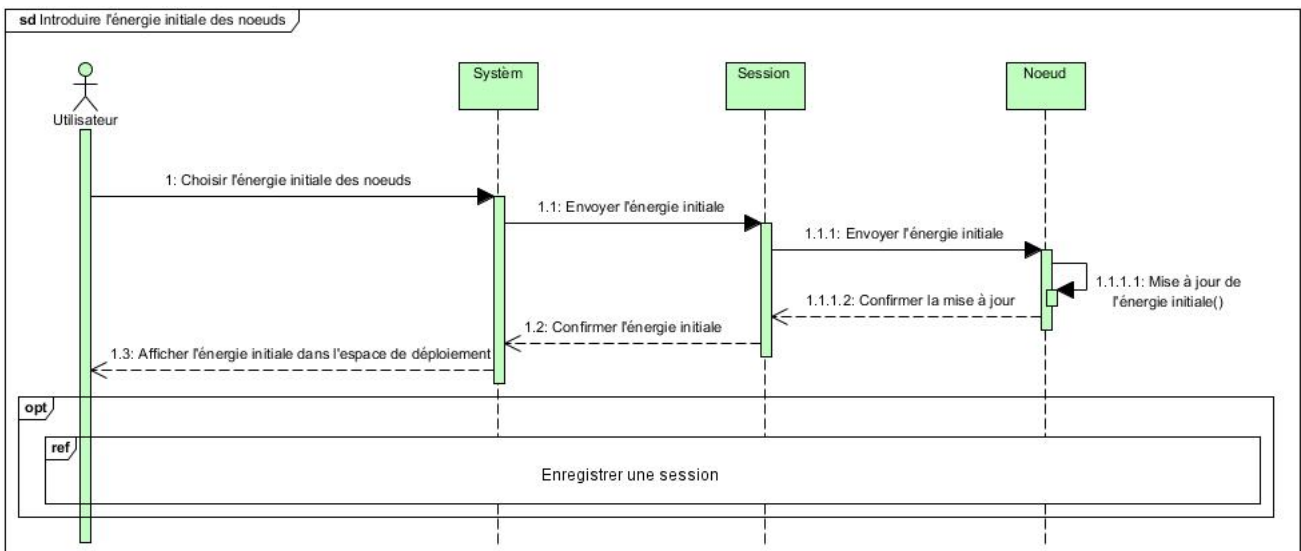


FIGURE 3.9 – Diagramme de séquence introduire l'énergie initiale des nœuds.

### 3.6.9 Diagramme de séquence du cas d'utilisation «Introduire les métriques du routage»

L'utilisateur peut effectuer un choix des paramètres de routage selon lesquels les données transitent entre les nœuds lors de la simulation. La Figure 3.12 suivante représente ce cas.

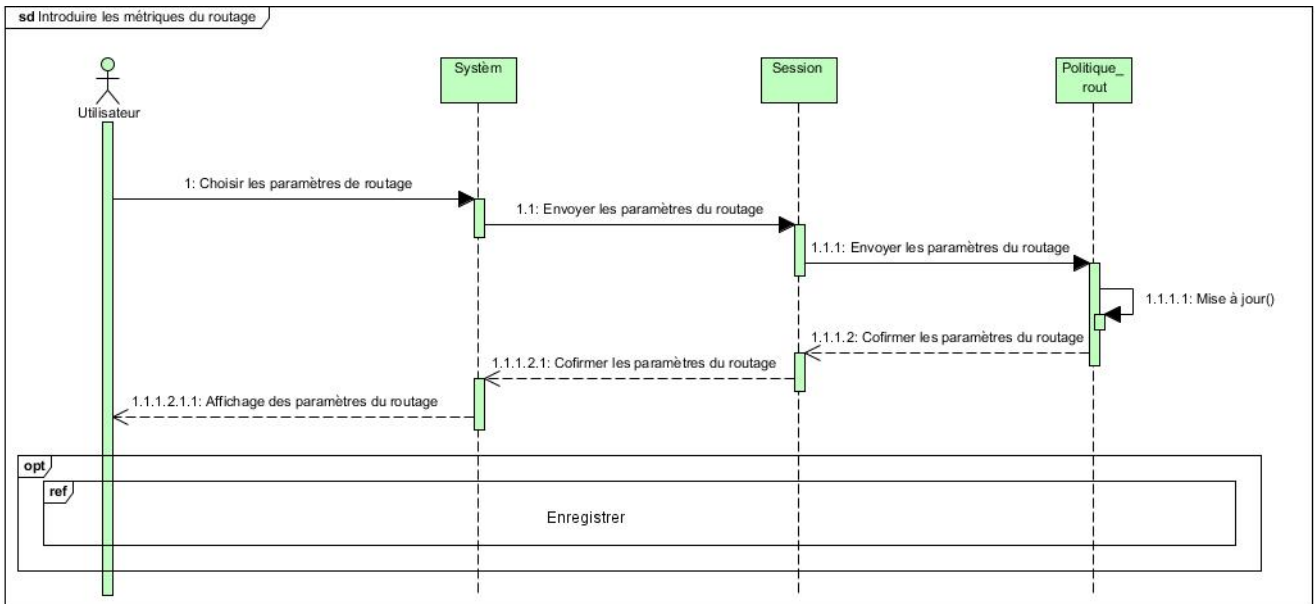


FIGURE 3.10 – Diagramme de séquence introduire les métriques du routage.

### 3.6.10 Diagramme de séquence du cas d'utilisation «Introduire la politique du routage»

Dans ce cas illustré dans la figure 3.11, l'utilisateur peut effectuer un choix des paramètres de routage selon lesquels les données transitent entre les nœuds lors de la simulation.

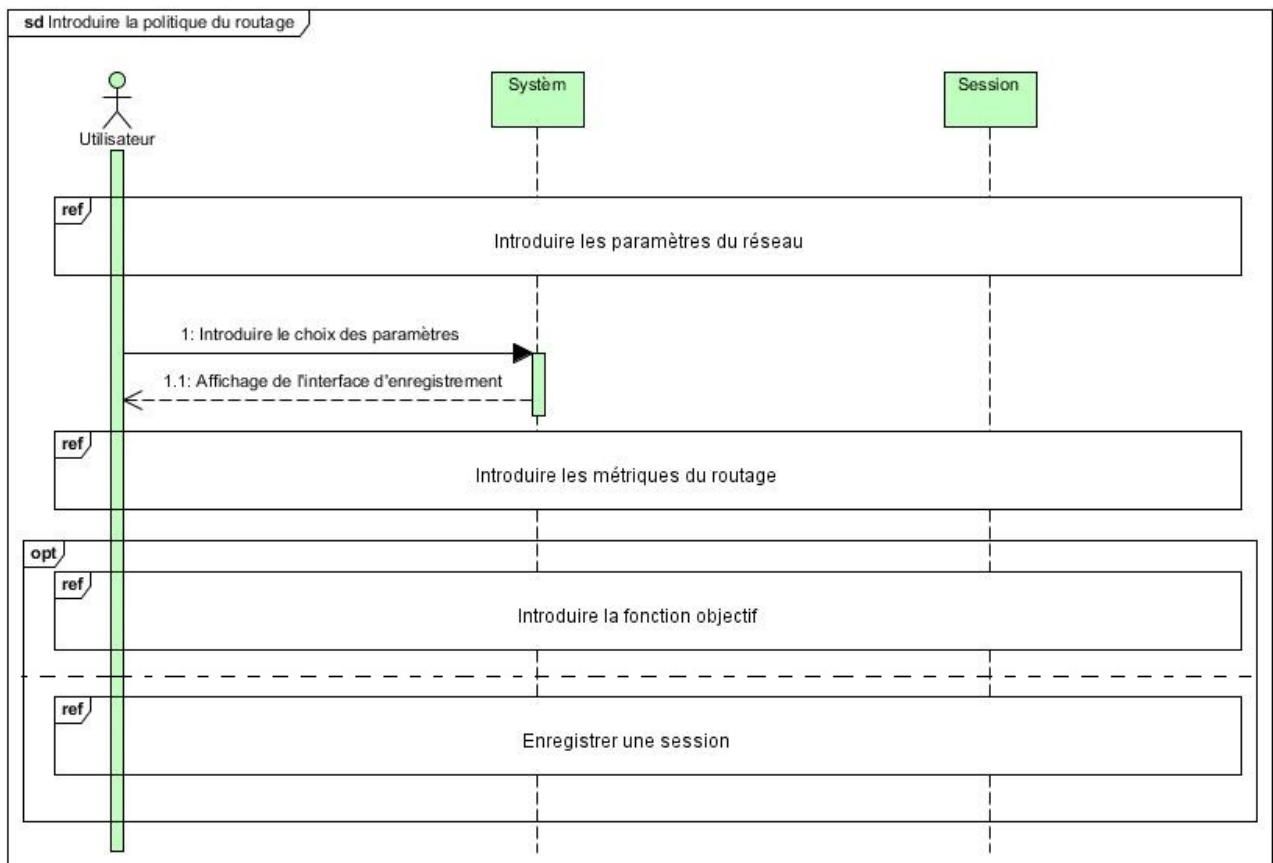


FIGURE 3.11 – Diagramme de séquence introduire la politique du routage.

