

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université de Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj  
Faculté des Mathématiques et d'Informatique  
Département d'Informatique



## MÉMOIRE

Présenté en vue de l'obtention du diplôme  
**Master en Informatique**  
Spécialité : Réseaux et Multimédia

## THÈME

Un nouveau protocole à état de lien basé sur l'énergie  
résiduelle des nœuds et la stabilité des liens pour les  
réseaux FANET

*Présenté par :*  
Bouderouaz Djouaria  
Bensefia Nadine

*Soutenu publiquement le : 12/06/2025*  
*Devant le jury composé de :*

<b>Président</b>	Dr Senouci Oussama
<b>Examineur</b>	Dr Benaouda Nadjib
<b>Encadreur</b>	Dr Mousaoui Ali

**Année Universitaire 2024/2025**

# Remerciement

Après avoir rendu grâce à **Allah** le tout puissant et le miséricordieux

**Nous tenons à remercier :**

**Dr Ali Moussaoui**, notre encadrant, pour sa disponibilité, sa patience et ses conseils surtout leurs suivis durant la préparation de ce mémoire.

Nos vifs remerciements aux membres de jury qui nous font l'honneur d'examiner notre travail : **Dr Benaouda Nadjib** et **Dr Senouci Oussama**.

Nos remerciement s'adresse également à tous nos professeurs de faculté de département de l'informatique pour leur générosité et la qualité de l'enseignement qu'ils ont prodigué au cours de notre formation à l'université de Bordj Bou Arreridj.

Les familles "**Bouderouaz**" et "**Bensefia**" qui nous ont été d'un encore soutien durant toute la période de cette recherche.

Merci à tous les chercheurs dont les travaux nous ont orientées et à toute personne ayant contribué à la réalisation de ce travail.

# Dédicace

Je dédie ce travail à *moi-même*, pour ma persévérance et mes efforts tout au long de ce parcours.

Je dédie ce mémoire à la mémoire de mon cher père, **qu'Allah lui fasse miséricorde**. Un homme formidable, exceptionnel, qui a toujours cru en moi, soutenu mes rêves et guidé mes pas avec sagesse et amour. Bien que tu ne sois plus à mes côtés, ton souvenir et ta force continuent de m'accompagner chaque jour. Que ce travail soit un hommage à tout ce que tu m'as transmis.

Je le dédie également à ma chère mère et à ma sœur, ainsi qu'à toute ma famille, qui m'ont toujours soutenue avec amour et encouragement.

Je tiens de remercier infiniment ma binôme **Djouaria Boudrouaz** pour sa coopération et patience et tout le temps travaille sans fatigue .

Je tiens aussi à dédier ce travail à mes précieuses amies : *Chaima Cheikh , Wissem Bendahman , Samah Zaharoui, Asma Bougetaya, Asma Khoudour, Maroua Djoudi, Hanane Benhamouda, Baya Mekhfi*, ainsi qu'à toutes mes amies qui ont été présentes derrière moi tout au long de ce chemin.

Je n'oublie pas les membres de l'**Infinity Club**, dont l'énergie, l'esprit d'équipe et l'engagement ont toujours été une source de motivation et d'inspiration pour moi.

**Nadine**

# Dédicace

Je dédie ce modeste travail : À mes très chers **parents**, qui ont toujours été à mes côtés, m'ont guidé durant les moments les plus pénibles de ce long parcours, m'ont soutenu sans relâche et ont sacrifié toute leur vie pour me voir devenir ce que je suis aujourd'hui, en particulier

**ma chère mère, Merci mes parents.**

À mes très chers frères : *Ayoub, Hicham et Abd-Elhafidh*, ainsi qu'à mes très chères sœurs : *Sanaa, Amina, Meriem, Houda, Chaima et Maroua*, qui m'ont toujours accompagné durant la réalisation de ce travail et m'ont encouragé tout au long de mon parcours.

À mes chers neveux et nièces : *Lokman, Yacine, Selma, Zakareia, Nadjla, Safaa, Rihab, Manar, Ibrahim, Bissan, Tasnim, Amdjad*, ainsi que *Assil*.

À ma binôme **Nadine Bensefia**, sans qui ce travail aurait été irréalisable.

À tous mes amis qui m'ont toujours encouragé, et à qui je souhaite encore plus de succès, tout particulièrement : *Achwek et Chaima. Chaima Cheikh, Wissem Bendahman*

À toute la famille **Bouderouaz**, sans exception.

c toute la promotion de **2025**.

Et à tous ceux que j'aime.

Je dédie aussi mon travail à mes frères et sœurs à **Gaza**

À qui nous souhaitons une victoire proche.

**Djouaria**

# Résumé

Les réseaux mobiles aériens (FANETs : Flying Ad-Hoc Networks) est une sous-classe des réseaux ad hoc dédiée à la communication entre les drones. Parmi les grands défis de ce type de réseau est le routage de données avec qualité de service. Malheureusement, les protocoles de routage ad hoc tel qu'ils sont ne sont pas adaptés au réseaux FANETs. Dans ce travail nous proposons une amélioration au protocole OLSR standard pour qu'il tienne en considération les contraintes spécifiques aux FANETS. L'idée proposée est d'améliorer la sélection des MPR (MultiPoint Relays) en intégrant des métriques sensibles à la QoS : l'énergie résiduelle des nœuds, le nombre de voisins à deux sauts, ainsi que la variance des distances entre les drones. Les résultats de simulations obtenus en comparaison avec OLSR standard montrent que le nouveau protocole réduit significativement la perte de paquets entre les communicants.

**Mots clés :** FANET (Flying Ad-Hoc Network), OLSR (Optimized Link State Routing), QoS (Qualité de Service),

---

# Abstract

Aerial mobile networks (FANETs : Flying Ad-Hoc Networks) are a subclass of ad hoc networks dedicated to communication between drones. Among the major challenges of this type of network is data routing with quality of service. Unfortunately, ad hoc routing protocols as they are are not suitable for FANETs. In this work, we propose an improvement to the standard OLSR protocol to take into account the specific constraints of FANETs. The proposed idea is to improve the selection of MPRs (MultiPoint Relays) by integrating QoS-sensitive metrics : the residual energy of the nodes, the number of two-hop neighbors, as well as the variance of distances between the drones. The simulation results obtained in comparison with standard OLSR show that the new protocol significantly reduces packet loss between communicators.

**Keywords** : FANET (Flying Ad-Hoc Network), OLSR (Optimized Link State Routing), QoS (Qualité de Service),

---

## ملخص

الشبكات الجوية المتنقلة (FANETs: Flying Ad-Hoc Networks) هي فئة فرعية من الشبكات المؤقتة مخصصة للتواصل بين الطائرات بدون طيار. من بين التحديات الكبرى لهذا النوع من الشبكات هو توجيه البيانات بجودة خدمة. للأسف، فإن بروتوكولات التوجيه المؤقتة كما هي ليست مناسبة لشبكات سارة. في هذا العمل، نقترح تحسيناً على بروتوكول OLSR القياسي ليأخذ في الاعتبار القيود الخاصة بشبكات سارة. الفكرة المقترحة هي تحسين اختيار MPR (MultiPoint Relays) من خلال دمج مقاييس حساسة لجودة الخدمة (QoS): الطاقة المتبقية للعقد، عدد الجيران على بعد قفزتين، وكذلك تباين المسافات بين الطائرات بدون طيار. أظهرت نتائج المحاكاة التي تم الحصول عليها بالمقارنة مع صشرضض القياسي أن البروتوكول الجديد يقلل بشكل كبير من فقدان الحزم بين المتواصلين.

الكلمات المفتاحية: FANET (شبكة طائرات بدون طيار)، QoS (جودة الخدمة)

# Table des matières

<b>Liste des abréviations</b>	<b>2</b>
<b>Liste des figures</b>	<b>3</b>
<b>Liste des tableaux</b>	<b>4</b>
<b>Introduction générale</b>	<b>5</b>
<b>1 Généralités sur les réseaux FANET</b>	<b>8</b>
1.1 Introduction . . . . .	8
1.2 Réseau Ad-hoc . . . . .	8
1.3 Les types de réseaux Ad-hoc . . . . .	9
1.3.1 VANET (Vehicular Ad-hoc Network) . . . . .	9
1.3.2 MANET (Mobile Ad-hoc Network) . . . . .	10
1.3.3 FANET . . . . .	10
1.4 Caractéristiques des FANETs . . . . .	11
1.5 Architecture des FANET . . . . .	12
1.6 Modèle de mobilité et de communication . . . . .	14
1.6.1 Modèle de communication . . . . .	14
1.6.1.1 UAV to UAV . . . . .	14
1.6.1.2 UAV à infrastructure . . . . .	15
1.6.2 Modèle de mobilité . . . . .	15
1.7 Comparaison MANET, VANET et FANET . . . . .	17
1.8 Conclusion . . . . .	18
<b>2 Le Routages avec QoS dans les FANETs</b>	<b>20</b>

2.1	Introduction . . . . .	20
2.2	Routage dans les reseaux FANETs et leur defis . . . . .	20
2.3	Classification des protocoles de routages FANETs . . . . .	21
2.3.1	Protocoles de routages Réactif . . . . .	22
2.3.1.1	RGR (Reactive Greedy Routing) . . . . .	22
2.3.1.2	RTOR (Reliable Topology and Opportunistic Routing) . . . . .	23
2.3.2	Protocole de routage Proactif . . . . .	23
2.3.2.1	P-OLSR(OLSR Améliorée) . . . . .	23
2.3.2.2	ML-OLSR . . . . .	24
2.4	La Qualité de service . . . . .	24
2.5	Les métriques de qualité du service . . . . .	25
2.6	Conclusion . . . . .	25
<b>3</b>	<b>Contribution et Validation</b>	<b>27</b>
3.1	Introduction . . . . .	27
3.2	Approche Proposée . . . . .	27
3.2.1	Choix du Protocole OLSR . . . . .	27
3.2.2	Le Mécanisme de Sélection des MPRs . . . . .	28
3.2.2.1	La Méthode de Sélection Standard . . . . .	28
3.2.2.2	La Méthode de Sélection des MPR Proposée . . . . .	29
3.2.2.3	Format de Paquet . . . . .	32
3.2.2.4	Fonction Objectif . . . . .	32
3.2.2.5	Algorithme de Sélection des MPR Proposé . . . . .	33
3.3	Avantages Prévus de la Méthode Proposée . . . . .	35
3.4	L'environnement de Travail . . . . .	35
3.4.1	La Configuration Matérielle . . . . .	35
3.4.2	Les Outils et les Plateformes Logicielles . . . . .	36
3.4.2.1	Ubuntu . . . . .	36
3.4.2.2	NS-3 . . . . .	36
3.4.3	Le Langage de Développement . . . . .	36
3.4.3.1	C++ . . . . .	36
3.5	Résultats de Simulation . . . . .	37
3.5.1	Paramètre de simulation . . . . .	37

3.5.2	Analyse Des resultats . . . . .	37
3.5.2.1	En foncton de l'Energie . . . . .	38
3.5.2.2	en fonction du Nombre de Noeud . . . . .	39
3.5.2.3	Nombre de paquets reçus en fonction de la vitesse . . . . .	40
3.5.2.4	Nombre de paquets reçus en fonction des coefficients $\alpha$ , $\beta$ et $\gamma$	41
3.6	Conclusion . . . . .	42
<b>Conclusion générale</b>		<b>43</b>
<b>Extrait de Code NS-3</b>		<b>44</b>
3.7	Fonction sendHello . . . . .	44
3.8	Fonction ProcessHello . . . . .	46
3.9	populateNeighborset . . . . .	48
3.10	CalculateVariance . . . . .	49
3.11	MprComputation . . . . .	50
3.12	GetRemainingEnergy . . . . .	52
3.13	ConsommationEnergie . . . . .	53
3.14	Normaliser . . . . .	54
3.15	sendTc . . . . .	54
<b>Références</b>		<b>56</b>
<b>Références Web</b>		<b>60</b>

# Liste des abréviations

**AODV** On Demand Distance Vector.

**FANET** Flying Ad-hoc Network.

**GMM** Gauss Markov mobility.

**MANET** Mobile Ad-hoc Network.

**ML-OLSR** Machine Learning-based Optimized Link State Routing.

**N1** neighbor one hop.

**N2** neighbor two hop.

**NS3** Network Simulator 3.

**OLSR** Optimized Link State Routing.

**P-OLSR** Predictive OLSR.

**QoS** Quality of Service.

**RGR** Reactive Greedy Routing.

**RTOR** Reliable Topology and Opportunistic Routing.

**SE\_ OLSR** Stabilité Énergie OLSR.

**UAV** Unmanned Aerial Vehicle.

**VANET** Vehicular Ad-hoc Network.

# Table des figures

1.1	Réseau vanet . . . . .	9
1.2	Réseau MANET . . . . .	10
1.3	Réseau fanet . . . . .	11
1.4	Point de passage aléatoire (RWP) . . . . .	16
1.5	Modèle de mobilité Gauss-Markov . . . . .	17
2.1	Routage Dans les reseaux Fanets . . . . .	22
3.1	Format de Paquet Nouveau . . . . .	32
3.2	Nombre de paquets reçus en fonction de l'énergie . . . . .	38
3.3	Nombre de paquets reçus en fonction du nombre de nœuds . . . . .	39
3.4	Nombre de paquets reçus en fonction de la vitesse . . . . .	40
3.5	Nombre de paquets reçus en fonction des coefficients $\alpha$ , $\beta$ et $\gamma$ . . . . .	41

# Liste des tableaux

1.1	Comparaison entre fanet, manet et vanet . . . . .	18
3.1	Mesures de distance sur 10 instants . . . . .	30
3.2	L'écart de la moyenne et carées . . . . .	31
3.3	Coût des nœuds candidats au rôle d'un MPR (ces valeurs sont normaliser entre (0 et 100) ) . . . . .	34
3.4	Spécifications de l'appareil utilisé . . . . .	35
3.5	Paramètres de simulation . . . . .	37

# Introduction générale

Les réseaux mobiles Ad hoc sont un type de réseau sans fils et mobiles dont la caractéristique la plus importante est l'absence d'infrastructure fixe. Dans ce type de réseau, la communication entre les nœuds se fait de saut à saut ou d'un nœud mobile à un autre. Plusieurs types de réseaux Ad hoc ont été apparus selon le besoin, l'environnement et le type d'utilisation comme réseau Ad-Hoc de véhicules, les réseaux de capteurs et les réseaux de drones.

