

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El-Bachir El-Ibrahimi - BBA
Faculté des Mathématiques et Informatiques



MÉMOIRE

Présenté en vue de l'obtention du diplôme

Master en Informatique

Spécialité : Réseaux & Multimédias

Thème

Modèle d'Apprentissage par Renforcement pour le Routage des Données dans l'Internet des Véhicules

Présenté par :

- BOUROUH Aymen
- SALIK Soundous

Soutenu le : 10/06/2025, devant la commission d'examen suivante :

Dr. Hassina BENSEFIA	Enseignant à l'université de BBA	Présidente
Dr. Nadjib BENAOUA	Enseignant à l'université de BBA	Examineur
Dr. Oussama SENOUCI	Enseignant à l'université de BBA	Encadrant

Promotion 2024 / 2025

**“Je n’ai jamais rêvé de succès,
J’ai travaillé pour ça.”**

- Estee Lauder -

Résumé

Cette thèse propose une approche innovante pour optimiser le routage dans les réseaux Internet of Vehicles (IoV) en intégrant l'apprentissage par renforcement, et plus précisément le Q-learning, afin d'améliorer la stabilité des connexions entre les véhicules. L'objectif principal est de réduire les handovers inutiles qui affectent négativement la Qualité de Service (QoS) dans des environnements dynamiques. Après une étude approfondie des architectures IoV et des limitations des protocoles existants, une amélioration du protocole DSRC/C-V2X est introduite. Le modèle développé anticipe intelligemment les décisions de handover en fonction des conditions du réseau, minimisant ainsi les basculements fréquents de protocole. Les performances du système ont été évaluées à travers des simulations réalistes. Les résultats montrent une précision de 97,50%, un rappel de 97,55%, et un F1-score de 97,23%, indiquant une grande fiabilité du modèle. La stabilité du processus d'apprentissage est confirmée par une perte de 0,0250 et une récompense cumulée de 160255340. Enfin, les tests démontrent une réduction significative des handovers inutiles (732 au total, dont 355 sont des ping-pong handovers), une latence moyenne de 6,34 ms, et un PDR moyen de 0,922, surpassant les méthodes traditionnelles basées sur des seuils fixes. Ces résultats soulignent l'efficacité de l'approche proposée pour améliorer la QoS dans les réseaux IoV.

Mots clés : Routage, Internet des véhicules, Apprentissage Automatique, Apprentissage par Renforcement, Q-Learning, DSRC, C-V2X, Handovers.

Abstract

This thesis focuses on optimizing data routing in Internet of Vehicles (IoV) networks by using reinforcement learning to improve the stability of connections between vehicles. The main objective is to design a routing protocol that reduces unnecessary handovers and improves the Quality of Service (QoS) in a dynamic environment. The work begins with an in-depth analysis of the IoV network architecture and the challenges related to routing in these mobile environments. Based on this, an enhancement of the DSRC/C-V2X protocol is proposed, integrating a reinforcement learning model based on Q-learning. This model aims to predict handover decisions more intelligently, based on network conditions, to minimize unnecessary protocol changes and ensure better stability of the connections. The model's performance was evaluated through simulations using a dataset simulating the characteristics of IoV links. The results are promising, with an accuracy of 97.50%, a recall of 97.55%, and an F1-score of 97.23%, showing high effectiveness in correctly detecting situations requiring a handover. The train loss and val loss stabilized at 0.0250, confirming the absence of overfitting. Additionally, the cumulative reward obtained by the agent during training reached 160255340, reflecting the quality of the decisions made by the agent. Tests performed in a realistic simulated environment also showed interesting results : the total number of handovers was 732, of which 355 were ping-pong handovers (repetitive and unnecessary changes). The average latency was 6.34 ms, while the PDR (Packet Delivery Ratio) averaged 0.922. These results indicate a significant improvement over classical approaches based on fixed thresholds, with a significant reduction in unnecessary handovers and better network stability.

Keywords : Routing, Internet of Vehicles, Machine Learning , Reinforcement Learning, Q-Learning, DSRC, C-V2X, Handovers.

ملخص

هذا البحث يركز على تحسين توجيه البيانات في شبكات الإنترنت الخاصة بالمركبات (IoV)، باستخدام التعلم المعزز لتحسين استقرار الاتصال بين المركبات. الهدف الرئيسي هو تصميم بروتوكول توجيه يتيح تقليل التحويلات غير الضرورية (Handovers) وتحسين جودة الخدمة (QoS) في بيئة ديناميكية. يبدأ العمل بتحليل معمق لهندسة شبكات (IoV) والتحديات المرتبطة بالتوجيه في هذه البيئات المتنقلة. بناءً على ذلك، تم اقتراح تحسين لبروتوكول (DSRC/C-V2X)، مع دمج نموذج يعتمد على التعلم المعزز باستخدام خوارزمية (Q-learning). يهدف هذا النموذج إلى التنبؤ بقرارات التحويل بشكل أكثر ذكاءً بناءً على ظروف الشبكة، من أجل تقليل التغييرات غير الضرورية في البروتوكولات وضمان استقرار أفضل للاتصالات. تم تقييم أداء النموذج من خلال محاكاة باستخدام مجموعة بيانات تحاكي خصائص الروابط في شبكات (IoV). النتائج كانت واعدة، حيث وصل النموذج إلى دقة 97.50%، واسترجاع 97.55%، ودرجة F1 بلغت 97.23%، مما يظهر كفاءة كبيرة في الكشف الصحيح عن الحالات التي تتطلب تحويلاً. استقر $train\ loss$ و $val\ loss$ عند 0.0250، مما يثبت أن التعلم كان موثوقاً دون الإفراط في التعلم. علاوة على ذلك، بلغت المكافأة التراكمية التي حصل عليها الوكيل أثناء التدريب 160255340، مما يعكس جودة القرارات التي تم اتخاذها. الاختبارات التي أجريت في بيئة محاكاة واقعية أظهرت نتائج مثيرة للاهتمام أيضاً: حيث كان إجمالي (Handovers) 732، منها 355 كانت ping-pong handovers (التغييرات المتكررة وغير الضرورية). كانت الكمون المتوسط 6.34 ms، في حين أن معدل تسليم الحزم (PDR) كان 0.922. هذه النتائج تظهر تحسناً ملحوظاً مقارنة بالنهج التقليدية القائمة على العتبات الثابتة، مع تقليل كبير للتحويلات غير الضرورية وتحقيق استقرار أفضل في الشبكة.

الكلمات المفتاحية: التوجيه، إنترنت المركبات، التعلم الآلي، التعلم المعزز، التعلم المعزز باستخدام Q-learning، الاتصالات المخصصة قصيرة المدى (DSRC)، الاتصال الخلوي بين المركبة وكل شيء (C-V2X)، عمليات التسليم (Handovers).

Remerciements

Nous remercions avant tout Allah de nous avoir donné la force et la patience nécessaires pour mener à bien ce travail. Nous exprimons nos remerciements à notre encadrant, Dr. SENOUCI Oussama, pour ses efforts inestimables, avec les plus hautes expressions de louange et d'appréciation.

Nous tenons également à remercier tout particulièrement Dr. BENAOUA Nadjib pour sa gentillesse et son soutien.

Nous adressons aussi nos sincères remerciements à Dr. BENSEFIA Hassina, membre du jury, pour l'honneur qu'elle nous a fait en acceptant d'évaluer notre travail.

Enfin, nous remercions toutes les personnes qui, de près ou de loin, ont apporté leur aide ou leurs conseils pour la réalisation de ce mémoire.

Dédicace

Ce mémoire est dédié :

En premier lieu, à l'âme de mon père décédé, qui a été toujours une source d'inspiration et de force dans ma vie. Il a laissé des souvenirs inoubliables et je veux exprimer combien je l'aime et combien il me manque. J'espère qu'il est dans un endroit paisible et qu'il sait qu'il occupe toujours une place spéciale dans mon cœur. Je n'oublierai jamais les précieuses leçons qu'il m'a enseignées et je garderai toujours ce qu'il m'a laissé en mémoire.

À ma mère, à qui je souhaite une longue vie pleine de santé et de bonheur.

À mon cher frère **Rayane** et à ma chère sœur **Lamia**, à qui je souhaite un avenir radieux, rempli de réussites et de bonheur.

À mon cher binôme **Soundous**, pour ton soutien constant et ton amitié précieuse.

À tous mes amis, qui ont toujours été présents pour moi.

★ *Aymen* ★

Dédicace

À mon père, **Djahid**, pilier inébranlable, qui incarne l'amour pur. Toi qui sacrifies tout, qui portes mes peines sans jamais faiblir, même malade, juste pour me voir réussir : tu es mon héros éternel.

À ma mère, **Akila**, source infinie de tendresse, qui veille sur moi malgré sa fatigue, qui m'aime sans limites et sans conditions.

À mes sœurs, **Zeynab** et **Cheyma**, mes plus grandes alliées, mes ombres fidèles, toujours là pour me relever et m'aimer sans faille.

À mes frères, **Sadam**, fierté de ma vie, qui protègent et chérissent leur sœur avec une fierté immense.

À **Misk**, petite étoile de notre maison, lumière douce et pure qui embellit nos vies.

À mon binôme, **Aymen**, merci d'être plus qu'un camarade : je suis reconnaissante d'avoir croisé un frère comme toi.

À mes précieuses amies : **Asma, Yasmina, Loubna, Amani, Yousra, Rim, Assala** – votre amitié est un trésor rare, une richesse inestimable.

Et enfin, à **moi-même**, pour ma force, ma persévérance et mes rêves : aujourd'hui, ce succès est le mien, car il est temps... *d'y être*.

★ *Soundous* ★

Table des matières

Liste des figures	xiv
Liste des tables	xv
Abréviations	xvi
Introduction Générale	1
1 Généralités sur les Réseaux IoVs	5
1.1 Introduction	5
1.2 Fondements de l’Internet des Véhicules	5
1.2.1 Définition de l’IoV	6
1.2.2 Architecture de l’IoV	6
1.2.2.1 Composants Principaux	6
1.2.2.2 Modèles de Communication	8
1.3 Routage dans l’IoV	10
1.3.1 Défis du Routage dans les Réseaux IoV	10
1.3.2 Techniques d’Optimisation du Routage	11
1.4 Apprentissage Automatique et Optimisation	13
1.4.1 Introduction à l’Apprentissage Automatique	13
1.4.2 Algorithmes d’Apprentissage Automatique pour le Routage . . .	13
1.4.3 Avantages de l’Utilisation de l’Apprentissage Automatique . . .	15
1.5 Défis et Enjeux	17
1.5.1 Défis Technologiques	17
1.5.2 Enjeux de Sécurité	17
1.5.3 Considérations Éthiques	17

1.6	Conclusion	18
2	État de l'art	19
2.1	Introduction	19
2.2	Méthodologie de Recherche	20
2.2.1	Approche Générale Adoptée pour la Revue de la Littérature . .	20
2.2.2	Sources Utilisées pour l'Analyse	20
2.2.3	Critères d'Évaluation des Protocoles et Approches	21
2.2.3.1	Latence	22
2.2.3.2	Densité	22
2.2.3.3	Stabilité	22
2.2.3.4	Surcharge (Overhead)	22
2.2.3.5	Chevauchement (Overlapping)	22
2.2.3.6	Complexité	23
2.3	Taxonomie des Protocoles Conventionnels de Routage	23
2.3.1	Protocoles de Routage Basés sur la Position	23
2.3.1.1	Protocoles de Routage Géographique (Geo-Routing) .	25
2.3.1.2	Protocoles de Routage par Zone Géographique (Zone-based Routing)	26
2.3.1.3	Comparaison et Limites des Protocoles Basés sur la Position	27
2.3.2	Protocoles de Communication Véhiculaires	28
2.3.2.1	Protocoles de Communication V2V	29
2.3.2.2	Protocoles de Communication V2I	31
2.3.2.3	Protocoles de Communication V2X	33
2.3.2.4	Comparaison et Limites des Protocoles de Communication Véhiculaires	35
2.3.3	Protocoles de Routage Ad-Hoc et Décentralisés	35
2.3.3.1	Protocoles de Routage Ad-Hoc pour IoV (AODV, DSR, etc.)	36
2.3.3.2	Protocoles de Clustering pour l'IoV	38
2.3.3.3	Protocoles Hybrides pour les Réseaux IoV	40