### 3.6.11 Diagramme de séquence du cas d'utilisation «Identifier la fonction objectif»

Dans ce cas d'utilisation (voir Figure 3.13), l'utilisateur introduit la fonction objectif.

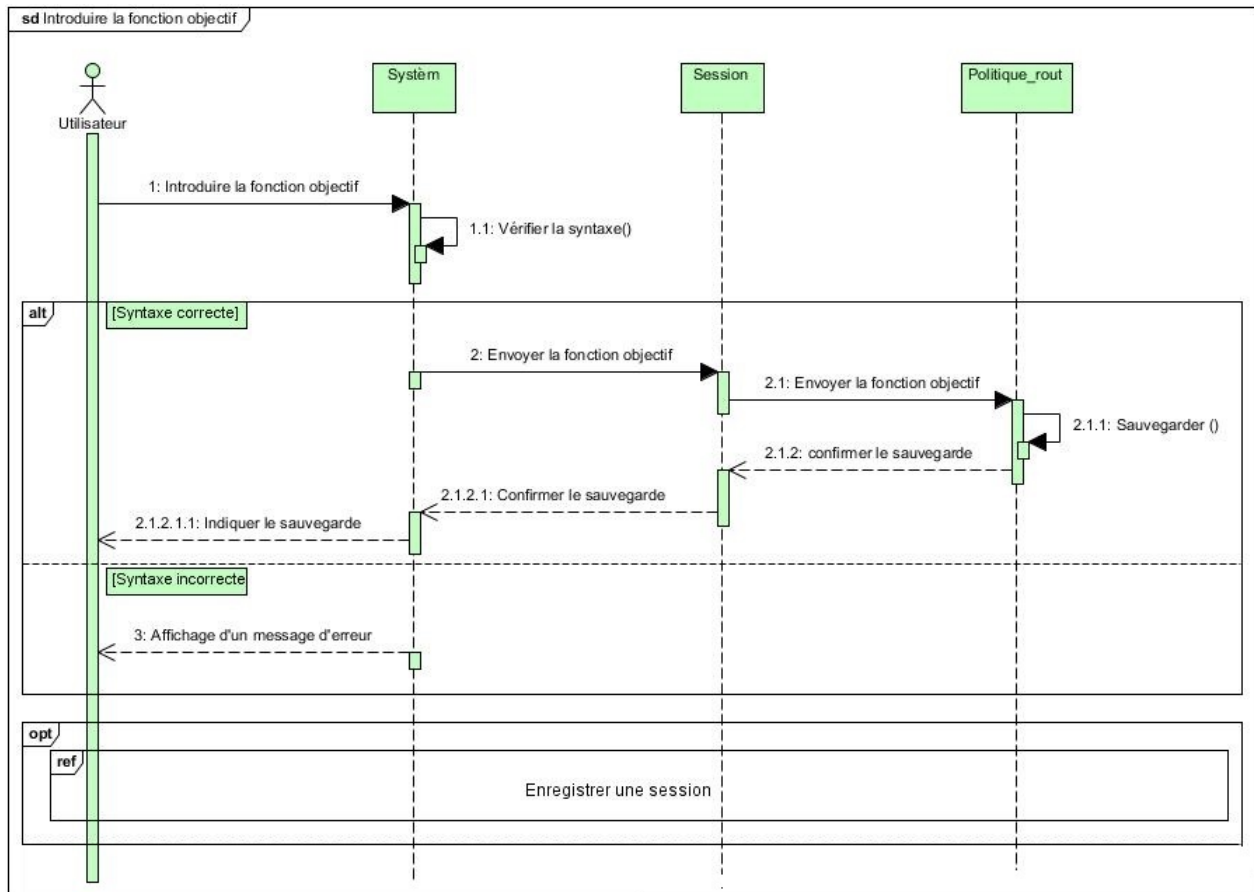


FIGURE 3.12 – Diagramme de séquence introduire la fonction objectif.

### 3.6.12 Diagramme de séquence du cas d'utilisation «Introduire les paramètres de simulation»

Dans ce cas l'utilisateur choisit les paramètres de simulation plus précisément les résultats qu'il veut obtenir. La figure 3.14 représente ce cas.

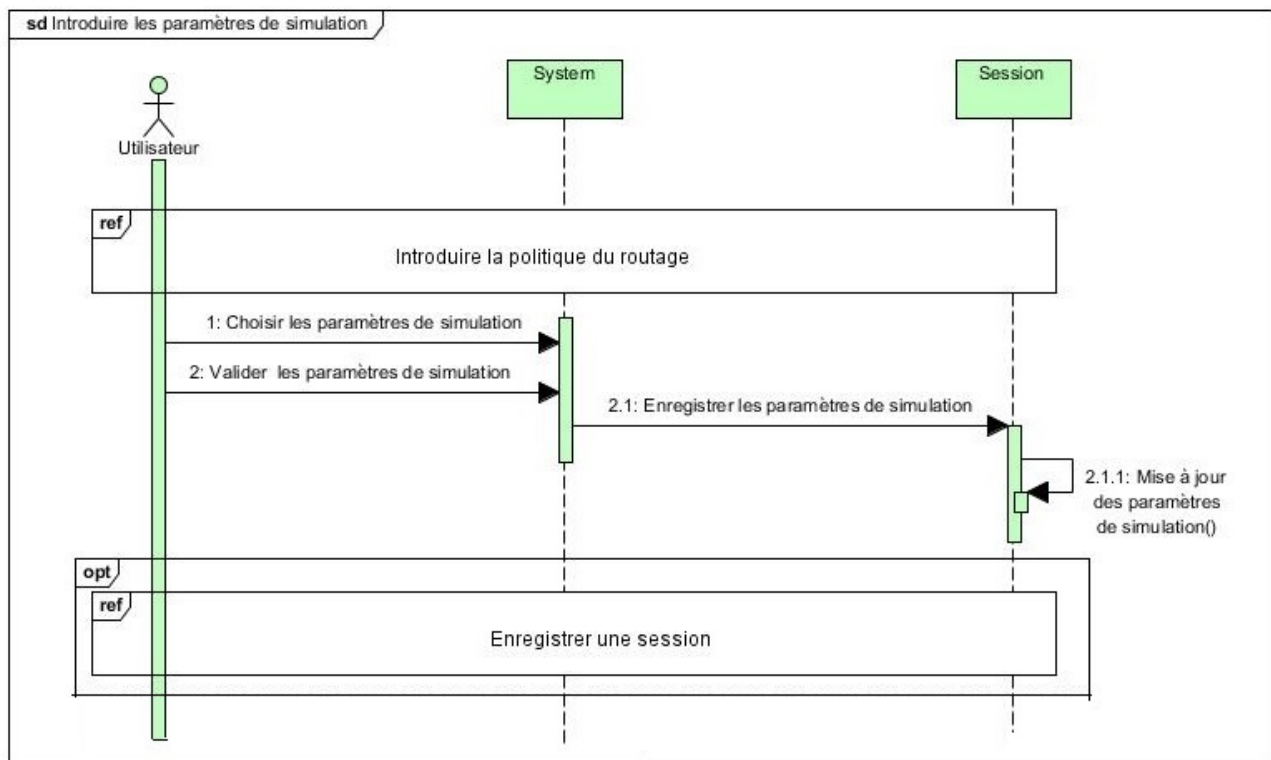


FIGURE 3.13 – Diagramme de séquence introduire les paramètres de simulation.

### 3.6.13 Diagramme de séquence du cas d'utilisation «simuler»

Dans ce cas d'utilisation (voir Figure 3.15), après avoir effectué la gestion des paramètres réseaux, l'introduction de la politique de routage et l'introduction des paramètres de simulation l'utilisateur peut simuler pour afficher les résultats de simulation d'un scénario déployé.