Les réseaux de drones ou les réseaux ad hoc aériens, FANETs (Flying Ad hoc network), sont composés de plusieurs drones qui collaborent entre eux pour accomplir une tâche bien spécifique. Un drone est un avion sans pilote qui communique avec une station de base. Parmi les caractéristiques les plus importantes des réseaux FANETs sont : la mobilité en 3D, topologie dynamique, communication à longue distance, ressources limitées, etc.

Les réseaux FANETs sont utilisés dans plusieurs domaines tel que : applications militaires et de défense, gestion des catastrophes, surveillance civile et environnementale, transmission vidéo en temps réel, relais de communication (UAV-to-Ground ou UAV-to-UAV), applications scientifiques et de recherche, logistique et livraison, etc. Les FANETs (Flying Ad Hoc Networks), bien qu'utiles et prometteurs, présentent plusieurs défis majeurs qui compliquent leur conception, leur déploiement et leur fonctionnement. Parmi les principaux défis est le routage des données tout en garantissant une certaine qualité de service. Les protocoles de routage ad hoc existants ne conviennent pas tels quels aux réseaux FANETs.

Ce mémoire s'inscrit dans le cadre des défis liés au routage avec qualité de service dans les réseaux FANETs, et propose une solution adaptée. Dans ce travail, nous proposons une amélioration du protocole OLSR standard, conçu pour les réseaux mobiles ad hoc.

L'objectif principal est d'optimiser l'efficacité énergétique, de renforcer la stabilité des connexions et de réduire la perte de paquets de données. Une nouvelle méthode innovante de sélection des MPRs a été proposée, afin de choisir des relais multipoints (MPRs) stables et durables. Cette méthode sélectionne les MPRs selon trois critères : le niveau d'énergie résiduelle, la variance des positions des nœuds, et le nombre de voisins qu'un nœud MPR peut couvrir. Les résultats de simulation ont montré que notre approche permet de réduire significativement la perte de paquets et d'offrir des chemins plus stables et plus fiables. Nous suggérons, dans le futur, d'implémenter notre protocole dans un environnement réel afin d'évaluer son efficacité sur le terrain.

### **Organisation du mémoire :**

Le premier chapitre présente les concepts généraux liés aux réseaux ad hoc, en mettant l'accent sur les FANETs, leurs caractéristiques distinctives, leur architecture, ainsi que leurs schémas de mobilité et d'échange de données. Le deuxième chapitre est consacré aux protocoles de routage avec qualité de service (QoS) dans les FANETs, avec une étude détaillée des protocoles proactifs et réactifs. Enfin, le troisième chapitre justifie le choix du protocole OLSR et décrit la démarche proposée, en fournissant une explication approfondie de l'environnement de simulation, du matériel et des outils logiciels utilisés, ainsi que des résultats obtenus. Une conclusion générale récapitule les contributions de ce travail et propose des pistes pour des recherches futures.

**Chapitre 1**  
*Généralités sur les réseaux*  
*FANET*

# Chapitre 1

## Généralités sur les réseaux FANET

### 1.1 Introduction

De nos jours, avec les avancées technologiques, les réseaux ad hoc représentent un progrès significatif dans le domaine des communications, en offrant des solutions flexibles ne nécessitant pas d'infrastructure fixe. Plusieurs sous-classes de réseaux ad hoc ont émergé et sont largement utilisées, notamment les MANET, VANET et FANET, conçues pour des contextes où l'installation d'infrastructures est complexe, voire impossible. Bien qu'ils partagent des caractéristiques communes telles que la mobilité et l'autonomie, chaque type de réseau présente ses propres spécificités.

Ce chapitre est consacré à l'étude des principales catégories de réseaux ad hoc, en mettant en évidence leurs caractéristiques distinctives.

### 1.2 Réseau Ad-hoc

Un réseau ad-hoc est un réseau sans fil capable de rendre transparentes aux utilisateurs mobiles les modifications de topologie qu'il subit. Autrement dit, un réseau ad hoc comprend des plates-formes mobiles (par exemple, un routeur interconnectant différents hôtes et équipements sans fil) appelées nœuds qui sont libres de se déplacer sans contrainte. Un réseau ad hoc est donc un système autonome de nœuds mobiles. Ce système peut fonctionner de manière isolée ou s'interfacer à des réseaux fixes au travers de passerelles. Dans ce dernier cas, un réseau

ad hoc est un réseau d'extrémité [1].

Les nœuds (ou dispositifs) de ce type de réseau sans infrastructure fixe communiquent directement entre eux sans nécessiter de point d'accès centralisé (un routeur ou un serveur). Ils fonctionnent de manière autonome et peuvent se connecter et se déconnecter dynamiquement. Chaque nœud agit non seulement comme un hôte, mais aussi comme un routeur pour transférer les données des autres nœuds, permettant ainsi une communication flexible et résiliente, même dans des environnements où une infrastructure de réseau conventionnelle est absente ou peu pratique à déployer.

## 1.3 Les types de réseaux Ad-hoc

### 1.3.1 VANET (Vehicular Ad-hoc Network)

VANET est un type de réseau ad hoc pour les communications entre les véhicules et l'infrastructure routière environnante, qui permet les communications au sein d'un groupe de véhicules à portée les uns des autres [2]. Ce réseau est destiné à améliorer la sécurité routière, optimiser la circulation et offrir des services aux conducteurs et passagers. Dans un VANET, les véhicules communiquent directement entre eux (communication véhicule-à-véhicule, ou V2V) ou avec des points d'accès fixes situés le long des routes (communication véhicule-à-infrastructure, ou V2I).

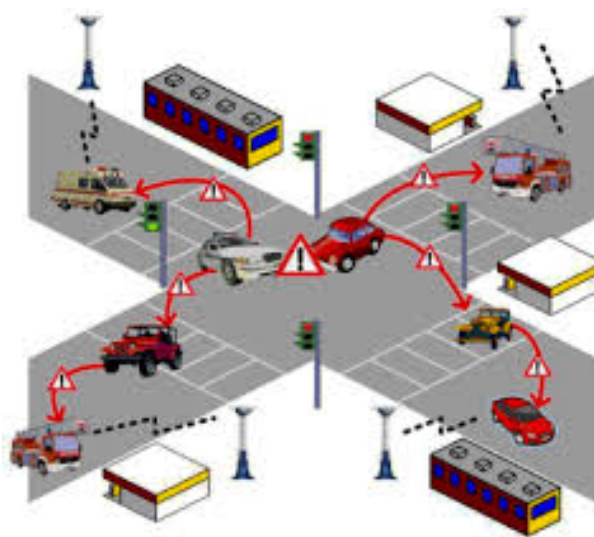


FIGURE 1.1 – Réseau VANET [2]

### 1.3.2 MANET (Mobile Ad-hoc Network)

MANET est un ensemble de nœuds mobiles reliés entre eux par des connexions sans fil et dont la gestion de la communication s'effectue d'une façon autonome sans administration centralisée [3]. MANET est un type de réseau sans fil autonome où les nœuds (généralement des dispositifs mobiles tels que des smartphones, des ordinateurs portables ou des capteurs) communiquent directement entre eux sans l'aide d'une infrastructure réseau fixe, comme des routeurs ou des points d'accès centralisés. Les MANET sont particulièrement utiles dans des environnements où une infrastructure traditionnelle est absente, endommagée ou impraticable à déployer.

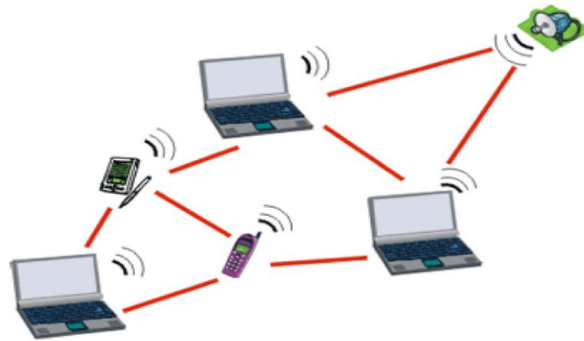


FIGURE 1.2 – Réseau MANET

### 1.3.3 FANET

Les FANET désignent des réseaux constitués de nœuds mobiles aériens, généralement appelés drones. Un drone est un véhicule aérien sans pilote, pouvant être contrôlé à distance ou télécommandé par un opérateur. Un réseau FANET a généralement une mission spécifique, à laquelle tous les nœuds collaborent pour en assurer la réalisation. Une station de base a pour rôle de guider les drones dans leurs trajets et de collecter les informations acquises par ces derniers. Les nœuds FANET communiquent entre eux dans les airs, ils transfèrent les données et les signaux sans intervention humaine et sans aucune connexion physique entre les nœuds [4].

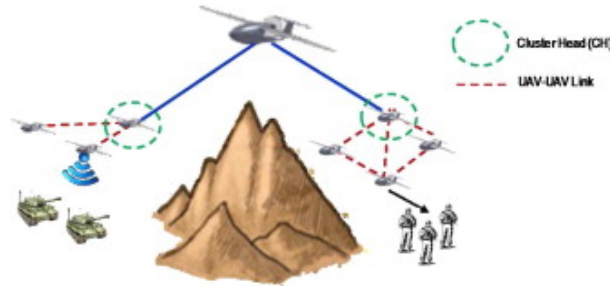


FIGURE 1.3 – Réseau FANET [5]

## 1.4 Caractéristiques des FANETs

- **Mobilité 3D élevée** : Dans les FANETs, les nœuds se déplacent dans un espace tridimensionnel avec des vitesses importantes, beaucoup plus élevées que dans les MANETs ou VANETs. Cette mobilité rapide et imprévisible implique que la topologie change constamment, ce qui nécessite des modèles de mobilité adaptés pour prévoir les trajectoires [6]. Ces déplacements rendent les communications sensibles aux pertes de liaison et nécessitent des protocoles capables de suivre efficacement ces changements.[7].
- **Topologie dynamique** : La structure du réseau évolue en permanence : les drones entrent et sortent de portée les uns des autres, compliquant le maintien des routes et la délivrance fiable des paquets. Les protocoles classiques conçus pour MANET ou VANET ne sont pas suffisants. Il faut des protocoles de routage très adaptatifs (géoroutage, prédiction, renforcement, IA...) pour s'ajuster à la topologie changeante en 3D[8]
- **Ressources limitées** : Les UAV sont souvent limités en batterie, capacités de calcul et mémoire embarquée. Le routage, le contrôle, la communication et le codage doivent être légers en consommation énergétique [9]. Par exemple, le protocole HIROL combine algorithme Abeille, DSR, OLSR et réseaux neuronaux pour minimiser la surcharge tout en maintenant de bonnes performances, montrant que l'optimisation énergétique est un enjeu majeur[10].
- **Couverture étendue** : Grâce à leur capacité à voler à différentes altitudes, les UAVs dans les FANETs peuvent couvrir des zones beaucoup plus vastes que les réseaux terrestres classiques [11]. Cette capacité est particulièrement précieuse dans les environnements difficiles d'accès (zones rurales, sinistrées ou montagneuses), là où les infrastructures réseau sont absentes ou détruites [12].

La couverture aérienne permet une visibilité directe entre les nœuds et réduit les obstacles physiques (bâtiments, reliefs), ce qui améliore les performances de communication[12]. Toutefois, cette étendue nécessite une gestion dynamique de la topologie pour maintenir la connectivité sur de longues distances [13].