2.3.3.4	Comparaison et Limites des Protocoles Ad-Hoc et Décentralisés dans l'IoV	41
2.3.4	Protocoles de Gestion de QoS et Transmission Multicast	42
2.3.4.1	Protocoles QoS dans les Réseaux IoV	43
2.3.4.2	Protocoles Multicast dans les Réseaux IoV	45
2.3.4.3	Comparaison et Limites des Protocoles QoS et Multicast dans l'IoV	47
2.4	Application de l'Intelligence Artificielle pour la Transmission de Données dans les Réseaux IoV	47
2.4.1	Rôle de l'Apprentissage Automatique dans l'Optimisation du Routage	48
2.4.2	Algorithmes d'Apprentissage Supervisé et Non Supervisé pour la Gestion du Routage	49
2.4.3	Applications de l'Intelligence Artificielle pour la QoS	49
2.4.4	Défis de l'IA dans les Réseaux IoV	50
2.5	Conclusion	50
3	Étude de cas d'un protocole de routage hybride basé sur DSRC et C-V2X	51
3.1	Introduction	51
3.2	Introduction au protocole DSRC	52
3.2.1	Définition de DSRC	53
3.2.2	Architecture et Composants du Protocole DSRC	53
3.2.3	Fonctionnement du Protocole DSRC	55
3.2.3.1	Transmission des Messages dans DSRC	55
3.2.3.2	Types de Communication dans DSRC	56
3.2.3.3	Gestion du Canal et Sécurité dans DSRC	56
3.2.4	Types de Messages Utilisés dans DSRC	56
3.2.4.1	Cooperative Awareness Messages (CAM)	56
3.2.4.2	Decentralized Environmental Notification Messages (DENM)	58
3.2.5	Métriques de Performance dans DSRC	58
3.2.5.1	Temps d'Attente du Canal (CWT - Channel Waiting Time)	58

3.2.5.2	Longueur de la File d'Attente (QL - Queue Length) . . .	59
3.2.5.3	Niveau d'Occupation du Canal (COL - Channel Occu- pancy Level)	59
3.2.5.4	Charge Réseau dans LTE pour la Sélection de DSRC .	59
3.2.6	Avantages du DSRC	60
3.2.7	Limitations de DSRC	60
3.3	Présentation de C-V2X	61
3.3.1	Définition de C-V2X	61
3.3.2	Fonctionnement de C-V2X	62
3.3.3	Normes techniques utilisées	62
3.4	Comparaison entre DSRC et C-V2X	63
3.5	Architecture hybride DSRC et C-V2X	64
3.5.1	Présentation du modèle hybride	64
3.5.2	Gestion des ressources radio	65
3.5.3	Sélection Adaptative du RAT et VHO	66
3.6	Conclusion	67
4	Optimisation du Routage DSRC/C-V2X par Apprentissage Renforcé	68
4.1	Introduction	68
4.2	Contexte et Motivation	69
4.2.1	Problématique des handovers inutiles	69
4.2.2	Limites des approches classiques (seuil RSSI, règles fixes)	69
4.2.3	Avantages de l'approche RL dans un contexte dynamique	70
4.2.4	Comparaison entre RL et les méthodes classiques de ML	71
4.3	Fondamentaux de RL	72
4.3.1	Définitions clés : état, action, récompense, politique	72
4.3.2	Q-Learning : principe et fonctionnement	73
4.3.2.1	Fonctionnement du Q-Learning	73
4.3.3	Tableau de Valeurs d'État-Action	74
4.4	Conception du Modèle RL pour le Handover	76
4.4.1	Objectifs du modèle	77
4.4.2	Architecture générale du système	78
4.4.3	Définition de l'environnement	79

4.4.4	Construction du Dataset pour l'entraînement	80
4.4.4.1	Format du Dataset et Prétraitement	80
4.4.5	Fonction de récompense et politiques d'exploration	83
4.4.5.1	Fonction de Récompense	83
4.4.6	Structure de l'Agent RL	84
4.4.7	Entraînement de l'Agent Q-Learning	84
4.4.8	Critères d'arrêt et convergence de l'apprentissage	85
4.5	Analyse et Évaluation des Performances du Modèle	87
4.5.1	Métriques de performance	87
4.5.2	Analyse des résultats des performances	89
4.6	Déploiement et simulation du protocole DSRC/C-V2X avec Q-Learning	96
4.6.1	Configuration de la Simulation	97
4.6.2	Topologie initiale des véhicules sur l'autoroute	97
4.6.3	Chronologie des handovers et indicateurs réseau	98
4.6.4	Évaluation des performances	99
4.6.4.1	Latence Moyenne	100
4.6.4.2	Taux de Livraison des Paquets (PDR)	101
4.6.4.3	Étude temporelle des handovers	102
4.6.4.4	Historique des Décisions par Véhicule	103
4.6.4.5	Historique des événements de handover	103
4.7	Discussion & Directions Futures	104
4.8	Conclusion	105
	Conclusion Générale	106
	Bibliographie	108
	Annexe	118
A.1	Exemple de transitions	118
A.2	Implémentation du Modèle RL	119
A.3	Étapes de la Simulation DSRC/C-V2X Basée sur le Q-Learning	135

Table des figures

1.1	Architecture de l’IoV.	7
1.2	Modèles de communication de l’IoV	9
2.1	Taxonomie des Protocoles de Routage.	24
3.1	Organigramme de protocole DSRC [1].	57
3.2	Architecture hybride DSRC et C-V2X	65
4.1	Organigramme du fonctionnement de l’algorithme de Q-learning	75
4.2	Aperçu des premières lignes de la Q-Table	76
4.3	Les résultats du dataset et des actions possibles	80
4.4	Aperçu des premières lignes du dataset	82
4.5	Paramètres d’apprentissage configurés	85
4.6	Progression de l’entraînement par Q-learning	86
4.7	Évolution des récompenses pendant l’apprentissage	90
4.8	Évolution du taux d’exploration ϵ	90
4.9	Évolution de Accuracy	92
4.10	Matrice de confusion.	93
4.11	Évolution de le precision et Rappel	94
4.12	Rapport de classification du modèle Q-Learning	94
4.13	Évolution de la perte (loss)	95
4.14	Topologie initiale des véhicules sur une autoroute à trois voies.	98
4.15	Sortie de la simulation avec les informations de performance.	99
4.16	Évolution de la latence moyenne pendant la simulation.	100
4.17	Évolution du PDR moyen pendant la simulation.	101
4.18	Nombre de Handovers par seconde pendant la simulation.	102

4.19 Exemple d'historique des décisions prises par les véhicules.	103
4.20 Exemple des handovers enregistrés dans la simulation.	104

Liste des tableaux

2.1	Comparaison des protocoles basés sur la position	28
2.2	Comparaison des protocoles de communication Véhiculaires	36
2.3	Comparaison des protocoles de Routage Ad-Hoc et Décentralisés	42
2.4	Comparaison des protocoles QoS et Multicast dans L'IoV	47
3.1	Comparaison entre DSRC et C-V2X	64
4.1	Synthèse des performances de l'agent	91
4.2	Résumé des Performances et Pertes du Modèle	96
4.3	Paramètres de simulation chargés	97
4.4	Exemple de la table des véhicules dans la simulation	98

Abréviations

IoV : Internet of Vehicles.

IoT : Internet of Things.

ITS : Intelligent Transportation Systems.

QoS : Quality of Service.

VCC : Vehicular Cloud Computing.

OBU : On-Board Unit.

RSU : Road-Side Unit.

TCC : Transport Control Center.

V2V : Vehicle to Vehicle.

V2I : Vehicle to Infrastructure.

V2N : Vehicle to Network.

V2P : Vehicle to Pedestrian.

RAVP : Routing Algorithm based on Vehicle Position.

GPS : Global Positioning System.

DSRC : Dedicated Short-Range Communications.

WAVE : Wireless Access in Vehicular Environment.

IEEE : Institute of Electrical and Electronics Engineers.

GPSR : Greedy Perimeter Stateless Routing.

C-V2X : Cellular Vehicle to Everything.

AOMDV : Ad hoc On-demand Multipath Distance Vector Routing.

IA : Intelligence Artificielle.

ML : Machine Learning

RL : Reinforcement Learning.

DQN : Deep Q-Network.
RNN : Recurrent Neural Network.
LSTM : Long Short-Term Memory.
VANETs : Vehicular Ad-hoc Networks.
GNN : Graph Neural Network.
NN : Neural Network.
DNN : Deep Neural Network.
AODV : Ad hoc On-Demand Distance Vector.
CH : Cluster Head.
MANETs : Mobile Ad hoc Networks.
V2X : Vehicle-to-Everything.
RAT : Radio Access Technology.
VHO : Vertical Handover.
RSSI : Received Signal Strength Indicator.
CIC : Communication Infrastructure Controller.
CS : Communication Server.
CCapp : Centralized Control Application.
CAM : Cooperative Awareness Message.
DENM : Decentralized Environmental Notification Message.
MSE : Mean Squared Error.

Introduction Générale

Dans le domaine de Internet of Vehicles (IoV), les systèmes de communication intelligents occupent une place centrale en facilitant l'interaction fluide des informations entre les véhicules, les infrastructures et les usagers. Devant les défis croissants posés par la mobilité, la densité du trafic et la qualité de service, il est actuellement envisagé d'explorer de nouvelles approches fondées sur l'intelligence artificielle afin d'améliorer les performances du réseau.

Notre étude s'est concentrée sur l'intégration de l'apprentissage automatique (Machine Learning - ML) dans ce contexte, à travers la modélisation et la simulation d'une solution intelligente visant à optimiser le processus de communication dans l'IoV. Dans cette introduction, nous présenterons les motivations et la problématique de cette recherche, avant de détailler les objectifs poursuivis et les contributions de notre mémoire. Enfin, un aperçu de la structure générale du document sera proposé.

Problématique & Motivations

IoV représente une avancée majeure dans les réseaux de communication entre véhicules, avec des avantages tels que l'amélioration de la sécurité routière et la gestion du trafic. Cependant, le routage efficace des informations dans ces réseaux dynamiques reste un défi. En effet, les méthodes de routage traditionnelles ne parviennent pas à répondre aux exigences des environnements IoV, où la topologie du réseau change fréquemment en raison de la mobilité des véhicules. Il devient donc crucial de trouver des stratégies de routage optimales qui garantissent une transmission rapide et fiable des données, tout en s'adaptant aux changements dynamiques du réseau.

L'intégration des algorithmes d'apprentissage automatique dans les techniques de routage constitue une opportunité prometteuse pour surmonter ces défis. En exploitant les capacités de ces algorithmes pour analyser des paramètres contextuels tels que la densité du trafic et la congestion, il devient possible d'optimiser le processus de routage. Cela permet non seulement d'améliorer la qualité de service (QoS) dans les réseaux IoV, mais aussi de réduire les temps de transmission et d'assurer une plus grande fiabilité dans la communication entre véhicules.

Objectifs & Contributions

Dans le cadre de ce mémoire, notre objectif principal est d'améliorer la transmission des données dans les réseaux de IoV en réduisant les handovers inutiles entre deux technologies de communication : le Dedicated Short-Range Communications (DSRC) et le Cellular Vehicle to Everything (C-V2X). En effet, les approches classiques de sélection de protocole, basées sur des seuils fixes de latence ou de RSSI, entraînent souvent des changements de protocole fréquents, instables et non pertinents, ce qui dégrade la qualité de service globale du réseau.

Pour remédier à cette problématique, nous proposons une approche basée sur l'apprentissage par renforcement (Reinforcement Learning – RL), capable de prendre des décisions de handover plus intelligentes et contextuelles. Notre solution vise à apprendre, à partir d'un environnement simulé, les meilleures décisions à prendre pour minimiser les commutations inutiles tout en maintenant un bon niveau de latence, de stabilité de la connexion, et de taux de livraison des paquets (PDR).

Nos contributions principales se déclinent comme suit :

1. Une revue critique de la littérature, permettant de contextualiser et d'identifier les limites des travaux existants.
2. Une étude de cas accompagnée d'une analyse approfondie de la problématique des handovers inutiles dans un réseau hybride DSRC/C-V2X.
3. La proposition d'un modèle d'apprentissage par renforcement, basé sur le Q-learning, afin d'optimiser les décisions de handover.
4. L'intégration du modèle proposé dans un environnement simulé, permettant d'évaluer les performances de l'optimisation apportée.

Ce travail contribue à ouvrir la voie vers des systèmes de communication véhiculaires plus intelligents, adaptatifs et efficaces dans des contextes dynamiques.