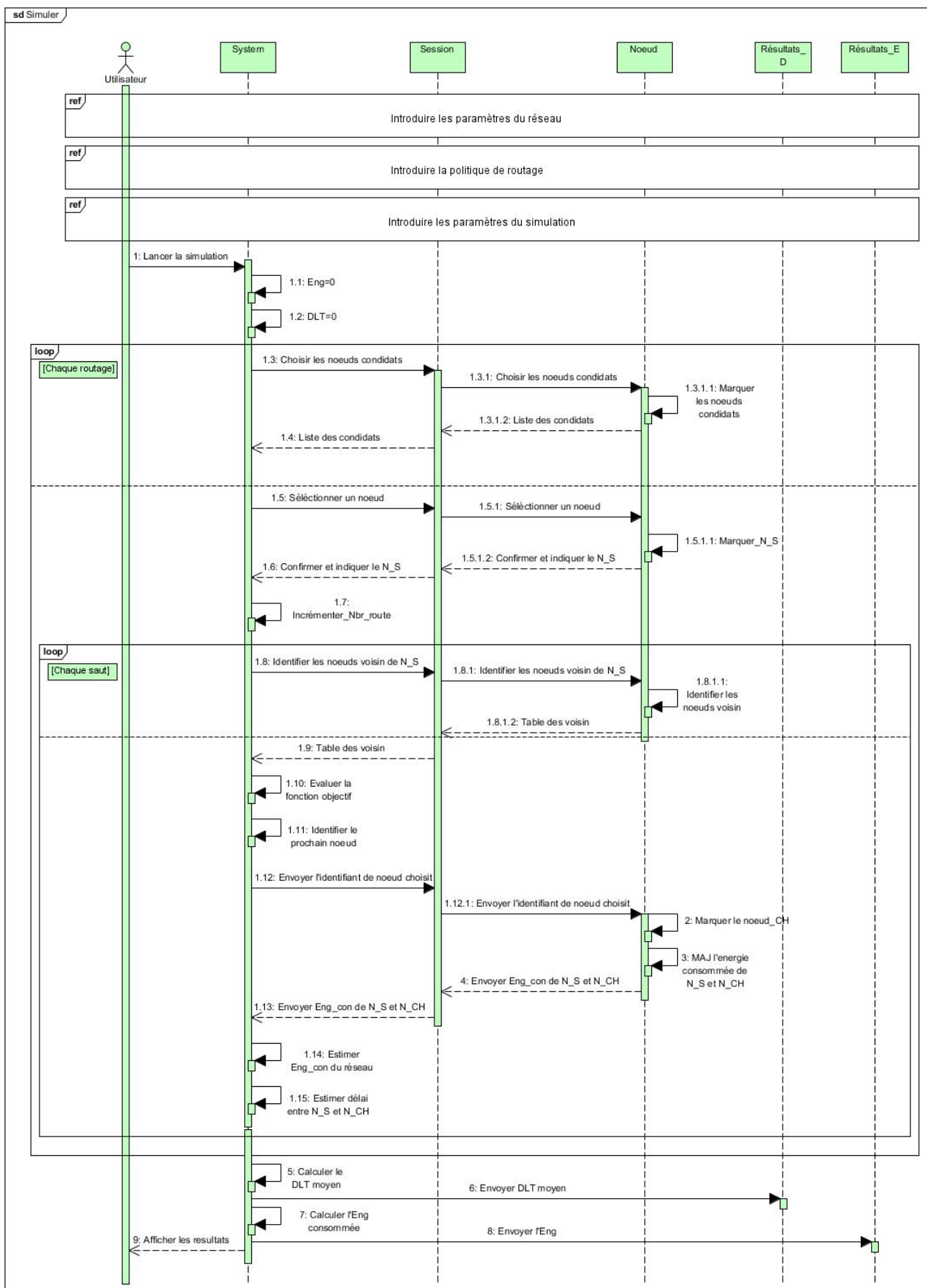


FIGURE 3.14 – Diagramme de séquence simuler

## 3.7 Diagramme d'activité

**Définition :** Le diagramme d'activité est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables. Le diagramme d'activité est également utilisé pour décrire un flux de travail (workflow) [31].

Dans notre cas lorsque l'utilisateur lance la simulation, ce dernier :

1. Applique un algorithme de sélection des candidats ;
2. Découvert les voisins de chaque nœud et les lui assigne ;
3. Applique une coordination des candidats afin de découvrir le nœud prioritaire ;  
Si la condition d'arrêt est vérifiée :
4. Calcule et estime l'énergie consommé et le délai moyen ;
5. Affiche les résultats de la simulation.  
Sinon il réitère la simulation.

la figure suivante illustre un diagramme d'activité associer au cas d'utilisation «simuler» :



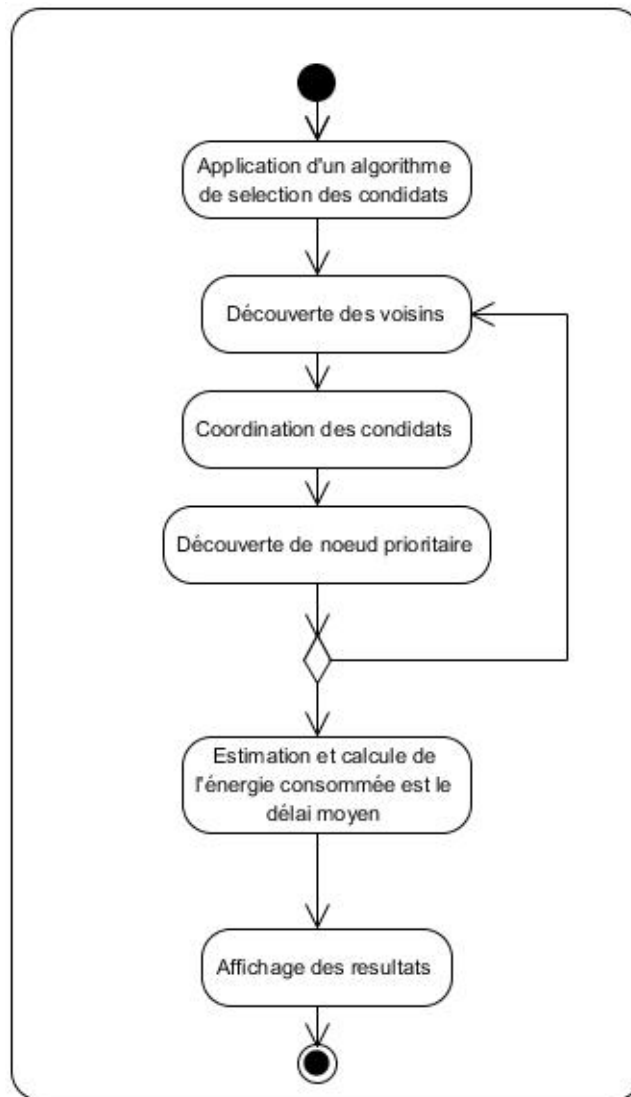


FIGURE 3.15 – Diagramme d'activité simulé.

### 3.8 Diagramme de classes

Le diagramme de classes est une version d'UML de la base de données, il s'agit d'un ensemble de classe contenant différentes opérations et attributs relié entre eux par des relations avec des conditions de cardinalités [32].

La figure 3.16 représente le digramme de classe correspond à notre conception :

