

UNIVERSITÉ DU QUÉBEC

MÉMOIRE PRÉSENTÉ À  
L'UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

COMME EXIGENCE PARTIELLE  
DE LA MAÎTRISE EN MATHÉMATIQUES ET INFORMATIQUE  
APPLIQUÉES

PAR  
HIND KHACHIAA

SIMULATION DE COMMUNICATION DES ROBOTS DE CHAMP DANS UN  
CHAMP DE BLEUETS

JUIN 2021

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire ou de cette thèse a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire ou de sa thèse.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire ou cette thèse. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire ou de cette thèse requiert son autorisation.

**Ce mémoire a été évalué par un jury composé de:**

➤ **François Meunier;**  
Professeur au département de Mathématiques et Informatique  
Université du Québec à Trois-Rivières

➤ **Mhamed Mesfioui;**  
Professeur au département de Mathématiques et Informatique  
Université du Québec à Trois-Rivières.

➤ **Fathallah Nouboud;**  
Professeur au département de Mathématiques et informatique  
Université du Québec à Trois-Rivières.

*À mes chers parents,*

*À mes chers frères,*

*À ma chère famille,*

*À tous ceux qui me sont chers, ...*

*Je dédie avec reconnaissance ce modeste travail*

*Hind*

## **REMERCIEMENTS**

Je tiens d'abord à remercier profondément mon directeur de recherche, professeur François Meunier, de m'avoir guidé tout au long de ma maitrise.

Je remercie infiniment l'ensemble des professeurs du département de mathématiques et informatiques qui ont contribué à l'avancement de mon parcours scolaire, ainsi que toute l'équipe du département.

Je tiens à remercier également madame Chantal Guimond, la secrétaire de notre département qui trouvait toujours des réponses à mes questions.

Enfin, je tiens à remercier de tout mon cœur tous ceux qui m'ont soutenu durant mon cursus. J'exprime ma gratitude envers vous à travers ce travail.

## TABLE DES MATIÈRES

REMERCIEMENTS .....	III
TABLE DES MATIÈRES .....	IV
LISTE DE TABLEAUX .....	VI
LISTE DES FIGURES.....	VII
LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES .....	VIII
Sommaire.....	III
Abstract .....	IV
Chapitre 1 : Introduction .....	- 1 -
1.1    Mise en contexte .....	- 1 -
1.2    Problématique .....	- 2 -
1.3    Projet et objectif du mémoire .....	- 2 -
1.4    Structure du mémoire.....	- 3 -
Chapitre 2 : Les robots sarclieurs et les réseaux Ad-Hoc.....	- 4 -
2.1    Les systèmes de pilotage pour robots sarclieurs.....	- 4 -
2.2    Les réseaux Ad-Hoc .....	- 5 -
2.2.1    Définition .....	- 5 -
2.2.2    Modes de communication des réseaux Ad-Hoc.....	- 6 -
2.2.2.1    V2I (Vehicule To Infrastructure) .....	- 6 -
2.2.2.2    V2V (Vehicule To Vehicule) .....	- 6 -
2.3    L'architecture du réseau Ad-Hoc .....	- 6 -
2.3.1    Connectivité intranet.....	- 7 -
2.3.2    TCP/IP .....	- 8 -
2.3.3    Connectivité globale .....	- 8 -
2.4    Les protocoles de routage Ad-Hoc.....	- 9 -
2.4.1    Protocoles topologiques proactifs .....	- 9 -
2.4.1.1    Optimized Link State Routing (OLSR) .....	- 9 -
2.4.1.2    Topology Broadcast based on also Reverse Path Forwarding (TBRPF).....	- 10 -
2.4.1.3    Fisheye state routing (FSR).....	- 10 -
2.4.1.4    Destination –also Sequenced a Distance Vector (DSDV) .....	- 11 -

2.4.2	Protocoles topologiques réactifs.....	- 11 -
2.4.2.1	AODV (Ad-Hoc On-Demand Distance Vector) .....	- 12 -
2.4.2.2	DSR (Dynamic Source Routing).....	- 12 -
2.4.2.3	TORA (Temporally Ordered Routing Algorithm) .....	- 13 -
2.4.3	Protocoles hybrides .....	- 13 -
2.5	Conclusion .....	- 14 -
CHAPITRE 3 : Environnements de simulations.....		- 15 -
3.1	SUMO (“Simulation of Urban Mobility”).....	- 15 -
3.2	Objective Modular Network Test-bed in C++ (OMNET++) .....	- 17 -
3.3	NS2 (Network Simulator 2).....	- 18 -
3.4	NS3 (Network Simulator 3).....	- 19 -
3.5	Conclusion .....	- 20 -
Chapitre 4 : SIMULATIONS ET RÉSULTATS .....		- 21 -
4.1	Environnements de simulations .....	- 21 -
4.2	Schématisation de l'environnement de simulation .....	- 22 -
4.3	La configuration du protocole de routage AODV .....	- 31 -
4.4	Résultats des simulations .....	- 33 -
4.4.1	La communication entre robots .....	- 34 -
4.4.2	L'importance du RSU .....	- 35 -
4.4.3	La communication entre RSU et robots.....	- 38 -
4.4.4	La bande passante et le nombre des mauvaises herbes dans le champ .....	- 41 -
4.5	Discussion des résultats .....	- 43 -
4.6	Conclusion .....	- 45 -
Chapitre 5 : Conclusions et Perspectives .....		- 46 -
LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES .....		- 48 -

## LISTE DE TABLEAUX

Tableau 4.1 : Paramètres de la simulation 1 .....	- 21 -
Tableau 4.2 : Paramètres de la simulation 2 .....	- 23 -
Tableau 4.3 : Paramètres de la simulation 3 .....	- 24 -
Tableau 4.4 : Paramètres de la simulation 4 .....	- 25 -
Tableau 4.5 : Paramètres de la simulation 5 .....	- 26 -
Tableau 4.6 : Paramètres de la simulation 6 .....	- 27 -
Tableau 4.7 : Paramètres de la simulation 7 .....	- 28 -
Tableau 4.8 : Paramètres de la simulation 8 .....	- 29 -

## LISTE DES FIGURES

Figure 2.1 : Robot sarceur autonome et électrique de conception québécoise [18] .....	- 5 -
Figure 3.1 : La carte de Trois-Rivières avant l'extraction des routes par SUMO .....	- 16 -
Figure 3.2 : La carte de Trois-Rivières après l'extraction des routes par SUMO .....	- 16 -
Figure 3.3 : Exemple de simulation sur OMNET ++ [21] .....	- 18 -
Figure 3.4 : Exemple de simulation sur NS2 [19] .....	- 19 -
Figure 3.5 : Exemple de simulation sur NS3 [20] .....	- 20 -
Figure 4.1 : Simulation du flux de la communication entre deux nœuds (Sans RSU) .....	- 22 -
Figure 4.2 : Simulation du flux de la communication entre deux nœuds (Avec RSU) .....	- 23 -
Figure 4.3 : Simulation du flux de la communication entre quatre nœuds (Sans RSU) .....	- 25 -
Figure 4.4 : Simulation du flux de la communication entre quatre nœuds (Avec RSU) .....	- 26 -
Figure 4.5 : Simulation du flux de la communication entre huit nœuds (Sans RSU) .....	- 27 -
Figure 4.6 : Simulation du flux de la communication entre huit nœuds (Avec RSU) .....	- 28 -
Figure 4.7 : Simulation du flux de la communication entre seize nœuds (Sans RSU) .....	- 29 -
Figure 4.8 : Simulation du flux de la communication entre seize nœuds (Avec RSU) .....	- 30 -
Figure 4.9 : Le fonctionnement du protocole AODV .....	- 33 -
Figure 4.10 : Le taux de consommation de la bande passante par rapport au nombre de mauvaises herbes ..	- 34 -
Figure 4.11 : Modèle de RSU [17] .....	- 35 -
Figure 4.12 : Le pourcentage des nœuds qui ont reçu une MAJ (sans RSU) par rapport au temps .....	- 36 -
Figure 4.13 : Le pourcentage des nœuds qui ont reçu une MAJ (avec RSU) par rapport au temps .....	- 37 -
Figure 4.14 : Nombre de paquets transférés par les nœuds par rapports à la distance avec RSU .....	- 38 -
Figure 4.15 : Pourcentage des paquets reçus avec succès par rapport à l'augmentation de la surface du terrain (Sans RSU) .....	- 39 -
Figure 4.16 : Pourcentage des paquets reçus avec succès par rapport à l'augmentation de la surface du terrain (Avec RSU) .....	- 39 -
Figure 4.17 : Pourcentage des paquets reçus avec succès par rapport à l'augmentation de la surface du terrain (Avec deux RSU) .....	- 40 -
Figure 4.18 : Nombre de paquets transférés par les nœuds par rapports a la distance avec RSU .....	- 41 -
Figure 4.19 : Le rapport entre la consommation de la bande passante et l'augmentation de la quantité des mauvaises herbes dans le champ .....	- 42 -
Figure 4.20 : Le rapport entre la consommation de la bande passante et l'augmentation de la quantité des mauvaises herbes dans le champ .....	- 43 -

## LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

**Anycast:** Un vers Un parmi plusieurs

**AODV:** Ad-Hoc On-Demand Distance Vector

**Broadcast:** Un vers Tous

**CEDAR:** Core-Extraction Distributed Ad-Hoc Routing Algorithm

**CLR:** Clear

**DSDV:** Destination-Sequenced also a Distance Vector

**DSR:** Dynamic Source Routing

**FSR:** Fisheye State Routing

**GPS:** Global Positioning System

**MANET:** Mobile Ad-Hoc Networks

**Multicast:** Un vers Plusieurs

**MPR:** MultiPoint Relay

**MAJ:** Mise À Jour

**NAM:** Network Animator

**NS2:** Network Simulator 2

**NS3:** Network Simulator 3

**OLSR:** Optimized Link State Routing

**OMNET++:** Objective Modular Network Test-bed in C++

**OSM:** OpenStreetMap

**QRY:** QUERY

**RSU:** Road Side Unit

**RIP:** Routing Information Protocol

**RERR:** Route Error

**RREQ:** Route Request

**RREP:** Route Response

**SUMO:** Simulation Of Urban Mobility

**TBRPF:** Topology Broadcast Based on also Reverse Path Forwarding

**TORA:** Temporally Ordered Routing Algorithm

**UDP:** Update

**UNICAST:** Un vers Un

**V2I:** Vehicle to Infrastructure

**V2V:** Vehicle to Vehicle

**VEINS:** Vehicles In Network Simulation

**WRAP:** Wireless Ad-Hoc Routing Protocol

**ZRP:** Zone Routing Protocol

# **SIMULATION DE COMMUNICATION ENTRE ROBOTS SARCLEURS DE CHAMPS (AD-HOC)**

**Hind Khachiaa**

## **Sommaire**

L'objectif de ce mémoire est de tester les communications entre robots sarclieurs de champs de bleuets sauvages en effectuant une série de simulations. Ce projet consiste à simuler un groupe de robots mobiles qui coopèrent pour permettre le sarclage des mauvaises herbes dans des champs de bleuets sauvages.

Il faudra d'abord simuler le processus de communication sur la base d'un protocole de communication sans fil, entre chaque robot mobile avec le serveur (station fixe). Le rôle du serveur est de maintenir la topologie dynamique du réseau. Cependant, le serveur sauvegarde une copie de la table de routage, ainsi que la carte géoréférencée mise à jour des positions des mauvaises herbes à sarcler pour ensuite transmettre à chaque robot les données à propos des emplacements des mauvaises herbes à sarcler (dans ces simulations, les données de position sont générées aléatoirement). En conséquence, nous voulons visualiser ces interactions sur une interface qui va se composer d'une carte, représentant notre champ. Deux types de marqueurs auxquels seront attribuées des couleurs et formes différentes représenteront les mauvaises herbes qui ne sont pas encore sarclées ( points rouges), et les robots (cercles bleus). Ainsi, après la réception de l'emplacement d'une mauvaise herbe du serveur, le robot concerné doit se déplacer vers la mauvaise herbe à sarcler, effectuer l'opération de sarclage et retourner un accusé de réception au serveur après avoir terminé l'opération de sarclage. Il sera alors possible de constater sur l'interface que le point rouge lié à un emplacement d'une mauvaise herbe disparaîtra de la simulation.

La portée du signal Ad-Hoc est limitée, alors nous devons assurer une communication entre les robots distants et le serveur qui est fixe. D'un autre côté, la création d'une communication entre les robots est essentielle pour permettre le transfert d'informations d'un robot à un autre. Les informations transférées entre les robots permettront d'abord à chaque robot de maintenir, tout comme le serveur, la topologie dynamique du réseau. Chaque robot aura une tâche de routage qui permettra la continuité de la communication entre l'ensemble du réseau même en situation de hors portées.

Mots-clés : Ad-hoc, sécurité, routage, simulation, robot

# COMMUNICATION SIMULATION BETWEEN FIELD WEEDING ROBOTS (AD-HOC)

Hind Khachiaa

## Abstract

The objective of this dissertation is to test communications between robot weeders in fields of wild blueberries by performing a series of simulations. This project consists of simulating a group of mobile robots which cooperate to enable weeding of weeds in fields of wild blueberries.

It will first be necessary to simulate the communication process on the basis of a wireless communication protocol, between each mobile robot with the server (fixed station). The role of the server is to maintain the dynamic topology of the network. However, the server saves a copy of the routing table, as well as the updated georeferenced map of the positions of the weeds to be weeded in order to then transmit to each robot the data about the locations of the weeds to be weed (in these simulations, position data is generated randomly). Consequently, we want to visualize these interactions on an interface which will consist of a map, representing our field. Two types of markers which will be assigned different colors and shapes will represent weeds that are not yet weeded (red dots), and robots (blue circles). Thus, after receiving the location of a weed from the server, the concerned robot must move to the weed to be weeded, perform the weeding operation, and return an acknowledgment to the server after completing the operation. It will then be possible to see on the interface that the red dot linked to a location of a weed will disappear from the simulation.

The range of the Ad-Hoc signal is limited, so we have to ensure communication between the remote robots and the server which is fixed. On the other hand, creating communication between robots is essential to enable the transfer of information from one robot to another. The information transferred between robots will first allow each robot to maintain, like the server, the dynamic topology of the network. Each robot will have a routing task that will allow the continuity of communication between the entire network even in an out-of-range situation.

Keywords: Ad-hoc, security, routing, simulation, robot

## Chapitre 1 : Introduction

### 1.1 Mise en contexte

L’agriculture est un domaine qui saurait profiter davantage de l’automatisation, et ce pour plusieurs raisons. D’une part, ce domaine souffre d’un manque criant de main-d’œuvre. Ceci est partiellement dû au fait que le travail y est en général physiquement demandant [25]. D’autre part, l’automatisation des opérations de contrôle des mauvaises herbes sous régie conventionnelle permettrait d’économiser temps et argent, tout en évitant l’utilisation excessive d’herbicides [28].

Dans les champs sous régie biologique, les opérations d’éradiation des mauvaises herbes sont souvent effectuées par sarclage manuel, ce qui rend ces opérations longues et coûteuses. Il serait donc possible de réaliser le même genre d’économies que dans les champs sous régie conventionnelle en automatisant certains processus [6].

C’est dans les années 1920 que la robotique fit son apparition dans le domaine de l’agriculture, accompagnée de recherches visant l’intégration du guidage automatique des véhicules dans les champs, comme les tracteurs et les moissonneuses-batteuses. Ces recherches couvrent un large éventail d’applications, dont la récolte automatisée, la pulvérisation ciblée autonome pour la lutte contre les parasites et les mauvaises herbes, et la conception optimale des techniques de localisation et de cartographie simultanées pour la taille des plantes [26].

Dans ce contexte de robotisation de l’agriculture, les robots agricoles contribuent entre autres à augmenter la fiabilité des opérations, à améliorer la santé des sols, et à améliorer le rendement des récoltes. Ces robots sont généralement équipés de plusieurs capteurs et caméras afin de contrôler la navigation, la localisation et la cartographie simultanées et les algorithmes de planification de trajectoire [26].

La robotique agricole a pour principal but de remplacer la main-d’œuvre humaine devenue de plus en plus rare, par des robots de terrain ou des systèmes automatisés capables de gérer les tâches de manière plus précise et uniforme à moindre coût et avec une plus grande efficacité [26].

Dans le cadre de notre recherche, nous nous intéressons précisément aux champs de bleuets nains qui, dus à leur nature sauvage et leur développement anarchique dans les champs, présentent certains défis aux agriculteurs québécois. Nous nous concentrerons donc sur les robots qui ont pour tâche d'éliminer les mauvaises herbes dans ces champs en particulier.

## **1.2 Problématique**

Comme certaines mauvaises herbes, le bleuet sauvage préfère pousser d'une manière anarchique dans des sols acides. Par conséquent, si on n'utilise pas des mesures de lutte contre les mauvaises herbes qui poussent dans les bleuetières, elles vont finir par coloniser les surfaces en culture et entrer alors en compétition avec le bleuet sauvage, en absorbant la lumière et les éléments nutritifs destinés aux bleuets, ceci pourrait avoir des effets nocifs sur la qualité et la quantité de la récolte [14].

En ce qui concerne la production, il existe deux types : une production traditionnelle (conventionnelle) et une production biologique.

Au Québec, un produit biologique ne peut venir que d'une entreprise certifiée par un organisme de certification, ce qui implique que l'agriculteur a besoin de respecter le cahier de charge de cet organisme afin de vendre des bleuets reconnus biologiques [1]. Contrairement à la production biologique, la production traditionnelle permet l'utilisation des engrains chimiques et herbicides, alors les bleuetières biologiques ont besoin de plus de précision en ce qui concerne l'opération de sarclage des mauvaises herbes.

Nous pouvons donc formuler la problématique qui se présente devant nous dans la question suivante : comment éliminer les mauvaises herbes à moindres coûts, tout en préservant une bonne qualité de la récolte [24] ?

## **1.3 Projet et objectif du mémoire**

Pour éliminer les mauvaises herbes et récolter un bleuet de qualité en grande quantité, il faut viser les mauvaises herbes directement.