Organisation du mémoire

Le mémoire est organisé en quatre chapitres distincts, chacun traitant d'un aspect spécifique de notre travail :

— **Chapitre 1 : Généralités sur les Réseaux IoVs**

Ce chapitre introduit le concept d'IoV, en présentant ses composants fondamentaux, son architecture globale, ainsi que les différents modèles de communication. Nous y abordons également les principaux défis liés au routage dans ces réseaux dynamiques. Enfin, nous introduisons le rôle de l'apprentissage automatique et les motivations de son intégration dans le domaine du routage IoV.

— **Chapitre 2 : État d'art**

Dans ce chapitre, nous présentons une étude approfondie de l'état de l'art des protocoles de routage dans IoV. Nous analysons les principales approches classiques utilisées pour assurer la connectivité et la transmission efficace des données dans des environnements hautement dynamiques. Par ailleurs, nous étudions l'intégration progressive du Machine Learning dans le processus de routage, en mettant en évidence son potentiel d'adaptation et de prise de décision contextuelle. Nous avons structuré notre analyse autour de plusieurs critères pertinents tels que la stabilité, la surcharge, la densité, la latence, la complexité et overlapping. En nous appuyant sur ces critères, nous avons comparé diverses approches récentes issues de la littérature, dans le but d'identifier leurs limitations structurelles et fonctionnelles. L'objectif de cette analyse est de dégager les opportunités d'amélioration à travers des techniques intelligentes, mieux adaptées aux contraintes spécifiques des réseaux IoV.

— **Chapitre 3 : Étude de cas d'un protocole de routage hybride basé sur DSRC et C-V2X**

Ce chapitre analyse les technologies DSRC et C-V2X en présentant leurs caractéristiques techniques et leurs différences principales, ainsi que les scénarios d'utilisation privilégiés pour chacune. Il introduit ensuite une architecture hy-

bride combinant ces deux technologies, en étudiant les méthodes classiques de sélection de la technologie d'accès radio (RAT) et en soulignant les problèmes liés aux handovers fréquents causés par l'utilisation de seuils fixes.

— **Chapitre 4 : Optimisation du Routage DSRC/C-V2X par Apprentissage Renforcé**

Ce chapitre présente notre contribution principale. Nous y détaillons les étapes de préparation des données simulées, la conception et l'entraînement du modèle basé sur l'apprentissage par renforcement, ainsi que l'intégration de l'agent de décision dans le processus de routage. Enfin, nous présentons et analysons les résultats de simulation obtenus, permettant d'évaluer la pertinence et les avantages de l'approche proposée face aux méthodes classiques.

Chapitre 1

Généralités sur les Réseaux IoVs

1.1 Introduction

L'IoV une extension de Internet des objets (IoT) appliquée aux véhicules, permet aux véhicules de communiquer entre eux et avec l'infrastructure pour améliorer la gestion du trafic, la sécurité routière, pour maximiser l'efficacité de ces réseaux, l'optimisation du routage est essentielle. Le routage dans l'IoV est confronté à des enjeux complexes tels que la vitesse de déplacement rapide des véhicules, la saturation du réseau et la variété des échanges. Les méthodes conventionnelles ont leurs contraintes, cependant les algorithmes d'apprentissage automatique proposent des alternatives prometteuses en facilitant l'analyse de données en temps réel et l'optimisation dynamique des choix de routage. Ce chapitre examine ces notions, en mettant l'accent sur l'amélioration du routage grâce aux algorithmes.

1.2 Fondements de l'Internet des Véhicules

une vue d'ensemble L'IoV symbolise une avancée majeure des réseaux intelligents dans le domaine du transport. C'est un milieu interconnecté où les véhicules partagent des données entre eux et avec les infrastructures routières en temps réel. Grâce à des technologies comme la 5G, l'intelligence artificielle (IA) et le traitement en périphérie.

1.2.1 Définition de l’IoV

L’IoV est une composante essentielle de Intelligent Transportation Systems (ITS) [2], permettant aux véhicules intelligents et connectés de former des réseaux basés sur Internet grâce à l’informatique Vehicular Cloud Computing (VCC).

Les véhicules collaborent pour échanger des ressources et des services en construisant un cloud à la périphérie du réseau [3]. Au moyen de technologies telles que la 5G, les véhicules se servent d’Unités Embarquées (OBU) et d’Unités Routières (RSU) pour collecter en direct des données sur le trafic et l’état des routes, contribuant ainsi à l’amélioration de l’efficacité du trafic et de la sécurité. Toutefois, cela suscite aussi des inquiétudes relatives à la collecte de données sensibles des utilisateurs, comme les itinéraires des véhicules et les données biométriques, susceptibles de mettre en péril la sécurité personnelle et financière [4].

1.2.2 Architecture de l’IoV

L’architecture de l’IoV est conçue pour permettre une communication fluide et efficace entre les véhicules, les infrastructures, et les utilisateurs. Elle est généralement divisée en plusieurs couches, chacune ayant des fonctions spécifiques. Pour illustrer cette organisation, La Figure 1.1 suivante met en évidence les principaux composants de l’IoV ainsi que leurs interactions à travers les différentes couches.

1.2.2.1 Composants Principaux

L’IoV, s’appuie sur plusieurs composantes qui rendent possible l’échange de données entre les véhicules et les infrastructures routières. Voici les principaux éléments qui entrent en jeu :

1. **Véhicules intelligents** : Ces véhicules embarquent divers capteurs, systèmes de communication et dispositifs électroniques. Grâce à ces outils, ils peuvent surveiller en temps réel le comportement du conducteur, détecter l’état de la route ou analyser ce qui se passe autour d’eux. Cette capacité d’observation permet d’adapter la conduite aux conditions rencontrées, ce qui améliore à la fois la sécurité et l’efficacité du transport [2].
2. **Cloud Computing et Edge Computing** : Face à la masse de données pro-

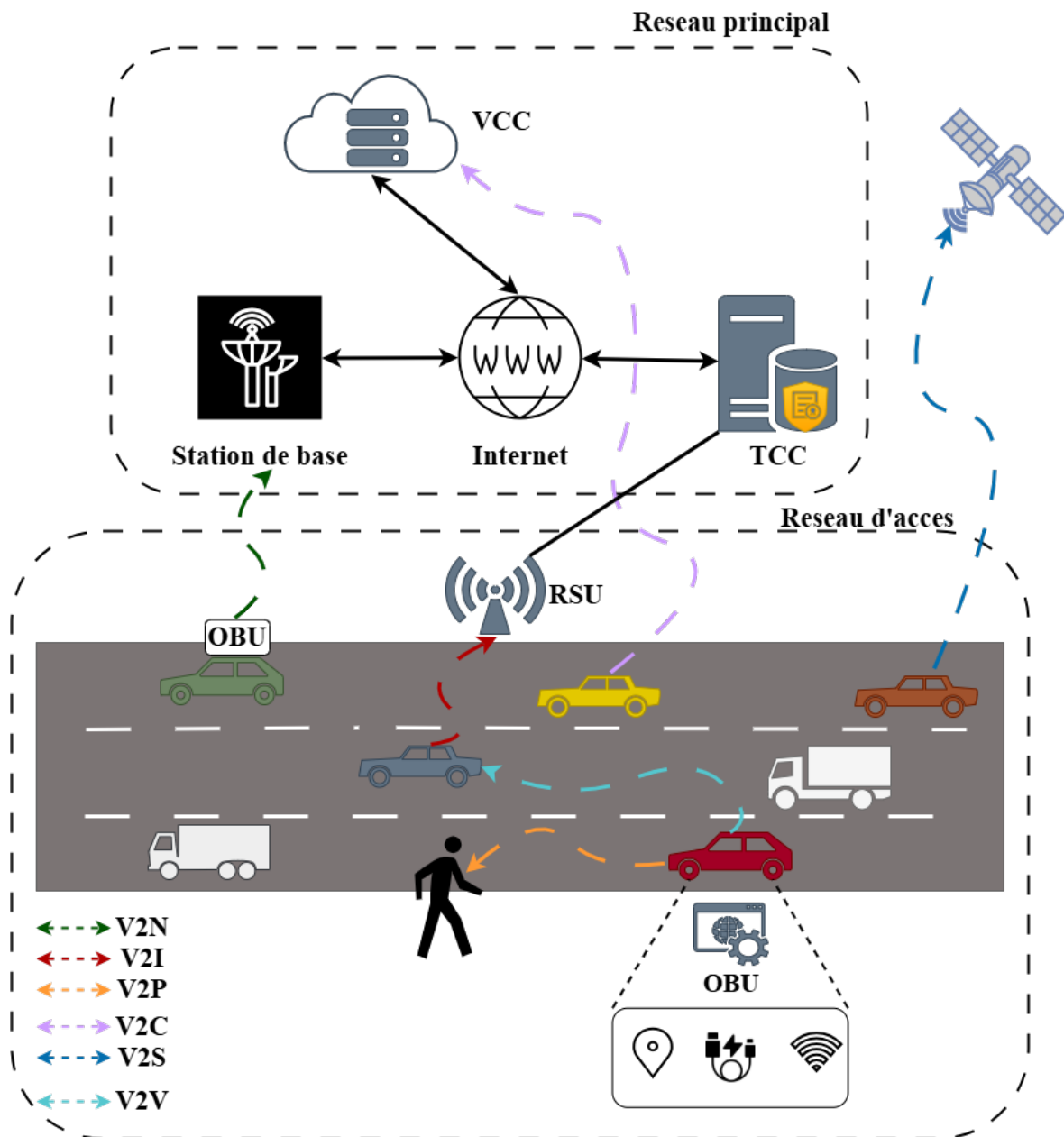


FIGURE 1.1 – Architecture de l'IoV.

duites par les véhicules, deux types de traitement sont utilisés. Le cloud permet d'analyser les données à grande échelle, tandis que l'edge intervient plus près de la source, c'est-à-dire directement au niveau du véhicule ou de l'infrastructure locale. Ensemble, ils permettent d'assurer des services comme la régulation du trafic, la maintenance anticipée ou encore les mises à jour à distance [2].

3. **Centre de Contrôle des Transports (TCC) :** Le TCC joue un rôle central dans le pilotage global du système. Il peut détecter les incidents, signaler les infractions ou alerter sur les bouchons. Il gère également l'accès à certains services,

notamment l'Internet embarqué pour les usagers, renforçant ainsi à la fois la sécurité et la connectivité [5].

4. **OBU** : Installée dans chaque véhicule, l'OBU permet la communication directe avec d'autres véhicules ou les infrastructures à proximité. Elle utilise la norme WAVE, issue du protocole IEEE 802.11p, et communique sur la bande DSRC à 5,9 GHz. C'est grâce à elle que les véhicules peuvent dialoguer entre eux de façon autonome [6].
5. **RSU** : La RSU est un appareil installé sur les routes, chargé de relayer les communications entre les véhicules et les serveurs distants. Elle est reliée au TCC via un réseau câblé, et elle utilise elle aussi la norme WAVE. En plus de sa fonction de relais, elle offre divers services comme le partage d'informations, l'accès à Internet et la géolocalisation. Elle joue aussi un rôle dans l'extension de la portée du réseau et dans la fluidification du trafic [6].

1.2.2.2 Modèles de Communication

Dans un monde où les véhicules deviennent de plus en plus connectés, différentes manières de communiquer ont été développées. Ces échanges ne se font pas seulement entre voitures, mais aussi avec leur environnement, comme les routes, les piétons ou même les satellites. La Figure 1.2 ci-dessous regroupe les différents types de communication établis entre les véhicules, les infrastructures, les piétons et d'autres entités du réseau.