- **Communication sans infrastructure** : Les FANETs fonctionnent de manière autonome, sans infrastructures fixes comme celles des réseaux cellulaires ou satellites. La communication s'effectue de drone à drone (multi-sauts) ou entre drone et station sol, selon les applications . Cette approche offre une grande flexibilité et autonomie, utile en zones sinistrées ou isolées, mais exige aussi des protocoles robustes pour maintenir la connectivité sans point central .

## 1.5 Architecture des FANET

- **Nœuds aériens**

Les drones sont les principaux nœuds du réseau FANET. Ils sont mobiles et interconnectés sans infrastructure préexistante. Chacun de ces drones peut être équipé de capteurs pour collecter des informations (vidéo, température, humidité, etc.) et les transmettre à la station de base. Les drones échangent les informations via des liaisons sans fil (Wi-Fi, LTE, ou autres technologies spécifiques) [14].

- **Topologie du réseau**

La topologie du réseau FANET fait référence à la structure spatiale et logique des liaisons entre les drones. À l'inverse des réseaux fixes, la topologie d'un FANET est très dynamique, puisqu'elle change en fonction des mouvements des drones. La topologie d'un réseau FANET est un facteur clé influençant ses performances. En pratique, le contrôle de la topologie d'un FANET se divise en deux phases : 1- la construction de la topologie et 2- son ajustement. La construction désigne l'établissement initial d'une structure logique adaptée aux besoins de la mission, tandis que l'ajustement fait référence à la modification dynamique de cette topologie [15]. On identifie généralement plusieurs modèles de topologies possibles dans un réseau FANET.

- **Topologie en étoile** : Un drone central fait fonction de point de coordination, et les drones restants assurent principalement leur communication à travers lui. Cette méthode est simple mais repose fortement sur le drone central, ce qui peut constituer

un enjeu de risque. Un drone central agit comme point de coordination, les autres drones communiquant principalement à travers lui [16].

- **Topologie hiérarchique ou en grappes (cluster-based)** : Les drones sont organisés en clusters, chaque grappe étant supervisée par un « drone chef » (cluster head) qui se charge des échanges locaux et transmet les données aux autres grappes. Cette structure facilite une gestion plus efficace de la mobilité et de la consommation énergétique. Les drones sont organisés en clusters [16].

La topologie choisie ou qui émerge automatiquement influence directement les performances du réseau en termes de latence, de robustesse, de bande passante et de consommation énergétique. Les protocoles de communication doivent donc être capables de s'adapter en temps réel à ces changements topologiques.

- **Protocoles de communication**

Le protocole de routage dans FANET est essentiel pour assurer que les informations atteignent leur destination malgré les mouvements constants des drones. Il peut s'agir de protocoles spécifiques comme OLSR (Optimized Link State Routing) adaptés à la mobilité aérienne. Les protocoles de routage dans les FANET présentent des exigences différentes de celles des réseaux ad hoc traditionnels, telles qu'une faible densité de nœuds, une mobilité élevée et des changements fréquents de topologie. Concevoir un protocole de routage pour les FANET en tenant compte de ses caractéristiques est un défi majeur [17].

- **Gestion de la mobilité**

La gestion de la mobilité dans FANET est plus complexe que dans les réseaux traditionnels, car les drones se déplacent rapidement. Des algorithmes de gestion de la mobilité sont donc utilisés pour assurer la stabilité du réseau et la connectivité entre les drones. Certains systèmes FANET intègrent des mécanismes pour coordonner les trajectoires des drones afin d'éviter les collisions et optimiser les performances du réseau. Un algorithme de contrôle de topologie (TCA) adapté aux essaims de drones, avec une surcharge raisonnable, permet d'optimiser à la fois les performances de la mission et de la communication dans un réseau FANET. Ainsi, le TCA assure une couverture plus large et garantit la qualité de service dans la connectivité aérienne. De plus, il contribue à une gestion efficace de l'énergie, une exploration plus performante des cibles, une meilleure stabilité des formations, tout en assurant l'évitement des collisions entre

drones, l'évitement des obstacles extérieurs, une interférence réduite, et une autonomie accrue des essaims de drones [18].

- **Applications**

- Les FANET sont utilisés pour fournir une surveillance en temps réel, particulièrement dans des zones vastes ou difficiles d'accès. Un FANET, capable de collecter des données d'image à partir de drones équipés de caméras vidéo, et un système de traitement d'image permettant la collecte et l'analyse des données sont les deux composants principaux du système. Basé sur la technique de traitement d'image, un prototype pour une extraction et une amélioration efficaces des données dans des situations FANET et des services possibles est illustré [19].
- Les drones collectent des données géospatiales pour générer des cartes précises.
- En cas de catastrophe, les FANET peuvent être utilisés pour coordonner les efforts de sauvetage en temps réel. Cet article fournit une revue des activités de recherche pertinentes sur les FANET dans la gestion des catastrophes et les techniques émergentes d'intelligence artificielle [20].

## **1.6 Modèle de mobilité et de communication**

### **1.6.1 Modèle de communication**

Un modèle de communication désigne la façon dont les informations sont échangées entre différents nœuds d'un réseau. Il définit les mécanismes et les protocoles utilisés pour transmettre, recevoir et gérer les données entre les appareils [21].

#### **1.6.1.1 UAV to UAV**

Dans les réseaux de drones, les UAV échangent souvent des données directement entre eux en utilisant différents algorithmes de routage. La topologie du réseau change fréquemment en raison de l'ajout ou de la suppression de nœuds. Pour maintenir la communication entre les UAV, il est nécessaire d'explorer plusieurs itinéraires. Cela peut être accompli grâce à des protocoles de routage adaptatifs, qui permettent aux UAV de découvrir de nouveaux chemins de communication et de les mettre à jour en temps réel en fonction des changements dans le réseau. Pour répondre aux besoins de différentes missions, les UAV communiquent directement entre

eux en échangeant fréquemment des paquets de données. Toutefois, en raison des limitations de portée de transmission, la communication multi-sauts est réalisée via d'autres UAV. Cela est essentiel pour étendre la couverture d'une zone spécifique [22].

### 1.6.1.2 UAV à infrastructure

Le modèle UAV à infrastructure consiste en la communication entre un drone (UAV) et des stations au sol, comme des bases de contrôle. Dans ce modèle, les drones envoient leurs données à ces stations, ce qui permet de centraliser la gestion des informations. Cela est utile pour des missions comme la surveillance ou la gestion des urgences. Ce modèle offre une connectivité plus stable grâce aux stations au sol, mais dépend de la couverture du réseau. Si le drone s'éloigne trop ou rencontre des obstacles, la connexion peut être perturbée. Pour un meilleur contrôle des UAV en vol, des infrastructures sous forme de stations de base sont installées au sol afin d'échanger des messages critiques de commande et de contrôle [22].

## 1.6.2 Modèle de mobilité

Un modèle de mobilité représente les déplacements des nœuds d'un réseau, c'est-à-dire les variations de leur position, vitesse et accélération au fil du temps dans un scénario délimité. La mobilité permet aux nœuds de s'adapter aux besoins spécifiques de chaque application, offrant ainsi un grand dynamisme et une meilleure performance selon le modèle de mobilité choisi [23]. Les modèles de mobilité jouent un rôle essentiel dans la simulation du comportement des drones au sein des réseaux FANET. Ils influencent directement les performances du réseau, en particulier en termes de routage, de latence et de gestion des ressources. Parmi les modèles de mobilité les suivants :

**Random Way Point** Le modèle Random Waypoint (RWP) est largement utilisé pour simuler la mobilité des nœuds dans les réseaux ad hoc comme les FANETs. Dans ce modèle, chaque nœud choisit aléatoirement un point de destination et une vitesse, s'y déplace, fait une pause, puis recommence, voir Figure (1.4). Ce modèle est simple et permet de simuler un

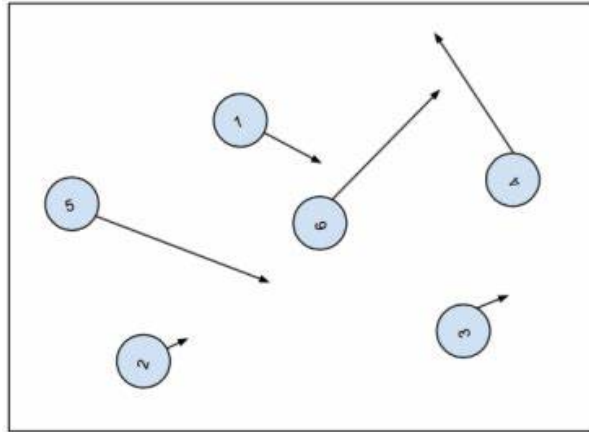


FIGURE 1.4 – Point de passage aléatoire (RWP) [24]

**Gauss-Markov Mobility (GMM)** Le modèle de mobilité Gauss-Markov permet de simuler des déplacements plus réalistes en introduisant une dépendance entre les mouvements successifs des nœuds. Dans ce modèle, l'orientation et la rapidité des nœuds dépendent de leur condition précédente, ce qui permet aux nœuds de suivre des trajectoires continues et prévisibles. Le modèle Gauss-Markov est particulièrement utilisé dans des scénarios où la coordination et la fluidité des déplacements des nœuds sont essentielles, comme dans les FANET.

Le modèle de mobilité Gauss-Markov utilise un seul paramètre d'ajustement pour accommoder différents niveaux d'aléatoire. Dans ce modèle, les mouvements des nœuds sont indépendants les uns des autres. Les nœuds se voient attribuer une direction et une vitesse initiales, et à intervalles constants, ces deux variables sont mises à jour. La vitesse et la direction précédentes de chaque nœud sont prises en compte pour les valeurs futures, avec un niveau d'aléatoire intégré [25].

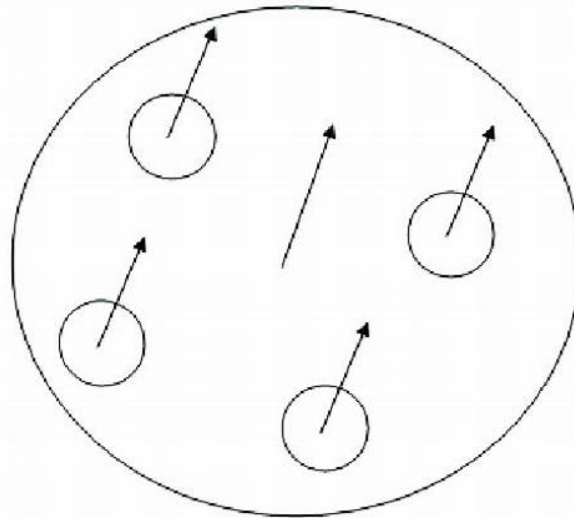


FIGURE 1.5 – Modèle de mobilité Gauss-Markov. [4]

## 1.7 Comparaison MANET, VANET et FANET

Le tableau suivant présente les différences et les similitudes entre les trois types de réseaux : MANET, VANET et FANET.

TABLE 1.1 – Comparaison entre fanet, manet et vanet .[26]

Critère	fanet	MANET	VANET
Définition	Réseau ad hoc avec des drones (véhicules aériens sans pilote)	Réseau ad hoc avec des nœuds mobiles (smartphones, ordinateurs)	Réseau ad hoc avec des véhicules en mouvement
Environnement	Aérien (drones volant à haute altitude)	Terrestre (zones urbaines, rurales, de guerre, etc.)	Routier (autoroutes, zones urbaines)
Mobilité des nœuds	Mobilité 3D (verticale et horizontale)	Mobilité 2D (terrestre, mais avec des déplacements variés)	Mobilité horizontale, principalement sur des routes
Applications principales	Surveillance aérienne, cartographie, gestion des urgences	Communication dans des environnements sans infrastructure (guerre, secours)	Sécurité routière, gestion du trafic, véhicules autonomes
Routage	Routage dynamique en 3D, complexe (prend en compte l'altitude)	Routage dynamique, protocoles comme AODV, OLSR	Routage dynamique avec protocoles comme GPSR, Geocast

## 1.8 Conclusion

En conclusion, les réseaux ad hoc (MANET, VANET et FANET) offrent des solutions flexibles et autonomes pour la communication sans infrastructure préétablie. Chaque type de réseau est adapté à des environnements et des besoins spécifiques : MANET pour des applications mobiles terrestres, VANET pour la communication entre véhicules, et FANET pour l'utilisation de drones dans des missions aériennes. Nous avons examiné dans ce chapitre l'architecture de ces types de réseaux, en particulier FANET, on effectuons une comparaison entre eux.

**Chapitre 2**  
*Le Routages avec QoS dans  
FANET*

# Chapitre 2

## Le Routages avec QoS dans les FANETs

### 2.1 Introduction

Dans les réseaux de drones, le routage joue un rôle très important pour permettre aux appareils de bien communiquer entre eux. À cause des mouvements fréquents et rapides des drones, il est nécessaire d'utiliser des protocoles de routage capables de s'adapter rapidement.