Dans le cas d'une production traditionnelle, les robots ont comme tâche de pulvériser des herbicides sur les mauvaises herbes, et les mauvaises herbes seulement.

Dans une production biologique, les robots devraient arracher ou déchiqueter les mauvaises herbes avec une toupie rotative.

Dans les deux cas, la précision est la solution qui peut garantir une meilleure qualité de récolte. Les productions traditionnelles éviteront la pulvérisation des herbicides sur le fruit, et les productions biologiques fourniront un environnement sain dans lequel les bleuets n'auront plus à rivaliser pour les ressources.

L'objectif de ce mémoire repose principalement sur l'établissement du mécanisme de communication pour ce réseau. Pour réaliser notre travail, nous avons essayé plusieurs types de simulateurs et de protocoles afin d'étudier le comportement des robots à travers des simulations.

#### **1.4 Structure du mémoire**

En ce qui concerne la structure du présent mémoire, nous considérons les chapitres 2 et 3 comme l'état de l'art. Ils résument les articles et les mémoires sur lesquelles nous nous sommes basés pendant nos recherches. Nous présenterons dans le chapitre 2 les différents types des robots sarclieurs, l'architecture des réseaux Ad-Hoc, ainsi que les différents types des protocoles de routage Ad-Hoc. Par la suite, le chapitre 3 présentera les différents environnements de simulations avec leurs avantages et inconvénients. Le chapitre 4 décrit des simulations conçues sur le simulateur NS2 avec le protocole AODV, et explique les différents résultats de ces simulations. Enfin, le chapitre 5 sert de conclusion pour ce mémoire.

## Chapitre 2 : Les robots sarceurs et les réseaux Ad-Hoc

Pour appliquer le concept des réseaux Ad-Hoc sur les robots sarceurs, nous avons besoin de comprendre le fonctionnement des robots et l'architecture des réseaux Ad-Hoc. Dans ce deuxième chapitre, nous allons voir les différents systèmes de pilotage pour les robots sarceurs. Ensuite, nous allons définir les réseaux Ad-Hoc et leurs différents types de protocoles.

### 2.1 Les systèmes de pilotage pour robots sarceurs

Il existe différents types de robots sarceurs, mais nous sommes plus intéressés par des robots qui possèdent des systèmes de pilotage qui permettent de donner le maximum de précision au robot.

L'auteur du [16] s'est concentré sur trois types de systèmes qui pilotent les robots sarceurs :

- Systèmes de guidage pilotés par caméra : L'auteur mentionne que ce type de guidage est utilisé depuis plus de 10 ans maintenant. Tandis que le guidage par caméra est l'un des systèmes les plus utilisés, il présente encore des problèmes de précision et de performance surtout pour les champs de grandes surfaces. Les résultats du guidage par caméra sont proches de ceux du guidage manuel avec l'avantage d'un travail confortable, ainsi l'adoption de ce système de pilotage pour robots sarceurs dépend principalement des besoins de l'agriculteur.
- Systèmes de guidage pilotés par satellite : Le concept pour ce type de guidage est inspiré par les systèmes de navigation pour les voitures. Le signal GPS offre en temps normal pour les voitures une précision de cinq à quinze mètres, cependant dans le milieu agricole cette précision est considérée faible. La solution adoptée en Suisse est d'installer des émetteurs partout dans le pays qui reçoivent et corrigent les signaux des satellites afin d'améliorer la précision jusqu'aux environs des deux centimètres [16]. Ce type de guidage très précis possède une sorte de mémoire qui détecte et mémorise les lignes habituelles des mauvaises herbes dans un champ. Ce système de guidage par GPS permet un sarclage précis même en cas de dysfonctionnement des caméras. Le système de guidage par GPS offre un plus grand confort que celui guidé par caméra, mais il coûte plus cher.

- Systèmes automatiques : Ce type de robots peuvent sarcler les mauvaises herbes avec précision sans l'utilisation du GPS, grâce à la reconnaissance optique. C'est un nouveau système comparé aux autres, ce qui explique ses erreurs de sarclage dans certaines occasions ainsi que son prix élevé.

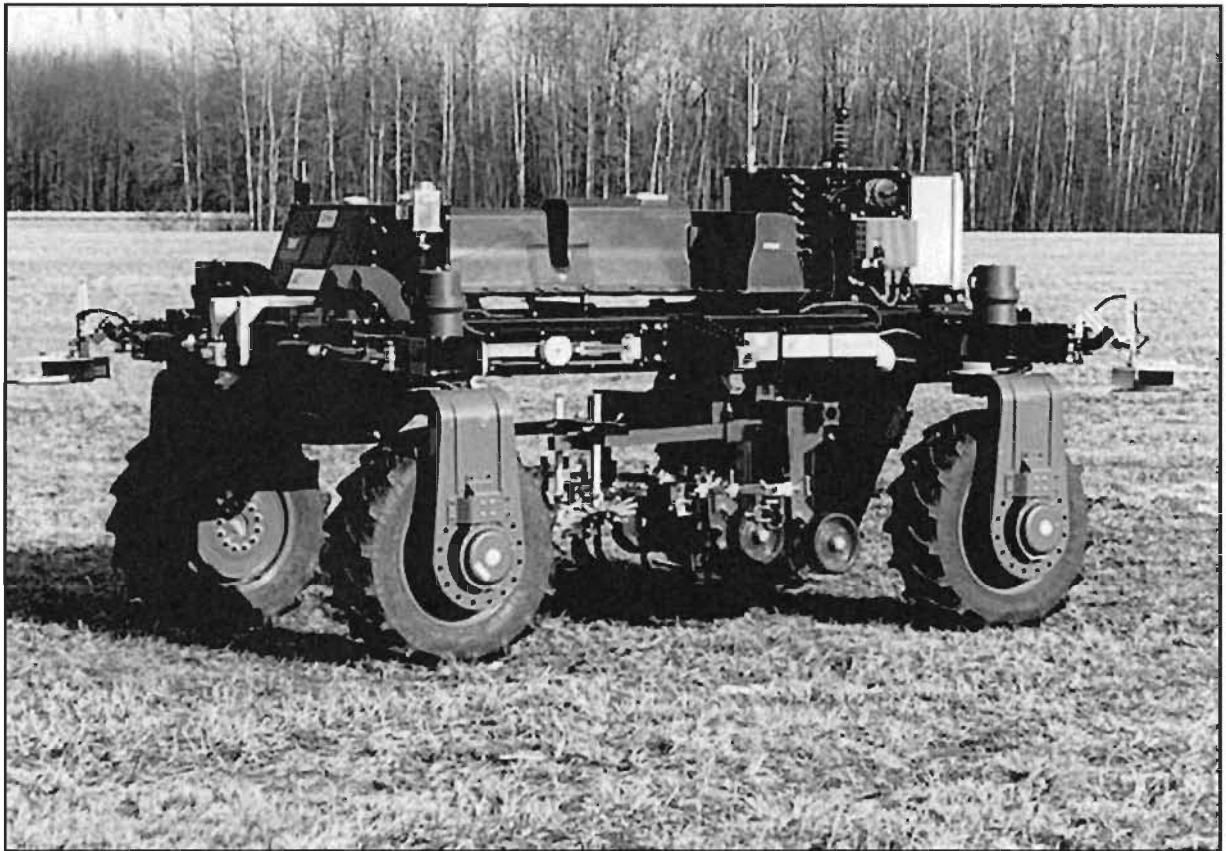


Figure 2.1 : Robot sarcleur autonome et électrique de conception québécoise [18]

## 2.2 Les réseaux Ad-Hoc

### 2.2.1 Définition

Un réseau Ad-Hoc mobile, ou un MANET (Mobile Ad-hoc Networks), est un réseau autonome sans fil qui est non centralisé. Un nœud peut passer des données aux autres nœuds voisins, ce qui permet à chaque nœud de devenir un routeur en cas de besoin. Cela est réalisé grâce aux protocoles

de routage Ad-Hoc, qui créent une communication entre chaque paire de nœuds du réseau [8]. Il existe deux modes de communication au sein des réseaux Ad-Hoc :

### **2.2.2 Modes de communication des réseaux Ad-Hoc**

#### **2.2.2.1 *V2I (Vehicule To Infrastructure)***

La communication entre les nœuds du réseau passe par le RSU qui est une unité électronique intermédiaire. Dans le cas où un nœud X a besoin de communiquer avec un nœud Y qui est hors de sa portée, il communiquera les données à échanger au RSU qui va les envoyer à son tour au nœud Y. Cette méthode est surtout efficace pour les grands réseaux de communication, en mettant des RSU dans des endroits stratégiques on garantit que la communication ne se perdra pas même si les nœuds sont hors de portée les uns des autres.

#### **2.2.2.2 *V2V (Vehicule To Vehicule)***

Ce type de communication permet aux nœuds de communiquer directement avec leurs nœuds voisins sans le biais de l'infrastructure. Le mode V2V est parfait pour les petits réseaux où tous les nœuds sont dans la même portée. Cependant, il peut aussi être utilisé pour les grands réseaux pour un échange entre deux nœuds voisins qui sont proches l'un de l'autre.

### **2.3 L'architecture du réseau Ad-Hoc**

La communication entre les hôtes est la mission principale d'un réseau Ad-Hoc. Supposant qu'il y a des nœuds qui sont hors de portée, un réseau Ad-Hoc implémente un mécanisme de routage complexe afin de réaliser des communications multi-sauts, offrant ainsi un routage pair à pair dans un domaine sans fil [8]. Ce modèle de routage pair à pair repose sur la liberté d'échanger des

fichiers de différents types, en permettant ainsi la conception de systèmes à forte disponibilité et de taille importante [10].

Le routage d'un réseau Ad-Hoc doit garantir plusieurs services, par exemple supporter les communications de type Unicast, mais aussi de type Broadcast ou Multicast. Dans notre modèle, nous devons aussi garantir les exigences de sécurité de communication du mécanisme :

- L'authentification : est la propriété des réseaux Ad-Hoc qui autorise les nœuds du réseau à participer au protocole de routage pour avoir accès à l'information de routage. C'est l'authentification qui gère les services de contrôle d'accès dans le réseau.
- L'intégrité : On satisfait l'intégrité dans un réseau quand tous les nœuds du réseau suivent correctement les démarches du protocole de routage. En suivant toutes les règles du protocole, on garantit que toutes les informations qui circulent dans le réseau sont correctes et utiles.
- La confidentialité : Afin de protéger l'information échangée entre les nœuds du réseau, généralement à l'aide d'autres nœuds intermédiaires, un chiffrement de données est nécessaire.
- La disponibilité : Chaque nœud a une liste de tâches qu'il est autorisé de faire, mais les nœuds ne sont pas tout le temps capable d'accomplir leurs tâches. La disponibilité d'un nœud réside dans le fait qu'il est autorisé et capable de faire une certaine tâche quand demandé.

### 2.3.1 Connectivité intranet

Les problèmes que soulève un réseau Ad-Hoc se résument généralement en deux parties :

- a) Pour maintenir une connexion à internet ou à un réseau cellulaire, le changement de topologie récurrent est la solution la plus proposée. Cependant, un réseau avec les caractéristiques dynamiques d'Ad-Hoc ne peut adapter cette solution à cause de son manque de stabilité.

- b) L'autre préoccupation est le choix du routage parfait selon le besoin. Parmi les solutions suggérées qui ont fonctionné avec succès pour les réseaux traditionnels est la création d'un système de bases de données qui s'occupe du routage à travers soit les nœuds du réseau ou ceux du routage. Cette solution ne garantit pas les types de communications broadcast et multicast, et ne respecte pas le principe de décentralisation des réseaux Ad-Hoc [10].

Les algorithmes de routage Ad-Hoc proactifs et réactifs, que nous allons décrire dans la quatrième section de ce chapitre, sont les approches les plus étudiées dans de nombreuses recherches. Cela revient au fait qu'un nœud Ad-Hoc qui utilise deux technologies différentes peut communiquer avec les nœuds qui utilisent au moins l'une de ces technologies [10].

Un réseau Ad-Hoc est capable d'offrir un service de broadcast. Les protocoles de routage utilisent ce type de communication pour envoyer un paquet à tous les nœuds du réseau. En complément du support efficace du service broadcast, les réseaux Ad-Hoc peuvent aussi fournir des services multicast et anycast [10].

### **2.3.2 TCP/IP**

La couche protocolaire TCP/IP doit avoir un support dans un réseau Ad-Hoc parce que chaque nœud de ce réseau doit se comporter comme s'il était dans un environnement IP classique. Parmi les services qu'IP peut offrir, il y a le service de broadcast ainsi que des règles de routage et d'adressage [10].

### **2.3.3 Connectivité globale**

Offrir une connectivité avec Internet à un réseau Ad-Hoc n'est peut-être pas essentiel, mais peut apporter un grand nombre d'avantages dans certaines situations. Connecter un nœud qui appartient à un réseau Ad-Hoc avec Internet pourrait lui donner la capacité de se comporter comme s'il était dans un environnement filaire classique, ainsi ses sessions de travail ne seront plus interrompues. La connectivité avec Internet va au-delà des services de mobilité et de routage, elle garantit la

continuité de service. En complément de tout cela, les réseaux Ad-Hoc consomment moins de bande passante et peuvent passer à l'échelle en supportant une montée en charge importante de nœuds mobiles [10].

## 2.4 Les protocoles de routage Ad-Hoc

Il existe trois principales catégories de protocoles pour un routage Ad-Hoc : Protocole topologique proactif, protocole topologique réactif et protocole hybride [31].

### 2.4.1 Protocoles topologiques proactifs

Chaque nœud du réseau enregistre un graphe complet du réseau. Par conséquent, chaque nœud finit par maintenir périodiquement toutes les routes du réseau [8]. Les protocoles topologiques proactifs ont en commun la maintenance de leurs réseaux et la possession d'une table de routage de tout le réseau, mais ils peuvent différer au niveau des services qu'ils offrent quand ils sont utilisés dans un réseau Ad-Hoc mobile. Nous allons présenter par la suite une série de protocoles proactifs pour pouvoir juger et choisir celui qui convient le plus à notre situation [2].

#### 2.4.1.1 *Optimized Link State Routing (OLSR)*

Ce protocole de communication est dédié aux réseaux mobiles Ad-Hoc. Le protocole utilise des nœuds qu'il sélectionne pour la diffusion générale des messages destinés à tout le réseau. Ce concept de clés des relais multipoints ou MPR évite au réseau une surcharge occasionnelle due à la diffusion classique par inondation où chaque nœud du réseau qui reçoit le message pour la première fois le retransmet à un autre nœud. OLSR diminue le nombre de messages de contrôle diffusés dans le réseau et ne mentionne que les états de liens importants du réseau. Seuls les liens entre un nœud MPR et les nœuds qu'ils l'ont choisi, comme clé de relais, sont mentionnés dans un message de contrôle. OLSR utilise cette information pour calculer les routes du réseau afin de sortir avec tous les chemins optimaux de la topologie du réseau qu'il gère [29] [15].

#### ***2.4.1.2 Topology Broadcast based on also Reverse Path Forwarding (TBRPF)***

Tout comme OLSR, TBRPF est un protocole à états de liens. Le TBRPF met en charge chaque nœud du réseau d'un arbre qui se compose de ses nœuds voisins, le nœud élu est la racine de cet arbre et en complément fourni les chemins les plus optimaux à tous les nœuds du réseau. Le routage pour ce protocole vient en deuxième place, parce que chaque nœud a besoin de découvrir ses voisins en premier. En envoyant au voisinage direct un paquet « hello » qui contient aussi la liste de ses voisins, un nœud est capable de connaître rapidement la topologie complète du réseau à deux sauts. Après avoir découvert le voisinage, le TBRPF passe au routage en échangeant les arbres de routage avec les nœuds voisins, ainsi l'information se diffuse graduellement dans l'ensemble du réseau. Cependant, chaque nœud a deux niveaux, et ne peut partager la liste des liens vers tous ses voisins directs qu'avec le premier niveau, le deuxième niveau reçoit habituellement une liste de liens uniques vers chaque voisin à deux sauts. Dans le cas où l'on veut économiser la bande passante, TBRPF peut regrouper les paquets « hello » avec les informations sur les liens [7] [15].

#### ***2.4.1.3 Fisheye state routing (FSR)***

FSR est un protocole de routage hiérarchique. Ce protocole utilise la technique « fisheye », proposée par Kleinrock et Stevens [23], qui réduit la taille des informations dont nous avons besoin pour présenter graphiquement des données. La technique de l'œil du poisson, « the eye of the fish », se concentre sur la capture de tous les détails qui entourent le point focal. L'attention au détail diminue au fur et à mesure que la distance depuis le point focal augmente. La technique de routage du FSR repose sur l'attention portée aux détails et à la qualité de l'optimisation des routes du voisinage direct d'un nœud choisi. En d'autres termes, chaque nœud possède des informations détaillées sur ses nœuds voisins et des informations plus vagues sur les nœuds distants du réseau [9].

Le routage de FSR partage des similarités avec celui à états de liens. Les deux protocoles permettent aux nœuds du réseau de maintenir des tables de routage du réseau complet. Le routage LS, ou à états de liens, inonde le réseau avec des paquets d'informations sur le réseau à chaque

fois que sa topologie change, tandis que FSR utilise une technique où chaque nœud du réseau maintient une table d'états de liens qui est basée sur les informations mises à jour et envoyées par les nœuds voisins pour ensuite l'échanger avec son voisinage [9] [15].