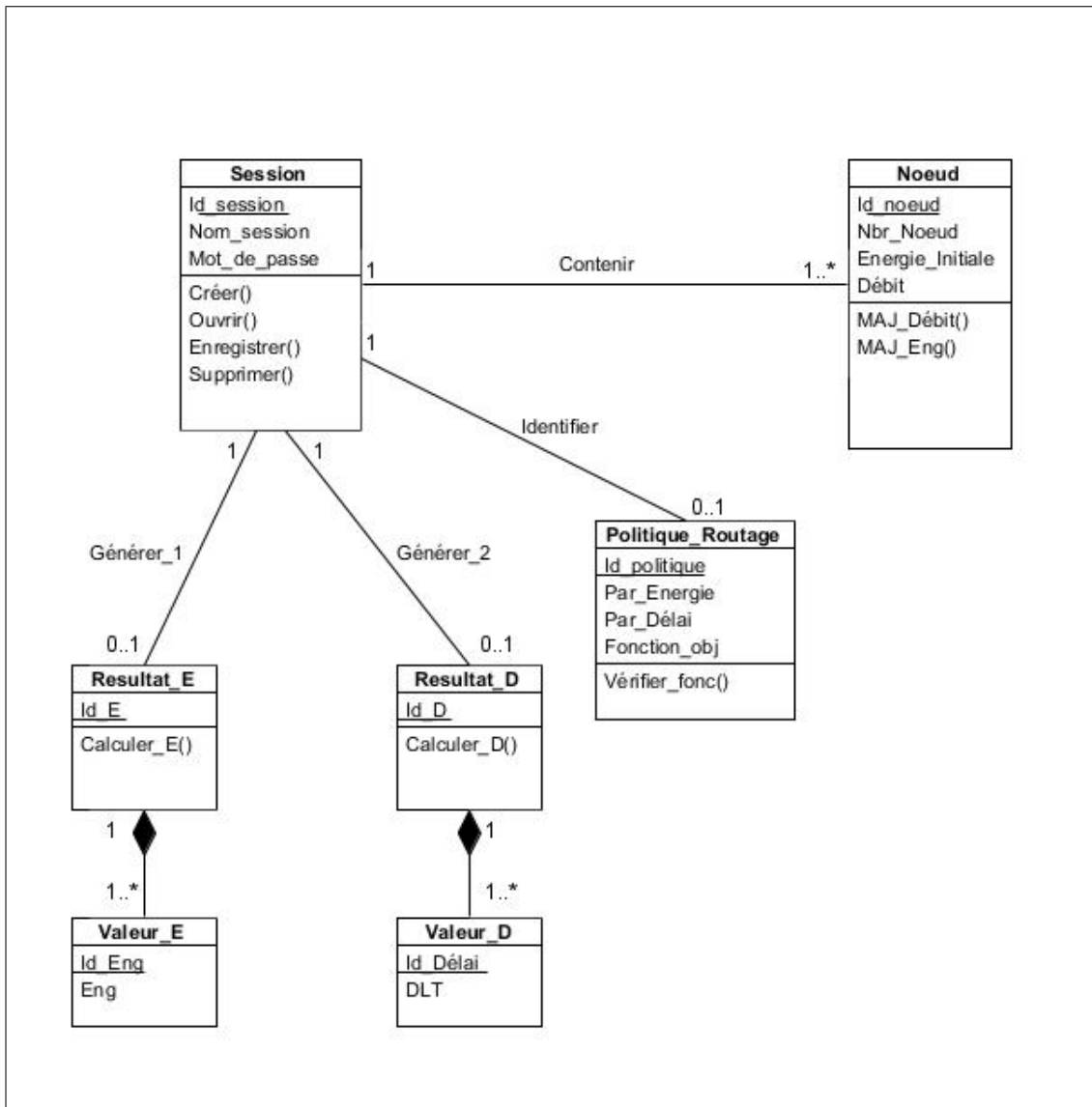


FIGURE 3.16 – Diagramme de classes.

### 3.8.1 Dictionnaire des données

| Classe            | Attributs        | Signification  | Type    |
|-------------------|------------------|--|---------|
| Session           | Id_session       | Identifiant de la classe (clé)   | Int     |
|                   | Nom_session      | Nom de la session  | Varchar |
|                   | Mot_de_passe     | mod de passe de la session   | Varchar |
| Nœud              | Id_nœud          | Identifiant de la classe (clé)   | Int     |
|                   | nbr_Nœud         | Le nombre de nœud  | Int     |
|                   | Energie_Initiale | Le niveau de batterie du noeud   | double  |
|                   | Débit            | Débit de transmission  | double  |
| Politique_Routage | Id_politique     | Identifiant de classe (clé)  | Int     |
|                   | Par_Energie      | Indique si le paramètre "énergie" est choisi pour effectuer le routage | Boolean |
|                   | Par_délai        | Indique si le paramètre "délai" est choisi pour effectuer le routage   | Boolean |
|                   | Fonction_obj     | La fonction objectif   | Varchar |
| Resultat_E        | Id_E             | Identifiant de la classe (clé)   | Int     |
| Resultat_D        | Id_D             | Identifiant de la classe (clé)   | Int     |
| Valeur_E          | Id_Eng           | Identifiant de la classe (clé)   | Int     |
|                   | Eng              | Les résultats de la simulation en termes d'énergie                     | Double  |
| Valeur_D          | Id_délai         | Identifiant de la classe (clé)   | Int     |
|                   | DLT              | Les résultats de la simulation en termes de délai                      | Double  |

TABLE 3.1 – Tableau de dictionnaire de données.

### 3.8.2 Modèle relationnel

Afin de pouvoir implémenter notre base de données, nous passons de la description conceptuelle au modèle relationnel à partir du diagramme de classe obtenu précédemment. Et ceci, en utilisons des règles de passages de l'UML vers le relationnel.

★ **Règle 1 : Transformation des classes**

- ✓ Chaque classe est transformée en table.
- ✓ Les propriétés de la classe deviennent les attributs de la table.

✓ L'identifiant de la classe devient la clé primaire de la table.

★ **Règle 2 : Associations binaires**

✓ Association binaire 1 - 1..\*

Cette règle consiste à ajouter la clé primaire de la relation père dans la relation fils de l'association comme une clé étrangère.

★ **Règle 3 : La composition**

La composition est une forte agrégation la classe composée est détruite lorsque la classe mère est disparait. Cette association est représentée par un losange plein.

Cette association consiste à ajouter la clé primaire de la classe mère dans la classe composée comme une clé étrangère.

En appliquant ces règles à notre modèle, nous obtenons les relations suivantes :

Session (Id\_session, Nom\_session, Mot\_de\_passe)

Nœud (Id\_nœud, nbr\_Nœud, Energie\_Initiale, Débit, Id\_session#)

Politique\_Routage (Id\_politique, Par\_Energie, Par\_délai, Fonction\_obj, Id\_session#)

Resultat\_E (Id\_E, Id\_session#)

Resultat\_D (Id\_D, Id\_session#)

Valeur\_E (Id\_Eng, Eng, Id\_E#)

Valeur\_D (Id\_Délai, DLT, Id\_D#)

## 3.9 Conclusion

Dans ce chapitre consacré à la spécification des besoins et la conception ,nous avons décrit les besoins fonctionnels et non fonctionnels et aussi nous avons donné une conception détaillée de notre simulateur au cours duquel nous avons présenté les différents diagrammes.

Le chapitre suivant sera consacré à la phase de la réalisation du simulateur.

# Chapitre 4

## *Réalisation*

### 4.1 Introduction

Ce chapitre constitue une description de l'environnement de développement utilisé ainsi que les outils et langage de développement dont a eu besoin pour l'implémentation, suivi de l'exposition de quelques interfaces graphiques de notre simulateur.

### 4.2 Environnement et outils de développement

#### 4.2.1 Java

Java est un langage de programmation orientée objet créée par l'entreprise Sun Microsystems, il permet de développer des applications client-serveur et des objets génériques pour que le code puisse être réutilisable [26].

#### ★ Le choix de java

Notre choix pour ce langage est motivé par les critères suivants [26] :

- ✓ Java est un langage de programmation moderne.
- ✓ Une de ses plus grandes forces par rapport à d'autres langages de programmation est son excellente portabilité, ce qui signifie qu'un même programme peut fonctionner sur différentes plates-formes et sous différents systèmes d'exploitation (Windows, Mac, Linux, etc.)
- ✓ Java est doté d'une bibliothèque de classes très riche, à savoir le graphisme qui est exploitée dans le développement des interfaces graphique (fenêtres et interactivité) et l'accès aux bases de données. Il existe pour ce besoin de nombreuses solutions sous

différentes formes.

- ✓ L'utilisation de ses bibliothèques facilite grandement la tâche du programmeur lors de la construction des applications complexes.
- ✓ Une variété de programmes avec Java : Des applications, sous forme de fenêtre ou de console ; Des applets, qui sont des programmes Java incorporés à des pages web ; Des applications pour appareils mobiles.

### 4.2.2 IDE Eclipse

Eclipse est un environnement de développement intégré libre, extensible, universel et polyvalent, permettant potentiellement de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions. La spécificité d'Eclipse vient du fait de son architecture totalement développée autour de la notion de plug-in : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in [33].

### 4.2.3 WampServer

WampServer est une plateforme de développement Web de type WAMP, à faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases de données MySQL [34].

### 4.2.4 Structured Query Language (MySQL)

MySQL ("Langage de requêtes structuré en français ") est un langage définition de données LDD, un langage de manipulation de données LMD, et un langage de contrôle de données LCD destiné à interroger ou piloter une base de données [27] .

### 4.2.5 Visual Paradigm for UML

Visual Paradigm est un logiciel de création de diagrammes dans le cadre d'une programmation, il possède plusieurs options permettant une large possibilité de modélisation en UML.