Les modèles de communication de l'IOV sont :

1. **Communication ad hoc Vehicle-to-vehicle (V2V)** : Ici, les voitures échangent directement des données entre elles. Pas besoin de passer par une borne ou un réseau central. Ce type de communication est souvent utilisé pour des alertes locales, comme un véhicule qui freine brusquement ou qui change de voie. C'est surtout utile pour éviter les accidents [7].
2. **Communication ad hoc Vehicle-to-Infrastructure (V2I)** : Les véhicules peuvent aussi communiquer avec des équipements placés sur ou autour des routes : feux de circulation, capteurs, bornes RSU, etc. Ces échanges passent généralement par des réseaux comme la 5G, le Wi-Fi ou le DSRC. Cela aide à mieux gérer la circulation en temps réel et à améliorer la sécurité [8]

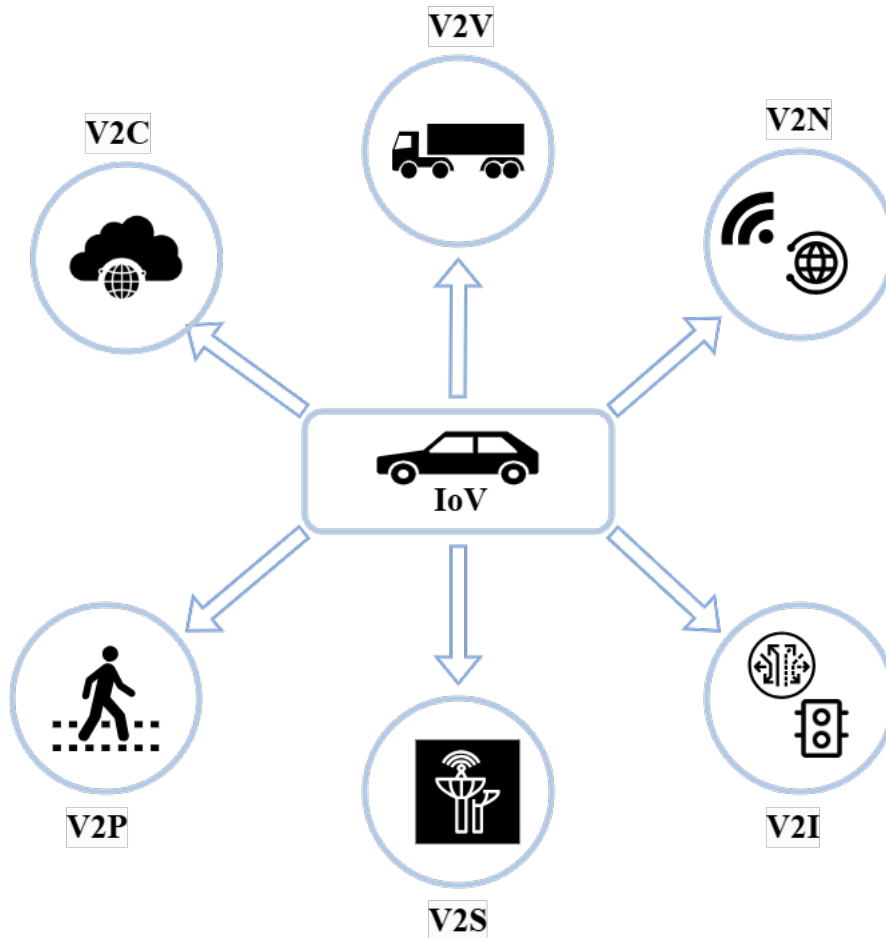


FIGURE 1.2 – Modèles de communication de l’IoV

3. **Communication Vehicle-to-Cloud (V2C)** : Les données générées par les véhicules sont souvent envoyées vers le cloud, c’est-à-dire vers des serveurs à distance. Cela permet d’utiliser des ressources puissantes pour analyser ces données, les stocker ou effectuer des mises à jour logicielles. Le véhicule reste ainsi léger, tout en étant intelligent [5].
4. **Communication Vehicle-to-Network (V2N)** : Ce mode permet au véhicule de se connecter directement aux éléments du réseau : feux connectés, serveurs de gestion du trafic, etc. Les conducteurs peuvent alors recevoir des infos en temps réel sur l’état des routes, les bouchons ou les incidents à venir. C’est un vrai plus pour rouler de façon fluide et sûre [5].
5. **Communication Vehicle-to-Pedestrian (V2P)** : Les voitures peuvent aussi interagir avec les appareils des piétons ou cyclistes – comme les smartphones. Cela permet, par exemple, d’envoyer des alertes de sécurité ou d’offrir des ser-

vices connectés aux passagers (musique, vidéo, partage de données). C'est un bon moyen d'améliorer l'expérience de chacun sur la route [5].

6. **Communication Vehicle-to-Satellite (V2S)** : Dans les zones mal couvertes par les réseaux terrestres (comme en montagne), les voitures peuvent passer par les satellites. La communication utilise des ondes radio pour rester connecté partout, même loin des villes. Pratique pour la navigation ou les services d'urgence.

1.3 Routage dans l'IoV

Le routage est un élément clé des systèmes de transport intelligents (ITS) dans l'IoV. L'algorithme RAVP (Routing Algorithm based on Vehicle Position) utilise plusieurs outils : une matrice qui donne la probabilité de la position des véhicules, une autre qui associe ces positions, ainsi que des facteurs implicites pour mieux prévoir leurs trajectoires. Cela permet de calculer une distance anticipée entre les véhicules, de l'ajuster selon la capacité du cache, et d'améliorer la prise de décision pour la transmission. Les simulations montrent que RAVP fait mieux que d'autres algorithmes en termes de taux de livraison, délai, et charge de routage [9].

1.3.1 Défis du Routage dans les Réseaux IoV

Le routage dans les réseaux IoV présente plusieurs difficultés dues à la complexité et à la mobilité des véhicules. En voici les principales :

1. **Connectivité intermittente et partitionnement du réseau** : Les véhicules bougent vite et des obstacles comme des bâtiments ou des collines peuvent couper les connexions. Cela cause des coupures fréquentes qui compliquent la communication et peuvent diviser le réseau en parties isolées [10].
2. **Topologie dynamique** : Le réseau change tout le temps avec les déplacements des véhicules. Il faut donc des algorithmes capables de s'adapter vite pour garder des routes fiables [11].
3. **Latence et QoS** : La latence est très importante, surtout pour les applications de sécurité. Si l'information arrive trop tard, elle peut ne plus être utile, voire dangereuse. Il faut donc minimiser ces délais pour assurer une transmission rapide [12].

4. **Sécurité et confidentialité** : Le routage doit aussi se protéger contre les attaques et la manipulation des données. Sinon, cela pourrait perturber les communications entre véhicules. Des systèmes d'authentification et de chiffrement solides sont nécessaires [13].
5. **Accès au canal radio et interférences** : Dans les zones urbaines, l'accès aux canaux radio est difficile à gérer à cause des interférences et des obstacles physiques, ce qui affecte la fiabilité des échanges [14].

1.3.2 Techniques d'Optimisation du Routage

Pour rendre la communication entre véhicules dans les réseaux IoV plus efficace, fiable et rapide, plusieurs techniques d'optimisation sont utilisées. Ces méthodes tiennent compte de la mobilité constante et des contraintes spécifiques à ce type de réseau :

1. **Routage basé sur la position géographique (Position-Based Routing :)**
 Cette méthode s'appuie sur les coordonnées Global Positioning Systems (GPS) des véhicules pour guider l'envoi des données. Par exemple, le protocole Greedy Perimeter Stateless Routing (GPSR) choisit le voisin le plus proche de la destination pour transmettre les paquets, ce qui diminue les délais [15].
 - (a) **Avantage** : Nécessite moins de mémoire pour garder la table de routage.
 - (b) **Défi** : Moins efficace dans les zones où les véhicules sont peu nombreux.
2. **Routage prédictif (Predictive Routing)** : Cette technique essaie d'anticiper où les véhicules se dirigeront, en analysant leurs mouvements actuels (vitesse, direction). Le protocole Connectivity-Aware Routing (CAR) en est un exemple, il estime combien de temps les véhicules resteront connectés [16].
 - (a) **Avantage** : Utile dans les villes ou sur des routes avec des trajets réguliers.
 - (b) **Défi** : Moins précis quand les véhicules changent souvent de direction.
3. **Routage par véhicules relais (Store-Carry-Forward)** : Dans les zones où la couverture réseau est faible, un véhicule conserve temporairement les données et les transmet quand il rencontre un autre véhicule. Le protocole Delay-Tolerant Network (DTN) suit cette approche [17].

-
- (a) **Avantage** : Permet de transmettre les données même si le réseau est fragmenté.
 - (b) **Défi** : Cela peut augmenter le temps d'attente et la quantité de stockage nécessaire.
4. **Routage à base de clusters (Cluster-Based Routing)** : Les véhicules sont regroupés en petits groupes où un chef de groupe (cluster head - CH) coordonne la communication. Cette méthode est notamment utilisée par le protocole Cellular Vehicle to Everything (C-V2X) [18].
- (a) **Avantage** : Facilite la gestion du réseau quand il y a beaucoup de véhicules.
 - (b) **Défi** : Le chef du groupe peut devenir un point de congestion.
5. **Routage multi-chemins (Multi-Path Routing)** : Cette technique envoie les données via plusieurs itinéraires différents pour garantir la livraison même si un chemin échoue. Le protocole Ad hoc On-demand Multipath Distance Vector Routing (AOMDV) utilise cette méthode [19].
- (a) **Avantage** : Améliore la résilience du réseau.
 - (b) **Défi** : Peut générer plus de trafic à cause des données envoyées en double.
6. **Optimisation basée sur les algorithmes d'IA (AI-Based Optimization)** : L'IA est utilisée pour adapter les routes en fonction du trafic en temps réel. Par exemple, les réseaux de neurones (Neural Networks -NN) peuvent prédire les embouteillages et ajuster les itinéraires automatiquement [20].
- (a) **Avantage** : S'adapte constamment aux changements du réseau pour optimiser les performances.
 - (b) **Défi** : Exige des ressources de calcul importantes.
7. **Routage basé sur les foules (Crowd-based Routing)** : Cette méthode utilise les informations collectées auprès des utilisateurs, via leurs smartphones ou applications, pour améliorer les décisions de routage. C'est particulièrement utile pour les véhicules autonomes [21].
- (a) **Avantage** : Permet d'obtenir des données en temps réel plus précises.
 - (b) **Défi** : Dépend fortement de la participation et de la qualité des données des utilisateurs.

1.4 Apprentissage Automatique et Optimisation

ML est une branche de l'IA qui vise à développer des systèmes capables d'apprendre et de s'adapter sans intervention humaine directe. Il joue un rôle clé dans l'optimisation des systèmes de routage, permettant d'anticiper les congestions, de gérer efficacement les ressources, et d'améliorer les performances globales du réseau en temps réel.

1.4.1 Introduction à l'Apprentissage Automatique

L'apprentissage automatique (ou machine learning, ML) est une branche de l'intelligence artificielle (IA). Selon l'American Association for Artificial Intelligence (AAAI) et l'ouvrage de Cornuéjols et al. (2002), l'IA se définit comme la capacité d'un système à acquérir et intégrer de manière autonome des connaissances [22].

Cette discipline regroupe l'ensemble des méthodes permettant de construire un modèle représentatif de la réalité à partir de données, soit en améliorant un modèle existant, soit en en créant un nouveau [22].

Le machine learning permet ainsi à une machine d'évoluer et d'accomplir des tâches liées à l'IA grâce à un processus d'apprentissage. Ce processus vise à optimiser le système en fonction de l'environnement, de l'expérience acquise et des résultats observés [22].

L'apprentissage est une capacité naturelle chez les êtres vivants, notamment chez l'humain, qui apprend à parler, lire, écrire ou encore à résoudre des problèmes complexes, tels que la reconnaissance de caractères manuscrits. La grande diversité des formes rend la simple mémorisation inefficace, d'où l'importance de la généralisation à partir d'exemples [22].

1.4.2 Algorithmes d'Apprentissage Automatique pour le Routage

Les algorithmes de ML jouent un rôle croissant dans le routage, en particulier pour optimiser l'acheminement du trafic dans les réseaux informatiques, les réseaux de transport, et même les réseaux de télécommunication. Voici un aperçu de quelques algorithmes de ML pertinents pour le routage.

1. Apprentissage par Renforcement (RL)

- (a) **Q-Learning** : Q-Learning est une approche populaire d'RL. Dans le routage, chaque routeur ou nœud peut être modélisé comme un agent qui essaie de maximiser une récompense cumulative en choisissant les chemins optimaux [23].
- (b) **Deep Q-Networks (DQN)** : Combinaison de Q-Learning et de réseaux neuronaux profonds (Deep Neural Networks - DNN) pour gérer de grands espaces d'états, particulièrement utile dans les réseaux étendus [23].
- (c) **SARSA et Policy Gradient** : D'autres méthodes d'RL, comme SARSA ou les méthodes basées sur le gradient de politique, peuvent également être adaptées pour des environnements de routage dynamiques et avec incertitude [23].

2. Apprentissage Supervisé

- (a) **Forêts Aléatoires et Arbres de Décision** : Utilisés pour prédire la congestion sur les chemins ou les temps de latence, à partir de données historiques [24].
- (b) **NN** : Les NN peuvent modéliser des relations complexes dans le réseau et prévoir les meilleures routes en fonction de paramètres en temps réel, comme la congestion ou la latence [24].
- (c) **Support Vector Machines (SVM)** : SVM peut être utilisé pour classer les chemins en fonction de leurs performances anticipées, utile pour le routage basé sur QoS [24].