#### **2.4.1.4 *Destination –also Sequenced a Distance Vector (DSDV)***

Le protocole DSDV est l'adaptation du protocole conventionnel RIP (Routing Information Protocol) pour les réseaux Ad-Hoc mobiles. Il ajoute un nouvel attribut, à savoir un numéro de séquence, à chaque entrée de table de routage du protocole RIP. L'utilité de ce numéro de séquence est de permettre aux nœuds de distinguer les anciens paquets des nouveaux afin d'éviter la formation d'une boucle de routage. DSDV permet à chaque nœud du réseau mobile Ad-Hoc de maintenir une table de routage qui liste tous les nœuds de destination disponible, la distance normale est à deux sauts pour chaque nœud de destination et un numéro de séquence qui est généré par le nœud de destination. Le maintien des tables de routage permet aux nœuds de transmettre des paquets d'informations qui ont été mises à jour afin de suivre le changement dynamique de la topologie du réseau Ad-Hoc. À chaque changement de topologie, chaque nœud transmet la nouvelle table de routage en utilisant la technique broadcast (un vers tous) ou multicast (un vers plusieurs). Tout comme le RIP, le protocole DSDV permet à chaque nœud d'envoyer la nouvelle table de routage à ses voisins directs, mais avec un petit ajout. Les nœuds du DSDV incrémentent le numéro de saut par 1 à chaque réception du nouveau paquet. Après la réception des nouvelles informations sur la topologie du réseau, chaque nœud garde le nouveau paquet en attendant les nouveaux chemins optimaux du réseau. Si jamais un nœud reçoit plusieurs chemins optimaux vers une même destination, il va choisir ceux envoyés dans les paquets les plus récents parce que ces paquets sont considérés comme des mises à jour de leurs précédents [11] [15].

### **2.4.2 Protocoles topologiques réactifs**

Les nœuds ne possèdent pas de graphes qui décrit les chemins du réseau, ainsi à chaque fois il faut diffuser une requête sur l'ensemble du réseau pour pouvoir trouver le chemin vers le destinataire [8]. Les protocoles réactifs ne maintiennent pas des tables de routage et ne possèdent pas un graphe qui décrit toute la topologie du réseau. Par conséquent, ils calculent le chemin vers chaque

destination à chaque fois avant d'effectuer le routage[4]. Cette technique évite aux nœuds de mettre à jour leurs chemins optimaux, mais le temps passé à découvrir les nouvelles routes à chaque fois retarde le démarrage du routage [30].

#### **2.4.2.1 AODV (*Ad-Hoc On-Demand Distance Vector*)**

Ce protocole multi-sauts s'adapte à la nature dynamique des réseaux mobiles Ad-Hoc ce qui évite la création des boucles de routage. Afin de transférer des données vers une destination, le nœud émetteur envoie en premier une requête RREQ (Route Request) à tout son voisinage en broadcast (un vers tous). Les nœuds qui ont reçu ce RREQ le transmettent jusqu'au destinataire. Le nœud destinataire valide le chemin en envoyant un message réponse RREP (Route Response) sur le même chemin, mais à l'envers en destination du nœud émetteur de la requête. Le protocole AODV compte surtout sur l'attribut de numéro de séquence pour savoir si les informations dans les tables de routage des nœuds sont à jour. Dans le cas d'une route défectueuse, AODV envoie un message d'erreur RERR (Route Error) à tous les nœuds qui utilisent ce chemin pour les prévenir de l'indisponibilité de cette route, ensuite une réparation de cette route est effectuée soit par le nœud qui a découvert la défectuosité ou par le nœud qui se trouve à l'autre bord du chemin brisé [30] [15].

#### **2.4.2.2 DSR (*Dynamic Source Routing*)**

Le protocole DSR comme tous les protocoles réactifs a besoin d'entreprendre une découverte de routes à chaque transfert de données. Il se distingue du protocole AODV par un mécanisme de maintenance de route qui offre une réparation automatique aux routes défectueuses qui sont dues aux changements de la topologie du réseau. DSR donne au réseau la chance de s'organiser par lui-même lors de certains changements de topologie comme la défaillance ou la mobilité des nœuds. Pour envoyer un paquet de données vers une destination, le nœud émetteur vérifie sa table de routage en premier avant d'entreprendre une découverte de route au cas où il ne trouverait pas de chemins vers sa destination. Le protocole DSR ajoute une entrée dans les tables de routage (Identificateur, cible) afin de faciliter la découverte de route. En premier, le nœud source envoie une requête à son voisinage en broadcast (Un vers tous), chaque nœud qui reçoit cette requête doit

vérifier si son adresse correspond à l'adresse du nœud cible. Si ce n'est pas le cas, le nœud transmet le paquet à ses voisins. Cependant, quand le destinataire reçoit enfin la requête, il génère une réponse de route à l'émetteur. Pour découvrir des routes, DSR utilise des paquets de requêtes et des paquets de réponses. Tandis que pour s'adapter aux changements de la topologie et maintenir les routes existantes, DSR utilise des paquets d'erreurs et des paquets d'acquittements [30] [15].

#### **2.4.2.3 *TORA (Temporally Ordered Routing Algorithm)***

Le protocole TORA est un protocole réactif adapté aux réseaux mobiles et dynamiques qui changent de topologie rapidement et d'une manière aléatoire. En cas de changement de topologie, TORA s'adapte en envoyant des messages de contrôle à un débit différent de la dynamique du réseau afin de limiter le nombre des messages propagés. TORA utilise trois types de paquets pour entretenir le réseau :

- Les paquets de requêtes QRY (QUERY) qui sont utilisés pour découvrir et créer des routes.
- Les paquets de mise à jour UDP (UPDATE) qui sont utilisés aussi pour créer de nouvelles routes, mais surtout pour maintenir les routes qui existent déjà.
- Les paquets de suppression CLR (CLEAR) qui ont pour mission de supprimer les routes défectueuses.

Dans le protocole TORA, chaque nœud qui envoie une requête QRY pour créer une route pousse les autres nœuds du réseau à activer leurs indicateurs de requête de route qui est en état inactif en temps normal. Les nœuds dans TORA maintiennent aussi les dates de création des dernières mises à jour et aussi les dates où leurs voisins deviennent actifs [30] [15].

### **2.4.3 Protocoles hybrides**

Chaque nœud connaît sa propre position géographique et celle du destinataire. Les nœuds utilisent ces deux informations pour créer un chemin à chaque transfert de données [8]. À chaque transfert de données, les protocoles hybrides vont commencer par adapter la technique des protocoles proactifs en calculant toutes les routes possibles pour ensuite changer à la méthode réactive pour

un meilleur routage [22]. On cite comme exemples de protocoles hybrides pour les réseaux Ad-Hoc : ZRP (Zone Routing Protocol), CEDAR (Core-Extraction Distributed Ad-Hoc Routing Algorithm) et WARP (Wireless Ad-Hoc Routing Protocol), etc.

## 2.5 Conclusion

Nous nous sommes concentrés dans ce deuxième chapitre sur les différents systèmes de pilotage des robots sarclieurs de champs, l'architecture des réseaux Ad-Hoc ainsi que les différents protocoles de routage. Dans le chapitre suivant, nous allons présenter et comparer les différents simulateurs pour choisir celui qui est le plus compatible avec notre besoin.

## CHAPITRE 3 : Environnements de simulations

Lors du développement de notre réseau de robots sarclieurs de champs, nous avons testé plusieurs simulateurs et protocoles dans le but de créer une simulation assez proche de la réalité qui nous permettra d'observer le comportement et la performance de notre réseau. Dans ce troisième chapitre, nous allons voir et analyser les environnements de simulations que nous avons expérimentés afin de choisir celui qui convient le plus à notre situation.

### 3.1 SUMO (“Simulation of Urban Mobility”)

SUMO (Simulation of Urban Mobility) est un simulateur open source à temps discret de trafic routier. SUMO aide à créer la carte du réseau à partir d'une vraie carte. En utilisant la plateforme OpenStreetMap (OSM) qui est une base de données géographiques, SUMO est capable de copier n'importe quelle carte de route en exportant ses données en format OSM. Dans les figures 3.1 et 3.2 ci-dessous, il est possible d'observer une carte de Trois-Rivières avant et après l'extraction des routes par SUMO. SUMO conserve seulement les routes, et l'utilisateur peut ajouter autant de composants qu'il veut, comme : des véhicules, des feux de circulation, des piétons, etc. Grâce à son grand ensemble d'outils pour la création de scénarios, il permet de simuler le comportement des composants d'un réseau qui change constamment. Dans notre cas, la carte de réseau générée conserve juste les routes et exclut tous les champs qui se trouvent à côté, ce qui nous laisse avec la seule solution de dessiner nous-mêmes le plan du réseau que nous voulons simuler. Le résultat est graphiquement limité et ne vaut pas les efforts déployés. Tandis que SUMO est un bon simulateur pour les réseaux VANETS, nous avons conclu qu'il n'est pas le meilleur choix pour notre réseau mobile de robots.



Figure 3.1 : La carte de Trois-Rivières avant l'extraction des routes par SUMO



Figure 3.2 : La carte de Trois-Rivières après l'extraction des routes par SUMO

### 3.2 Objective Modular Network Test-bed in C++ (OMNET++)

OMNET ++ est un outil de simulation modulaire open source qui est basé sur des composants. Il est souvent utilisé pour simuler les systèmes distribués comme les réseaux de communication. OMNET ++ se compose de plusieurs modules qui sont placés hiérarchiquement selon leurs complexités. Ces modules se divisent en deux types : simples et composés. Les modules simples sont supportés par des fichiers C++, en forme de librairies, qui contiennent des algorithmes dictant aux modules comment agir pour satisfaire les conditions de la simulation requise. Les modules composés représentent un ou des ensembles de modules simples qui sont supportés, mais de façon indirecte, par des fichiers C++ et leurs directives [3]. OMNET ++ est utilisé principalement pour simuler les réseaux véhiculaires, nous avons essayé de l'adapter à nos besoins. La figure 3.3 représente un exemple de simulation sur OMNET++. Afin d'utiliser les modules de OMNET ++, il faut passer en premier par SUMO. Avec la carte du réseau générée par SUMO, les modules d'OMNET ++ et en implémentant un Framework VEINS qui aide à exécuter les simulations des réseaux véhiculaires, nous nous sommes rendu compte que ces outils ne sont pas les plus adaptés à nos besoins. Le simulateur SUMO est très limité du côté graphique comme nous l'avons expliqué ci-dessus, et les modules d'OMNET ++ respectent leurs fichiers C++ à la lettre ce qui rend difficile de sortir avec une simulation fonctionnelle adaptée avec un protocole qui respecte nos besoins dans une représentation graphique adéquate. Pour les autres réseaux sans fil comme les réseaux Ad-Hoc, OMNET ++ manque de protocoles comparés aux autres simulateurs [13]. Le combo SUMO/OMNET ++/VEINS est peut-être le meilleur choix pour les simulations des réseaux véhiculaires et le trafic routier, mais dans un autre scénario, qui est légèrement différent, ces mêmes simulateurs ne fonctionnent pas au maximum de leurs capacités.

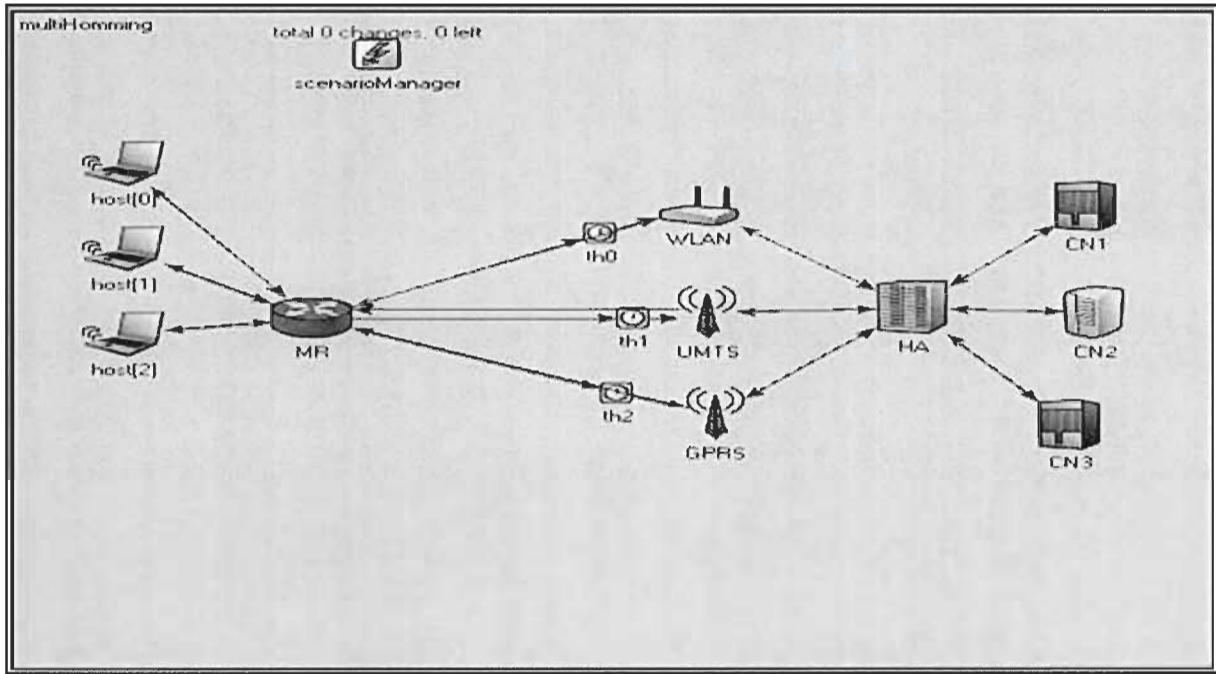


Figure 3.3 : Exemple de simulation sur OMNET ++ [21]

### 3.3 NS2 (Network Simulator 2)

NS est l'un des simulateurs de réseaux de communication les plus populaires et les plus utilisés pour la recherche dans les réseaux mobiles en raison de sa gratuité, de la disponibilité de sa documentation en ligne et de son adaptation aux besoins des simulations grâce à son modèle libre qui permet de forger n'importe quel modèle désiré [13]. Alors que les simulations de NS2 sont implémentées avec le langage C++, le simulateur est basé sur un autre langage qui est le langage de script Tcl/Tk. NS2 est un simulateur difficile à manipuler, mais pour les utilisateurs avancés, il donne la possibilité d'écrire leurs propres fichiers C++ ce qui leur offre plus de contrôle sur les comportements des composants de leurs simulations. NS2 ne vient pas avec une interface graphique, les simulations sont alors créées via les lignes de commandes. Cependant, il est possible de visualiser les résultats d'une simulation sur NAM (Network Animator) qui est l'outil le plus proche d'une interface graphique sur NS2 [27] [12]. Les nœuds dans les réseaux simulés sur NS2 sont équipés de capteurs qui leur permettent d'éviter toute collision avec d'autres nœuds ou avec des objets qui se trouvent sur leurs trajectoires. Dans la figure 3.4 ci-dessous, nous pouvons voir un exemple de simulation sur NS2.

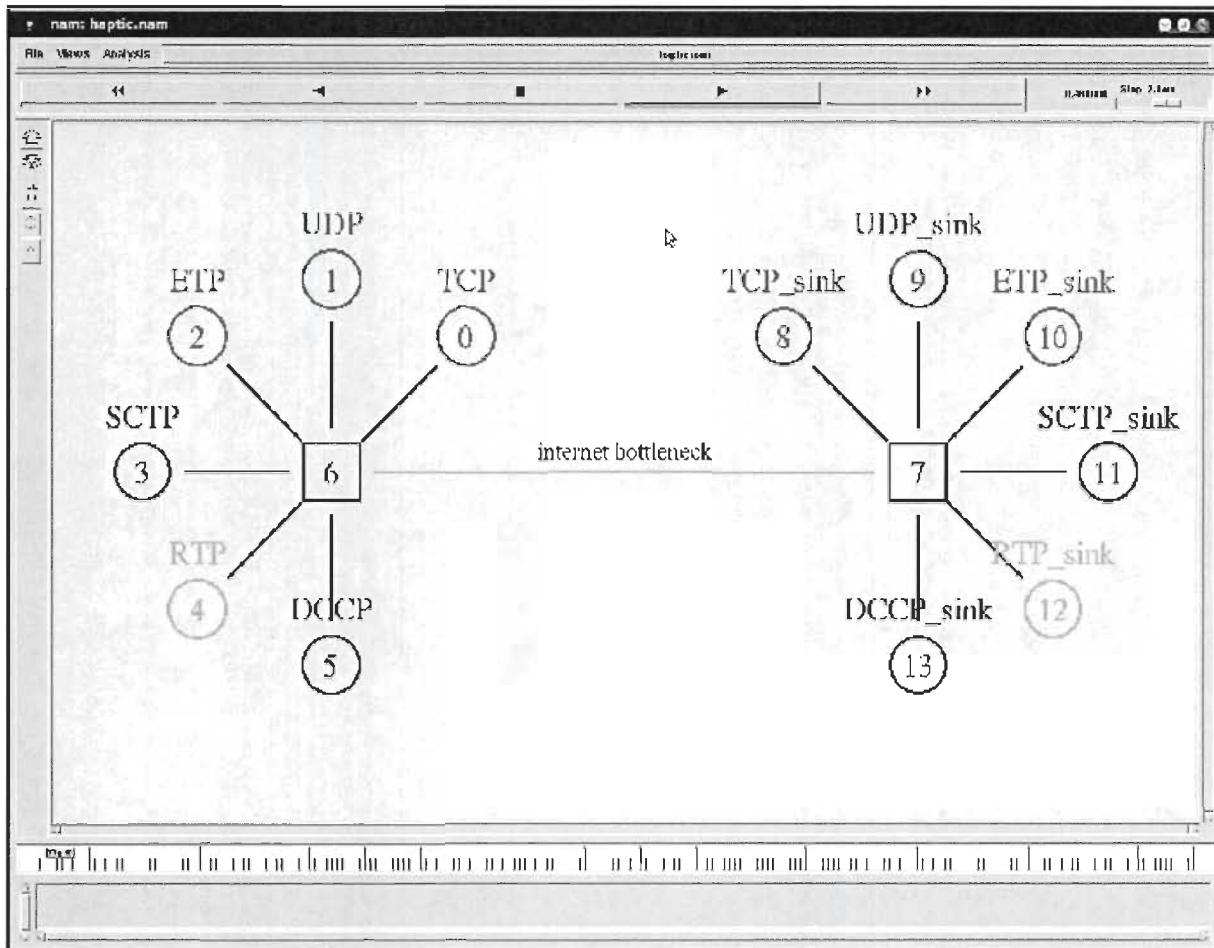


Figure 3.4 : Exemple de simulation sur NS2 [19]

### 3.4 NS3 (Network Simulator 3)

Tandis que NS2 est partiellement écrit en C++, la version NS3 est entièrement écrite en C++ avec la possibilité d'implémenter les simulations en Python aussi. Certes, NS3 est développé après NS2, mais cela ne veut pas dire qu'il est meilleur. Les deux simulateurs répondent à des besoins différents et dans certaines situations comme les simulations des réseaux de capteurs sans fil, il est mieux d'opter pour NS2. Cependant, NS3 est facile à manipuler et est plus adapté pour la conception de protocoles ou l'animation de réseaux. Il est inutile de comparer NS2 et NS3, il faut essayer les deux pour pouvoir choisir celui qui répond le mieux à nos besoins [5]. La figure 3.5 représente une simulation sur NS3.