Le logiciel Visual Paradigm offre de nombreux outils pour créer différents types de schémas comme les diagrammes de cas d'utilisation et de séquence. Il permet aussi de générer des codes sources en divers langages comme le Java ou C++ à partir du modèle créé, inversement, il permet de produire un modèle à partir de codes sources [35].

### 4.2.6 TEXstudio

Pour la rédaction du présent document TEXstudio a été utilisé, c'est un environnement open source de rédaction intégré pour la création de documents LATEX.

TEXstudio possède de nombreuses fonctionnalités telles que coloration syntaxique, visionneuse intégrée, la vérification des références et divers assistants [35].

## 4.3 Présentation de «RoSim»

RoSim est un simulateur conçu spécialement pour le routage opportuniste dans les réseaux de capteur sans fil. Il est composé de quatre principales interfaces à savoir l'interface principale (de gestion des sessions), l'interface de gestion des paramètres du réseau, l'interface de gestion des paramètres de routage (Identifier la politique) et l'interface de gestion des paramètres de simulation. Dans ce qui suit, nous présenterons les interfaces principales de notre simulateur. Cette partie a comme objectif de faciliter la prise en main du simulateur.

### 4.3.1 Interface de gestion des sessions

L'interface de gestion des sessions est constituée d'une barre de Menu, Barre des tâches et un bouton pour quitter (voir la figure 4.1).

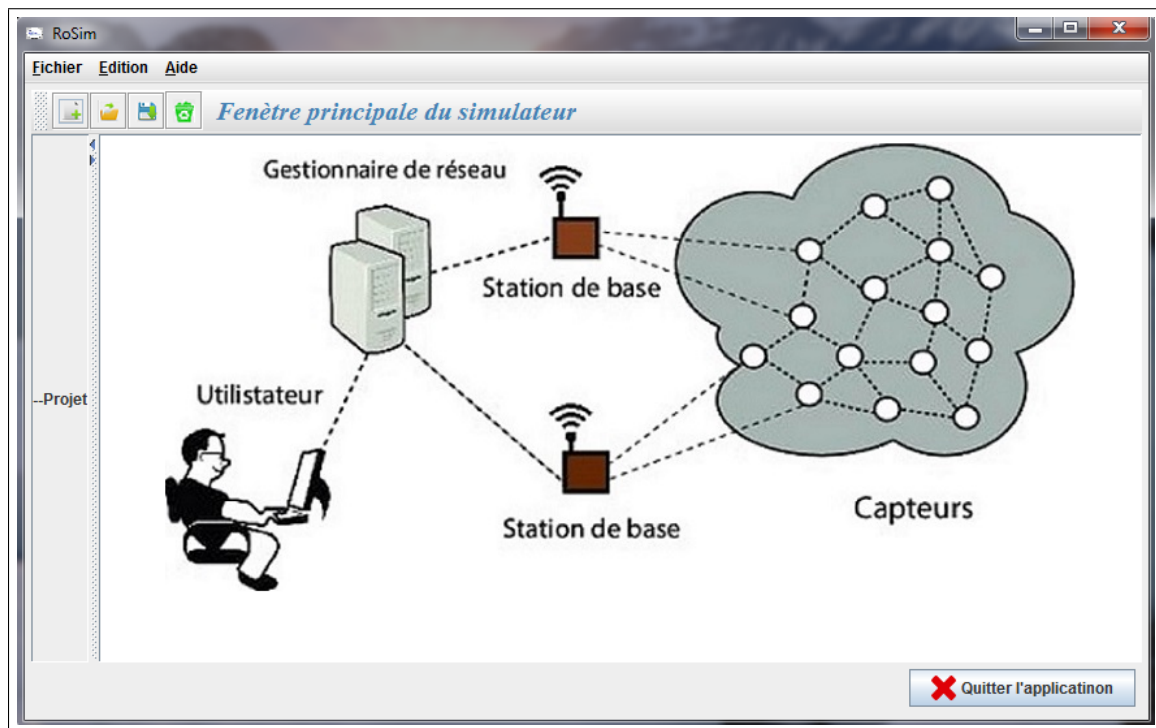


FIGURE 4.1 – Fenêtre de gestion des sessions.

★ La barre de menu contient :

✓ Menu "Fichier" :

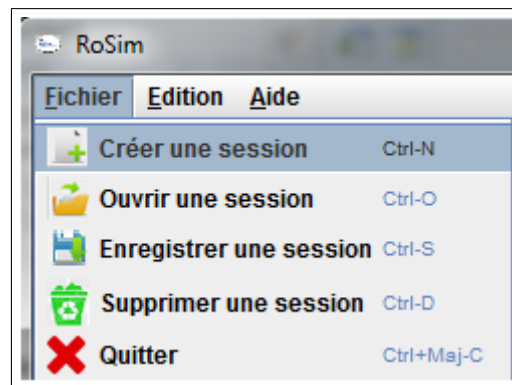


FIGURE 4.2 – Fenêtre menu fichier.

- Le menu "Créer une session" : permet créer une nouvelle session de simulation.
- Le menu "Ouvrir une session" : permet à l'utilisateur de visualiser une session de simulation.
- Le menu "Enregistrer" : permet à l'utilisateur d'enregistrer une session de simulation.
- Le menu "Supprimer" : permet à l'utilisateur de supprimer une session de simulation.
- Le menu "Quitter" : permet de quitter l'application.



- ✓ Menu "Edition" : ce menu offre à l'utilisateur la possibilité de "Simuler" ou de "Valider" les modifications effectués (voir la figure 4.3).

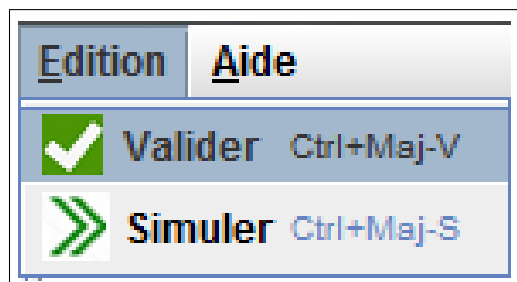


FIGURE 4.3 – Fenêtre menu édition.

- ✓ Menu "Aide" : Ce dernier est une page web statique qui donne plus d'informations sur «RoSim».

RoSim est un simulateur conçu spécialement pour le routage opportuniste dans les réseaux de capteur sans fil. Il est composé de quatre principales interfaces à savoir l'interface principale (de gestion des sessions), l'interface de gestion des paramètres du réseau, l'interface de gestion des paramètres de routage (Identifier la politique) et l'interface de gestion des paramètres de simulation.

**Interface de gestion des paramètres du réseau**

Cette interface permet à l'utilisateur de gérer les paramètres du réseau en lui offrant la main à remplir les champs suivant : Position des noeuds, le débit des noeuds, le nombre de noeuds et l'énergie initiale.

**Interface d'identification des paramètres du routage**

cette interface permet à l'utilisateur de gérer la politique du routage plus précisément choisir selon quel critère il souhaite effectuer le routage des données entre les noeuds du réseau et même d'introduire la fonction objectif.

**Interface de gestion des paramètres de la simulation**

L'interface de gestion des paramètres de simulation permet à l'utilisateur d'effectuer son dernier choix qui consiste à choisir selon quoi il voudra avoir les résultats de la simulation.