3. Algorithmes d'Apprentissage Non-Supervisé

- (a) **K-Means** : Pour le regroupement de données de trafic similaires et l'optimisation de la bande passante en temps réel [25].
- (b) **Algorithmes de clustering comme Density-Based Spatial Clustering of Applications with Noise (DBSCAN)** : Utilisé pour détecter des anomalies dans le trafic réseau, par exemple pour repérer des changements soudains dans la demande ou des pannes dans le réseau [25].

4. Apprentissage par Renforcement Multi-Agent (MARL) : Dans les réseaux distribués, plusieurs agents (comme les routeurs ou commutateurs) prennent

des décisions indépendantes. Le MARL permet aux agents d'apprendre de façon coopérative ou compétitive pour optimiser le routage global. [26]

5. **Recurrent Neural Networks (RNN) et Long Short-Term Memory (LSTM)**

Les réseaux de neurones récurrents (RNN) et la mémoire à long terme (LSTM) sont utilisés pour prédire les tendances du trafic en se basant sur les données historiques. Cette capacité est essentielle pour ajuster le routage en fonction des flux anticipés [27].

6. **Algorithmes Génétiques et Optimisation Evolutionnaire** sont employés pour identifier les meilleures combinaisons de chemins dans des réseaux complexes, où les méthodes déterministes échouent à cause de l'interaction complexe entre les nœuds. Ces méthodes d'apprentissage automatique permettent aux systèmes de routage de s'adapter de manière dynamique, de prédire les zones de congestion, de réduire les délais de latence et d'optimiser l'utilisation des ressources. Dans les applications de routage en temps réel, des algorithmes tels que le RL et DNNs sont particulièrement prometteurs, notamment dans les réseaux très dynamiques tels que les réseaux ad hoc véhiculaires (VANETs) ou les réseaux de capteurs [28].

Ces méthodes ML permettent aux systèmes de routage de s'adapter dynamiquement, d'anticiper les congestions, de réduire les temps de latence et d'optimiser l'utilisation des ressources. Dans les applications de routage en temps réel, des algorithmes comme le RL et DNNs sont particulièrement prometteurs, surtout pour les réseaux hautement dynamiques comme les réseaux Vehicular Ad-hoc Networks (VANETs) ou les réseaux de capteurs.

1.4.3 **Avantages de l'Utilisation de l'Apprentissage Automatique**

ML présente de nombreux avantages dans le domaine des algorithmes de routage, notamment :

1. **Meilleure évolutivité** : Les réseaux deviennent de plus en plus complexes avec l'augmentation du trafic et des appareils connectés. Les techniques de ML, comme les réseaux neuronaux graphiques (Graph Neural Networks - GNN), permettent

de gérer ces complexités en modélisant le réseau sous forme de graphe. Cela permet une gestion efficace des ressources même dans des réseaux à grande échelle, garantissant ainsi que les performances ne se dégradent pas avec l'augmentation de la taille du réseau [29].

2. **Adaptation rapide** : Dans un environnement de réseau en constante évolution, où les conditions peuvent changer rapidement (comme la congestion, les pannes de lien ou les pics de trafic), les algorithmes de routage basés sur ML peuvent s'adapter en temps réel. Par exemple, les algorithmes d'apprentissage supervisé peuvent ajuster les chemins de transmission en fonction des données en temps réel, ce qui permet de maintenir des performances optimales et de minimiser les interruptions [29].
3. **Réduction de la congestion** : L'un des problèmes majeurs dans les réseaux est la congestion, qui peut entraîner des retards et des pertes de paquets. Les techniques de ML, comme DNN, sont capables de prédire les zones de congestion et de rediriger le trafic de manière proactive pour éviter ces goulots d'étranglement. Cela permet une transmission plus fluide des données, même dans des conditions de trafic élevé [29].
4. **Optimisation de l'utilisation des ressources** : Les réseaux doivent souvent fonctionner sous des contraintes de ressources, telles que la bande passante et la capacité de traitement. Les algorithmes de ML peuvent optimiser l'utilisation des ressources en allouant dynamiquement la bande passante et en ajustant les routes pour minimiser les délais et maximiser le débit global du réseau. Cela améliore l'efficacité du réseau, notamment dans les environnements où les ressources sont limitées [29].
5. **Flexibilité et adaptation des politiques de routage** : Les algorithmes centralisés de routage basés sur ML permettent une grande flexibilité dans la modification et l'ajustement des politiques de routage. Grâce à une vue globale de l'état du réseau, il est possible d'ajuster les routes en fonction des changements de conditions en temps réel, ce qui améliore non seulement la résilience du réseau mais aussi sa capacité à répondre aux nouvelles exigences en matière de QoS et de sécurité [29].

Ces avantages contribuent à rendre les systèmes de routage plus efficaces, réactifs et

adaptés aux besoins des utilisateurs, tout en améliorant l'expérience de conduite globale.

1.5 Défis et Enjeux

Les défis et enjeux liés à l'IoV sont nombreux et complexes, car cette technologie repose sur une infrastructure robuste et des systèmes interconnectés qui doivent assurer à la fois sécurité, efficacité et fiabilité.

1.5.1 Défis Technologiques

L'IoV nécessite l'intégration de différentes technologies pour permettre une communication efficace entre les véhicules et les infrastructures. Les défis incluent la gestion d'énormes volumes de données en temps réel, la faible latence pour assurer une réactivité rapide, et l'interopérabilité entre divers fabricants et systèmes. Le besoin de réseaux plus performants, comme la 5G, est essentiel pour garantir une connexion continue et stable [30] [31].

1.5.2 Enjeux de Sécurité

Les véhicules connectés sont confrontés à de nombreux risques, notamment les cyberattaques visant à accéder à des données sensibles ou à interférer avec les systèmes de contrôle des véhicules. La protection de ces systèmes requiert des méthodes avancées de sécurité, telles que IA et ML, pour identifier et neutraliser les menaces rapidement. La gestion de la confiance entre les véhicules est également cruciale pour garantir la fiabilité du système [30] [31].

1.5.3 Considérations Éthiques

Pour que l'IoV soit adopté à grande échelle, il est nécessaire de développer des infrastructures routières intelligentes et de standardiser les protocoles de communication. Les utilisateurs devront aussi être sensibilisés aux avantages et aux risques potentiels liés à ces véhicules, pour encourager une acceptation sociale. L'élaboration de cadres

réglementaires adaptés est essentielle pour protéger la vie privée et assurer la sécurité des données [31].

1.6 Conclusion

L'optimisation du routage dans l'IoV est essentielle pour améliorer les performances des réseaux véhiculaires. Face à des défis tels que la mobilité rapide et la congestion, les algorithmes d'apprentissage automatique offrent des solutions prometteuses en analysant les données en temps réel pour optimiser les routes. Ce chapitre a exploré les techniques d'optimisation du routage et les avantages de ML. Malgré les progrès, des défis technologiques, de sécurité, et d'éthique subsistent, nécessitant une attention particulière pour garantir des communications fiables et sécurisées.

Le prochain chapitre de ce mémoire présentera l'état de l'art de notre travail.

Chapitre 2

État de l'art

2.1 Introduction

Les réseaux IoV peuvent être vus comme une évolution des VANETs, car ils intègrent des technologies plus avancées pour permettre une communication plus efficace et plus sûre entre les véhicules et les infrastructures. Ces réseaux sont essentiels pour optimiser la gestion du trafic, prévenir les accidents et offrir des services connectés de haute qualité aux conducteurs. Néanmoins, de nombreux obstacles demeurent. L'acheminement des données dans ces réseaux est particulièrement complexe en raison de la vitesse élevée des véhicules, des modifications fréquentes de la topologie du réseau et des exigences strictes en matière de qualité de service. Pour surmonter ces obstacles, diverses méthodes ont été mises en place. Certaines reposent sur des protocoles de routage conventionnels, tandis que d'autres font appel à des techniques plus récentes et plus sophistiquées. Ce chapitre débutera par une analyse détaillée des protocoles de transmission de données les plus répandus dans le domaine de IoV, en mettant en lumière leurs principes fondamentaux, les critères qu'ils emploient et leur classification. Ensuite, Nous proposerons une taxonomie des grandes familles d'approches, suivie d'une comparaison entre leurs performances. Enfin, on discutera des limites actuelles et des défis encore à relever, en mettant en avant comment l'intelligence artificielle peut justement aider à améliorer le routage dans les réseaux IoV.

2.2 Méthodologie de Recherche

Dans ce travail, on a essayé de rester le plus simple et logique possible pour aborder le problème du routage dans les réseaux IoV. On a utilisé des techniques de machine learning pour essayer d'améliorer les choses, mais avant tout, on a suivi une méthode en trois étapes : d'abord on s'est documenté à fond, ensuite on a construit un modèle basé sur ce qu'on a appris, et enfin on a testé ce modèle pour voir s'il tenait la route.

2.2.1 Approche Générale Adoptée pour la Revue de la Littérature

Avant de se lancer dans les détails techniques, on a pris le temps de bien comprendre ce qui existe déjà dans ce domaine. C'était important pour ne pas repartir de zéro et pour éviter de refaire ce qui a déjà été fait. Voilà comment on a procédé :

- On a commencé par les bases, en regardant les protocoles de routage classiques, comme Ad hoc On-Demand Distance Vector (AODV) et Dynamic Source Routing (DSR). Ces protocoles sont connus et beaucoup utilisés dans les réseaux ad hoc, donc c'était logique de commencer par là.
- Ensuite, on s'est intéressé à ce que le machine learning pouvait apporter, en cherchant des études qui utilisent par exemple des réseaux de neurones ou l'apprentissage par renforcement pour rendre le routage plus intelligent.
- Enfin, on a cherché à voir ce qui clochait dans les approches actuelles. On a remarqué que beaucoup de méthodes classiques avaient du mal à s'adapter quand les conditions du réseau changent rapidement, ce qui est fréquent dans un environnement avec des véhicules en mouvement.

2.2.2 Sources Utilisées pour l'Analyse

Pour assurer la pertinence et la fiabilité des informations collectées nous avons exploité plusieurs bases des données académique ainsi que des plateformes de recherche reconnues, La méthodologie adoptée repose sur une sélection rigoureuse des sources, comme détaillé ci-dessous.

— Bases de Données Consultées

Dans le cadre de cette étude, nous avons effectué une recherche approfondie en consultant plusieurs bases de données académiques afin d'identifier les travaux les plus pertinents en matière d'optimisation du routage dans l'IoV et l'intégration de ML dans ce domaine. La sélection des articles s'est basée sur une revue de la littérature existante ainsi que sur une exploration ciblée des approches récentes.

Nous avons privilégié les articles publiés dans des revues scientifiques et conférences spécialisées disponibles sur des plateformes telles que IEEE Xplore, ScienceDirect et SpringerLink, qui offrent une large gamme de publications traitant des réseaux véhiculaires intelligents et des techniques de ML appliquées au routage. En complément, nous avons consulté Google Scholar et ResearchGate pour accéder à des études récentes et à des discussions entre chercheurs spécialisés dans le domaine.

Afin de cibler efficacement notre recherche, nous avons employé des termes spécifiques en lien direct avec notre problématique, en combinant des mots-clés relatifs aux réseaux IoV, aux protocoles de communication, ainsi qu'aux algorithmes de machine learning tels que VANETs, IoT, IoV, ITS, etc. Les résumés des articles sélectionnés ont été soigneusement analysés afin d'en évaluer la pertinence avant leur inclusion dans notre étude. Cette démarche nous a permis d'identifier les travaux les plus récents dans le domaine et de constituer une base de connaissances solide pour la conception de notre propre modèle d'optimisation du routage.

2.2.3 Critères d'Évaluation des Protocoles et Approches

L'évaluation des protocoles dans les réseaux IoV repose sur plusieurs critères permettant d'analyser leur efficacité et leur adaptation aux environnements dynamiques. Ces critères, tels que la latence, la densité, la stabilité, la surcharge, le chevauchement des clusters et la complexité, sont essentiels pour mesurer la performance des approches proposées. Leur analyse permet de comparer les protocoles et de choisir ceux qui répondent le mieux aux exigences du réseau, en optimisant la connectivité, la fiabilité et l'utilisation des ressources.