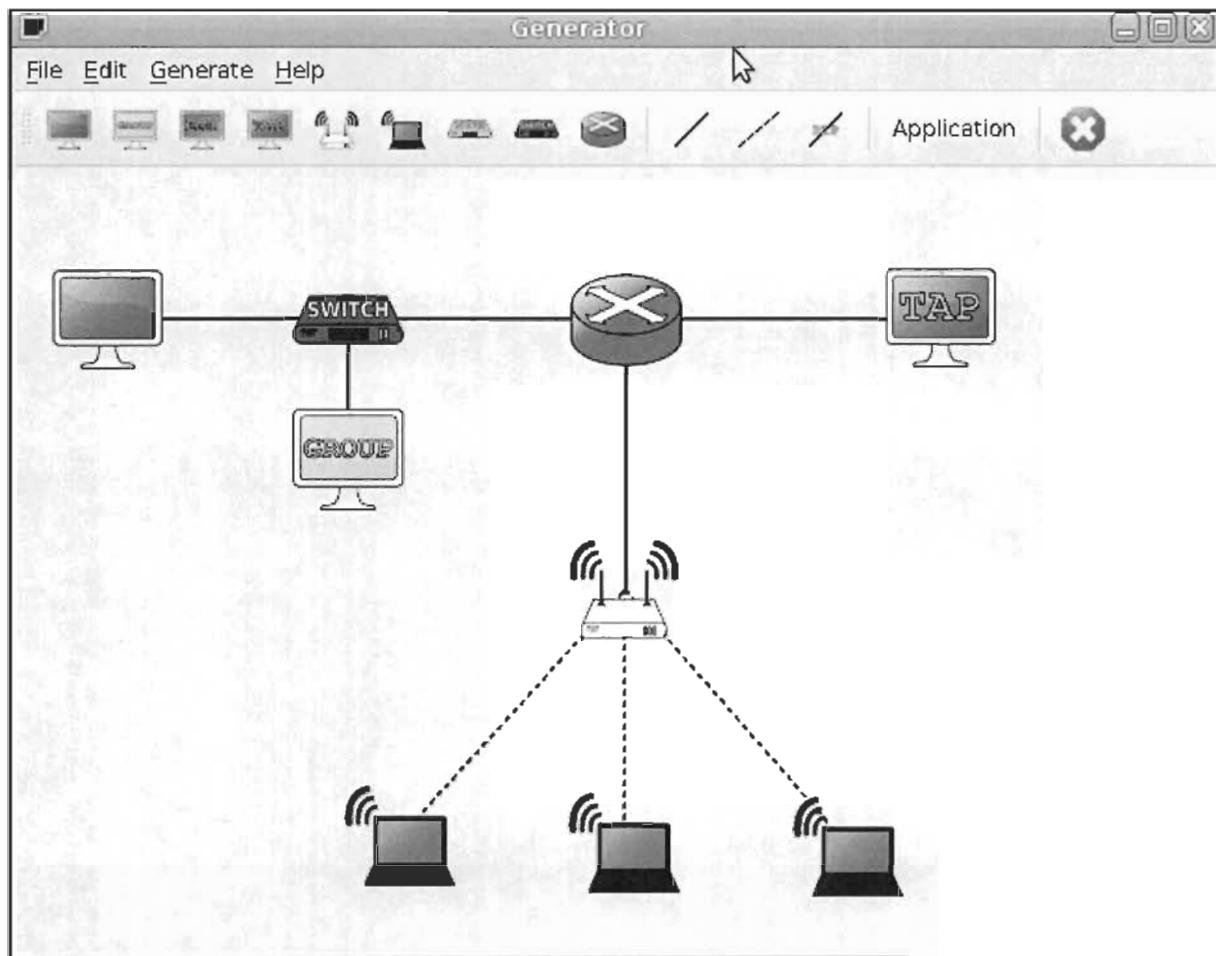


Figure 3.5 : Exemple de simulation sur NS3 [20]

### 3.5 Conclusion

Après avoir testé les simulateurs présentés dans ce troisième chapitre, nous avons réussi à choisir le simulateur ainsi que le protocole de routage qui convient le plus à nos besoins. Dans le chapitre 4, nous allons expliquer les choix des outils avec lesquels nous avons travaillé et nous allons voir et discuter les différents résultats obtenus à travers nos simulations.

## Chapitre 4 : SIMULATIONS ET RÉSULTATS

Ce chapitre décrit les simulations que nous avons réussi à exécuter sur le simulateur NS2 avec le protocole AODV. Nous allons voir en premier les paramètres principaux des simulations, ensuite nous allons présenter et discuter les résultats de ces simulations. L'objectif principal de ce chapitre est de démontrer que nous avons réussi à créer et entretenir une communication entre tous les composants du réseau.

### 4.1 Environnements de simulations

La raison pour laquelle nous avons choisi NS2 est la suivante. Nous avons essayé les simulateurs OMNET ++, NS2 et NS3 pour implémenter plusieurs scénarios et essayer de différents protocoles afin d'étudier le comportement de notre réseau. NS2 s'est montré le simulateur le plus performant du groupe grâce à sa flexibilité à définir les protocoles. En ce qui concerne les protocoles, AODV avait montré de bons résultats contrairement à DSDV et DSR qui étaient plus difficiles à définir et/ou configurer. Pour les simulations, nous avons considéré un réseau Ad-Hoc sans fil d'environ 25 nœuds (robots) mobiles placés dans un terrain de 2km x 2km, avec un RSU qui est placé au milieu du terrain. La portée de communication des nœuds (robots) mobiles est de 800m.

Les paramètres de la première simulation sont présentés dans le tableau 4.1 :

Paramètre	Valeur
Superficie	2km x 2km
Durée de la simulation	3600s
Nombres de nœuds mobiles (robots)	2
Vitesse des nœuds	5m/s
Trafic	Aléatoire
Temps de simulation	1 heure
Taille de paquet	1024 bytes
Portée de communication des nœuds	800 m
RSU	0

Tableau 4.1 : Paramètres de la simulation 1

#### 4.2 Schématisation de l'environnement de simulation

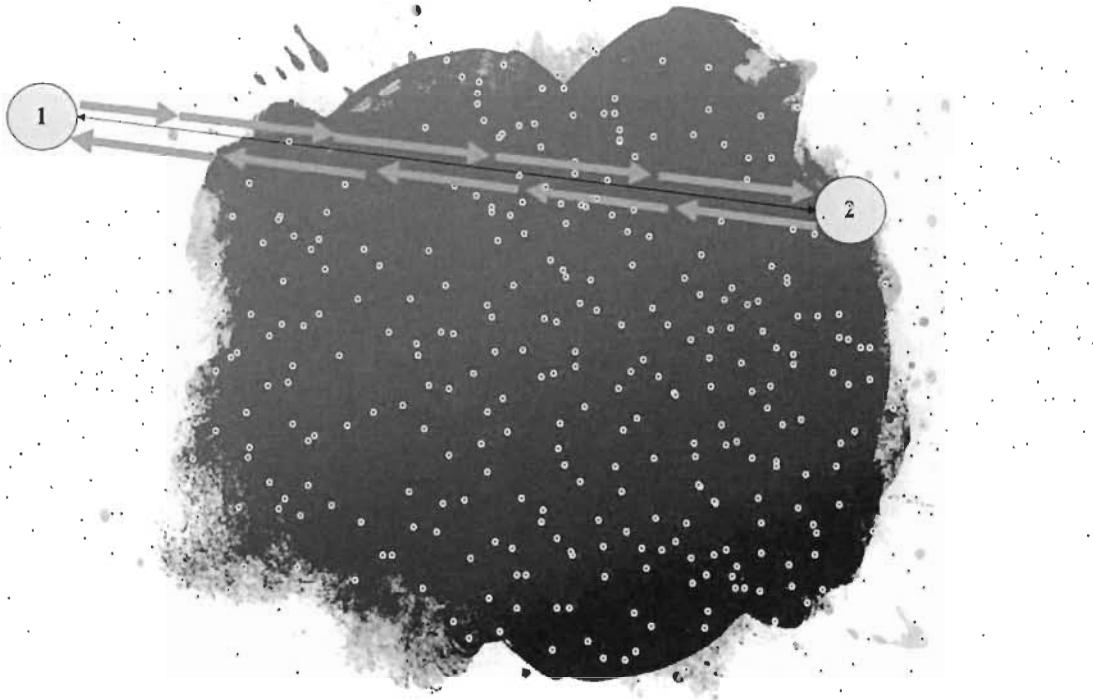


Figure 4.1 : Simulation du flux de la communication entre deux nœuds (Sans RSU)

La figure 4.1 est un schéma qui représente le flux de communication entre les nœuds du réseau. Les paramètres de cette simulation sont mentionnés dans le tableau 4.1 ci-dessus. La tâche bleue symbolise la portée de la communication entre les robots qui est de 800m. Les flèches en noir représentent la possibilité de communiquer entre les nœuds qui se trouvent dans les deux extrémités de la flèche. Deux nœuds sont capables de communiquer quand ils ne dépassent pas la portée de 800m. Les flèches en vert représentent le partage de données entre les nœuds, et les points rouges symbolisent les mauvaises herbes dans le champ. Les robots bougent tout au long du champ vers les mauvaises herbes (points rouges) afin de les sarcler. Dans la simulation, dès qu'un robot touche un point rouge, ce dernier disparaît. Ceci est l'équivalent du sarclage d'une mauvaise herbe par un robot en temps réel. Vu que nous travaillons avec un réseau Ad-Hoc mobile, la mobilité des nœuds peut mener à briser des liens de connexion quand deux nœuds s'éloignent

jusqu'au point de sortir du périmètre de 800m. Ceci entraîne à la perte de certains paquets qui ont été envoyés avant la rupture du chemin.

Les paramètres de la deuxième simulation sont présentés dans le tableau 4.2 :

Paramètre	Valeur
Superficie	2km x 2km
Durée de la simulation	3600s
Nombres de nœuds mobiles (robots)	2
Vitesse des nœuds	5m/s
Trafic	Aléatoire
Temps de simulation	1 heure
Taille de paquet	1024 bytes
Portée de communication des nœuds	800 m
RSU	1

Tableau 4.2 : Paramètres de la simulation 2

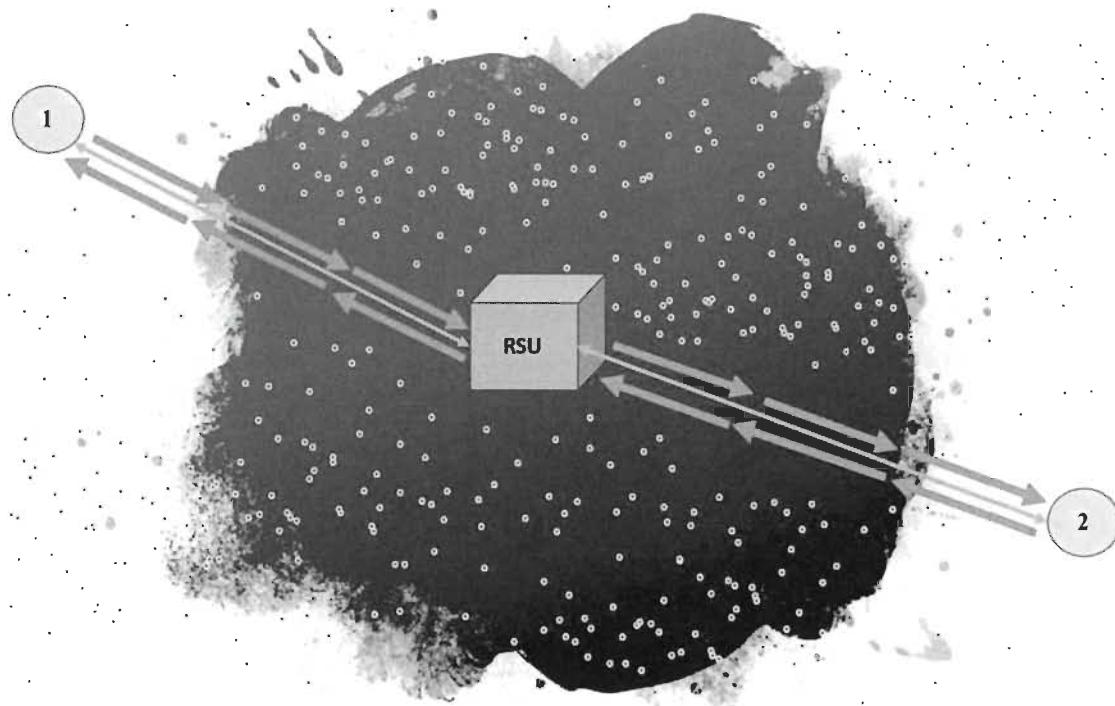


Figure 4.2 : Simulation du flux de la communication entre deux nœuds (Avec RSU)

La figure 4.2 montre que le mouvement des robots garantit la rupture du lien de la communication. Les deux robots avancent aléatoirement vers les mauvaises herbes, et dans certains cas ils finissent par sortir de la portée de communication de 800m. Par conséquent, la perte des paquets envoyés avant l'interruption de la connexion est inévitable. L'ajout du RSU est crucial afin d'assurer une connectivité continue en tout temps entre les nœuds du réseau.

Les paramètres de la troisième simulation sont présentés dans le tableau 4.3 :

Paramètre	Valeur
Superficie	2km x 2km
Durée de la simulation	3600s
Nombres de nœuds mobiles (robots)	4
Vitesse des nœuds	5m/s
Trafic	Aléatoire
Temps de simulation	1 heure
Taille de paquet	1024 bytes
Portée de communication des nœuds	800 m
RSU	0

Tableau 4.3 : Paramètres de la simulation 3

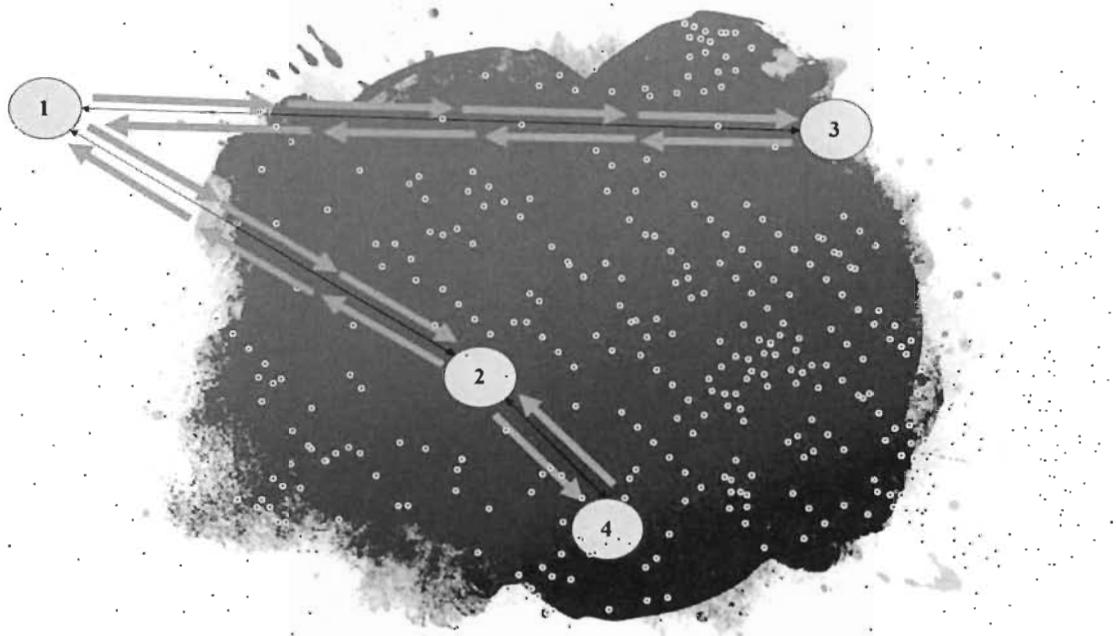


Figure 4.3 : Simulation du flux de la communication entre quatre nœuds (Sans RSU)

Les paramètres de la quatrième simulation sont présentés dans le tableau 4.4 :

Paramètre	Valeur
Superficie	2km x 2km
Durée de la simulation	3600s
Nombres de nœuds mobiles (robots)	4
Vitesse des nœuds	5m/s
Trafic	Aléatoire
Temps de simulation	1 heure
Taille de paquet	1024 bytes
Portée de communication des nœuds	800 m
RSU	1