Ce chapitre présente les principales approches de routage utilisées dans les FANETs (proactif et réactif). Il aborde aussi la notion de Qualité de Service (QoS : Quality of Service) et les critères essentiels pour assurer une bonne performance du réseau, comme l'énergie, la stabilité ou encore le délai.

### 2.2 Routage dans les reseaux FANETs et leur défis

Le routage dans les réseaux FANETs est crucial pour assurer une communication fiable et efficace entre les UAVs [27]. Étant donné la nature dynamique et mobile de ces réseaux, les protocoles de routage doivent s'adapter rapidement aux changements fréquents de topologie [28]. Le routage dans les réseaux FANETs est un processus essentiel permettant aux drones de transmettre des données malgré les changements fréquents de la topologie du réseau. Différentes approches de routage sont utilisées pour s'adapter aux défis posés par ces réseaux .Le routage dans les FANETs repose sur plusieurs approches :

- Le routage basé sur la **topologie** utilise des tables de routage pour gérer les connexions

entre les noeuds, soit de manière proactive en maintenant constamment des informations de route, soit de manière réactive en découvrant les routes à la demande. - Le routage basé sur la **localisation** exploite les coordonnées GPS des UAVs pour optimiser l'acheminement des paquets et réduire les interruptions de connexion. - L'approche de routage axée sur les données ne repose pas sur des tables de routage statiques. Elle offre aux nœuds consommateurs la possibilité de solliciter des données spécifiques grâce à des messages d'abonnement, ce qui est particulièrement approprié pour les applications de surveillance et de collecte d'informations.

Dans les FANETs, le défi majeur du routage consiste à améliorer l'acheminement des informations selon divers indicateurs de performance, comme la stabilité des connexions, le temps de latence, la saturation du réseau et l'efficacité énergétique. Une bonne stratégie de routage doit donc s'adapter aux changements fréquents de topologie et garantir une communication fluide et fiable entre les drones et les autres entités du réseau. Les protocoles de routage sont responsables de la découverte, de la création et du maintien des routes de communication entre deux nœuds. La surcharge et l'utilisation de la bande passante de ces protocoles doivent être réduites au minimum. En raison de la forte mobilité des UAV, la topologie du réseau peut varier dans le temps, rendant la découverte et le maintien des routes comme l'un des défis majeurs à relever [29].

### 2.3 Classification des protocoles de routages FANETs

Afin d'expliquer les protocoles de routage pour les FANETs, nous présentons une classification proposée dans [30], illustrée à la Figure 2.1.

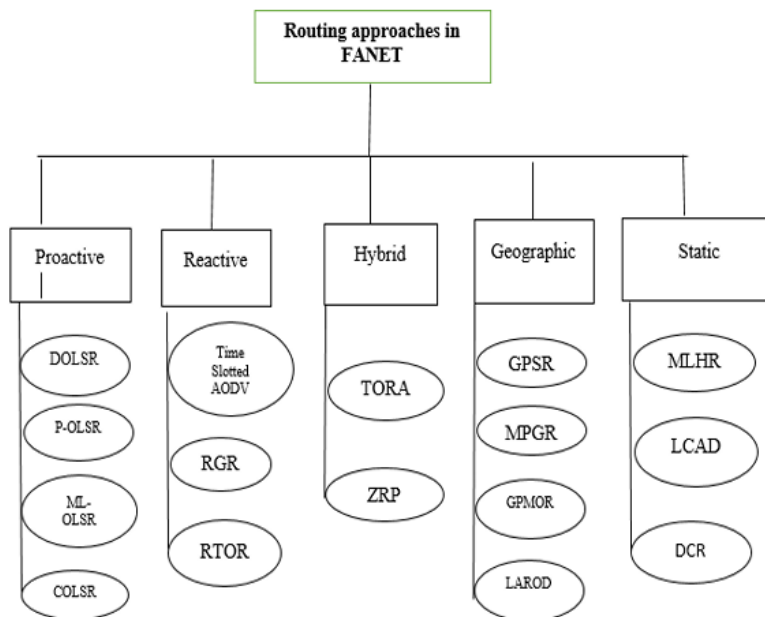


FIGURE 2.1 – Routage Dans les reseaux Fanets.[30]

### 2.3.1 Protocoles de routages Réactif

Les techniques de routage réactives, également appelées le routage sur demande, adoptent une approche très différente par rapport aux protocoles proactifs. Une grande partie des frais de contrôle des protocoles proactifs proviennent du besoin de chaque nœud de maintenir un itinéraire vers chaque autre nœud du réseau et à tout moment [31]. Un protocole de routage réactif est un type de protocole qui établit une route uniquement lorsque cela est nécessaire,

#### 2.3.1.1 RGR (Reactive Greedy Routing)

Le protocole RGR (Reactive Greedy Routing) est une solution de routage optimisée pour les FANETs (Flying Ad-Hoc Networks), proposée par [32]. Contrairement aux approches traditionnelles, RGR utilise une méthode hybride réactive-géographique qui permet :

- Une découverte de route **à la demande** (réactive), réduisant la surcharge du réseau,
- Une sélection du prochain saut basée sur la **position géographique** (mode "greedy"),
- L'élimination des tables de routage permanentes, adaptée aux topologies dynamiques des drones.

**Améliorations clés** (d'après [32]) :

- Gestion des **"trous de routage"** par mécanisme de récupération,

- Adaptation à la **mobilité 3D** des drones,
- Optimisation énergétique pour les UAVs.

### 2.3.1.2 RTOR (Reliable Topology and Opportunistic Routing)

Le protocole RTOR (Reliable Topology and Opportunistic Routing) est un protocole réactif conçu spécifiquement pour les réseaux FANET afin d'assurer une communication fiable dans des environnements hautement dynamiques. Il combine deux stratégies complémentaires : la sélection d'une topologie stable basée sur la prédiction de la mobilité et l'acheminement opportuniste qui permet de choisir dynamiquement le meilleur nœud de relais à chaque transmission. Cette approche réduit la fréquence des reconstructions de routes et améliore la tolérance aux ruptures de liens. RTOR évalue les liens à l'aide d'un indicateur de fiabilité basé sur les positions GPS, la direction et la vitesse relative des UAVs [33]. Grâce à cette double stratégie, il offre de meilleures performances en termes de délai de transmission, taux de livraison de paquets et robustesse du réseau.

### 2.3.2 Protocole de routage Proactif

Les approches proactives de routage, conçues pour les réseaux ad hoc, sont dérivés des protocoles traditionnels du vecteur d'état et à vecteur de distance développés pour l'usage dans les réseaux filaires. La caractéristique primaire des approches proactives est que chaque nœud dans le réseau maintient un itinéraire à tous les autres nœuds dans le réseau à tout moment[31].c'est un protocole de routage qui fournissent constamment une mise à jour des tables de routage

#### 2.3.2.1 P-OLSR(OLSR Améliorée)

Le protocole P-OLSR intègre des mécanismes de prédiction basés sur les informations de localisation et les vecteurs de vitesse des nœuds pour anticiper les changements de topologie du réseau. En estimant la durée de vie des liens entre les nœuds, le protocole sélectionne des routes plus stables, améliorant ainsi la fiabilité et la performance du réseau, notamment en réduisant les ruptures de lien et la latence[34] Le protocole P-OLSR (Predictive Optimized Link State Routing) représente une amélioration significative du protocole OLSR, spécialement conçue pour les environnements caractérisés par une forte mobilité, tels que les réseaux ad hoc aériens (FANETs). Alors qu'OLSR repose principalement sur la connectivité instantanée entre les nœuds, P-OLSR intègre une approche prédictive en tenant compte de paramètres de mobilité

comme la position GPS, la vitesse et la direction des nœuds. Cette capacité d'anticipation des changements topologiques permet de privilégier des routes plus stables et durables, réduisant ainsi le risque d'interruptions fréquentes. En adaptant dynamiquement la sélection des relais multipoints (MPR) en fonction de la stabilité prévisible des liens, P-OLSR diminue le besoin de reconstructions fréquentes de routes, limite la surcharge de contrôle et améliore la qualité de service (QoS) dans des contextes réseau hautement dynamiques. Ce protocole se distingue ainsi comme une solution robuste et performante pour assurer des communications fiables dans les réseaux mobiles modernes.

### 2.3.2.2 ML-OLSR

Le protocole ML-OLSR (Machine Learning-based Optimized Link State Routing) est une version améliorée du protocole proactif OLSR, spécifiquement conçue pour les réseaux FANET. Il utilise des techniques d'apprentissage automatique pour prédire la stabilité des liens et sélectionner les meilleurs relais dans des environnements où la topologie varie rapidement. Contrairement au OLSR classique qui repose uniquement sur des métriques comme la qualité de lien, ML-OLSR intègre des caractéristiques comme la vitesse relative, la position, l'altitude et la direction de mouvement des UAVs. Ces données sont exploitées par un modèle de machine learning (régression logistique) pour estimer la probabilité de stabilité d'un lien. En adaptant dynamiquement les MPR (MultiPoint Relays), ML-OLSR améliore la fiabilité et la réactivité du réseau dans les scénarios aériens critiques [35].

## 2.4 La Qualité de service

La Qualité de Service l'effet global de la performance d'un service qui détermine le degré de satisfaction d'un utilisateur vis-à-vis de ce service [36]. En effet, on définit la qualité de service comme l'impact global de la performance du service qui influence sur degré de satisfaction d'un utilisateur envers le système [27]. c'est l'ensemble des mécanismes et des techniques permettant d'assurer un niveau de performance optimal pour la transmission des données, elle englobe plusieurs paramètres comme la latence, l'énergie, la stabilité du réseau, ...etc. Une bonne QoS garantit une transmission efficace des données, surtout dans les réseaux critiques comme les FANETs.

## 2.5 Les métriques de qualité du service

Plusieurs métriques de qualité de service peuvent être exigées par les applications. Les principales sont les suivantes

- La Bande Passante : quantité de données pouvant être transmises sur le réseau par unité de temps.[37].
- Perte de Paquets : mesure la fiabilité de la transmission des données [38]
- temps nécessaire pour qu'un paquet atteigne sa destination depuis la source[37]

## 2.6 Conclusion

Les réseaux FANETs présentent des défis en matière de routage en raison de la mobilité élevée des drones et de la topologie dynamique et en 3D. Dans ce chapitre, nous analysons les principales approches de routage (proactif, réactif) et leurs adaptations spécifiques aux FANETs, en mettant l'accent sur des protocoles .

**Chapitre 3**  
*Contribution et Validation*

# Chapitre 3

## Contribution et Validation

### 3.1 Introduction

Le problème du routage avec qualité de service constitue un défi majeur dans les FANETs. Dans ce cadre, nous proposons dans ce chapitre une nouvelle approche de routage visant à garantir une transmission de données efficace au sein des FANETs

### 3.2 Approche Proposée

Au lieu de proposer un nouveau protocole, nous avons choisi de modifier un protocole standard bien connu de la communauté Ad hoc.

#### 3.2.1 Choix du Protocole OLSR

Après une analyse approfondie des différents protocoles de routage utilisés dans les réseaux ad hoc, notre choix s'est porté sur le protocole OLSR (*Optimized Link State Routing*). Ce choix est motivé par sa structure proactive, qui garantit une disponibilité constante des routes, ainsi que par sa robustesse face aux changements fréquents de topologie, caractéristiques des environnements mobiles.

## 3.2.2 Le Mécanisme de Sélection des MPRs

### 3.2.2.1 La Méthode de Sélection Standard

Dans le protocole OLSR (*Optimized Link State Routing*), la sélection des relais multipoints (MPRs) repose sur une heuristique visant à réduire la redondance des messages tout en assurant une couverture complète des nœuds à deux sauts.

#### La méthode se déroule en plusieurs étapes :

- **Sélection immédiate par meilleur :** Les nœuds à un saut N1 ayant une volonté maximale (`WILLINGNESS_ALWAYS`) sont sélectionnés en priorité comme MPRs.
- **Couverture directe des N2 :** Pour chaque nœud N1, le protocole évalue le nombre de voisins à deux sauts (N2) qu'il permet de couvrir. Les nœuds N1 qui assurent une liaison directe vers des N2 encore non couverts sont ajoutés à l'ensemble MPR, et les N2 couverts sont retirés de la liste à couvrir.
- **Sélection itérative basée sur l'accessibilité :** Tant que certains N2 restent non couverts, les nœuds N1 restants sont évalués selon :
  - Leur volonté (`WILL`),
  - Leur accessibilité (nombre de N2 qu'ils peuvent couvrir),
  - Leur degré (nombre de voisins à 1 saut),
  - Et, en cas d'égalité, leur accessibilité est à nouveau utilisée comme critère final.
- **Phase d'optimisation :** Une phase finale tente de réduire l'ensemble MPR en supprimant les nœuds redondants, à condition que la couverture complète des N2 soit toujours assurée.