2.2.3.1 Latence

La latence désigne le délai entre l'envoi d'un paquet de données et sa réception par le destinataire. Elle est influencée par le nombre de sauts, la distance entre les nœuds, et la congestion du réseau. Une faible latence est essentielle pour les applications en temps réel, comme la prévention des collisions ou la conduite autonome.

2.2.3.2 Densité

La densité d'un réseau représente le nombre de nœuds présents dans une zone spécifique. Une forte densité améliore la connectivité mais peut entraîner des collisions de paquets et une congestion du canal. Une faible densité, en revanche, peut générer des problèmes de couverture et d'isolement des nœuds.

2.2.3.3 Stabilité

La stabilité représente la capacité d'un protocole de routage à maintenir des chemins de communication efficaces et cohérents sur une période prolongée. Elle est généralement évaluée en fonction de la fréquence des changements de routes et du taux de reconfiguration du réseau. Une bonne stabilité permet une transmission plus fluide des données, limite les interruptions et réduit la consommation de ressources engendrée par les mises à jour fréquentes des tables de routage.

2.2.3.4 Surcharge (Overhead)

La surcharge correspond au volume de messages de contrôle nécessaires pour maintenir la structure du réseau. Un protocole efficace doit minimiser cette charge afin de ne pas surcharger la bande passante ni consommer trop de ressources énergétiques.

2.2.3.5 Chevauchement (Overlapping)

Le chevauchement désigne la présence de nœuds appartenant simultanément à plusieurs zones ou routes de communication. Bien qu'un chevauchement modéré puisse favoriser l'interconnexion et la résilience du réseau, un chevauchement excessif complique la gestion du routage, augmente la redondance des transmissions et peut engendrer une surcharge inutile sur certains nœuds.

2.2.3.6 Complexité

La complexité d'un protocole mesure la quantité de ressources computationnelles et mémoires requises pour son exécution. Elle est souvent exprimée en nombre d'opérations par nœud et détermine la capacité du protocole à fonctionner efficacement dans des réseaux de grande échelle.

2.3 Taxonomie des Protocoles Conventionnels de Routage

Les protocoles de transmission de données jouent un rôle clé dans l'acheminement efficace des informations à travers un réseau. Parmi les différentes catégories, ceux qui s'appuient sur la position géographique des nœuds permettent d'optimiser le routage. On distingue notamment les dits géographiques, qui utilisent directement les coordonnées des destinations, et ceux à base de zones, qui segmentent le réseau en régions pour structurer l'envoi des données. Cette classification aide à mieux cerner leurs différences et leur impact sur la performance des réseaux. La Figure 2.1 synthétise cette classification en présentant une vue d'ensemble des principales catégories de protocoles conventionnels selon leurs approches de transmission.

2.3.1 Protocoles de Routage Basés sur la Position

Les protocoles de routage basés sur la position sont essentiels dans VANETs et l'IoV, car ils utilisent directement la localisation des véhicules pour transmettre les données. Contrairement aux méthodes classiques qui reposent sur des tables de routage statiques, ces protocoles prennent des décisions en temps réel en fonction de la position des nœuds, ce qui les rend plus adaptés à l'environnement dynamique des véhicules en mouvement. Cependant, ils doivent surmonter plusieurs défis, comme les interférences sur les canaux de communication et les variations de la connectivité. Pour mesurer leur performance, des outils comme NS-2, NS-3 et SUMO sont utilisés afin d'évaluer des aspects comme la rapidité et la fiabilité de la transmission des informations. [32].

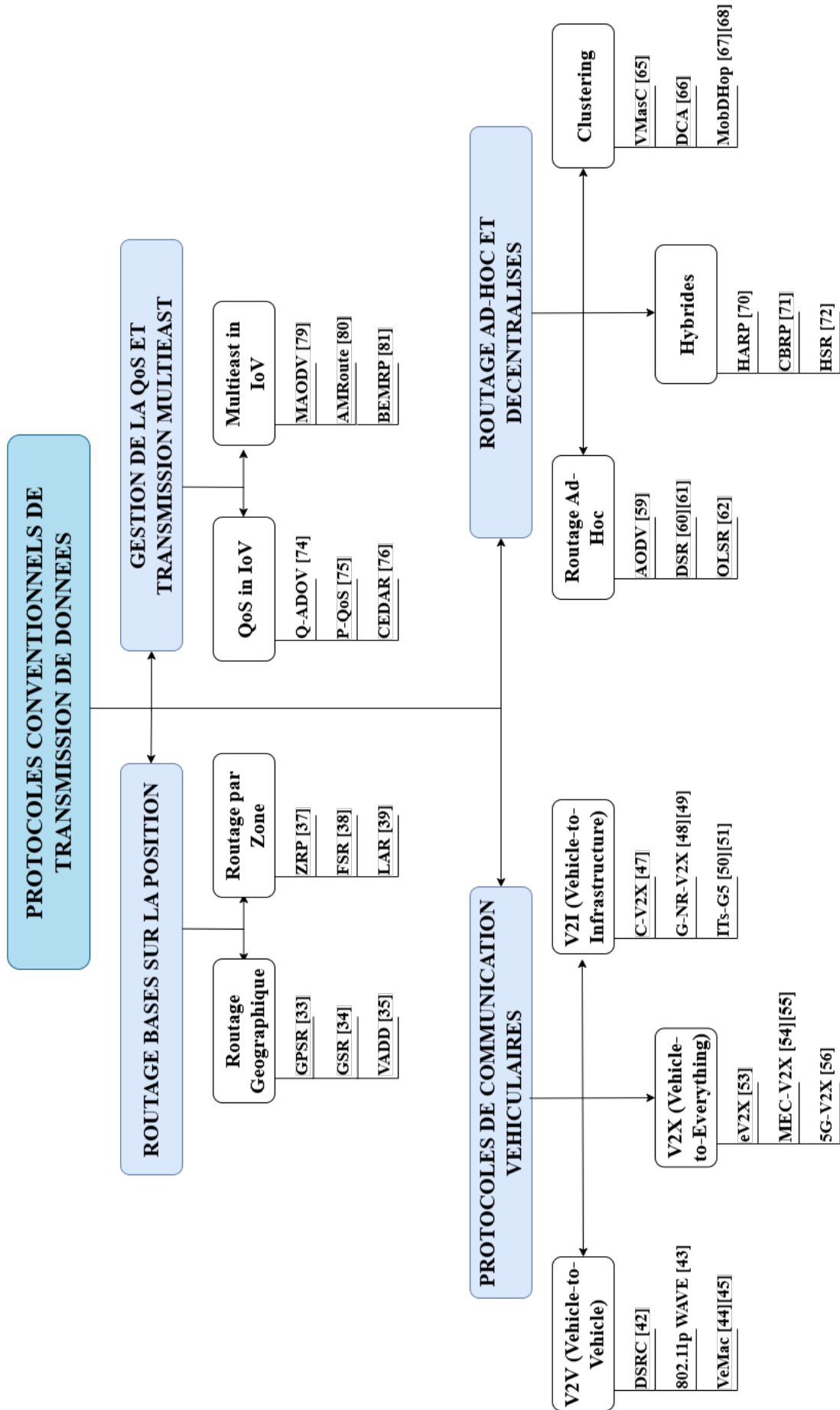


FIGURE 2.1 – Taxonomie des Protocoles de Routage.

2.3.1.1 Protocoles de Routage Géographique (Geo-Routing)

Les protocoles de routage fondés sur la position géographique s'imposent comme des solutions efficaces dans les réseaux VANET, en exploitant directement la localisation des véhicules. Contrairement aux approches classiques basées sur des tables statiques, ces protocoles adaptent le chemin des données en fonction de la position actuelle des nœuds, ce qui les rend particulièrement adaptés aux environnements mobiles et dynamiques [33].

1. GPSR

Le protocole GPSR, repose sur une combinaison de deux techniques. D'abord, la méthode Greedy Forwarding, qui consiste à faire transiter les paquets via le voisin le plus proche de la destination. Si cette approche échoue, GPSR bascule sur la stratégie Perimeter Forwarding, qui contourne les zones bloquées à l'aide d'un algorithme inspiré de la règle de la main droite. L'un des avantages majeurs de GPSR réside dans l'absence de tables de routage complexes, ce qui en fait un choix judicieux pour les réseaux où la topologie évolue rapidement. Une version avancée, appelée Advanced GPSR (AGPSR), a été proposée afin de renforcer la performance : elle introduit des critères supplémentaires, tels que la vitesse des véhicules ou la densité du trafic, pour améliorer la livraison des paquets et réduire les délais [34].

2. GSR

Conçu pour les milieux urbains, le protocole Geographic Source Routing (GSR) combine un service de localisation réactif (RLS) avec des cartes routières numériques pour tracer le chemin entre la source et la destination. L'algorithme de Dijkstra permet alors de sélectionner les meilleures intersections à suivre. Bien que ce protocole permette une planification efficace des trajets, il présente une limite importante : il ne prend pas en compte la densité du trafic. Cela peut engendrer des retards, voire une saturation des itinéraires choisis [35].

3. VADD

Le protocole Vehicle-Assisted Data Delivery (VADD) adopte une stratégie opportuniste pour pallier les interruptions de communication. Lorsqu'un véhicule ne trouve pas de relai immédiat, il conserve temporairement le paquet et attend une meilleure opportunité pour le transmettre. Cette approche, connue sous le

nom de Carry-and-Forward (CAF), est particulièrement utile dans les zones où les connexions sont peu fréquentes. Contrairement à d'autres solutions opportunistes comme MaxProp ou SKVR, VADD intègre une analyse des schémas de déplacement des véhicules afin d'anticiper les itinéraires les plus prometteurs, ce qui favorise une transmission plus rapide et plus fiable des données [36].

2.3.1.2 Protocoles de Routage par Zone Géographique (Zone-based Routing)

Ce sont des solutions hybrides conçues pour améliorer l'efficacité du routage dans les réseaux dynamiques comme les Unmanned Aerial Vehicle (UAV). Leur principe repose sur une division du réseau en plusieurs zones. À l'intérieur d'une zone, le routage fonctionne de manière proactive, c'est-à-dire que les chemins sont préétablis et régulièrement mis à jour. En revanche, lorsqu'un nœud doit envoyer des données vers une destination en dehors de sa zone, un routage réactif est utilisé, ne cherchant un chemin que lorsqu'il est nécessaire. Cette approche permet de réduire la surcharge des mises à jour globales tout en garantissant une transmission efficace des données, même dans des environnements où les connexions sont instables ou sujettes à des interruptions fréquentes [37].

1. ZRP

Le Zone Routing Protocol (ZRP) est un protocole de routage hybride conçu pour les réseaux Mobile Ad hoc (MANETs) et largement utilisé dans les VANETs. Son fonctionnement repose sur une division du réseau en zones locales, permettant une gestion plus efficace du routage.

À l'intérieur de chaque zone, le protocole adopte une approche proactive, où chaque nœud stocke et met à jour régulièrement les routes vers les autres nœuds proches. Pour les communications en dehors de la zone, une méthode réactive est utilisée afin de limiter la diffusion des requêtes de routage et d'optimiser la bande passante.

Cette combinaison offre un équilibre entre réactivité et efficacité, réduisant la latence tout en minimisant la surcharge du réseau. Grâce à ces caractéristiques, ZRP est particulièrement adapté aux environnements dynamiques comme VANETs, où la mobilité élevée des véhicules rend les protocoles classiques moins

performants [38].

2. FSR

Le protocole Fisheye State Routing (FSR) est une méthode de routage proactive utilisée dans les réseaux ad hoc, notamment dans les VANETs. Il repose sur une mise à jour progressive des tables de routage, où les informations concernant les nœuds proches sont actualisées fréquemment, tandis que celles des nœuds plus éloignés le sont moins souvent. Ce mécanisme permet de limiter la surcharge de transmission des mises à jour et d'optimiser l'utilisation de la bande passante. Toutefois, dans des réseaux à forte mobilité, la modification rapide de la topologie peut provoquer des erreurs de transmission, en particulier pour les destinations lointaines. Par rapport aux protocoles réactifs comme DSR, FSR peut afficher une latence plus élevée et une diminution du débit lorsque le nombre de nœuds augmente [39].

3. LAR

Le Location-Aided Routing (LAR) est un protocole de routage réactif utilisé dans VANETs, exploitant les données de localisation pour améliorer l'efficacité du routage. Contrairement aux protocoles classiques qui diffusent largement les requêtes de recherche de route, LAR limite cette diffusion à une zone spécifique, basée sur la position estimée du véhicule de destination.