Tableau 4.4 : Paramètres de la simulation 4

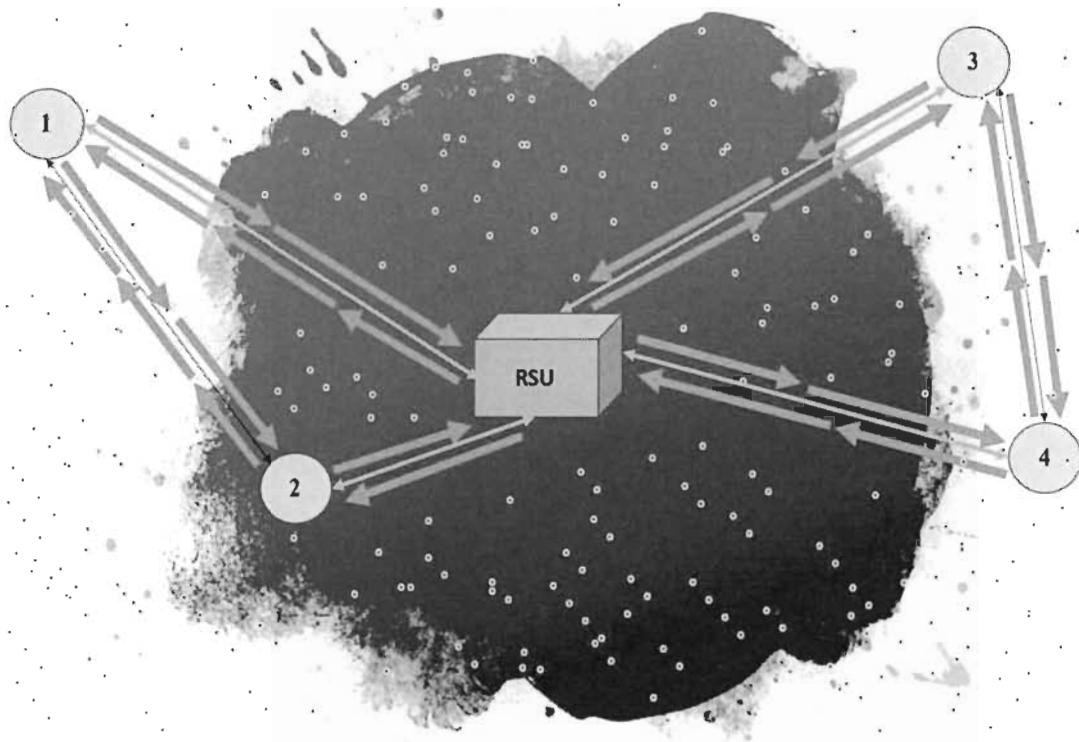


Figure 4.4 : Simulation du flux de la communication entre quatre nœuds (Avec RSU)

Les paramètres de la cinquième simulation sont présentés dans le tableau 4.5 :

Paramètre	Valeur
Superficie	2km x 2km
Durée de la simulation	3600s
Nombres de nœuds mobiles (robots)	8
Vitesse des nœuds	5m/s
Trafic	Aléatoire
Temps de simulation	1 heure
Taille de paquet	1024 bytes
Portée de communication des nœuds	800 m
RSU	0

Tableau 4.5 : Paramètres de la simulation 5

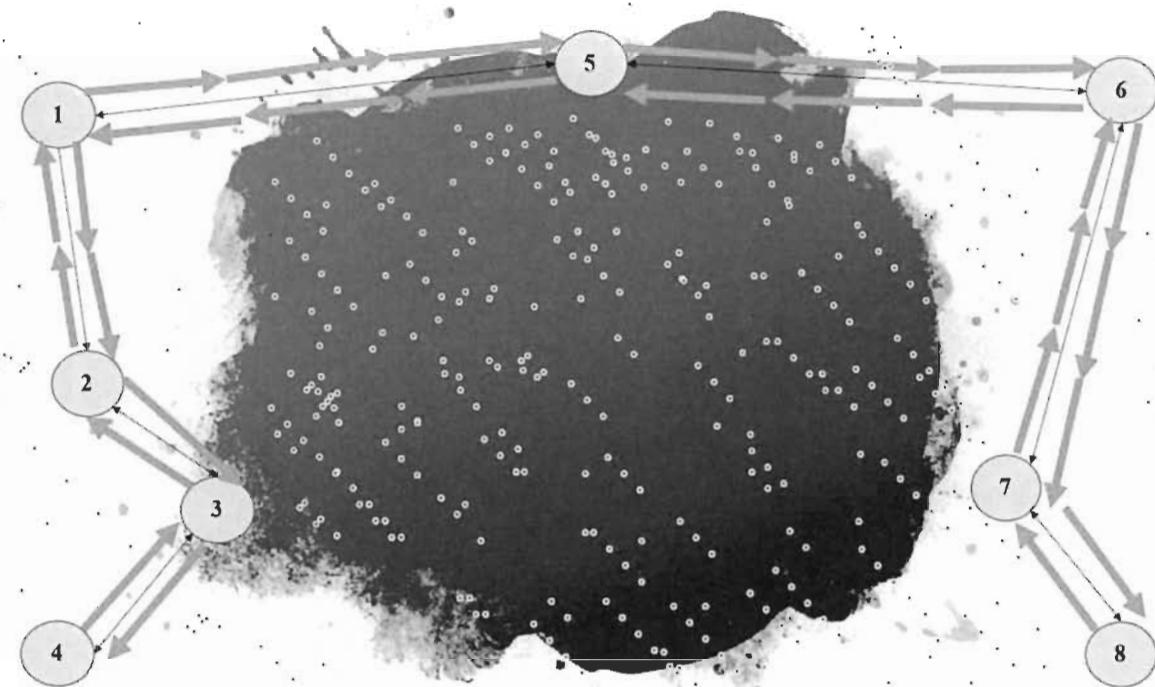


Figure 4.5 : Simulation du flux de la communication entre huit nœuds (Sans RSU)

Les paramètres de la sixième simulation sont présentés dans le tableau 4.6 :

Paramètre	Valeur
Superficie	2km x 2km
Durée de la simulation	3600s
Nombres de nœuds mobiles (robots)	8
Vitesse des nœuds	5m/s
Trafic	Aléatoire
Temps de simulation	1 heure
Taille de paquet	1024 bytes
Portée de communication des nœuds	800 m
RSU	1

Tableau 4.6 : Paramètres de la simulation 6

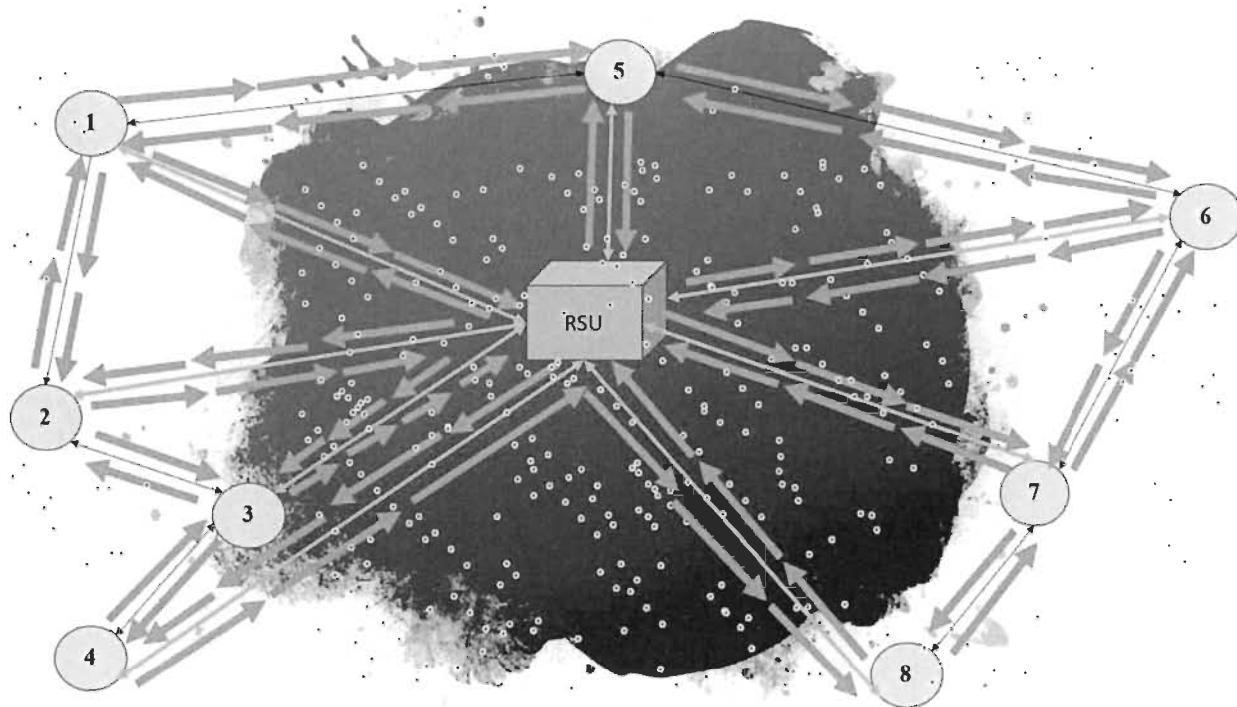


Figure 4.6 : Simulation du flux de la communication entre huit nœuds (Avec RSU)

Les paramètres de la septième simulation sont présentés dans le tableau 4.7 :

Paramètre	Valeur
Superficie	2km x 2km
Durée de la simulation	3600s
Nombres de nœuds mobiles (robots)	16
Vitesse des nœuds	5m/s
Trafic	Aléatoire
Temps de simulation	1 heure
Taille de paquet	1024 bytes
Portée de communication des nœuds	800 m
RSU	0

Tableau 4.7 : Paramètres de la simulation 7

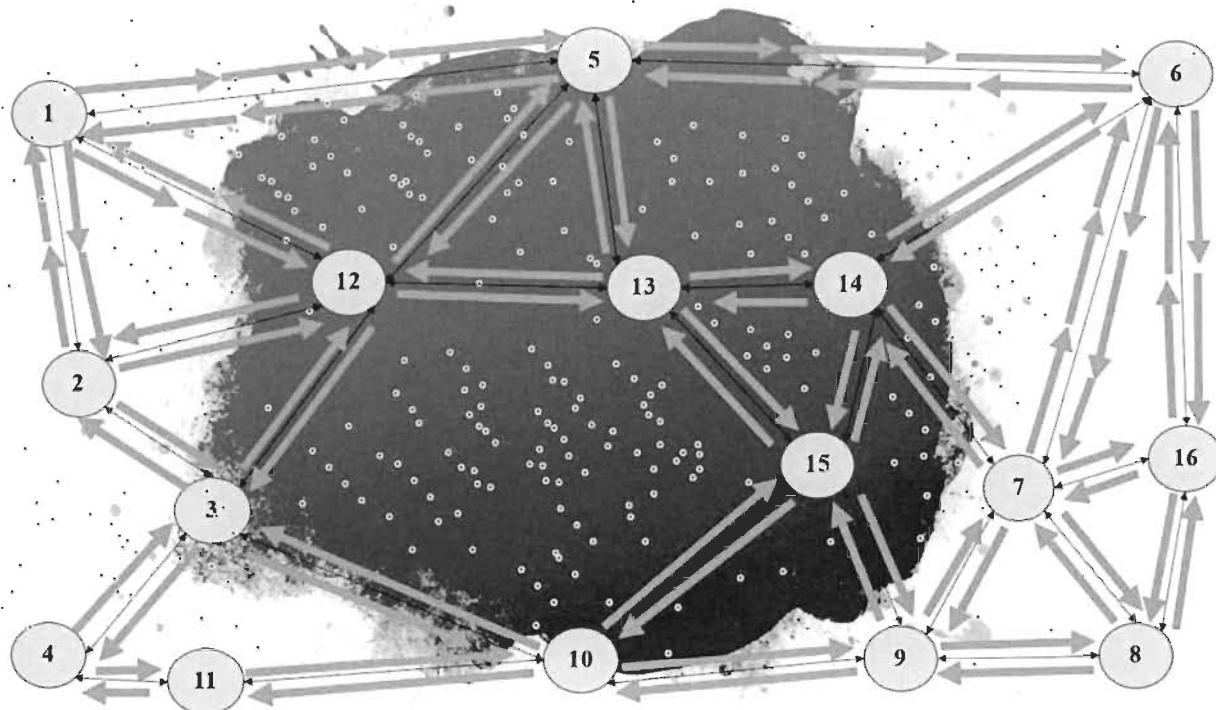


Figure 4.7 : Simulation du flux de la communication entre seize nœuds (Sans RSU)

Les paramètres de la huitième simulation sont présentés dans le tableau 4.8 :

Paramètre	Valeur
Superficie	2km x 2km
Durée de la simulation	3600s
Nombres de nœuds mobiles (robots)	16
Vitesse des nœuds	5m/s
Traffic	Aléatoire
Temps de simulation	1 heure
Taille de paquet	1024 bytes
Portée de communication des nœuds	800 m
RSU	1

Tableau 4.8 : Paramètres de la simulation 8

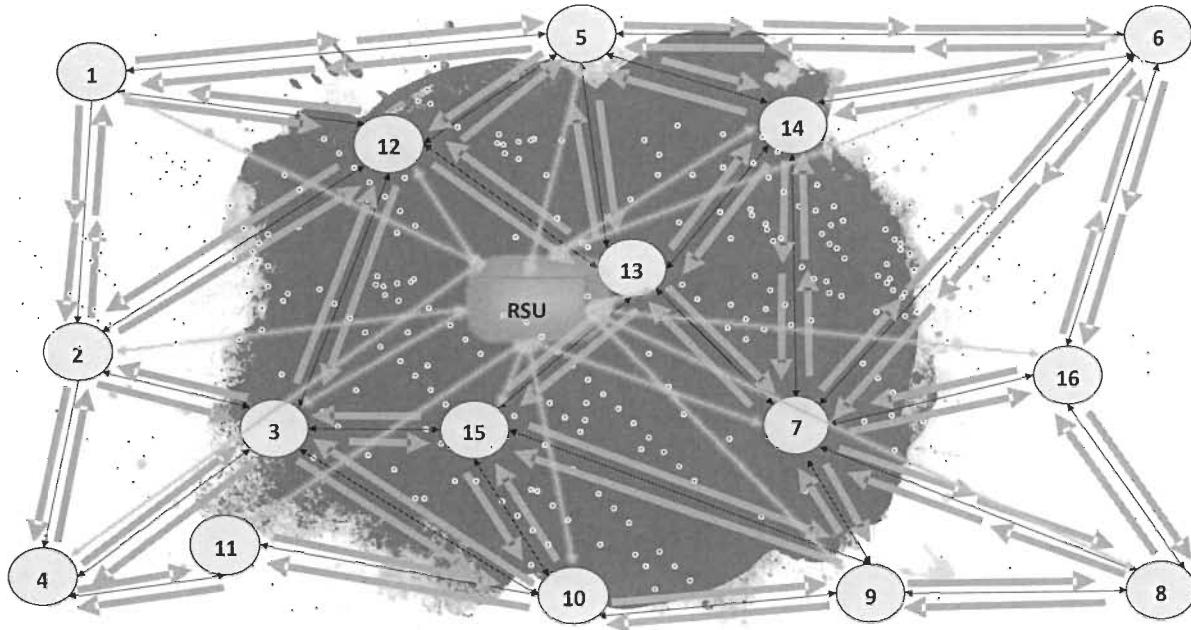


Figure 4.8 : Simulation du flux de la communication entre seize nœuds (Avec RSU)

Nous avons commencé par simuler deux robots afin de pouvoir analyser leurs comportements dans un environnement plus primitif. Dans la figure 4.1, les deux robots sont les composants principaux du réseau. En rajoutant deux autres robots à la simulation, comme le représente la figure 4.3, nous pouvons remarquer que cette petite expansion du réseau a affecté la communication sur deux niveaux. D'un côté, quand deux robots sont hors portée l'un de l'autre mais partagent le même nœud voisin, ce dernier veille à passer l'information d'un voisin à l'autre ce qui évite la perte des données. Mais d'un autre côté, la consommation de la bande passante augmente vu que le nombre des composants du réseau a augmenté.

Cependant, ajouter des robots au réseau ne vient pas entièrement résoudre le problème de l'interruption de la communication, car les quatre robots peuvent bouger simultanément dans le champ et finir par s'éloigner l'un de l'autre jusqu'à l'interruption de la connectivité. Dans cette situation aussi, le RSU se montre capital pour la continuité de la connexion (Figure 4.4).

Dans les figures 4.5 et 4.7, nous avons rajouté davantage des robots dans la simulation. Dans le cas de 8 robots, nous n'avons pas changé la surface du terrain (2km x 2km). Les robots étaient capables de transférer les paquets sans l'aide du RSU, mais nous avons constaté certaines pertes de données. L'ajout du RSU a garanti une communication parfaite sans perte de paquets. En agrandissant la surface du terrain à 5km x 5km, la perte de données est devenue plus sérieuse, et la présence du RSU est devenue obligatoire.

Nous avons obtenu les mêmes résultats avec 16 robots dans le champ. Bien évidemment, l'ajout des robots (nœuds) garantit l'expansion du réseau et l'ajout de plusieurs voisinages à chaque nœud. Ceci entraîne à l'augmentation des mises à jour, à l'augmentation des paquets transférés et à l'augmentation de la consommation de la bande passante. Dans ce cas, le RSU vient libérer notre protocole AODV de toute communication inutile. Au lieu de demander à tous les nœuds du réseau le chemin vers un nœud en particulier, le nœud émetteur peut passer par le RSU afin de transférer un paquet directement au nœud destinataire. Également, si on augmente la surface du terrain le RSU garantit une communication sans perte de données.

### 4.3 La configuration du protocole de routage AODV

Nous allons voir dans cette section la configuration du protocole avec lequel nous avons travaillé pour simuler nos scénarios. Contrairement aux protocoles proactifs qui maintiennent à jour leurs tables de routage en tout temps, AODV (Ad-Hoc On-demand Distance Vector routing) est un protocole de routage réactif qui maintient ses tables de routage sur demande.