#### Limites :

- Ne tient pas compte de la mobilité, ni de l'énergie résiduelle.
- Susceptible de sélectionner des nœuds instables ou bientôt hors de portée (*reachability*).
- Risque accru de pertes de paquets et de recalculs fréquents.

### 3.2.2.2 La Méthode de Sélection des MPR Proposée

Afin de pallier les limites du protocole OLSR standard présenté précédemment, et d'améliorer l'efficacité ainsi que les performances dans des scénarios à haute mobilité, nous proposons une approche innovante de sélection des MPRs, basée sur trois critères :

1. **Reachability (nombre de voisins à 2 saut) :** Ce critère garantit la couverture des nœuds à deux sauts, comme dans la méthode standard (nous avons conservé la structure reachability. Ces nœuds sont accessibles via les MPRs choisis. Il assure la couverture nécessaire au bon fonctionnement du routage.
2. **Stabilité du lien :** Ce paramètre est calculé en déterminant la variance des distances entre un nœud et chacun de ses voisins à un saut. Une faible variance indique un lien plus fort et plus stable (faible variance = haute stabilité).

- **Calcul de la moyenne des distances :** La moyenne est calculée comme suit :

$$\text{mean} = \frac{1}{n} \sum_{i=1}^n d_i$$

où  $d_i$  est la  $i$ -ème distance mesurée, et  $n$  le nombre total de mesures.

- **Calcul de la variance :** La variance est obtenue par :

$$\text{variance} = \frac{1}{n} \sum_{i=1}^n (d_i - \text{mean})^2$$

Ainsi, la variance correspond à la moyenne des carrés des écarts de chaque distance par rapport à la moyenne. Une variance faible indique une variation réduite des distances, signe d'une connexion stable.

**Exemple explicatif :**

Considérons un nœud local qui enregistre les distances avec un de ses voisins à 1 saut sur 10 instants. Nous utilisons les 10 derniers éléments les plus récents pour calculer la moyenne. Voici les mesures :

TABLE 3.1 – Mesures de distance sur 10 instants

Mesure $i$	Distance $d_i$ (m)
1	9,5
2	10,0
3	10,2
4	9,8
5	10,1
6	9,7
7	10,3
8	9,9
9	10,0
10	9,6

**Étape 1 : Calcul de la moyenne**

$$\text{mean} = \frac{1}{10} \sum_{i=1}^{10} d_i = \frac{9,5 + 10,0 + 10,2 + 9,8 + 10,1 + 9,7 + 10,3 + 9,9 + 10,0 + 9,6}{10} = \frac{99,1}{10} = 9,91$$

**Étape 2 : Calcul des écarts à la moyenne et des carrés**

TABLE 3.2 – L'écart de la moyenne et carées

$d_i$	$d_i - \text{mean}$	$(d_i - \text{mean})^2$
9,5	-0,41	0,1681
10,0	0,09	0,0081
10,2	0,29	0,0841
9,8	-0,11	0,0121
10,1	0,19	0,0361
9,7	-0,21	0,0441
10,3	0,39	0,1521
9,9	-0,01	0,0001
10,0	0,09	0,0081
9,6	-0,31	0,0961

**Étape 3 : Calcul de la variance**

$$\text{variance} = \frac{1}{10} \sum_{i=1}^{10} (d_i - \text{mean})^2 = \frac{0,609}{10} = 0,0609$$

**Interprétation :** Une faible variance ( $\approx 0,06$ ) indique que les distances varient peu autour de la moyenne, reflétant une connexion stable. Ce résultat sera intégré dans l'algorithme de choix des MPRs : une variance plus faible indique un nœud plus fiable et donc préféré.

- Énergie résiduelle :** Ce critère, permet d'estimer l'énergie résiduelle des nœuds. En choisissant des MPRs ayant une capacité énergétique optimale, on contribue à prolonger la durée de vie globale du réseau.

### 3.2.2.3 Format de Paquet

Pour récupérer l'énergie résiduelle de chaque nœud avec leurs positions (X, Y et Z), nous avons modifié le format standard du paquet HELLO afin de répondre à nos besoins. Nous avons ajouté quatre champs au format standard du paquet HELLO :

- **Énergie** : Ce champ de 16 bits détermine l'énergie actuelle du nœud.
- **XPosition** : Détermine la position du nœud sur l'axe X (16 bits).
- **YPosition** : Détermine la position du nœud sur l'axe Y (16 bits).
- **ZPosition** : Détermine la position du nœud sur l'axe Z (16 bits).

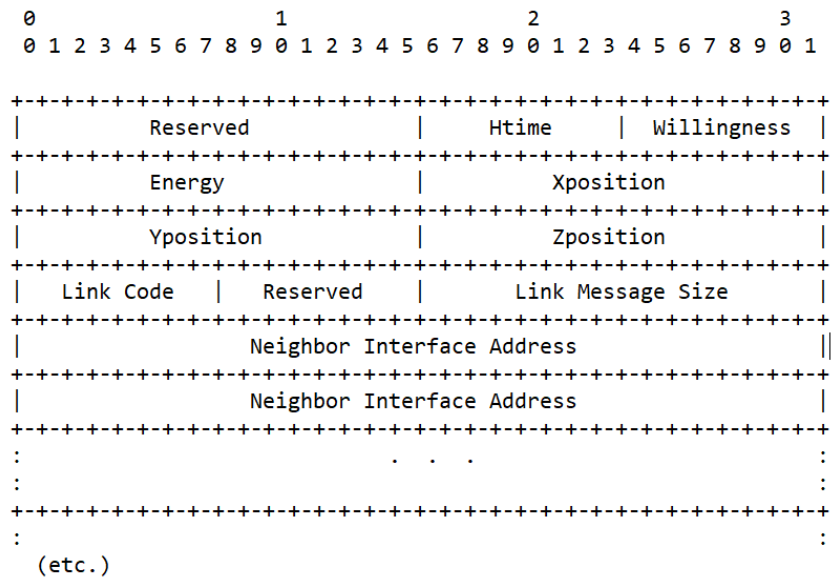


FIGURE 3.1 – Format de Paquet Nouveau

### 3.2.2.4 Fonction Objectif

Dans notre approche, le choix des MPRs se base sur trois critères : l'énergie résiduelle, la stabilité des liens et la portée de connectivité (reachability). Pour cela, nous avons proposé une fonction de coût qui combine ces trois critères, ajustés en fonction de leur pertinence :

$$\text{Coût}(N_i) = \alpha \times \text{Reachability} + \beta \times \text{Stabilité} + \gamma \times \text{Énergie} \quad (3.1)$$

$$\text{Stabilité} = 1/\text{variance} \quad (3.2)$$

Avec les poids  $\alpha$ ,  $\beta$  et  $\gamma$ , tels que :

- $\alpha$  : Priorité à la *reachability* (couverture efficace des  $N_2$ ),
- $\beta$  : Accordée à la stabilité des liens,
- $\gamma$  : Prise en compte de l'énergie.

De sorte que :  $\alpha + \beta + \gamma = 1$ .

### 3.2.2.5 Algorithme de Sélection des MPR Proposé

**Entrées :**

- $N_1$  : Ensemble des voisins à un saut.
- $N_2$  : Ensemble des voisins à deux sauts à couvrir.
- mprSet : Ensemble des MPRs sélectionnés.
- Pour chaque nœud  $n_i \in N$  :
  - energy : Énergie résiduelle.
  - variance : Variance des distances entre le nœud local et  $n_i$ .
  - reachability : Le nombre de voisins à 2 sauts couvert par un voisin à 1 saut.
- Les coefficients :
  - Double  $\alpha = 0,3$
  - Double  $\beta = 0,3$
  - Double  $\gamma = 0,4$

**Étapes de l'algorithme :**

1. MPR\_sélectionnée  $\leftarrow$  NULL

2. coût\_meilleur  $\leftarrow$  0

3. **Tant que** ( $N_2 \neq \emptyset$ ) **faire**

3.1 **Pour chaque** voisin  $n_i \in N_1$  **faire**

3.1.1 Calculer son degré de *reachability*  $r_i$

3.1.2 Récupérer son énergie résiduelle

3.1.3 Calculer sa variance

3.1.4 Calculer la fonction de coût :

$$\text{Coût}(n_i) = \alpha \cdot \text{reachability}(n_i) + \beta \cdot \text{variance}(n_i) + \gamma \cdot \text{energy}(n_i)$$

3.1.5 **Si**  $\text{Coût}(n_i) < \text{coût\_meilleur}$  **alors**

$$\text{coût\_meilleur} \leftarrow \text{Coût}(n_i)$$

$$\text{MPR\_sélectionnée} \leftarrow n_i$$

### 3.2 Fin Pour

3.3 Ajouter MPR\_sélectionnée à mprSet

3.4 Supprimer de  $N_2$  les nœuds couverts par MPR\_sélectionnée

## 4. Fin Tant que

**Exemple explicatif :** Dans cet exemple, nous considérons qu'un nœud "A" à 5 voisins à 1 saut et plusieurs voisins à 2 saut. Nous voulons calculer les MPR de "A" parmi la liste des voisins à un saut.

### Paramètre de simulations :

- **Poids utilisés dans la formule du score :**
  - $\alpha = 0,3$  : accessibilité (*reachability*)
  - $\beta = 0,3$  : variance des distances (inversée)
  - $\gamma = 0,4$  : énergie résiduelle
- **Énergie initiale des nœuds :** entre 100 J et 400 J
- **Vitesse moyenne des nœuds :** 10 m/s

TABLE 3.3 – Coût des nœuds candidats au rôle d'un MPR (ces valeurs sont normaliser entre (0 et 100) )

Nœud	Reachability( $n_i$ )	Variance( $n_i$ )	Énergie( $n_i$ )	Score( $n_i$ )
$n_1$	60	50	70	61
$n_2$	90	20	30	45
$n_3$	40	100	60	66
$n_4$	80	25	90	67.5
$n_5$	50	100	80	77

Ainsi, le nœud  $n_5$  obtient le Coût le plus élevé (**77**). Donc ce nœud est sélectionné comme **MPR (MultiPoint Relay) pour le nœud  $n_5$**  .

### 3.3 Avantages Prévus de la Méthode Proposée

L'intégration des critères de QoS permet d'améliorer significativement les performances du protocole OLSR dans des environnements mobiles :

- **Réduction des recalculs de MPRs** grâce à la sélection de nœuds plus stables,
- **Optimisation de la consommation énergétique** en évitant les nœuds en fin de vie,
- **Amélioration du taux de livraison des paquets** due à une meilleure continuité des liens,
- **Diminution de la surcharge de messages de contrôle** grâce à une sélection plus efficace et durable des relais.

### 3.4 L'environnement de Travail

#### 3.4.1 La Configuration Matérielle

Les simulations ont été effectuées sur un ordinateur portable DELL, dont les principales caractéristiques techniques sont présentées ci-dessous :

TABLE 3.4 – Spécifications de l'appareil utilisé

Élément	Spécification
Marque et modèle	DELL
Système d'exploitation	ubuntu
Processeur	Intel(R) Core(TM) i5-6200U CPU @ 2.30GHz
Fréquence du processeur	2.40 GHz
Mémoire vive (RAM)	8,00 Go
Type de système	Système 64 bits, processeur x64

## 3.4.2 Les Outils et les Plateformes Logicielles

### 3.4.2.1 Ubuntu

Ubuntu est une distribution Linux open source, basée sur Debian, conçue pour être conviviale, sécurisée et régulièrement mise à jour [Réf. Web 1]. Très populaire pour les postes de travail, les serveurs et les environnements de développement. Il est largement utilisée dans les domaines de l'éducation, de la recherche et de l'ingénierie [Réf. Web 2].

### 3.4.2.2 NS-3

NS3 est un simulateur de réseaux open source utilisé pour la recherche et le développement dans le domaine des réseaux informatiques. Cet outil propose la fonctionnalité de modéliser, simuler et étudier le comportement des protocoles de communication à travers divers environnements réseau [Réf. Web 4]

## 3.4.3 Le Langage de Développement

### 3.4.3.1 C++

Le langage C++ est apparu au début des années 90 et figure aujourd'hui parmi les langages de programmation les plus répandus à l'échelle mondiale. Utilisé dans des domaines allant des applications scientifiques au développement de logiciels commerciaux, il constitue une évolution du langage C.

Le C++ se distingue par son efficacité, tout en offrant des fonctionnalités avancées comme la programmation orientée objet, permettant la création de programmes modulaires, réutilisables et faciles à maintenir [Réf. Web 3]

En C++, les programmes sont généralement divisés en deux types de fichiers :

- Fichier d'en-tête (.h) : Ils déclarent les éléments du programme comme les classes et les fonctions.
- Fichier source (.cc) : Ils contiennent le code qui définit la logique des classes et les fonctions.