Cette approche permet de réduire la surcharge du réseau, d'accélérer la découverte des routes et d'optimiser la consommation des ressources. Cependant, dans un environnement VANET, où les véhicules se déplacent rapidement et où les obstacles urbains peuvent affecter la précision du GPS, LAR peut rencontrer des difficultés pour maintenir une efficacité optimale. Pour pallier ces limites, des améliorations ont été proposées, telles que des mises à jour périodiques des positions ou des techniques de clustering, permettant d'ajuster dynamiquement la zone de recherche et d'améliorer la fiabilité du routage. [40].

2.3.1.3 Comparaison et Limites des Protocoles Basés sur la Position

Le Tableau 2.1 présente une comparaison des protocoles de routage basés sur la position, en analysant les protocoles géographique et par zone géographique selon des critères clés.

1. Limitations de Protocoles de Routage Géographique (Geo-Routing)

- **Problème du maximum local** : Un nœud peut ne pas avoir de voisin plus proche de la destination, entraînant une impasse dans la transmission des paquets.
- **Dépendance à la précision du GPS** : La précision de la position est essentielle, mais elle peut être affectée par des erreurs de signal.
- **Vulnérabilité aux interférences et aux obstacles** : Les bâtiments et tunnels peuvent bloquer les signaux GPS et affecter le routage.

2. Limitations de Protocoles de Routage par Zone Géographique (Zone-based Routing))

- **Latence élevée pour le routage inter-zone** : La commutation entre le routage proactif et réactif introduit des retards.
- **Surcharge de gestion des zones** : Nécessite des mises à jour fréquentes des tables de routage, augmentant la consommation des ressources réseau.

TABLE 2.1 – Comparaison des protocoles basés sur la position

Type	Protocole	Densité	Stabilité	Latence	Surcharge	Overlapping	Complexité
Geo-Routing	GPSR [34]	Moyenne	Faible	Faible à Moyenne	Élevée	Faible	Moyenne
	GSR [35]	Moyenne	Moyenne	Moyenne à Élevée	Moyenne	Moyenne	Moyenne
	VADD [36]	Faible	Élevée	Élevée	Moyenne	Faible	Moyenne à Élevée
Zone-based Routing	ZRP [38]	Moyenne	Faible	Moyenne	Élevée	Moyenne	Moyenne
	FSR [39]	Moyenne	Moyenne	Moyenne à Élevée	Moyenne	Moyenne	Moyenne
	LAR [40]	Faible	Élevée	Élevée	Moyenne	Faible	Élevée

2.3.2 Protocoles de Communication Véhiculaires

Les protocoles de communication véhiculaire sont cruciaux dans les systèmes de transport intelligents car ils facilitent la transmission d'informations entre les véhicules et leur environnement. Ces technologies se regroupent en trois catégories principales : la communication V2V, qui autorise un échange direct de données en temps réel entre les véhicules ; la communication V2I, établissant une connexion entre les véhicules et les infrastructures routières afin d'optimiser la gestion du trafic ; et la communication Vehicle-to-Everything (V2X), qui englobe toutes les formes d'interactions, y compris

celles avec les piétons et les objets connectés. Ces protocoles ont pour objectif d'améliorer la sécurité routière, d'accroître l'efficacité du trafic et d'optimiser la connectivité des véhicules autonomes, ce qui favorise le développement des transports intelligents.

2.3.2.1 Protocoles de Communication V2V

Les protocoles de communication véhicule à véhicule (V2V) offrent la possibilité aux automobiles de transmettre directement des données cruciales relatives au trafic et aux circonstances routières, sans avoir recours à une infrastructure fixe. Ces systèmes sont élaborés pour renforcer la sécurité routière en facilitant la transmission rapide des alertes relatives aux accidents, aux freinages soudains ou encore aux obstructions sur la chaussée. Néanmoins, la crédibilité de ces protocoles est conditionnée par plusieurs éléments tels que la stabilité des liaisons, le temps de latence et les perturbations susceptibles d'influer sur la qualité des communications [41].

De plus, la protection des transmissions constitue un défi majeur, étant donné que ces échanges sont exposés à différentes attaques informatiques. Des moyens de chiffrement, d'authentification et de détection d'intrusions ont été instaurés pour sécuriser les transmissions d'informations entre véhicules, assurant ainsi une communication plus sûre et digne de confiance [42].

1. DSRC

Le DSRC, abréviation de Dedicated Short-Range Communication, représente une technologie de communication sans fil développée spécifiquement pour les véhicules connectés dans le but de faciliter la transmission rapide d'informations relatives à la circulation routière. En se conformant à la norme IEEE 802.11p, ce système opère à une fréquence de 5,9 GHz et emploie la technique de multiplexage par répartition Orthogonal Frequency Division Multiplexing (OFDM) afin d'assurer une transmission efficace, même en cas de mobilité élevée. Son architecture repose sur divers canaux de communication, chacun étant adapté à des besoins spécifiques tels que la sécurité routière et la gestion du trafic. Un des principaux atouts du DSRC réside dans sa faible latence, ce qui le rend spécialement approprié pour des applications critiques telles que les avertissements de collision ou les alertes de freinage d'urgence. Cependant, l'implémentation de cette technologie peut être confrontée à des défis en termes de gestion des inter-

férences et de congestion du spectre, particulièrement dans des environnements urbains denses où de multiples véhicules communiquent simultanément. Ainsi, des recherches sont actuellement menées pour comparer le DSRC à d'autres solutions telles que le C-V2X, dans le but d'identifier la stratégie optimale pour les véhicules autonomes et les infrastructures intelligentes [1].

2. 802.11p (WAVE)

Le 802.11p, connu sous le nom de WAVE, constitue une extension de la norme Wireless Fidelity (Wi-Fi) conçue spécifiquement pour les communications V2V et V2I. Conçu pour opérer dans des environnements routiers changeants, ce système utilise la fréquence de 5,9 GHz et est basé sur une architecture dérivée du DSRC. Contrairement aux technologies Wi-Fi classiques, la norme 802.11p a été conçue pour gérer efficacement des connexions de courte durée et à haute mobilité, ce qui facilite les échanges d'informations critiques entre les véhicules, tels que les alertes de sécurité.

La précision et l'état du trafic sont améliorés, tout en réduisant la latence. Il renforce également la fiabilité des communications en incorporant des mécanismes de gestion des interférences et des pertes de paquets, indispensables pour garantir un fonctionnement optimal dans des environnements de circulation réelle. Les caractéristiques de la norme 802.11p en font un élément essentiel des systèmes de systèmes de Transport Intelligents (ITS), contribuant ainsi à la progression des infrastructures connectées et des véhicules autonomes [43].

3. VeMAC

Le protocole Vehicular MAC Protocol (VeMAC) est une solution appropriée pour les VANETs, permettant de gérer de manière efficace l'accès au canal de communication. Il utilise une méthode d'accès multiple par division de temps (TDMA), qui segmente le temps en intervalles assignés aux véhicules en fonction de leur trajectoire. Cette technique contribue à minimiser les interférences et à renforcer la fiabilité des communications entre les véhicules et les RSUs [44].

Cependant, dans des contextes de circulation dense, la gestion des créneaux s'avère plus ardue, ce qui peut susciter un risque augmenté d'accidents et restreindre les performances du réseau. Malgré ces obstacles, VeMAC persiste à être une option efficace par rapport aux protocoles conventionnels, en privilégiant

un transfert plus stable et conforme aux besoins des communications véhiculaires [45].

2.3.2.2 Protocoles de Communication V2I

Le protocole de communication V2I est une technologie de communication qui permet aux véhicules d'échanger des informations avec les infrastructures routières, telles que les feux de circulation, les caméras de surveillance, les capteurs routiers et les panneaux de signalisation intelligents. Cette interaction vise à améliorer la sécurité routière, optimiser la gestion du trafic et réduire la consommation de carburant grâce à une communication en temps réel entre les véhicules et leur environnement.

Le V2I améliore la sécurité routière en permettant aux véhicules de communiquer avec les infrastructures routières. Il comprend des systèmes tels que l'évitement des collisions, qui envoie des alertes aux conducteurs pour prévenir les accidents, l'avertissement de non-respect des feux rouges, qui aide à éviter les infractions aux intersections, et la détection des incidents routiers, qui analyse le trafic en temps réel pour signaler les dangers potentiels. Grâce à ces applications, le V2I contribue à une conduite plus sécurisée et à une réduction significative des accidents [46].

1. C-V2X

Le protocole C-V2X est une technologie de communication conçue pour permettre l'échange d'informations en temps réel entre les véhicules, les infrastructures routières, V2P et V2N. Il se décline en deux versions principales : Long-Term Evolution Vehicle-to-Everything (LTE-V2X), qui repose sur la 4G, et New Radio Vehicle-to-Everything (NR-V2X), basé sur la 5G, offrant ainsi une amélioration significative en termes de latence réduite, de fiabilité accrue et de gestion optimisée des interférences. Le fondamental du C-V2X est l'allocation des ressources radio, qui joue un rôle clé dans la gestion efficace du spectre et l'optimisation des communications. Le mode sidelink permet aux véhicules de communiquer directement entre eux sans nécessiter d'infrastructure réseau, améliorant ainsi la rapidité des échanges, en particulier dans les environnements où une connectivité stable n'est pas garantie.

Par la sécurité est un enjeu majeur du C-V2X. Des mécanismes avancés d'authentification et de gestion des identités ont été développés pour garantir la confi-

dentialité et l'intégrité des données échangées. Ces solutions visent à protéger les utilisateurs contre les attaques potentielles, telles que l'usurpation d'identité et l'interception des communications . Grâce à ces avancées s'impose comme une technologie clé pour le développement des véhicules autonomes et des infrastructures intelligentes, contribuant à une gestion plus efficace et sécurisée du trafic routier [47].

2. 5G NR-V2X

Le 5G New Radio for V2X (5G NR-V2X) est une technologie avancée de communication véhiculaire introduite avec la norme 3GPP Release 16, visant à améliorer la connectivité entre les véhicules, les infrastructures, les piétons et les réseaux [48].

Contrairement aux générations précédentes comme LTE-V2X, il exploite les capacités de la 5G, notamment une bande passante élargie, une latence ultra-faible et une gestion optimisée du spectre radio, essentielles pour les applications critiques telles que la conduite autonome et les systèmes avancés d'aide à la conduite Advanced Driver Assistance Systems (ADAS). Grâce à son mode sidelink amélioré, les véhicules peuvent communiquer directement sans dépendre d'une station de base, réduisant ainsi la latence et améliorant la fiabilité des échanges [49].

De plus, le 5G NR-V2X intègre des technologies avancées telles que le Network Slicing, qui permet d'attribuer dynamiquement des ressources aux différentes applications, ainsi que le Multiple Input Multiple Output (MIMO) massif et l'Edge Computing, garantissant une transmission efficace et stable même dans des environnements à forte densité de trafic. En combinant ces innovations, 5G NR-V2X devient une pierre angulaire des réseaux de transport intelligents, offrant des performances accrues pour une mobilité plus sûre et plus efficace [48].

3. ITS-G5 (IEEE 802.11p en Europe)

Le protocole ITS-G5 (IEEE 802.11p en Europe) est une technologie de communication sans fil dédiée aux systèmes de ITS, permettant l'échange d'informations entre véhicules V2V et avec les infrastructures routières V2I. Basé sur la norme IEEE 802.11p, il fonctionne sur la bande de fréquence 5,9 GHz et est conçu pour assurer une faible latence et une transmission fiable des messages critiques liés à la sécurité routière, comme les alertes de collision ou les conditions météorologiques

défavorables [50].

Contrairement aux technologies cellulaires comme le C-V2X, ITS-G5 ne dépend pas des réseaux d'opérateurs et permet une communication directe et décentralisée entre les véhicules, ce qui le rend particulièrement adapté aux environnements où la connectivité réseau peut être limitée [51].

Toutefois, bien que ce protocole offre une grande robustesse en milieu urbain et autoroutier, il présente certaines limitations en matière de portée et de gestion de la congestion du spectre radio, en particulier dans les zones à forte densité de trafic. Malgré ces défis, ITS-G5 reste une technologie clé pour l'amélioration de la sécurité routière et l'optimisation du trafic dans les infrastructures de transport intelligentes en Europe [50].