Vu que la mise à jour des routes se fait sur demande, nous avons utilisé un Ipv4NetworkConfigurator afin de distribuer des adresses IP aux nœuds pour pouvoir les contacter en tout temps quand c'est nécessaire. Le protocole AODV considère les nœuds du réseau comme des routeurs. Chaque nœud joue alors le rôle d'un hôte et d'un client en même temps. Un réseau qui utilise AODV comme protocole crée des connexions au besoin et est habituellement calme. Quand un nœud désire communiquer avec un autre nœud au sein du réseau, il émet un message requête à son voisinage qui, à son tour, transmet le message jusqu'au nœud destinataire. Ce

message requête vise à découvrir le ou les chemins entre le nœud émetteur et le nœud destinataire. Dès la réception de la requête, le nœud destinataire émet une réponse qui est partagée avec son voisinage dans le but d'atteindre le nœud émetteur. Généralement, le nœud émetteur utilise le chemin avec le moins de sauts possible pour maintenir sa connexion avec le nœud destinataire concerné. Les routes qui ne sont pas utilisées seront recyclées afin que d'autres nœuds puissent les utiliser dans leurs communications. La figure 4.9 ci-dessous est une présentation graphique du fonctionnement du protocole AODV.

Dans le chapitre 2, nous avons présenté les différents messages utilisés par AODV, à savoir : RREQ ( message requête), RREP (message réponse) et RRER (message erreur). Ces messages apparaissent dans deux situations :

- Au début de la simulation à chaque demande de connexion entre deux nœuds.
- À chaque changement de topologie qui est dû à la mobilité des nœuds.

AODV est capable de détecter les chemins qui ne sont plus capables d'envoyer ou de recevoir des paquets. Quand le lien est brisé entre deux nœuds à cause de leur mobilité, AODV reconfigure les routes afin de transmettre tout paquet qui est envoyé avant la rupture du lien. La continuité de la connexion entre deux nœuds avec un lien brisé est garantie par d'autres nœuds qui viennent prendre le relais pour assurer la réception du paquet envoyé. La détection des chemins défectueux ainsi que leurs reconfigurations ne prennent pas beaucoup de temps généralement, et après cette brève interruption le trafic UDP continue normalement.

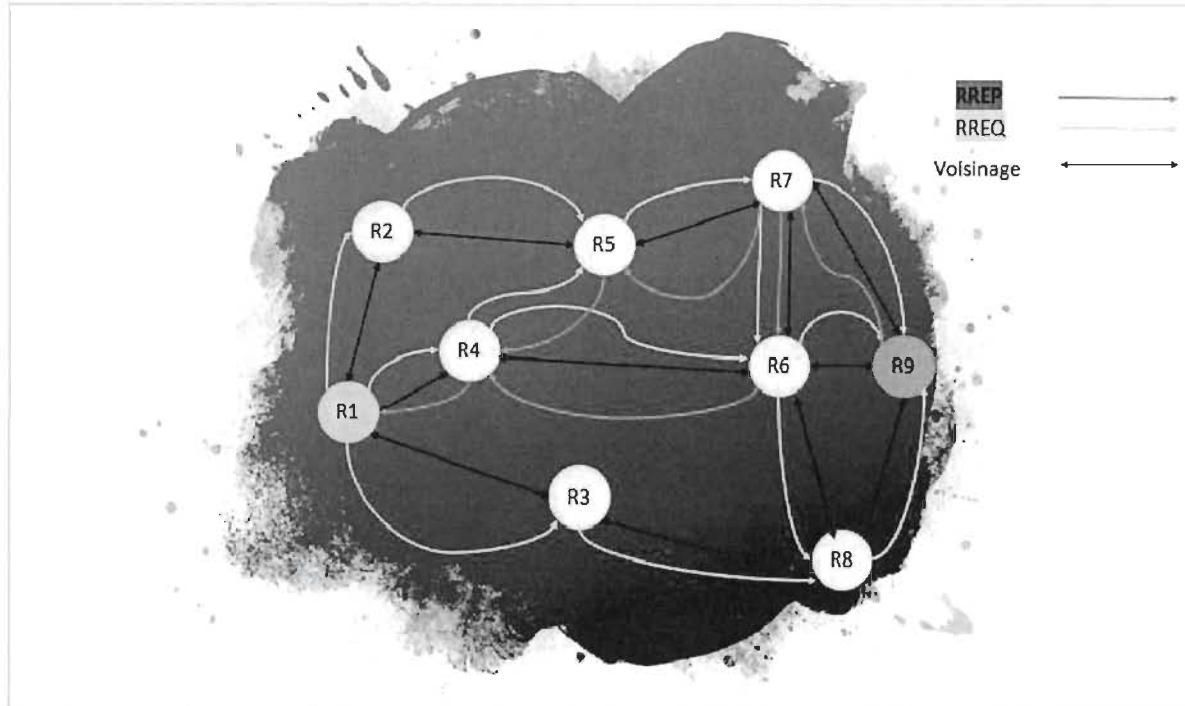


Figure 4.9 : Le fonctionnement du protocole AODV

#### 4.4 Résultats des simulations

Nous avons étudié les résultats de simulation en se concentrant sur trois points principaux :

- La communication entre robots.
- L'importance du RSU.
- La communication entre RSU et robots.

#### 4.4.1 La communication entre robots

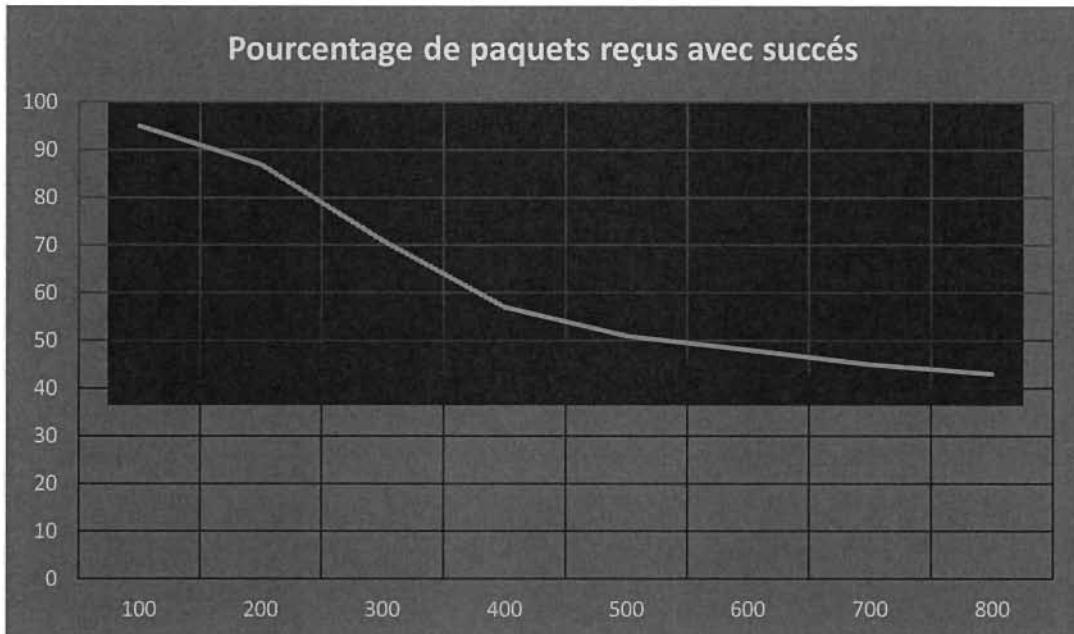


Figure 4.10 : Le taux de consommation de la bande passante par rapport au nombre de mauvaises herbes

La figure 4.10 est la représentation graphique de la première simulation que nous avons faite. Ce graphe nous montre le pourcentage de paquets reçus avec succès par rapport à la quantité des mauvaises herbes dans le terrain. Nous constatons qu'au fur et à mesure que le nombre des mauvaises herbes augmente, le nombre des paquets reçus avec succès diminue. Nous n'avons pas considéré le RSU dans ce scénario, ce qui explique la perte de certains paquets quand les robots sont hors de portée (+800m) l'un de l'autre.

Nous avons augmenté le nombre des mauvaises herbes dans le champ et comme résultat, nous avons constaté que le pourcentage des paquets perdus d'un nœud à l'autre avait augmenté aussi. Ces résultats sont évidents parce qu'en augmentant le nombre d'informations à partager entre les nœuds, sans ajouter de nouveaux nœuds ou une RSU dans le réseau, on risque de dégrader la qualité de la communication. Au fur et à mesure que l'information à partager augmente, mais pas les ressources qui partagent cette information, les paquets vont continuer à se perdre.

#### 4.4.2 L'importance du RSU

Afin d'assurer la disponibilité de notre système de communication, nous avons implémenté un RSU dans la deuxième simulation. Nous avons comparé les résultats de pourcentages des nœuds mobiles qui reçoivent un paquet de mise à jour dans un scénario où le RSU n'est pas installé, et dans un deuxième scénario où il l'est.

Le RSU, en anglais Road Side Unit, est un dispositif qui sert d'habitude dans les réseaux VANET d'intermédiaire entre l'infrastructure et les nœuds du réseau. Il est habituellement placé dans les bords de la route pour permettre aux véhicules mobiles de rester en communication avec tous les composants du réseau, à savoir les autres véhicules et l'autorité de certification. La figure 4.11 nous montre à gauche de la photo un modèle de RSU, et à droite un RSU qui est placé au bord d'une route.

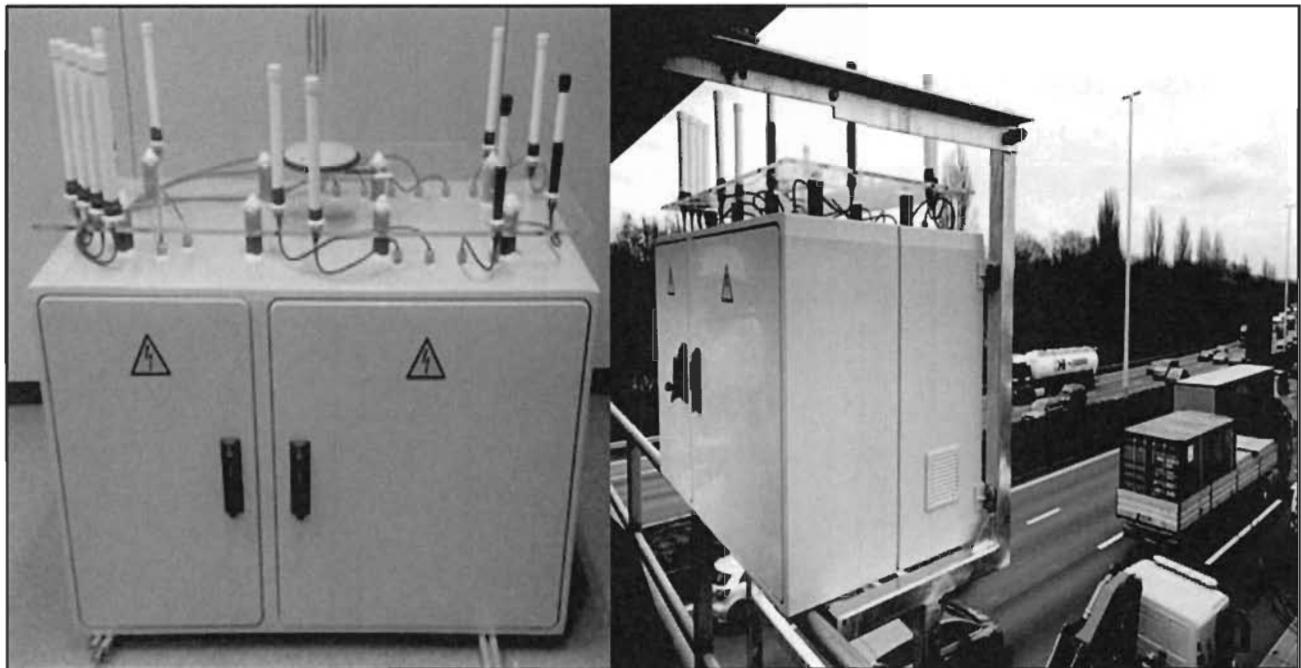


Figure 4.11 : Modèle de RSU [17]

Nous avons mentionné dans les paramètres des simulations que le terrain était d'une surface de 2km x 2km, et dans certains scénarios nous avons placé le RSU en plein milieu du terrain dans le but d'assurer une communication non interrompue dans tout le périmètre du terrain. Un RSU est

une antenne de bord de route, habituellement utilisé dans les réseaux VANET pour assurer la continuité de la communication dans le sein du réseau mobile. Ainsi, si la surface du terrain augmente, il faudra placer plus de RSU dans le nouveau terrain pour continuer à offrir la même qualité de communication qu'avant.

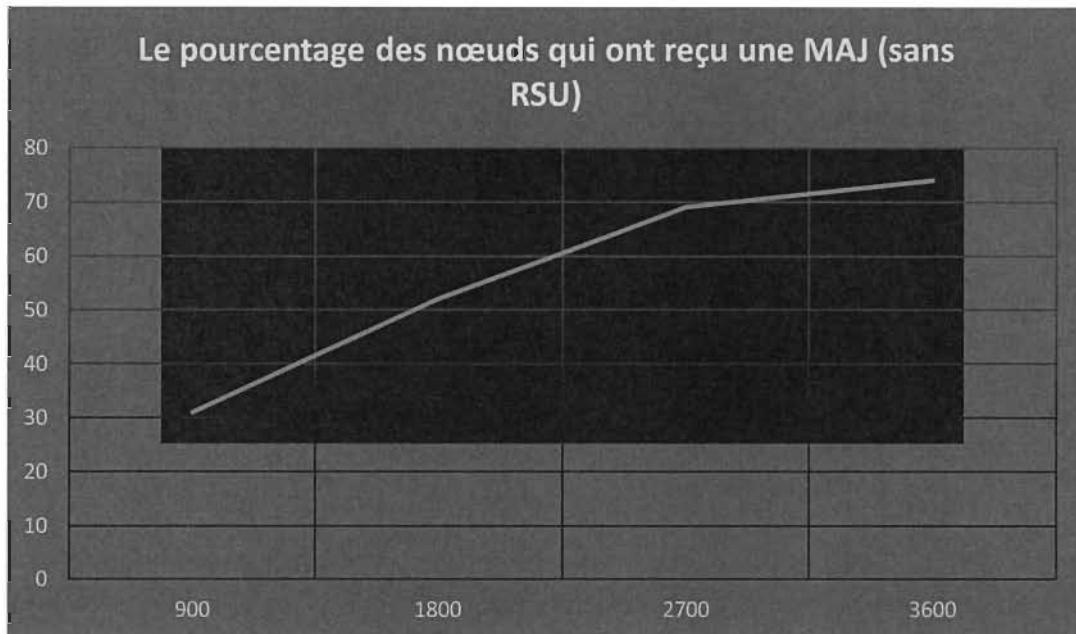


Figure 4.12 : Le pourcentage des nœuds qui ont reçu une MAJ (sans RSU) par rapport au temps

La figure 4.12 représente le pourcentage des nœuds qui ont reçu une mise à jour dans un réseau sans RSU pendant une simulation de 3600 secondes. Nous pouvons constater que le nombre de nœuds qui reçoivent le paquet de mise à jour augmente par rapport au temps, mais ils n'atteignent pas le 80%. Certes les paquets de mise à jour sont reçus par 80% des nœuds du réseau, mais ce sondage nous montre aussi qu'il y a une faille dans notre mécanisme de communication. En augmentant le nombre des mauvaises herbes dans la simulation précédente, nous avons perdu davantage des paquets lors du partage de l'information. La perte de l'information est directement liée au fait que les robots sont hors portée les uns des autres dans certains cas.

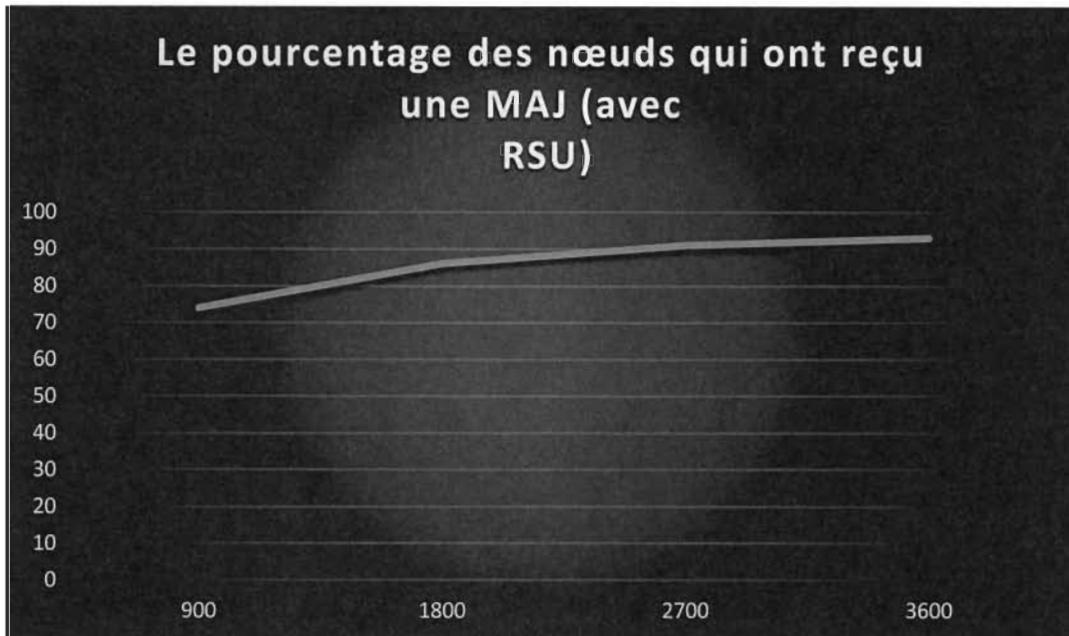


Figure 4.13 : Le pourcentage des nœuds qui ont reçu une MAJ (avec RSU) par rapport au temps

La figure 4.13 montre que la présence du RSU permet de propager davantage les mises à jour dans un réseau où les nœuds risquent d'être hors portée l'un de l'autre. Les paquets de mise à jour sont envoyés et reçus par l'ensemble des nœuds du réseau. Le RSU assure une communication qui ne coupe pas même en longue portée. Vers la fin de la simulation, nous remarquons que tous les nœuds ont reçu la mise à jour. L'ajout du RSU avait garanti une communication continue entre les nœuds du réseau, le concept de l'hors portée est inexistant dans la présence du RSU. Le RSU est affecté plus par la surface du terrain que par le nombre d'informations à partager. Dans le cas où le terrain devient plus grand, il faudra ajouter un ou plusieurs RSU. Ces RSU vont travailler ensemble afin de garder tous les nœuds connectés en tout temps.