## 3.5 Résultats de Simulation

Dans cette section, nous allons présenter un ensemble de figures qui illustre les résultats de notre protocole baptisé **SE\_OLSR** (Stabilité\_Energie OLSR).pour voir l'efficacité SE\_OLSR nous l'avons comparer avec protocole standard OLSR.

### 3.5.1 Paramètre de simulation

TABLE 3.5 – Paramètres de simulation

Paramètres	Valeur
Modèle de mobilité	Gauss_Markov
Dimension de Topologie	1000*1000*300 (m)
Le débit	8192 (Mbits/s)
Nombre de nœuds	50, 60,70
Temps de simulation	200.0 (s)
Énergie initiale	100 - 500 (J)
Puissance de transmission	7.5 (dBm)

### 3.5.2 Analyse Des resultats

Dans cette section, nous présentons les résultats de simulation obtenus pour notre protocole SE\_ OLSRet nous les comparons à ceux obtenus pour le protocole OLS\_STANDARD. Pour les paramètres de simulation, nous considérons :

- **Énergie** : Cette métrique représente le niveau d'énergie résiduelle des nœuds. Elle permet d'évaluer l'efficacité énergétique du protocole, en analysant sa capacité à prolonger la durée de vie du réseau.
- **Nombre de Nœuds** : Cette métrique indique le nombre total de nœuds dans le réseau. Elle influence directement sur la densité du réseau et peut influence les performances du protocole, notamment en termes de surcharge de calcul et de trafic.
- **Vitesse** : Cette métrique fait référence à la mobilité des nœuds (vitesse de déplacement). Une vitesse élevée peut entraîner des changements fréquents de topologie, affectant ainsi la stabilité et la réactivité du protocole.

- **Coefficient** : Ce terme regroupe les poids utilisés dans la fonction de coût (comme  $\alpha$ ,  $\beta$ ,  $\gamma$ ) pour pondérer les différentes métriques (connectivité, stabilité, énergie). Ces coefficients permettent d'ajuster le comportement du protocole selon les priorités du réseau (ex : privilégier la stabilité ou l'énergie).

### 3.5.2.1 En fonction de l'Énergie

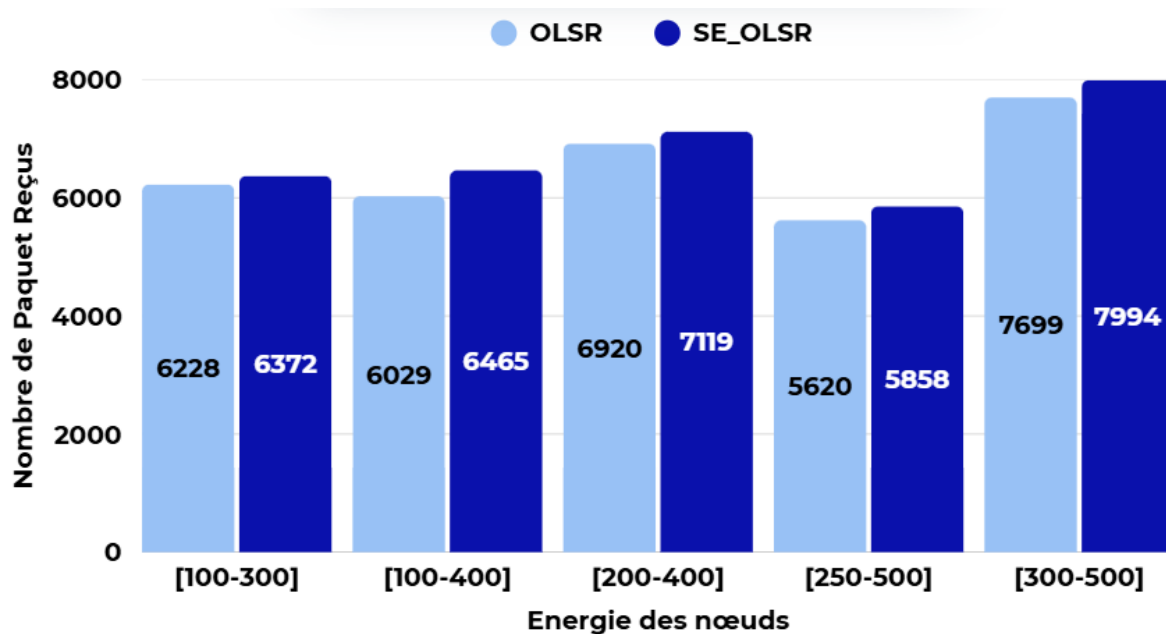


FIGURE 3.2 – Nombre de paquets reçus en fonction de l'énergie

#### Analyse :

SE\_OLSR reçoit plus de paquets que OLSR dans toutes les plages d'énergie proposées. Cela indique que notre protocole optimise le routage selon l'énergie, offrant des chemins plus stables et durables que OLSR.

### 3.5.2.2 en fonction du Nombre de Noeud

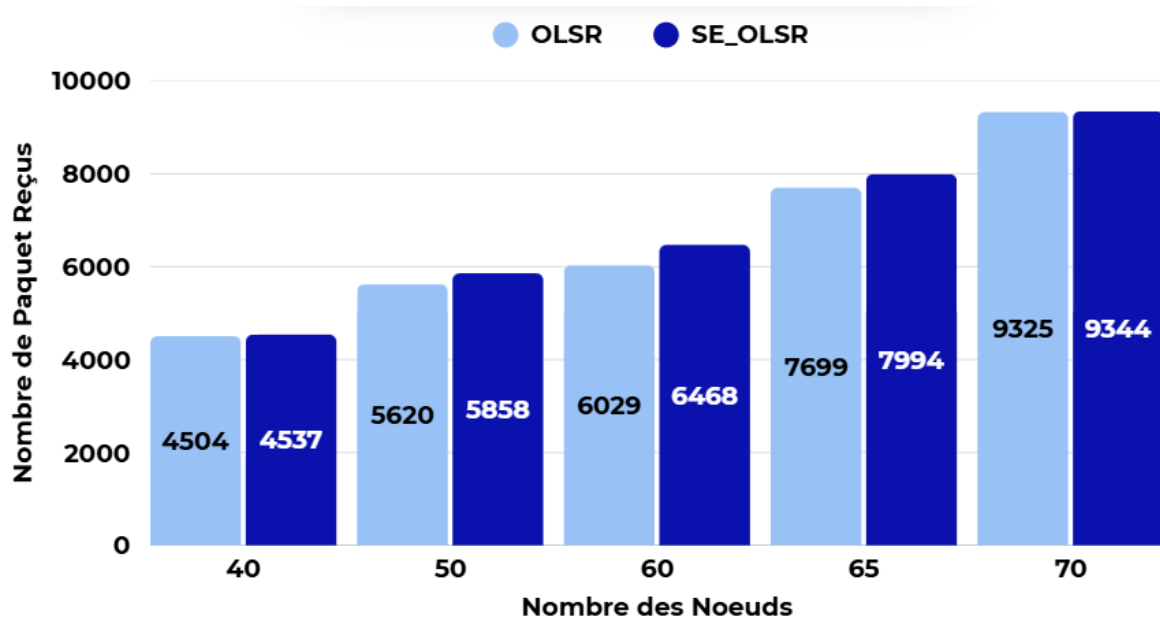


FIGURE 3.3 – Nombre de paquets reçus en fonction du nombre de nœuds

#### Analyse :

Le nombre de paquets reçus augmente avec le nombre de nœuds dans le réseau. Cela s'explique par le fait que les nœuds se rapprochent les uns des autres, ce qui réduit la probabilité de défaillance d'un lien. On peut remarquer sur la figures 3.3 que SE\_OLSR dépasse OLSR\_Standard en nombre de paquets reçus. Ces résultats s'expliquent par le fait que notre protocole prend simultanément en compte les trois critères : l'énergie résiduelle, la stabilité des liens et la portée de connectivité (reachability).

### 3.5.2.3 Nombre de paquets reçus en fonction de la vitesse

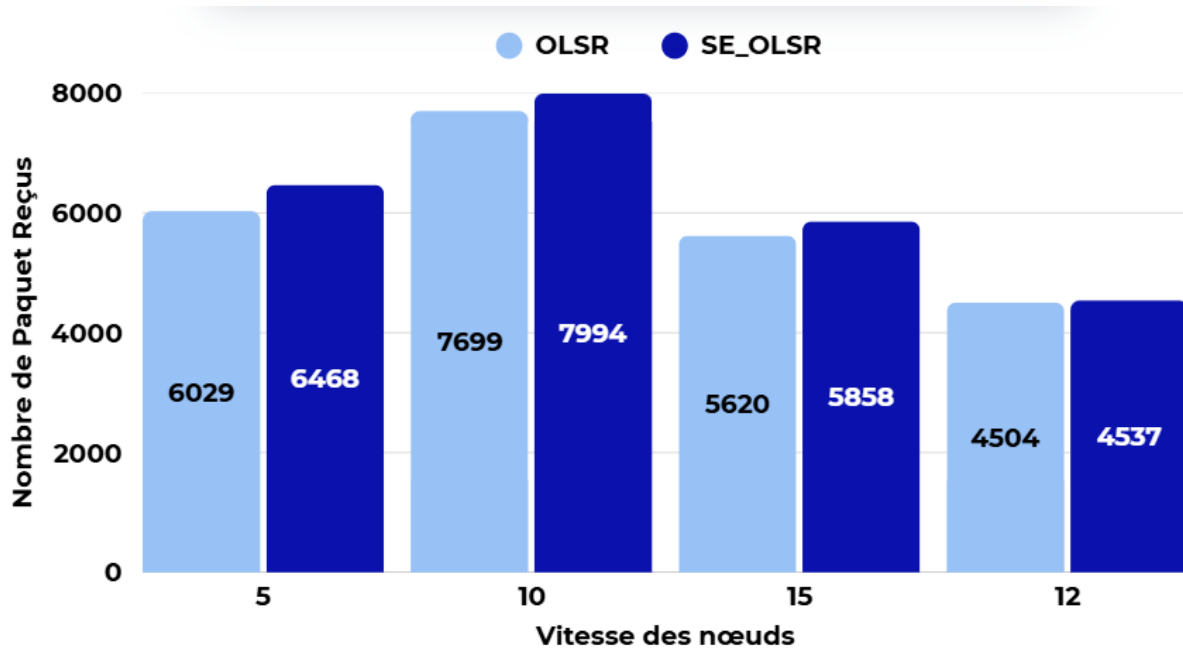


FIGURE 3.4 – Nombre de paquets reçus en fonction de la vitesse

#### Analyse :

La vitesse des UAV est un facteur important qui influence directement le comportement du protocole de routage. En modifiant la vitesse des nœuds dans plusieurs scénarios, la Figure XXX montre que le protocole SE\_OLSR obtient de meilleurs résultats que OLSR\_Standard. Ces résultats s'expliquent par le fait que, dans notre approche, les MPRs sont sélectionnés selon un critère primordial : la stabilité des liens

### 3.5.2.4 Nombre de paquets reçus en fonction des coefficients $\alpha$ , $\beta$ et $\gamma$

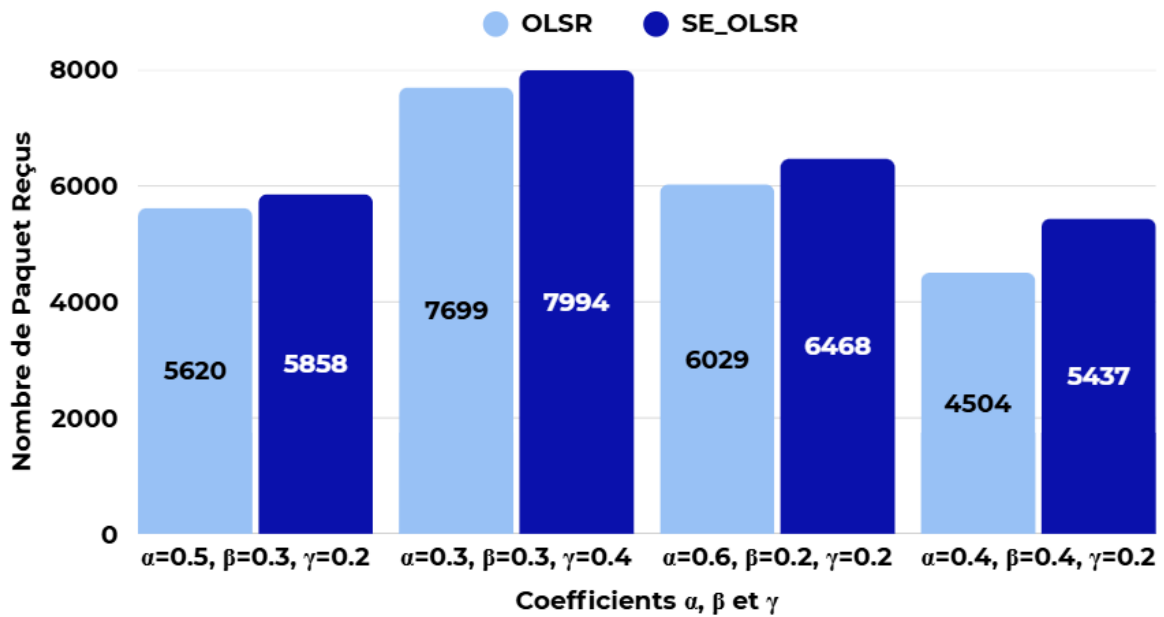


FIGURE 3.5 – Nombre de paquets reçus en fonction des coefficients  $\alpha$ ,  $\beta$  et  $\gamma$

#### Analyse :

Après plusieurs scénarios de simulation, on peut confirmer que les meilleurs coefficients sont :  $\alpha = 0,3$ ,  $\beta = 0,3$ ,  $\gamma = 0,4$ . Ces coefficients indiquent l'importance relative de chaque critère.