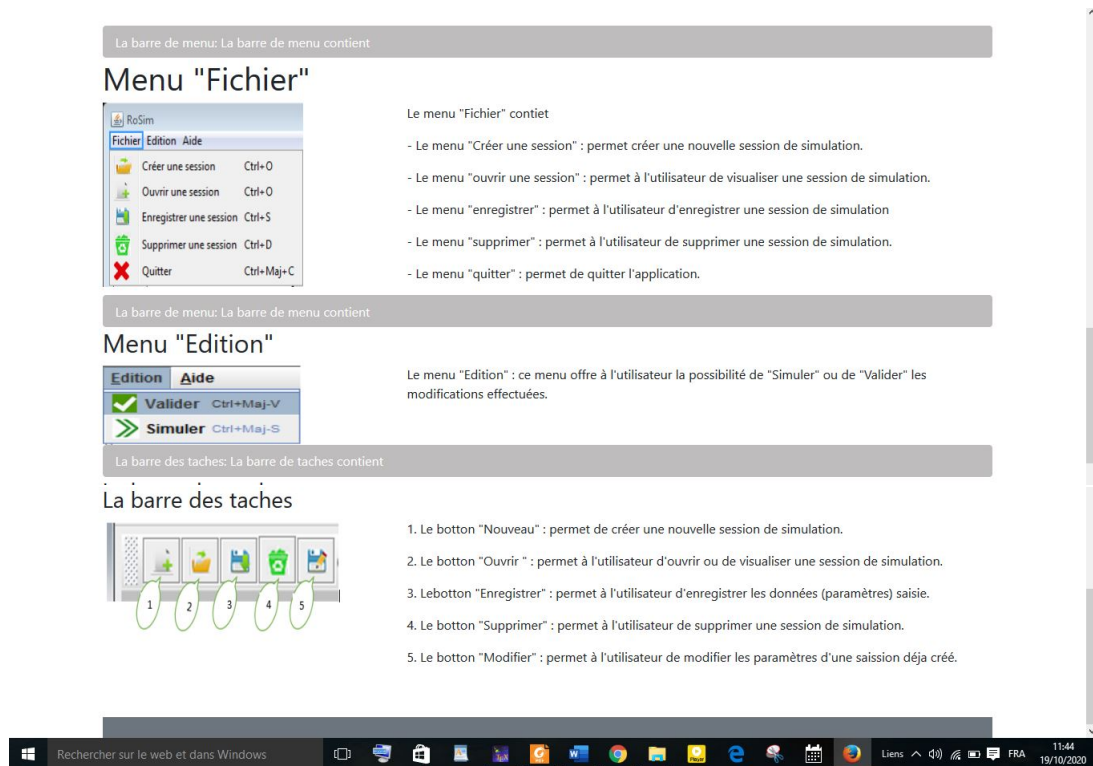


FIGURE 4.4 – Page web.

- ★ La barre des tâches contient des boutons pour la gestion des session : "Créer", "Ouvrir", "Enregistrer" et "supprimer".

### 4.3.2 Ouverture d'une session

Cette interface (présenté dans la figure 4.5) est constituée de deux champs à remplir (le nom et le mot de passe correspond à la session que l'on souhaite ouvrir) et deux bouton un pour "Valider" c'est à dire ouvrir la session demandée et l'autre pour l'annulation : "Annuler".

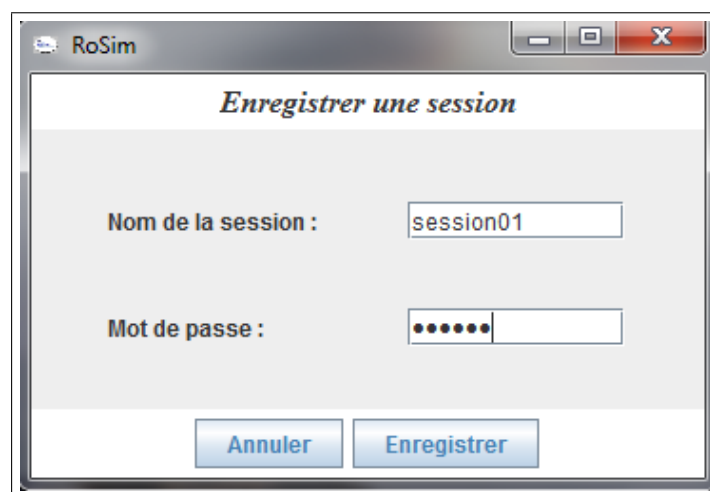


FIGURE 4.5 – Fenêtre ouvrir un session.

### 4.3.3 Gestion les paramètres du réseau

En cliquant sur le bouton dans la barre des tâches "Nouveau" ou dans le menu "Fichier" ou en ouvrant un session déjà créé l'interface représenté dans la figure 4.6 apparait.

Cette interface permet à l'utilisateur de gérer les paramètres du réseau en lui offrant la main à remplir les champs suivant :

- Le nombre de nœuds.
- Le débit des nœuds.
- L'énergie initiale.

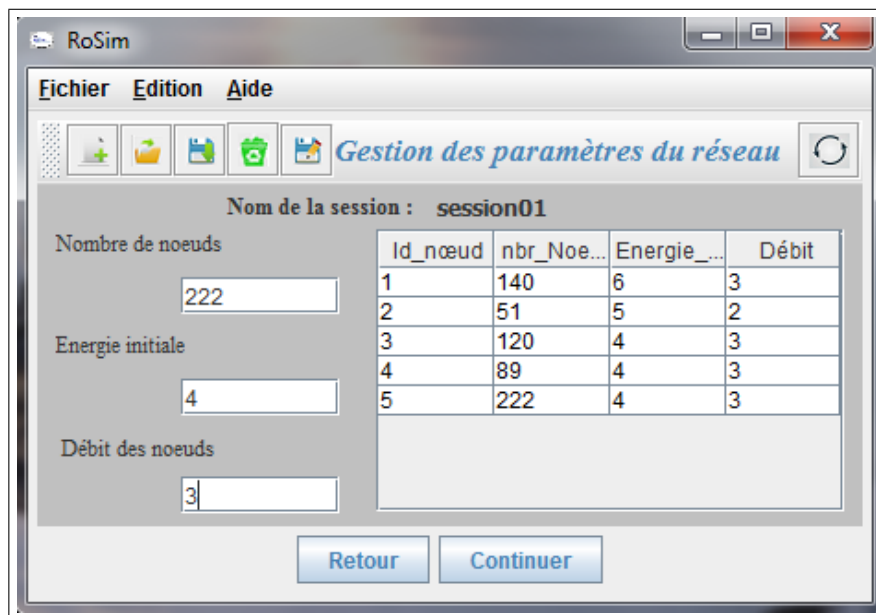


FIGURE 4.6 – Fenêtre gestion des paramètres du réseau

### 4.3.4 Identification des paramètres du routage

L'interface de la gestion des paramètres du routage (voir la figure 4.7) permet à l'utilisateur de gérer la politique du routage plus précisément choisir selon quel critère il souhaite effectuer le routage des données entre les nœuds du réseau et même d'introduire la fonction objectif.

A partir de cette interface il peut continuer et aller à l'étape suivante grâce au bouton "Continuer" ou faire un retour vers l'interface précédente.

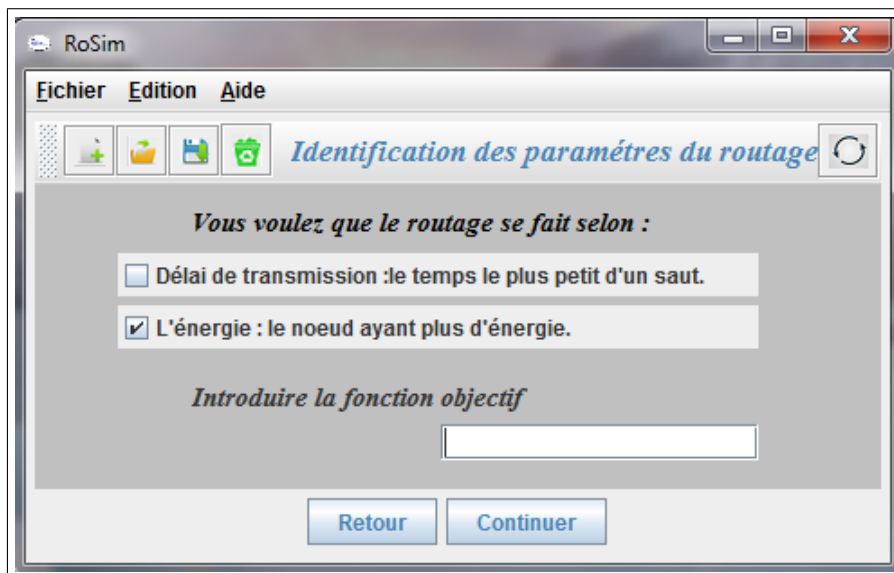


FIGURE 4.7 – Fenêtre de gestion des paramètres du routage

#### 4.3.5 Gérer les paramètres de la simulation

Dans cette interface l'utilisateur effectue son dernier choix qui consiste à choisir selon quoi il voudra avoir les résultats de la simulation.