#### 4.4.3 La communication entre RSU et robots

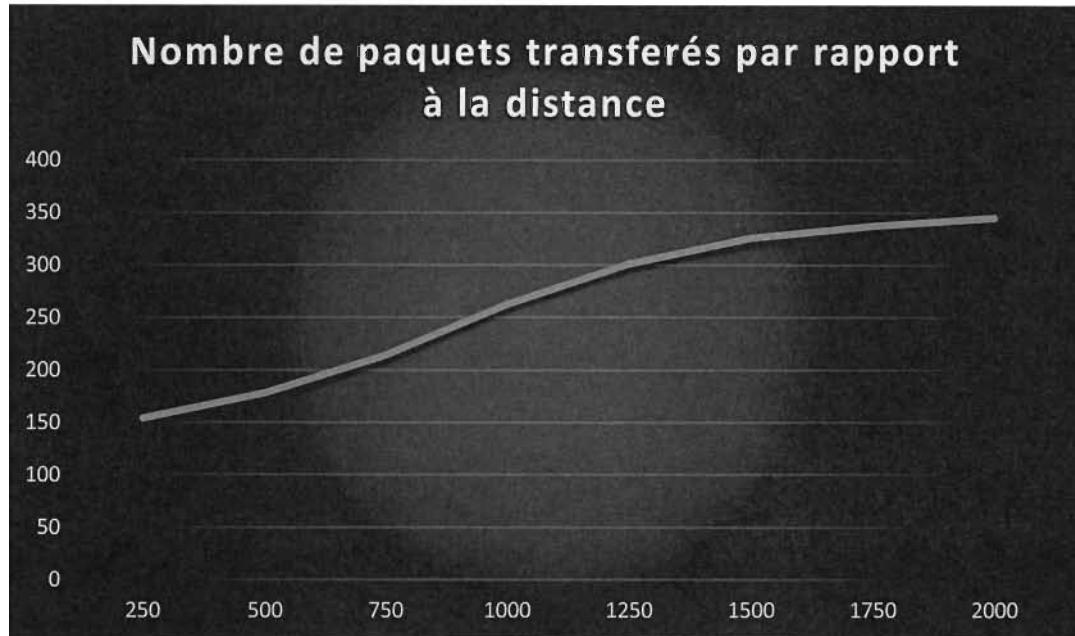


Figure 4.14 : Nombre de paquets transférés par les nœuds par rapport à la distance avec RSU

La figure 4.14 représente le nombre de paquets transférés par les nœuds par rapport à la distance du terrain qui est d'une Superficie de 2000x2000m. Grâce au RSU qui se trouve au milieu du terrain, nous avons évité la perte de plusieurs paquets, une perte qui aurait été garantie sans le RSU.

Afin de montrer l'importance du RSU, nous avons augmenté la surface du terrain jusqu'à 5km x 5km. La figure 4.15 représente une perte importante de paquets, ceci est dû au fait que la portée des robots n'est que de 800m. Sans RSU qui joue le rôle d'intermédiaire entre les robots, la perte des paquets est garantie.

La figure 4.16 prouve l'utilité du RSU dans la transmission des données. Cependant, un RSU qui n'est pas bien placé dans le terrain ne serait pas capable de livrer tous les paquets à l'ensemble des nœuds du réseau qui sont hors portée. Également, un RSU ne serait pas capable de gérer un terrain d'une grande surface tout seul. La figure 4.17 montre qu'en ajoutant un deuxième RSU, le pourcentage des paquets reçus avec succès augmente remarquablement.

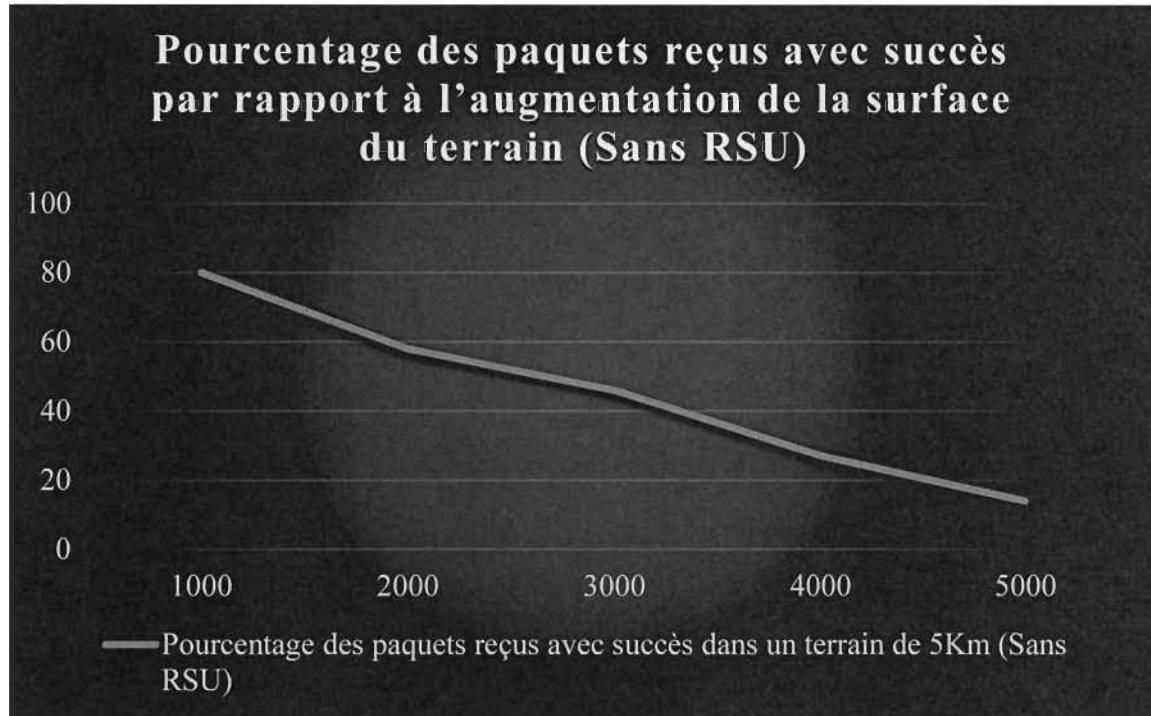


Figure 4.15 : Pourcentage des paquets reçus avec succès par rapport à l'augmentation de la surface du terrain (Sans RSU)

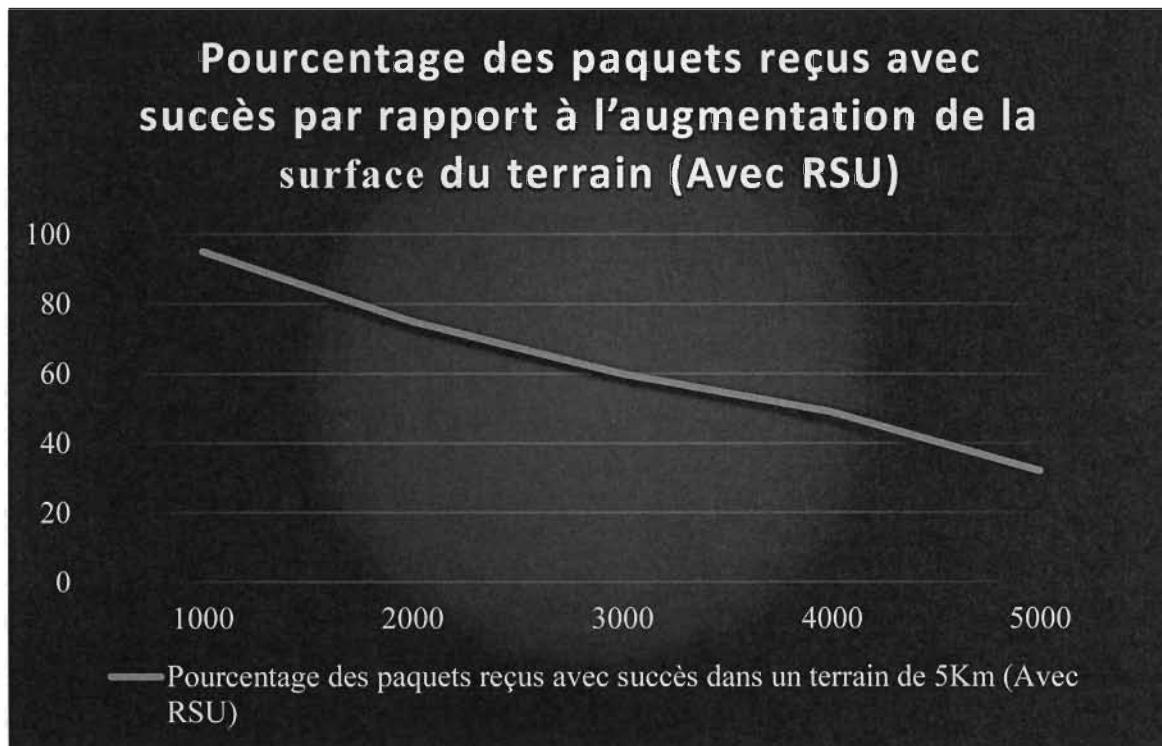


Figure 4.16 : Pourcentage des paquets reçus avec succès par rapport à l'augmentation de la surface du terrain (Avec RSU)

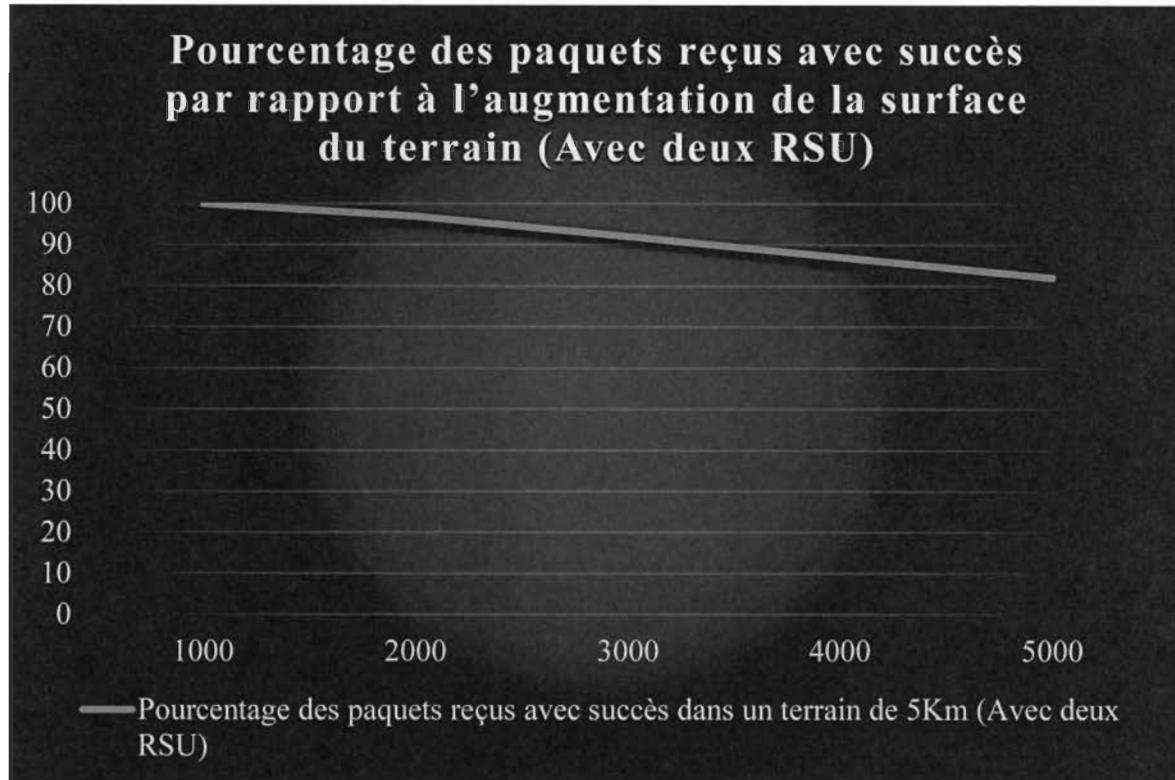


Figure 4.17 : Pourcentage des paquets reçus avec succès par rapport à l'augmentation de la surface du terrain (Avec deux RSU)

Le RSU a prouvé son importance dans notre modèle à travers les simulations précédentes. Comme le montrent les graphes ci-dessus, la présence du RSU est cruciale pour offrir une communication continue aux nœuds (robots) du réseau. Cette communication qui assure la satisfaction des exigences d'un réseau VANET, telles que : la confidentialité , l'authentification , l'intégrité et surtout la disponibilité de tous les nœuds du réseau en tout temps. Et enfin, le RSU aide à ne pas perdre des paquets qui peuvent contenir des informations sur les mauvaises herbes à sarcler.

#### 4.4.4 La bande passante et le nombre des mauvaises herbes dans le champ

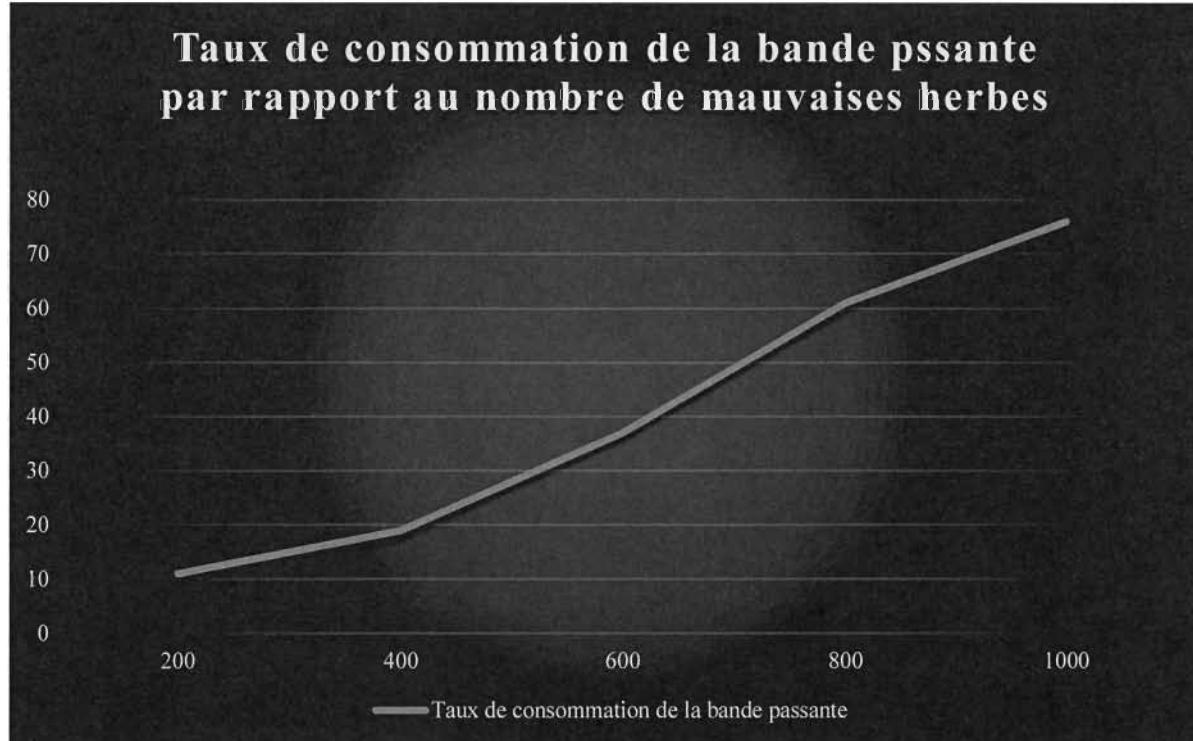


Figure 4.18 : Nombre de paquets transférés par les nœuds par rapport a la distance avec RSU

La figure 4.18 représente le lien entre la consommation de la bande passante et la quantité des mauvaises herbes dans le champ. On remarque sur le graphe qu'en augmentant le nombre de mauvaises herbes dans le champ, les nœuds robots consomment plus de bande passante. Ceci peut être expliqué par l'augmentation des paquets envoyés et reçus par les robots à propos des mauvaises herbes additionnelles à sarcler. La présence de plusieurs nouvelles mauvaises herbes dans le champ crée un besoin de communication entre les nœuds, car ils doivent partager toutes les informations et mettre à jour tout le réseau. Également, le protocole AODV choisi évite la création des boucles de routage en envoyant des messages requêtes à chaque fois qu'un nœud souhaite communiquer avec un autre nœud. Comme nous l'avons expliqué dans le deuxième chapitre, AODV repose surtout sur ces messages requêtes pour s'assurer que le chemin entre les deux nœuds concernés est fonctionnel, mais AODV a d'autres messages qui circulent dans le

réseau, comme : RREP ( messages réponse), RRER ( messages erreur). Ainsi, nous avons remarqué que AODV a un modèle de routage qui dépend sur le travail d'équipe des nœuds du réseau, ce qui explique aussi la grande consommation de la bande passante à chaque ajout d'information.