**Attention** : ces coefficients dépendent étroitement d'autres paramètres, tels que la vitesse des nœuds, la densité du réseau, etc.

## 3.6 Conclusion

Dans ce chapitre, nous avons suggéré une méthode innovante pour le choix des MPRs dans le cadre du protocole OLSR. En intégrant des critères tels que la stabilité des connexions et la vitalité des nœuds, notre méthode améliore la fiabilité du réseau, diminue les pertes de paquets et augmente sa durée de vie. Les simulations effectuées indiquent que notre technique dépasse l'approche conventionnelle, particulièrement dans des contextes où les nœuds présentent une grande mobilité.

# Conclusion générale

Ce travail de recherche s'est concentré sur l'étude et l'amélioration des protocoles de routage dans les Réseaux Ad Hoc Volants (FANETs), en raison des enjeux associés à leur forte mobilité et à leurs limites énergétiques. Suite à une analyse détaillée et approfondie des réseaux ad hoc, des FANETs et des protocoles de routage, nous avons développé une version améliorée du protocole OLSR, que nous L'avons nommée **SE\_OLSR**. Cette la nouvelle approche repose sur l'amélioration des choix des Multi-Point Relays (MPRs), dans le but d'optimiser les performances énergétiques, de conseder la stabilité des liaisons entre les noeuds et de réduire les délais de transmission.

Les résultats obtenus sous le simulateur NS-3 montrent que SE\_OLSR minimise la perte de paquets comparativement à OLSR standard. Ces performances s'expliquent par la capacité de SE\_OLSR à offrir des chemins plus stables et plus durables entre les nœuds communicants. Nous suggérons, à l'avenir, d'implémenter notre protocole sur une plateforme réelle

# Annexe

## 3.7 Fonction sendHello

```
1 void
2 RoutingProtocol::SendHello()
3 {
4     NS_LOG_FUNCTION(this);
5
6     if (hors_fonction)
7     {
8         return;
9     }
10
11     // Get the node
12     Ptr<Node> node = m_ipv4->GetObject<Node>();
13     if (!node)
14     {
15         NS_LOG_UNCOND("Node " << m_mainAddress << " has no associated
16 Node object.");
17         return;
18     }
19
20     // Check adjusted energy
21     double mon_energie = ConsommationEnergie();
22     if (mon_energie <= 10.0) // 20% of 50 J minimum
23     {
24         /*std::cout << "[SendHello] Time: " << Simulator::Now().
25 GetSeconds() << " s"
26         << " | Node: " << m_mainAddress
27         << " | Adjusted Energy: " << mon_energie << " J <<<<
```

```
20%. Skipping Hello." << std::endl;*/
26     return;
27 }
28
29 olsr::MessageHeader msg;
30 Time now = Simulator::Now();
31
32 // Try to retrieve the energy source for raw energy
33 Ptr<EnergySource> energySource;
34 Ptr<Object> object = node->GetObject<EnergySourceContainer>();
35 if (object)
36 {
37     Ptr<EnergySourceContainer> container = DynamicCast<
EnergySourceContainer>(object);
38     if (container && container->GetN() > 0)
39     {
40         energySource = container->Get(0);
41     }
42 }
43
44 double energy = 0.0;
45 if (energySource)
46 {
47     energy = energySource->GetRemainingEnergy();
48     //NS_LOG_UNCOND("Node " << node->GetId() << " has remaining
energy: " << energy << " J");
49 }
50 else
51 {
52     //NS_LOG_UNCOND("Node " << node->GetId() << " has no energy
source.");
53
54 }
55 // Normalize and store energy in hello.energy
56 double normalized_energy = Normaliser(50, 200, energy);
57 uint8_t energyInt = static_cast<uint8_t>((normalized_energy /
100.0) * 255.0);
58 hello.energy = energyInt;
59
```

```

60 // Get node position
61 Ptr<MobilityModel> mobility = node->GetObject<MobilityModel>();
62 if (!mobility)
63 {
64     NS_LOG_WARN("MobilityModel not found for node " << node->GetId
65     ());
66     return;
67 }
68 Vector pos = mobility->GetPosition();
69 hello.x = pos.x;
70 hello.y = pos.y;
71 hello.z = pos.z;

```

Listing 3.1 – Envoi du message Hello avec énergie et position

## 3.8 Fonction ProcessHello

```

1 void
2 RoutingProtocol::ProcessHello(const olsr::MessageHeader& msg,
3                               const Ipv4Address& receiverIface,
4                               const Ipv4Address& senderIface)
5 {
6     NS_LOG_FUNCTION(this << msg << receiverIface << senderIface);
7
8     if (hors_fonction)
9     {
10        return;
11    }
12
13    double mon_energie = ConsommationEnergie();
14    if (mon_energie <= 10.0)
15    {
16        /*std::cout << "[ProcessHello] Time: " << Simulator::Now().
17        GetSeconds() << " s"
18        << " | Node: " << m_mainAddress
19        << " | Adjusted Energy: " << mon_energie << " J <<<<
20        20%. Skipping Hello processing." << std::endl;*/
21        return;

```

```
20     }
21
22     uint8_t receivedEnergyInt = hello.energy;
23     double energyInJoules = 50.0 + (static_cast<double>(
24     receivedEnergyInt) / 255.0) * 150.0;
25     if (energyInJoules <= 10.0)
26     {
27         /*std::cout << "[ProcessHello] Time: " << Simulator::Now().
28         GetSeconds() << " s"
29         << " | Node: " << msg.GetOriginatorAddress() << " has
30         low energy: "
31         << energyInJoules << " J <<<< 20%. Skipping Hello
32         processing." << std::endl;*/
33         return;
34     }
35
36     Ptr<Node> node = m_ipv4->GetObject<Node>();
37     if (!node)
38     {
39         NS_LOG_WARN("No node found for receiver interface " <<
40         receiverIface);
41         return;
42     }
43
44     Ptr<MobilityModel> mobility = node->GetObject<MobilityModel>();
45     if (!mobility)
46     {
47         NS_LOG_WARN("MobilityModel not found for node " << node->GetId
48         ());
49         return;
50     }
51
52     Vector myPos = mobility->GetPosition();
53
54     double remoteX = hello.x;
55     double remoteY = hello.y;
56     double remoteZ = hello.z;
57
58     m_currentDistance = std::sqrt(
59         std::pow(remoteX - myPos.x, 2) +
60         std::pow(remoteY - myPos.y, 2) +
```

```

53     std::pow(remoteZ - myPos.z, 2));
54     Mettre     jour la table de voisinage
55     PopulateNeighborSet(msg, hello);
56     LinkSensing(msg, hello, receiverIface, senderIface);

```

Listing 3.2 – Réception Hello : extraction énergie et distance

### 3.9 populateNeighborset

```

1
2 void
3 RoutingProtocol::PopulateNeighborSet(const olsr::MessageHeader& msg,
4                                     const olsr::MessageHeader::Hello&
5     hello)
6 {
7     Ipv4Address senderAddress = msg.GetOriginatorAddress();
8     NeighborTuple* nb_tuple = nullptr;
9     for (auto& neighbor : N)
10    {
11        if (neighbor.neighborMainAddr == senderAddress)
12        {
13            nb_tuple = &neighbor;
14            break;
15        }
16
17        if (nb_tuple == nullptr)
18        {
19            NeighborTuple newTuple;
20            newTuple.neighborMainAddr = senderAddress;
21            N.push_back(newTuple);
22            nb_tuple = &N.back();
23        }
24
25        // Store energy in joules (50 200 J)
26        nb_tuple->energy = 50.0 + (static_cast<double>(hello.energy) /
27        255.0) * 150.0;
28
29        if (m_currentDistance >= 0)

```

```

29     {
30         nb_tuple->distanceHistory.push_back(m_currentDistance);
31         if (nb_tuple->distanceHistory.size() > 10)
32         {
33             nb_tuple->distanceHistory.erase(nb_tuple->distanceHistory.
begin());
34         }
35         nb_tuple->variance = CalculateVariance(nb_tuple->
distanceHistory);
36     }
37     else
38     {
39         NS_LOG_WARN("Invalid m_currentDistance: " << m_currentDistance)
;
40     }
41
42     /* std::ostringstream os;
43     os << "Noeud: " << m_ipv4->GetAddress(1, 0).GetLocal()
44         << ", Voisin: " << senderAddress
45         << ", Energy: " << nb_tuple->energy << " J"
46         << ", Distance: " << m_currentDistance
47         << ", Tableau: [";
48     for (size_t i = 0; i < nb_tuple->distanceHistory.size(); ++i)
49     {
50         os << nb_tuple->distanceHistory[i];
51         if (i < nb_tuple->distanceHistory.size() - 1)
52         {
53             os << ", ";
54         }
55     }
56     os << "], Variance: " << nb_tuple->variance << " m ";
57     NS_LOG_DEBUG(os.str()); */
58 }

```

Listing 3.3 – Mise à jour de la table des voisins avec distance

## 3.10 CalculateVariance

```
1 double
```

```

2 RoutingProtocol::CalculateVariance(const std::vector<double>& distances
  )
3 {
4   if (distances.size() < 2)
5       return 0.0;
6
7       double mean = 0.0;
8       for (double d : distances)
9           mean += d;
10          mean /= distances.size();
11
12          double variance = 0.0;
13          for (double d : distances)
14              variance += std::pow(d - mean, 2);
15          variance /= distances.size();
16
17          return variance;
18      }

```

Listing 3.4 – Calcul de la variance sur l’historique des distances pour mesurer la stabilité du lien

### 3.11 MprComputation

```

1
2   // 4.1. Pour chaque n ud dans N, calcule la reachability, c est
3   - -dire
4   // le nombre de n uds dans N2 qui ne sont pas encore couverts par
5   au moins
6   // un n ud dans l ensemble MPR, et qui sont accessibles via ce
7   voisin un saut
8
9   std::map<double, std::vector<const NeighborTuple*>> reachability;
10  double w1 = 0.3;
11  double w2 = 0.3;
12  double w3 = 0.4;
13
14  for (const auto& nb_tuple : N)

```

```

12     {
13         int r = 0;
14         for (const auto& nb2hop_tuple : N2)
15             {
16                 if (nb_tuple.neighborMainAddr == nb2hop_tuple.
neighborMainAddr &&
17                     mprSet.find(nb2hop_tuple.twoHopNeighborAddr) == mprSet.
end())
18                     {
19                         r++;
20                     }
21             }
22 // Normalisation dans [0, 100]
23 // 1.  energie  (d j  normalis e dans [0, 100])
24 // Normalize energy (50 200 J to 0 100 ), reachability, and
variance
25 double norm_energy = Normaliser(50, 200, nb_tuple.energy);
26 double norm_r = Normaliser(0.0, 10, r);
27 double norm_variance = nb_tuple.variance > 0 ? Normaliser(0.0, 1,
1.0 / nb_tuple.variance) : 0.0;
28
29 // Calculate score
30 double score = w1 * norm_r + w2 * norm_variance + w3 * norm_energy;
31 reachability[score].push_back(&nb_tuple);
32
33
34     reachability[score].push_back(&nb_tuple);
35 }
36
37 const NeighborTuple* max = nullptr;
38 double max_score = 0.0;
39
40 for (const auto& entry : reachability) {
41     double score = entry.first;
42     const auto& neighbors = entry.second;
43     for (const auto* nb_tuple : neighbors) {
44         if (max == nullptr || score > max_score) {
45             max = nb_tuple;
46             max_score = score ;

```

```

47     }
48 }
49 }
50
51 if (max != nullptr)
52 {
53     mprSet.insert(max->neighborMainAddr);
54     CoverTwoHopNeighbors(max->neighborMainAddr, N2);
55     NS_LOG_LOGIC("MPR: " << max->neighborMainAddr << ", score: " <<
max_score);
56     NS_LOG_LOGIC(N2.size() << " 2-hop neighbors left to cover!");
57 }
58 }
59
60
61 }