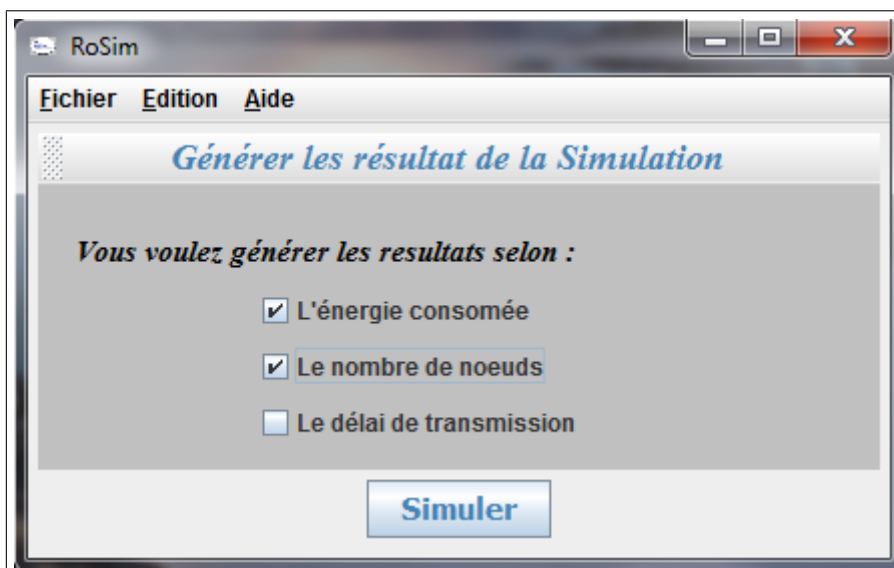


FIGURE 4.8 – Fenêtre de gestion des paramètres de simulation

### 4.3.6 Résultat de simulation

Après avoir géré les paramètres de réseau, routage et de simulation l'utilisateur peut visualiser les résultats de cette dernière sous forme du graphe.

#### ★ Premier scénario

Dans ce premier scénario nous initialisons les paramètres à

- ✓ Nombre de nœuds = 20
- ✓ Énergie initiale = 5
- ✓ Débit = 3

Pour la métrique du routage nous avons choisis l'énergie et comme paramètres de simulation nous avons choisis :

- ✓ L'énergie consommé
- ✓ Le nombre de nœuds

En simulant ce scénario nous obtenons les résultats représenté dans la figure 4.9 suivante :

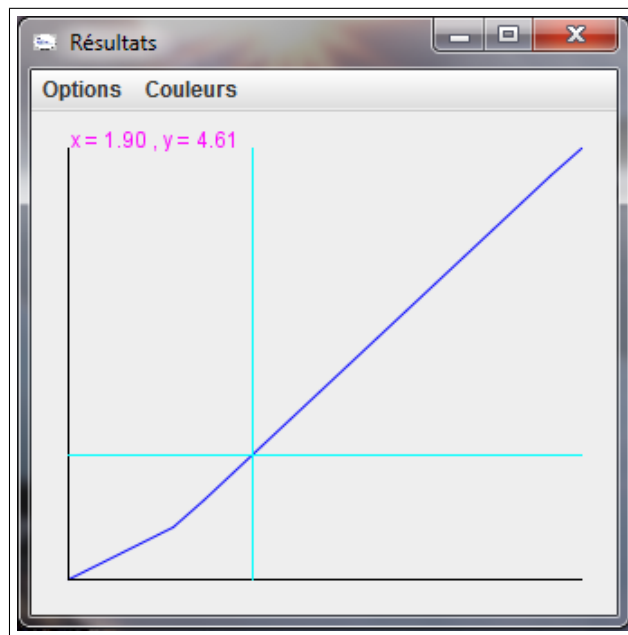


FIGURE 4.9 – Fenêtre de résultat de simulation du premier scénario.

#### ★ Deuxième scénario

Dans ce deuxième scénario nous avons gardé les mêmes paramètres de routage et de simulation alors que nous avons modifié la valeur des paramètres de routage à :

- ✓ Nombre de nœuds = 360
- ✓ Énergie initiale = 4
- ✓ Débit = 2

En simulant ce scénario nous obtenons les résultats représenté dans la figure 4.10 suivante :

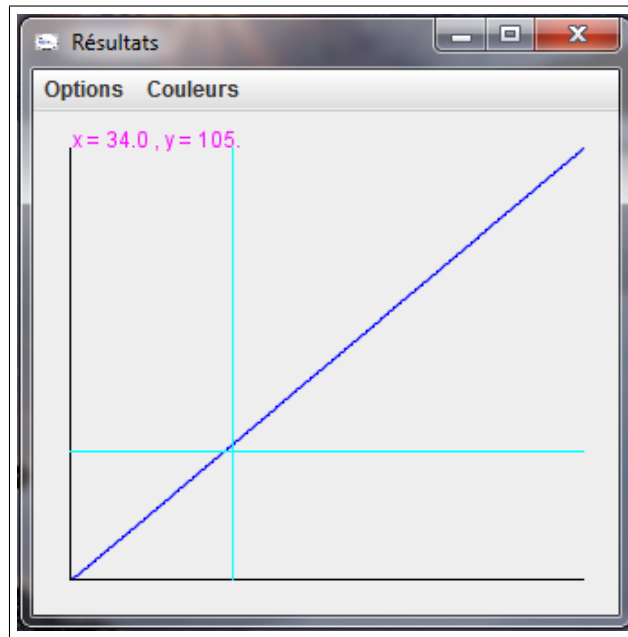


FIGURE 4.10 – Fenêtre de résultat de simulation du premier scénario.

D'après les résultats obtenue nous résumons que lorsque le nombre de nœuds est plus élevé la consommation de l'énergie deviennent plus intéressante(s'augmente).

## 4.4 Conclusion

Dans ce dernier chapitre nous avons décrit brièvement les outils de réalisation de notre application en spécifiant l'environnement et le langage de développement, puis nous avons enchainé par la présentation des principales fenêtres du simulateur, lors de la programmation nous avons privilégié la qualité du code et amélioré sa lisibilité afin de simplifier la contribution et la compréhension.

# Conclusion générale et perspectives

Les réseaux de capteurs sont composés d'un très grand nombre de dispositifs de communication ultra petits, autonomes avec des ressources de calcul et d'énergie limitées. Ils sont actuellement considérés comme l'une des technologies qui bouleverse notre façon de vivre, grâce à leur utilisation dans différents domaines d'application.

L'utilisation des batteries par les capteurs est une contrainte critique dans les réseaux de capteurs. La deuxième contrainte est environnementale : les capteurs sont parfois déployés sans surveillance et en grand nombre, de sorte qu'il est difficile de changer ou de recharger leurs batteries. Pour cela, les algorithmes et les protocoles de routage pour les réseaux de capteurs doivent minimiser la consommation d'énergie. En effet, plusieurs protocoles de routage ont été proposés dans la littérature afin d'améliorer les performances des RCSFs et de répondre aux besoins des applications considérées. En outre, plusieurs classifications de ces protocoles ont été proposées. Dans ce contexte plusieurs outils de simulation ont été développés pour évaluer les performances de ses nouveaux protocoles.

Nous nous sommes intéressés dans ce mémoire à la simulation du routage opportuniste comme principal objectif est de concevoir et réaliser un nouveau simulateur dédié au routage opportuniste dans les RCSFs qui vise à simplifier l'étude et la compréhension de ce type de réseau.

Effectivement, le travail consigné dans ce mémoire a été le fruit d'une étude menée dans le contexte des réseaux de capteurs sans fil, ce qui nous a permis de découvrir les propriétés de ces derniers, de leurs contraintes et des domaines variés qui les utilisent. Nous avons enchainé par la présentation du routage opportuniste dans les RCSFs afin d'améliorer certains critères de performances tel que la consommation d'énergie, la fiabilité, le débit, etc. Nous avons donc présenté ses principaux composants, ses métriques et puis quelques protocoles présentés dans la littérature. Par la suite, nous avons défini les besoins fonctionnels et non fonctionnels de notre simulateur, en décrivant les différents cas d'utilisations avec les diagrammes appropriés. Au final, nous avons donné une présentation de quelques interfaces de notre simulateur RoSim

et son fonctionnement ainsi que les différents outils utilisés pour sa réalisation.

Cette expérience a été très bénéfique et enrichissante pour nous, car elle nous a permis d'enrichir nos connaissances que ça soit dans la partie théorique ou pratique. Ça nous a permis également de découvrir et d'acquérir de nouvelles connaissances en matière de programmation et de développement.

Notre projet, bien que les objectifs soient atteints, est amené à évoluer, ce qui laisse la place aux perspectives suivantes :

- ★ Ajouter d'autres paramètres du RCSF (Taille de paquet, Taux de livraison...).
- ★ Intégrer d'autres protocoles du routage opportuniste (ExOR, ORTR...).



# Bibliographie

- [1] Y. YASER, Routage pour la gestion de l'énergie dans les réseaux de capteurs sans fil. Thèse du doctorat, Université de Haute Alsace Faculté des Sciences et Techniques. Année : 2010-2011.
  