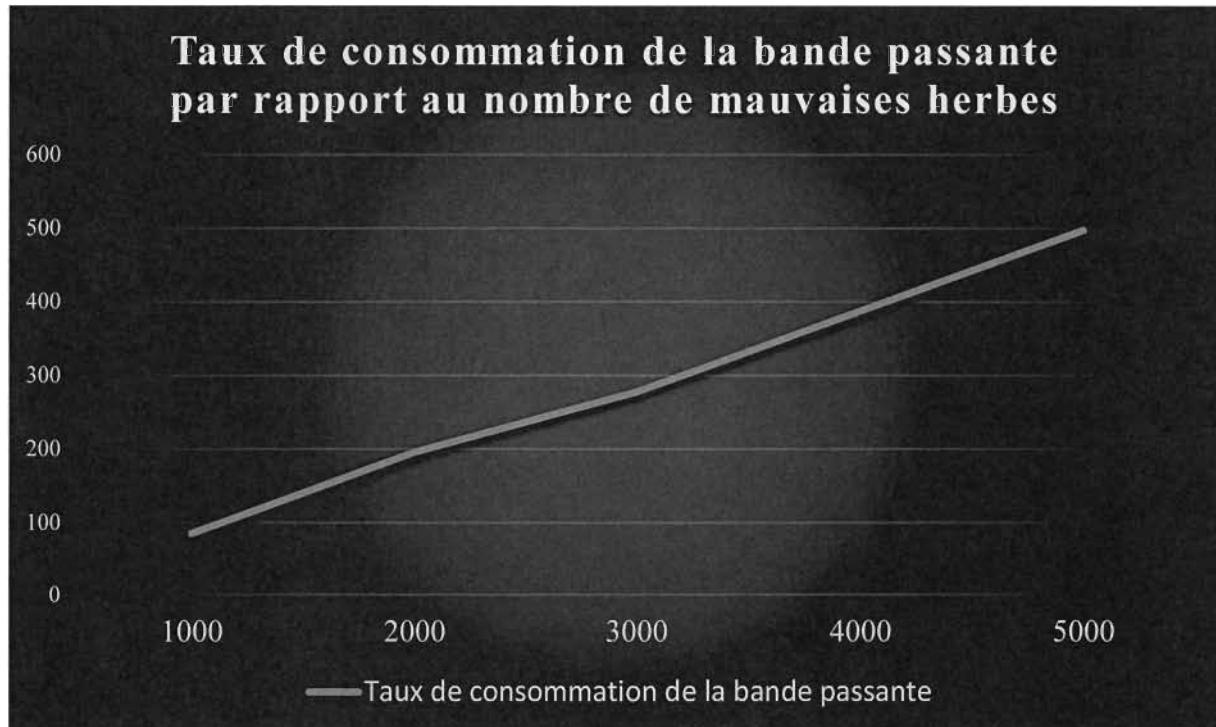


Figure 4.19 : Le rapport entre la consommation de la bande passante et l'augmentation de la quantité des mauvaises herbes dans le champ

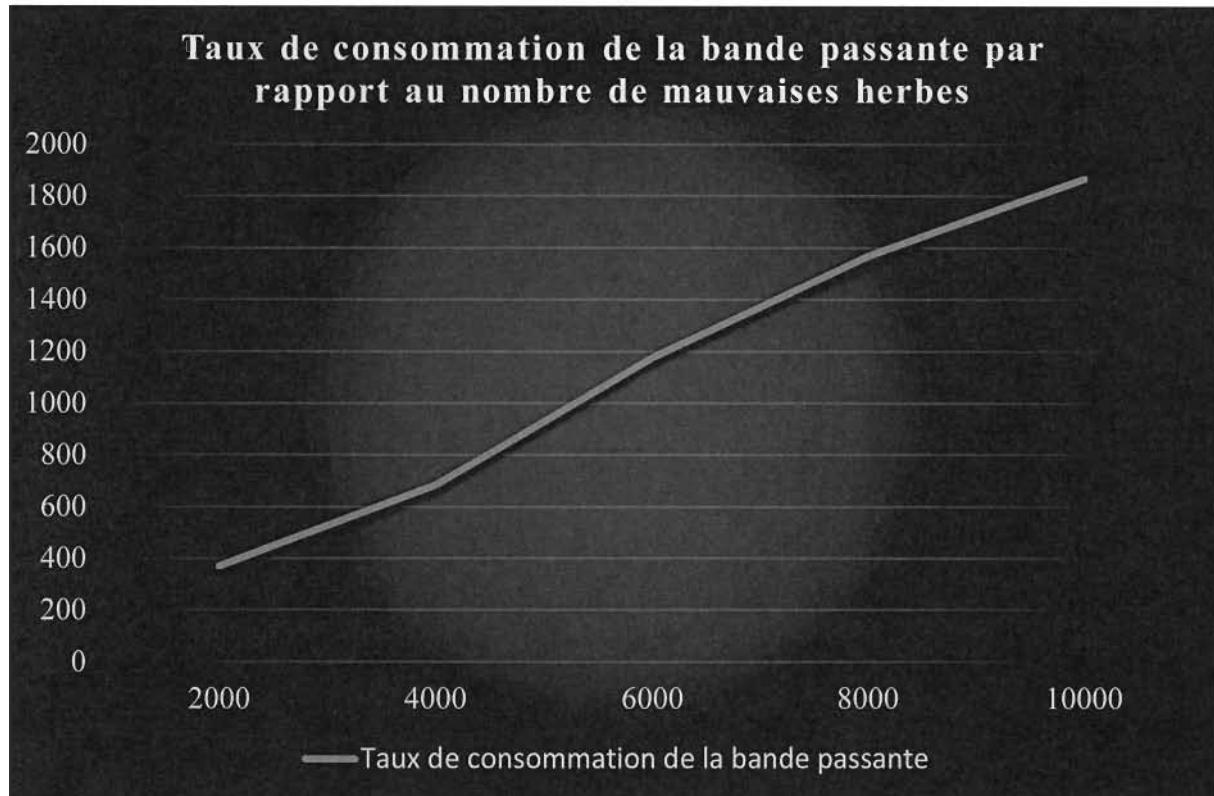


Figure 4.20 : Le rapport entre la consommation de la bande passante et l'augmentation de la quantité des mauvaises herbes dans le champ

Les figures 4.19 et 4.20 représentent l'augmentation de la consommation de la bande passante par rapport à l'augmentation de la quantité des mauvaises herbes dans le champ. Nous avons constaté qu'au fur et à mesure que le nombre des mauvaises herbes augmente dans le champ, la consommation de la bande passante augmente aussi. Ce résultat est normal, parce que les mauvaises herbes représentent des informations que les robots (nœuds) doivent partager entre eux.

#### 4.5 Discussion des résultats

Le résultat de la première simulation représenté dans la figure 4.12 nous montre que l'établissement d'une communication entre les nœuds du réseau n'est pas toujours suffisant. La surface du terrain et la quantité d'information à partager affectent grandement la qualité de la communication. Par exemple, en augmentant la surface du terrain les nœuds ont plus de risque à

circuler hors portée des autres nœuds qui sont de l'autre côté du terrain. Nous allons prendre comme deuxième exemple l'augmentation de la quantité des mauvaises herbes, car en augmentant les mauvaises herbes dans le terrain nous augmentons la quantité d'informations et les paquets de mise à jour à partager. Ces augmentations de surface et/ou d'informations mènent à la perte des paquets supposés être reçus par tous les nœuds du réseau, ce qui crée un déséquilibre au niveau des mises à jour des chemins entre les nœuds comme le montre la figure 4.13. Nous avons suggéré d'ajouter un RSU en plein milieu du terrain comme solution au problème de distance entre les nœuds même quand ils sont hors portée l'un de l'autre. Comme le montre la figure 4.14, le RSU joue le rôle d'un intermédiaire entre les nœuds qui sont séparés par une distance qui peut discontinue leur communication. Sans RSU, nous avons remarqué une perte exponentielle de données. Cependant, après l'implémentation du RSU, nous avons remarqué que tous les nœuds ont reçu leurs paquets de mise à jour. La figure 4.15 montre également que tous les nœuds ont réussi à recevoir leurs paquets sans aucune perte tout en étant dispersés sur toute la surface du terrain.

Le RSU couvre un périmètre de 2km x 2km. Donc en agrandissant le terrain, il est crucial d'ajouter un ou des RSU pour pouvoir bénéficier de la continuité de connexion entre les nœuds et sans aucune perte de données.

En ce qui concerne la bande passante, nous avons remarqué que sa consommation augmente avec l'augmentation des paquets à partager. Ceci est dû partiellement à l'augmentation des informations et mises à jour à partager, et d'un autre côté au protocole de routage AODV. Le protocole avec lequel nous travaillons repose sur une demande de chemins à chaque demande de connexion entre deux nœuds. Les voisinages des deux nœuds concernés sont impliqués dans les requêtes de chemins afin de pouvoir mettre à jour tous les chemins du réseau en tout temps. L'avantage de l'AODV réside dans la mise à jour des tables de chemins de tous les nœuds du réseau, ce qui évite d'avoir des chemins défectueux. Cependant, cette stratégie de routage consomme de la bande passante, car tous les nœuds du réseau doivent mettre à jour leurs tables très régulièrement.

#### **4.6 Conclusion**

Dans ce quatrième chapitre, nous avons expliqué le choix de simulateur ainsi que celui du protocole de routage avec lesquels nous avons travaillé. Nous avons présenté et discuté les résultats obtenus des simulations exécutées. Le cinquième et dernier chapitre marquera la fin de ce mémoire et servira de conclusion.

## Chapitre 5 : Conclusions et Perspectives

Les robots sarclieurs de champs existent et sont déjà utilisés par des agriculteurs pour éliminer les mauvaises herbes et rendre leurs récoltes plus rentables. Cependant, dans une situation où l'agriculteur possède plusieurs robots sarclieurs et qu'il souhaite les utiliser ensemble dans un grand champ, afin de recréer une ambiance de travail d'équipe tout en évitant des dégâts matériels, il aura besoin d'un mécanisme qui va assurer la communication entre les robots concernés.

L'objectif de ce travail était de trouver la solution qui rendra possible la création de cette communication. Pour arriver à ce but, nous avons suggéré de prendre le concept des réseaux mobiles Ad-Hoc et de l'appliquer sur ce groupe de robots pour les transformer en des nœuds, à l'aide d'un RSU et d'un serveur central, afin de construire un réseau capable de partager des informations.

Pour mettre notre suggestion à l'épreuve, nous avons essayé plusieurs simulateurs et protocoles de routage. Notre choix final s'est constitué de NS2 comme simulateur et AODV comme protocole. Cette combinaison était celle qui nous a offert les résultats les plus satisfaisants sans erreurs et sans problèmes techniques.

Les résultats, discutés dans le chapitre précédent, nous montrent que notre suggestion peut être adoptée dans un cadre réel. Cependant, pour améliorer la performance de notre mécanisme en temps réel, il faut considérer certaines variables, comme : le type des robots sarclieurs utilisés (systèmes pilotés par caméra, GPS, etc.), la capacité et le temps de décharge des batteries des robots, et la représentation des localisations géographiques des mauvaises herbes.

En résumé, le but de ce projet était d'introduire le concept des réseaux Ad-Hoc au milieu de l'agriculture, afin de créer une communication non interrompue entre les robots qui sarclent les mauvaises herbes dans un champ de bleuets nains, pour améliorer la performance des robots sarclieurs et aussi pour améliorer la qualité de la récolte tout en économisant du temps, de l'argent, de l'énergie, et pour pallier le manque de mains d'œuvre dans ce domaine.

Les perspectives que nous suggérons sont de réaliser des simulations réelles en s'appuyant sur les simulations de ce mémoire. En adoptant le fonctionnement du protocole AODV, nous pouvons commencer par un réseau qui contient 2 robots afin d'analyser leurs communications et réactions, mais en temps réel.

## LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

[1] Andrée Tremblay, "La production du bleuet sauvage dans une perspective de développement durable", <http://perlebleue.ca/images/documents/amenagement/guideproduction/f010.1-2010.pdf>, 2010.

[2] Brakni Taher et Bentaher Atif, " Les protocoles de routages dans les réseaux pair-à-pair" – Université de Larbi Tébessi, Tébessa, Algérie, 29 Mai 2016.

[3] C. Mallanda, A. Suri, V. Kunchakarra, S.S. Iyengar, R. Kannan et A. Durresi, " Simulating Wireless Sensor Networks with OMNeT++" – Université de l'état de Louisiane, Bâton rouge, Louisiane, États-Unis, 24 Janvier 2005.

[4] Claude Chaudet, "Autour de la réservation de bande passante dans les réseaux Ad-Hoc" - Réseaux et télécommunications [cs.NI]. L'Institut National des Sciences Appliquées de Lyon, 2004.

[5] David Cavin, Yoav Sasson et André Schiper, " On the Accuracy of MANET Simulators" - École polytechnique Fédérale de Lausanne, Suisse, 2002.

[6] Davinia Font, Tomàs Pallejà, Marcel Tresanchez, David Runcan, Javier Moreno, Dani Martínez, Mercè Teixidó et Jordi Palacín, "A Proposal for Automatic Fruit Harvesting by Combining a Low-Cost Stereovision Camera and a Robotic Arm", Université de Lérida, Espagne, 30 juin 2014.

[7] Dominique Dhoutaut, "Étude du standard IEEE 802.11 dans le cadre des réseaux Ad-Hoc : de la simulation à l'expérimentation" - L'Institut National des Sciences Appliquées de Lyon, 11 Décembre 2003.

[8] Florent Kaisser, " Communications dans les réseaux fortement dynamiques" – Université Paris-Sud, Paris, 21 Juin 2010.

[9] Guangyu pei, Mario Gerla et Tsu-Wei Chen, "Fisheye state routing: A routing scheme for Ad-Hoc Wireless Networks" – Publié sur : 2000 IEEE International Conference on Communications. ICC 2000. Global Convergence Through Communications. Conference Record, 2000.

[10] Guillaume CHELIUS, "Architectures et communications dans les réseaux spontanés sans fil" – INSA, Lyon, 26 Avril 2004.

[11] Guoyou He, "Destination-Sequenced Distance Vector (DSDV) Protocol" – L'université Technologique d'Helsinki, Espoo, Finlande, 2002.

[12] Guthemberg Silvestre, "Simulation de systèmes en réseau avec ns2", Cours de Master ISTR, Spécialité ASTR à L'université Paul Sabatier, Toulouse, France, 2015.

[13] HELAILI Nabila et MEKHNACHE Salima, " Simulation du Routage dans les Réseaux de Capteurs Sans Fils" – Université de Béjaia, Algérie, 2017.

[14] <http://perlebleue.ca/images/documents/amenagement/guideproduction/f004.1-2010.pdf> vu le 20 juin 2020 (Guide de lutte intégrée contre les mauvaises herbes dans les bleuetières, 2010).

[15] <https://networksimulationtools.com/protocols-used-in-ns2/> vu le 12 juillet 2020 (Les protocoles utilisés par NS2).

[16] [https://orgprints.org/34067/1/Bioactualités\\_9\\_2018\\_H.Dierauer.pdf](https://orgprints.org/34067/1/Bioactualités_9_2018_H.Dierauer.pdf) Vu le 22 octobre 2020.

[17] <https://medium.com/predict/edge-computing-is-so-much-more-fun-ac2a8a23e696> Vu le 9 novembre 2020.

[18] <https://www.laterre.ca/utiliterre/equipement/des-robots-quebecois-bientot-dans-les-champs> Vu le 2 janvier 2021.

[19] [https://www.researchgate.net/figure/Simulation-environment-of-Network-Simulator-2-NS2\\_fig1\\_329757869](https://www.researchgate.net/figure/Simulation-environment-of-Network-Simulator-2-NS2_fig1_329757869) Vu le 4 septembre 2019.

[20] [https://www.researchgate.net/figure/ns-3-Simulator-Interface\\_fig2\\_234018317](https://www.researchgate.net/figure/ns-3-Simulator-Interface_fig2_234018317) Vu le 6 septembre 2019.

[21] [https://www.researchgate.net/figure/Multi-homed-NEMO-simulation-environment-in-OMNeT\\_fig3\\_254036540](https://www.researchgate.net/figure/Multi-homed-NEMO-simulation-environment-in-OMNeT_fig3_254036540) Vu le 7 septembre 2019.

[22] L. Abusalah, A. Khokhar, and M. Guizani, "A survey of secure mobile Ad-Hoc routing protocols," *Communications Surveys & Tutorials*, IEEE, vol. 10, pp. 78-93, 2008.

[23] L. Kleinrock et K. Stevens, "Fisheye: A lenslike computer display transformation" – Technical report, UCLA, Computer science department, 1971.

[24] M. Sujarithaa, S. Annaduraia, J. Satheeshkumarb, S. Kowshik Sharana et L. Mahesha, "Weed detecting robot in sugarcane fields using fuzzy real time classifier" – ELSEVIER, *Computers and Electronics in Agriculture*, 7 Février 2017.

[25] Mireille Bellemare, Véronique Moreau, Pierre-Olivier Martel, Andrée Tremblay et Gérard Bouchard, "Développement d'une régie biologique de lutte contre les mauvaises herbes dans une bleuetière aménagée sans hexazinone", [https://www.mapaq.gouv.qc.ca/SiteCollectionDocuments/Recherche\\_Innovation/Petitsfruits/121NNO304.pdf](https://www.mapaq.gouv.qc.ca/SiteCollectionDocuments/Recherche_Innovation/Petitsfruits/121NNO304.pdf), 11/2012.

[26] Redmond Ramin Shamshiri, Cornelia Weltzien, Ibrahim A. Hameed, Ian J. Yule, Tony E. Grift, Siva K. Balasundram, Lenka Pitonakova, Desa Ahmad et Girish Chowdhary, "Research and development in agricultural robotics: A perspective of digital farming" – *International Journal of Agricultural and Biological Engineering*, Juillet 2018.

[27] Samyak Shah, Amit Khandre, Mahesh Shirole et Girish Bhole, "Performance Evaluation of Ad-Hoc Routing Protocols Using NS2 Simulation" – L'institut technologique de Veermata Jijabai, Mumbai, Inde, 2008.

[28] Shraddha S. Durugkar, Priyanka S. Jadhav, Shraddha S. Zade, Vijay S. Bhong, "Design of farming robot for weed detection and herbicides applications using image processing" -IJRET: International Journal of Research in Engineering and Technology, 19 septembre 2018.

[29] T. Clausen, P. Jacquet, "Optimized Link State Routing Protocol (OLSR)" – Project Hipercom, INRIA, Octobre 2003.

[30] Youcef ziani, "Étude comparative de méthodes de routage dans les réseaux de capteurs sans fil pour le domaine résidentiel" – Université de Québec à Trois-Rivières, Juin 2013.

[31] Z. Haas, J. Deng, B. Liang, P. Papadimitratos et S. Sajama "Wireless Ad-Hoc Networks" – Wiley Encyclopedia of telecommunications, 15 Avril 2003.