```

Listing 3.5 – Sélection des MPRs selon une fonction de score combinant énergie

### 3.12 GetRemainingEnergy

```

1     double
2 RoutingProtocol::GetRemainingEnergy()
3 {
4     Ptr<Node> node = m_ipv4->GetObject<Node>(); // Get the current
node
5     if (!node)
6     {
7         NS_LOG_WARN("[Energy] No node found!");
8         return -1.0;
9     }
10
11     // Retrieve Energy Source attached to the node
12     Ptr<EnergySourceContainer> energyContainer = node->GetObject<
EnergySourceContainer>();
13
14     if (!energyContainer || energyContainer->GetN() == 0)
15     {

```

```

16     NS_LOG_WARN("[Energy] No energy source found for Node ID: " <<
node->GetId());
17     return -1.0;
18 }
19
20 // Get first energy source in the container
21 Ptr<EnergySource> energySource = energyContainer->Get(0);
22
23 // Ensure it's a 'BasicEnergySource'
24 Ptr<BasicEnergySource> basicSource = DynamicCast<BasicEnergySource
>(energySource);
25 if (!basicSource)
26 {
27     NS_LOG_WARN("[Energy] No BasicEnergySource found for Node ID: "
<< node->GetId());
28     return -1.0;
29 }
30
31 // Get the remaining energy
32 double remainingEnergy = basicSource->GetRemainingEnergy();
33 NS_LOG_INFO("[Energy] Node ID: " << node->GetId() << " | Remaining
Energy: " << remainingEnergy << " J");
34
35 return remainingEnergy;
36 }

```

Listing 3.6 – Récupération de l'énergie résiduelle à partir de la source d'énergie du nœud

### 3.13 ConsommationEnergie

```

1     double
2 RoutingProtocol::ConsommationEnergie()
3 {
4     Ptr<Node> node = m_ipv4->GetObject<Node>();
5     if (!node)
6     {
7         NS_LOG_WARN("No node found for energy calculation");
8         hors_fonction = true;
9         return 0.0;

```

```
10     }
11
12     double energy = GetRemainingEnergy();
13     if (energy < 0.0)
14     {
15         NS_LOG_WARN("Energy retrieval failed for Node ID: " << node->
16         GetId());
17         hors_fonction = true;
18         return 0.0;
19     }
20
21     double mon_energie = energy - (m_nbrTc * 3.0);
22     mon_energie = std::max(0.0, mon_energie);
23
24     return mon_energie;
25 }
```

Listing 3.7 – Calcul de l'énergie consommée par le nœud en fonction du nombre de messages TC

### 3.14 Normaliser

```
1 double
2 RoutingProtocol::Normaliser(double a, double b, double x)
3 {
4     const double c = 0.0;
5     const double d = 100.0;
6
7
8     return (c + (x - a) * d / (b - a));
9 }
```

Listing 3.8 – Normalisation

### 3.15 sendTc

```
1 void
```

```
2 RoutingProtocol::SendTc()
3 {
4     NS_LOG_FUNCTION(this);
5
6     olsr::MessageHeader msg;
7
8     msg.SetVTime(OLSR_TOP_HOLD_TIME);
9     msg.SetOriginatorAddress(m_mainAddress);
10    msg.SetTimeToLive(255);
11    msg.SetHopCount(0);
12    msg.SetMessageSequenceNumber(GetMessageSequenceNumber());
13
14    olsr::MessageHeader::Tc& tc = msg.GetTc();
15    tc.ansn = m_ansn;
16
17    for (auto mprsel_tuple = m_state.GetMprSelectors().begin();
18         mprsel_tuple != m_state.GetMprSelectors().end();
19         mprsel_tuple++)
20    {
21        tc.neighborAddresses.push_back(mprsel_tuple->mainAddr);
22    }
23    QueueMessage(msg, JITTER);
24    m_nbrTc++;
25 }
```

Listing 3.9 – Envoi d'un message TC contenant la liste des MPR selectors et mise à jour du compteur

# Bibliographie

- [1] J.-M. Percher and B. Jouga, “Détection d’intrusions dans les réseaux ad hoc,” *Projet*, 2003.
- [2] D. Zekri, “Agrégation et extraction des connaissances dans les réseaux inter-véhicules,” Ph.D. dissertation, Institut National des Télécommunications, 2013.
- [3] S. A. H. Belkhira, “Optimisation de la qos dans les réseaux adhoc mobiles,” Ph.D. dissertation, Université de Haute Alsace-Mulhouse ; Université Djillali Liabès (Sidi Bel . . . , 2020.
- [4] S. Makkar, Y. Singh, and R. Singh, “Performance investigation of olsr and aodv routing protocols for 3d fanet environment using ns3,” *J. Commun. Eng. Syst*, vol. 8, no. 2, pp. 1–10, 2018.
- [5] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, “Flying ad-hoc networks (fanets) : A survey,” *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, 2013.
- [6] K. Kashyap, A. Agrawal *et al.*, “Fanet : Survey on design challenges, application scenario and communication protocols,” *IJRAR-International Journal of Research and Analytical Reviews (IJRAR)*, vol. 5, no. 4, pp. 1024–1032, 2018.
- [7] C. A. T. Romero, J. H. Ortiz, L. C. Rodríguez Timaná, F. V. Varela, A. H. Aristizábal Montufar, and J. Gamboa-Cruzado, “Fanet and manet, a support and composition relationship.” *Computers, Materials & Continua*, vol. 82, no. 2, 2025.
- [8] M. F. Khan, K.-L. A. Yau, R. M. Noor, and M. A. Imran, “Routing schemes in fanets : A survey,” *Sensors*, vol. 20, no. 1, p. 38, 2019.
- [9] ———, “Routing schemes in fanets : A survey,” *Sensors*, vol. 20, no. 1, p. 38, 2019.

- 
- [10] C. N. K. Reddy and M. Anusha, “Hybrid intelligent routing with optimized learning (hirol) for adaptive routing topology management in fanets,” *arXiv preprint arXiv :2406.15105*, 2024.
- [11] J. Wang and C. Jiang, “Seamless coverage strategies of fanet,” in *Flying Ad Hoc Networks : Cooperative Networking and Resource Allocation*. Springer, 2021, pp. 41–119.
- [12] M. Tropea, P. Fazio, F. De Rango, and N. Cordeschi, “A new fanet simulator for managing drone networks and providing dynamic connectivity,” *Electronics*, vol. 9, no. 4, p. 543, 2020.
- [13] Y. Liu, J. Xie, C. Xing, and S. Xie, “Topology construction and topology adjustment in flying ad hoc networks for relay transmission,” *Computer networks*, vol. 228, p. 109753, 2023.
- [14] M. DJEDIAI and K. Bellahcene, “Evaluation des performances des différents protocoles de routage des réseaux ad hoc pour les réseaux multi-uav (fanet) etude comparative des protocoles dsdv et aodv,” Ph.D. dissertation, UNIVERSITY OF KASDI MERBAH OUARGLA.
- [15] Y. Liu, J. Xie, C. Xing, and S. Xie, “Topology construction and topology adjustment in flying ad hoc networks for relay transmission,” *Computer networks*, vol. 228, p. 109753, 2023.
- [16] A. H. Wheeb, R. Nordin, A. Samah, M. H. Alsharif, and M. A. Khan, “Topology-based routing protocols and mobility models for flying ad hoc networks : A contemporary review and future research directions,” *Drones*, vol. 6, no. 1, p. 9, 2021.
- [17] T. Kim, S. Lee, K. H. Kim, and Y.-I. Jo, “Fanet routing protocol analysis for multi-uav-based reconnaissance mobility models,” *Drones*, vol. 7, no. 3, p. 161, 2023.
- [18] M. M. Alam, M. Y. Arafat, S. Moh, and J. Shen, “Topology control algorithms in multi-unmanned aerial vehicle networks : An extensive survey,” *Journal of network and computer applications*, vol. 207, p. 103495, 2022.
- [19] B. Olaniyi Sadiq, M. Yusuf Abiodun, S. Olayinka Zakariyya, and M. Dahiru Buhari, “Fanet experiment : Real-time surveillance applications connected to image processing sys-

- tem,” *arXiv e-prints*, pp. arXiv–2306, 2023.
- [20] S. Abbas, M. Abu Talib, I. Ahmed, and O. Belal, “Integration of uavs and fanets in disaster management : A review on applications, challenges and future directions,” *Transactions on Emerging Telecommunications Technologies*, vol. 35, no. 12, p. e70023, 2024.
- [21] M. de Graaf, “Energy efficient networking via dynamic relay node selection in wireless networks,” *Ad hoc networks*, vol. 11, no. 3, pp. 1193–1201, 2013.
- [22] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain, “Routing in flying ad hoc networks : Survey, constraints, and future challenge perspectives,” *IEEE access*, vol. 7, pp. 81 057–81 105, 2019.
- [23] A. Guillen-Perez and M.-D. Cano, “Flying ad hoc networks : A new domain for network communications,” *Sensors*, vol. 18, no. 10, p. 3571, 2018.
- [24] J. Kerdsri and T. Veeraklaew, “Visualization of spatial distribution of random waypoint mobility models.” *J. Comput.*, vol. 12, no. 4, pp. 309–316, 2017.
- [25] G. Amponis, T. Lagkas, P. Sarigiannidis, V. Vitsas, P. Fouliras, and S. Wan, “A survey on fanet routing from a cross-layer design perspective,” *Journal of Systems Architecture*, vol. 120, p. 102281, 2021.
- [26] RF Wireless World, “MANET vs VANET vs FANET – Key Differences,” 2024, consulté en juin 2025. [Online]. Available : <https://www.rfwireless-world.com/Terminology/MANET-vs-VANET-vs-FANET.html>
- [27] N. E. Belhadj, L. Chetioui *et al.*, “Etudes et simulation des protocoles de routages dans les réseaux fanets.”
- [28] H. E. Ablaoui and N. O. Bentayeb, “Un nouveau protocole de routage à état de lien basé sur l’énergie et le débit pour les réseaux fanets,” Ph.D. dissertation, UNIVERSITY BBA, 2024.
- [29] A. H. Wheeb, “Flying ad hoc networks (fanet) : Performance evaluation of topology based routing protocols.” *Int. J. Interact. Mob. Technol.*, vol. 16, no. 4, pp. 137–149, 2022.
- [30] X. Wang, H. Li, and J. Zhang, “Routing schemes in FANETs : A survey,” *Wireless Com-*

- munications and Mobile Computing*, vol. 2021, pp. 1–19, 2021, figure7 : classification des protocoles de routage.
- [31] A. MOUSSAOUI, “Routage qos dans les réseaux mobiles ad hoc,” Ph.D. dissertation, Université de Béjaia-Abderrahmane Mira.
- [32] Y. Li, M. St-Hilaire, and T. Kunz, “Enhancing the rgr routing protocol for unmanned aeronautical ad-hoc networks,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 4, pp. 2430–2447, 2013.
- [33] X. Yang, J. Wu, Y. Zhang, B. Qin, and Y. Li, “RTOR : Reliable topology and opportunistic routing for flying ad-hoc networks,” *IEEE Access*, vol. 8, pp. 147 805–147 816, 2020.
- [34] A. T. Le, Y. Ghamri-Doudane, and A.-L. Beylot, “Predictive-olsr : Routing protocol for mobile ad hoc networks using mobility prediction,” *IEEE 73rd Vehicular Technology Conference (VTC Spring)*, pp. 1–5, 2011.
- [35] M. S. Oubbati, A. Lakas, M. Atiquzzaman, and F. Filali, “Intelligent uav routing protocol for flying ad hoc networks using machine learning,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6840–6850, 2019.
- [36] X. Masip-Bruin, M. Yannuzzi, J. Domingo-Pascual, A. Fonte, M. Curado, E. Monteiro, F. Kuipers, P. Van Mieghem, S. Avallone, G. Ventre *et al.*, “Research challenges in qos routing,” *Computer communications*, vol. 29, no. 5, pp. 563–581, 2006.
- [37] S. A. H. Belkhira, “Optimisation de la qos dans les réseaux adhoc mobiles,” Ph.D. dissertation, Université de Haute Alsace-Mulhouse ; Université Djillali Liabès (Sidi Bel . . . , 2020.
- [38] D. Djenouri, N. Badache, and A. Bouabdallah, “Qos routing protocols in wireless multimedia sensor networks : Survey and challenges,” *Sensors*, vol. 14, no. 3, pp. 5293–5334, 2014.

# Références Web

1. Canonical Ltd., *Official Ubuntu Documentation*, 2024. Disponible sur : <https://ubuntu.com>. Consulté le 15 juillet 2024.
2. Canonical Ltd., *Ubuntu Features Overview*, 2024. Disponible sur : <https://ubuntu.com/features>. Consulté le 15 juillet 2024.
3. ISO C++, *About C++*. Disponible sur : <https://isocpp.org/about>. Consulté le 3 juin 2025.
4. NS-3 Project, *NS-3 Network Simulator*, 2023. Logiciel libre sous licence GNU GPLv2. Disponible sur : <https://www.nsnam.org/>.