- [2] S.ABI-AYAD épouse BRAHAMI, Evaluation du Protocole Directed Diffusion dans un réseau de capteurs sans fil, Mémoire master en Informatique, Université Abou Bakr Belkaid- Tlemcen, Année : 2013-2014.
  
- [3] G.PUJOLLE, Les réseaux. Eyrolles, 2008.
  
- [4] L.KHELLADI and N.BADACHE, Réseaux de capteurs : Etat de l'art, Rapport de recherche, Laboratoire LSI USTHB, Année : 2003-2004.
  
- [5] Z.Bouzidi and A.Benameur, Mise en place d'un réseau de capteur sans fil pour l'irrigation intelligente, Mémoire de fin d'étude pour l'obtention du diplôme de Master en Informatique, Année : 2013-2012.
  
- [6] N.KHOULALENE, Regroupement avec équilibrage de charge dans les réseaux de capteurs sans fil, Mémoire de fin d'étude pour l'obtention du diplôme de Magister en Informatique, Université Abderahmane Mira de Béjaïa, Année : 2006-2007.
  
- [7] E.DHIB, Routage avec QoS temps réel dans les réseaux de capteurs, Technical report, Ecole supérieur des communications de Tunis, Année 2006-2007.

- [8] A.Delye, V.Gauthier, M.Marot et M.Becker, "Etat de l'art sur les réseaux de Capteurs", Rapport de Recherche, Institut National des Télécommunications. Année 2004-2005.
  
- [9] K.MESSAD et L.RAHMANI, Le routage opportuniste dans les réseaux de capteurs sans fil, Mémoire de fin d'étude, Université Abderahmane Mira de Béjaïa, Année : 2015 -2016.
  
- [10] C-T.Kone, Conception de l'architecture d'un réseau de capteurs sans fil de grande dimension. Thèse du doctorat, Université Henri Poincaré-Nancy I, Année : 2010-2011.
  
- [11] D.ANKI et L.LEBIB, Routage avec conservation d'énergie dans les réseaux de capteurs sans fil, Mémoire fin de cycle en informatique, Université Abderahmane Mira de Béjaïa, Année : 2015-2016.
  
- [12] S.TALEB et A.HOCINE, Simulation du Routage dans les Réseaux de Capteurs Sans Fils, Mémoire fin de cycle en informatique, Université Abderahmane Mira de Béjaïa. Année : 2015-2016.
  
- [13] Y.CHALLAL. " Réseaux de capteurs sans fil ", cours Université de Technologie de Compiègne, France, Année : 2008.
  
- [14] M.KHELIFI, Optimisation de la consommation de l'énergie et maximisation de la durée de vie des réseaux capteurs sans fil, Mémoire de Magister en Informatique, Université Abderahmane Mira de Béjaïa. Année : 2007-2008.
  
- [15] S.belkheyr, Etude d'un protocole de routage basé sur les colonies de Fourmis dans les réseaux de capteurs sans fil, Mémoire de fin d'études Pour l'obtention du diplôme de Master en Informatique ,Université Abou Bakr Belkaid- Tlemcen, Année : 2012-2013.
  
- [16] R.MOHAMED, Problèmes de sécurité dans les Réseaux de capteurs avec prise en charge de l'énergie, Mémoire de magister, Université de Saad Dahlab de Blida. Année : 2012-2013.

- [17] A.Bouabdallah, H.Betthahar, Y.Challal, Les Réseaux de capteurs (WSN : Wireless Sensor Networks), Cours, Université de Technologie de Compiègne, France, 2008.
- [18] N.HELAILI et S.MEKHNACHE .Simulation du Routage dans les Réseaux de Capteurs Sans Fils. Mémoire de master 2 en Informatique. Option Administration et Sécurité des Réseaux. Faculté des Sciences Exacte. Université Abderrahmane Mira – Bejaia. 2016 – 2017.
- [19] L.Loiseau. De l'exploitation des réceptions opportunistes dans les mécanismes de relaying pour les réseaux sans-fil. PhD thesis, l'Université de Rennes 1, 2013
- [20] L.babu and P.Balasubramanie. A survey on opportunistic routing protocols. International Journal of Innovative Research in Computer and Communication Engineering, 2014
- [21] Jadhav. A survey on opportunistic routing protocols for wireless sensor networks. Elsevia journal, procedia computer science, 2016
- [22] S. Chachulski, M. Jennings, S. Katti and D. Katabi, Trading structure for randomness in wireless opportunistic routing, Conference of the Special Interest Group on Data Communication, Kyoto, Japan, 2007, pp. 169-180.
- [23] A.TIAB. Routage dans les RCSFs en environnement industriel : économie d'énergie et qualité de service. Thèse de doctorat. Domaine Mathématique et Informatique. Filière Informatique. Spécialité : Réseaux et Systèmes Distribués. Unité de recherche de rattachement LaMOS (Modélisation et Optimisation des Systèmes). Université Abderrahmane Mira – Bejaia. 2016 – 2017.
- [24] A.Darehshoorzadeh, L. Cerda-Alabern, and V. Pla. Opportunistic routing in wireless mesh networks. springer science and business media, 2013.
- [25] L.Debrauwer, F.Vander hayde, "UML 2 Modélisation des objets", ENI, 2eme édition, 2010. 19, 20, 21, 30, 32

[26] C.Delannoy, "Programmer en Java", EYROLLES, 2006. 34

[27] M.Olivier, "SQL Server 2014", ENI, 2015. 35

# Webographie

- [28] [www.supinfo.com/articles/single/1860-ngage-uml](http://www.supinfo.com/articles/single/1860-ngage-uml)
- [29] [www.irisa.fr/triskell/members/pierre-alain.muller/teaching/demarche](http://www.irisa.fr/triskell/members/pierre-alain.muller/teaching/demarche).
- [30] [www.sparxsystems.fr/resources/uml2\\_tutorial/uml2\\_usecasediagram.html](http://www.sparxsystems.fr/resources/uml2_tutorial/uml2_usecasediagram.html).
- [31] [Dictionnaire.sensagent.leparisien.fr/Diagramme d'activit](http://Dictionnaire.sensagent.leparisien.fr/Diagramme_d%27activit).
- [32] [laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes](http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes).
- [33] "Definition d'eclipse". [www.techno-science.net/onglet=glossaire/definition=517](http://www.techno-science.net/onglet=glossaire/definition=517), 2016.
- [34] [www.encyclopedie.fr/definition/WampServer](http://www.encyclopedie.fr/definition/WampServer)
- [35] "telecharger-34058460-visual-paradigm ", [www.commentcamarche.net](http://www.commentcamarche.net)
- [36] "Site officiel de texstudio". [www.texstudio.org/](http://www.texstudio.org/), 2016.

# Résumé

Le routage opportuniste est un mécanisme de routage qui prend en compte l'exploitation de son environnement. Il utilise le concept de diffusion qui caractérise les Réseaux de Capteurs Sans Fil (RCSFs) pour acheminer convenablement les paquets. Dans ce mémoire, nous nous sommes intéressés à la simulation du routage opportuniste dans les RCSFs, où nous avons dressé un état de l'art des RCSFs et du routage opportuniste dans ce type de réseau. Nous avons également mis l'accent sur quelques simulateurs des RCSFs des plus connus et nous avons constaté l'absence de modules dédiés au Routage opportuniste. Afin d'y remédier nous avons conçu et réaliser un nouveau simulateur que nous avons nommé RoSim dédié au routage opportuniste dans les RCSF. RoSim permet de simuler en tenant compte de délai de transmission ainsi que l'énergie comme métriques et visualiser les résultats sous forme de graphe.

**Mots clés :** Réseaux de Capteurs Sans Fil (RCSF), Simulation, Routage opportuniste, Simulateur, Java.

## Abstract

Opportunistic routing is a new routing mechanism, which takes into account the exploitation of its environment. It uses the broadcast concept that characterizes Wireless Sensor Networks (WSNs) to properly route packets. In this thesis, we interested in the simulation of opportunistic routing in RCSFs, where we have drawn up a state of the art of RCSFs and opportunistic routing in this type of network. We also focused on some of the most well-known WSNs simulators and noted the lack of modules dedicated to Opportunistic Routing. In order to remedy this, we designed and built a new simulator that we named RoSim dedicated to opportunistic routing in RCSF. RoSim allows simulating taking into account transmission delay as well as energy as metrics and visualizing the results as a graph.

**Keywords :** Wireless Sensor Networks (WSN) , Simulation, Opportunistic Routing, Simulator, Java.