2.3.2.3 Protocoles de Communication V2X

Les protocoles de communication V2X permettent aux véhicules de communiquer avec divers éléments de leur environnement, notamment d'autres véhicules, les infrastructures routières, les piétons et les réseaux cellulaires. Cette technologie vise à améliorer la sécurité routière, l'efficacité du trafic et à faciliter le développement des véhicules autonomes. Grâce aux échanges en temps réel, les protocoles V2X permettent de transmettre des informations critiques telles que les alertes de collision, les conditions météorologiques ou les obstacles sur la route. Ils reposent sur deux principales technologies : le DSRC basé sur la norme IEEE 802.11p, et le C-V2X utilisant les réseaux 4G LTE et 5G pour assurer une couverture plus étendue et une communication plus fiable. Cependant, malgré leurs avantages, ces protocoles doivent relever plusieurs défis, notamment en matière de latence, de sécurité des données et de gestion des interférences dans les environnements à forte densité [52].

1. eV2X

Le protocole Enhanced V2X communication (eV2X) représente une évolution des communications V2X, intégrant les avancées des réseaux 5G pour améliorer la latence, la fiabilité et l'efficacité du partage de données entre les véhicules, les infrastructures et les autres éléments connectés. Grâce à des mécanismes optimisés de gestion des ressources radio, notamment l'approche Full Duplex Semi-Persistent Scheduling (FD-SPS), eV2X permet une transmission plus rapide et plus stable

des informations critiques, ce qui est essentiel pour les véhicules autonomes et les systèmes de transport intelligents. Ce protocole vise également à réduire les interférences et la congestion du réseau, en allouant dynamiquement les canaux de communication selon les besoins des véhicules et les conditions du trafic [53].

2. MEC-V2X

Le protocole Mobile Edge Computing for V2X (MEC-V2X) combine les technologies V2X et l'informatique en périphérie MEC pour améliorer la rapidité et l'efficacité des communications entre les véhicules et leur environnement. En intégrant des capacités de calcul à proximité des réseaux routiers, cette approche permet de réduire la latence des transmissions, un élément essentiel pour les véhicules autonomes et ITSs. Contrairement aux architectures cloud traditionnelles, où les données doivent transiter par des centres de données distants, MEC-V2X traite l'information localement, ce qui optimise la prise de décision en temps réel et améliore la fiabilité des échanges d'informations critiques, comme les alertes de collision ou les changements de signalisation [54].

De plus, cette technologie tire parti des réseaux 5G pour gérer efficacement un grand volume de connexions simultanées, garantissant ainsi une meilleure prise en charge des applications coopératives de conduite autonome et des services connectés à forte exigence de performance [55].

3. 5G-V2X

Le protocole 5G-V2X repose sur les technologies 5G pour améliorer les communications entre les véhicules et leur environnement, incluant les interactions V2V, V2I, V2P et V2N. Il offre des performances supérieures en matière de latence réduite, fiabilité accrue et capacité de transmission à haut débit, ce qui le rend particulièrement adapté aux véhicules autonomes et aux ITS. Contrairement aux générations précédentes comme le LTE-V2X, la 5G-V2X exploite le slicing réseau pour attribuer dynamiquement des ressources spécifiques aux applications critiques, garantissant ainsi une meilleure gestion des priorités et une qualité de service optimale. De plus, l'intégration du Mobile Edge Computing permet un traitement des données plus rapide en rapprochant les capacités de calcul des véhicules, réduisant ainsi la dépendance aux infrastructures cloud centralisées. Grâce à ces avancées, la 5G-V2X joue un rôle clé dans le développement des

communications intelligentes, facilitant la conduite coopérative et la gestion efficace du trafic en temps réel [56].

2.3.2.4 Comparaison et Limites des Protocoles de Communication Véhiculaires

Le Tableau 2.2 compare les protocoles de communication véhiculaires V2V, V2I et V2X en fonction de critères clés tels que la densité, la stabilité, la latence et la complexité. Cette analyse aide à choisir le protocole le plus adapté selon les besoins de performance et de fiabilité des réseaux véhiculaires.

1. Limitations de Protocoles de Communication V2V

- **Fiabilité dépendante de la densité des véhicules** : En trafic faible, le manque de relais peut empêcher la communication efficace.
- **Sensibilité aux attaques** : Vulnérable aux cyberattaques, notamment les attaques Sybil et le brouillage.
- **Latence et interférences** : Les connexions sans fil peuvent être affectées par l'interférence électromagnétique et la congestion du spectre.

2. Limitations Protocoles de Communication V2I

- **Infrastructure coûteuse** : Nécessite des investissements importants pour déployer et maintenir les équipements de communication.
- **Dépendance aux infrastructures fixes** : En cas de panne ou d'absence de couverture, le système devient inefficace.

3. Limitations de Protocoles de Communication V2X

- **Interopérabilité et standardisation** : La compatibilité entre différents constructeurs et systèmes reste un défi.
- **Sécurité et confidentialité des données** : L'échange de données sensibles en temps réel expose le système à des risques de piratage.

2.3.3 Protocoles de Routage Ad-Hoc et Décentralisés

Les protocoles de routage ad-hoc et décentralisés sont des mécanismes essentiels pour assurer la communication dans les réseaux sans fil mobiles, tels que VANETs et les

TABLE 2.2 – Comparaison des protocoles de communication Véhiculaires

Type	Protocole	Densité	Stabilité	Latence	Surcharge	Overlapping	Complexité
V2V	DSRC [1]	Moyenne	Moyenne	Faible	Moyenne	Faible	Moyenne
	802.11p (WAVE) [43]	Moyenne	Élevée	Faible	Moyenne	Faible	Élevée
	VeMAC [44], [45]	Élevée	Moyenne	Moyenne	Faible	Moyenne	Moyenne
V2I	C-V2X [47]	Élevée	Moyenne à élevée	Moyenne	Moyenne	Moyenne	Moyenne
	5G NR-V2X [48], [49]	Très Élevée	Très Élevée	Très Faible	Élevée	Moyenne	Très Élevée
	ITS-G5 [50], [51]	Moyenne	Moyenne	Faible	Faible à Moyenne	Élevée	Moyenne
V2X	eV2X [53]	Élevée	Élevée	Faible	Moyenne	Faible	Moyenne
	MEC-V2X [54], [55]	Élevée	Très Élevée	Très Faible	Moyenne	Moyenne	Élevée
	5G-V2X [56]	Très Élevée	Très Élevée	Très Faible	Élevée	Moyenne	Très Élevée

réseaux de capteurs. Ils permettent aux nœuds de s'auto-organiser et de transmettre les données sans infrastructure fixe, s'adaptant ainsi aux environnements dynamiques où la connectivité est instable. Ces protocoles se divisent en trois catégories : proactifs, où les routes sont préétablies, réactifs, qui découvrent les routes à la demande, et hybrides, qui combinent les avantages des deux approches. Utilisés dans divers domaines comme l'Internet des objets, les communications d'urgence et les véhicules autonomes, ces protocoles garantissent une transmission efficace et résiliente des informations. Leur intégration avec des technologies émergentes comme la 5G et le C-V2X renforce encore leur potentiel dans les systèmes intelligents de nouvelle génération [57].

2.3.3.1 Protocoles de Routage Ad-Hoc pour IoV (AODV, DSR, etc.)

Les protocoles de routage ad-hoc pour l'IoV permettent la communication entre véhicules sans infrastructure fixe, s'adaptant ainsi aux environnements dynamiques et mobiles des VANETs. Des protocoles comme AODV, Optimized Link State Routing (OLSR) et Destination-Sequenced Distance Vector (DSDV) sont utilisés pour gérer la transmission des données, mais ils rencontrent des défis liés à la stabilité, la latence et la congestion du réseau. Pour améliorer leur efficacité, des solutions comme l'intégration de drones ont été proposées afin de renforcer la connectivité et optimiser le routage dans les réseaux véhiculaires [58].

1. AODV

Le protocole AODV est un protocole de routage réactif conçu pour MANETs et VANETs. Contrairement aux protocoles proactifs, AODV établit une route uniquement lorsqu'un nœud a besoin d'envoyer des données, ce qui réduit la

consommation de bande passante et le stockage des tables de routage. Il fonctionne via deux mécanismes principaux : l'envoi Route Request (RREQ) pour découvrir les chemins et l'envoi de Route Reply (RREP) pour établir une liaison. Une fois la communication terminée, les routes sont supprimées si elles ne sont plus utilisées. Dans les réseaux modernes, notamment ceux intégrant la 5G, des améliorations comme l'RL sont utilisées pour optimiser AODV en améliorant la QoS en termes de débit, latence et de gestion du signal. Cette évolution est essentielle pour répondre aux exigences croissantes des systèmes de communication sans fil, notamment dans les environnements urbains et les véhicules autonomes [59].

2. DSR

DSR est un protocole de routage réactif conçu pour MANETs et VANETs, permettant une communication sans infrastructure fixe. Il repose sur un mécanisme d'acquisition et de maintien de routes dynamiques, où chaque paquet de données contient l'intégralité du chemin à suivre jusqu'à sa destination. Contrairement aux protocoles proactifs, DSR ne stocke pas de tables de routage mises à jour en permanence, ce qui le rend adapté aux environnements à forte mobilité et aux topologies changeantes.

Une évolution du DSR, appelée Multi-Metric Scoring Dynamic Source Routing (MMS-DSR), a été proposée pour optimiser ses performances dans les réseaux intelligents. Cette amélioration intègre des modèles de ML, notamment des réseaux de neurones CNN-LSTM, pour ajuster dynamiquement les chemins en fonction de plusieurs métriques comme la qualité du signal, la congestion du réseau et la latence [60].

Par ailleurs, une étude a implémenté DSR dans VANETs en utilisant le modèle de propagation Friis Free Space afin d'évaluer son efficacité selon différentes configurations routières. Les résultats ont montré que les performances du DSR varient significativement en fonction de la structure du réseau et des interférences présentes dans l'environnement urbain. Grâce à sa capacité d'adaptation et à sa flexibilité, DSR reste un protocole pertinent pour les communications véhiculaires, bien qu'il puisse être optimisé pour mieux gérer la latence et la surcharge dans des scénarios à haute densité de trafic [61].

3. OLSR

Le protocole Optimized Link State Routing (OLSR) est un protocole de routage proactif conçu pour MANETs et largement étudié pour VANETs. Il repose sur un mécanisme de mise à jour périodique des informations de liaison afin de maintenir une vue globale et actualisée de la topologie du réseau. Son fonctionnement est basé sur trois étapes principales : la découverte des voisins via l'échange de messages HELLO, la sélection MultiPoint Relay (MPR) pour optimiser la diffusion des messages de contrôle et la construction de la topologie du réseau à partir des informations collectées.

Dans un environnement dynamique tel que les VANETs, où les véhicules sont en mouvement constant et où la connectivité varie rapidement, l'OLSR peut souffrir de limitations liées à la surcharge du routage et aux changements fréquents de la topologie. Pour pallier ces problèmes, une approche Multi-Objective Particle Swarm Optimization (MOPSO) a été proposée, permettant d'ajuster les paramètres clés du protocole, comme les intervalles des messages HELLO et TC. Cette optimisation vise à trouver un équilibre entre la performance de livraison des paquets (débit, latence, taux de perte) et la charge du réseau liée au routage. Les simulations effectuées avec Python et Network Simulator version 2 (NS2) ont montré que l'OLSR optimisé grâce à MOPSO permet une meilleure efficacité du routage, en réduisant le coût de transmission tout en augmentant la fiabilité des connexions entre véhicules. Ces résultats soulignent l'importance d'adapter les protocoles de routage aux contraintes spécifiques des réseaux véhiculaires, afin d'assurer une communication stable et rapide dans les systèmes de transport intelligents [62].

2.3.3.2 Protocoles de Clustering pour l'IoV

Les protocoles de clustering pour l'IoV jouent un rôle essentiel dans l'organisation des véhicules en groupes dynamiques afin d'améliorer l'efficacité des communications et la gestion des ressources réseau. Une approche consiste à utiliser un clustering hiérarchique, où RSU facilitent la connectivité entre les clusters et optimisent la transmission des données, réduisant ainsi la latence et la congestion du réseau [63].

Par ailleurs, l'intégration de la blockchain et d'un mécanisme de gestion de la

confiance permet de renforcer la sécurité des communications entre les véhicules en tenant compte de critères comme la réputation des nœuds et la distance de sécurité. Cette approche améliore non seulement la QoS, mais aussi la fiabilité des échanges d'informations dans des environnements à forte mobilité [64].

1. VMaSC

Le protocole Vehicular Mobility-aware Stable Clustering (VMaSC) est conçu pour améliorer la stabilité des clusters dans les VANETs en prenant en compte la mobilité des véhicules. Il utilise des critères tels que la vitesse, la direction et la densité du trafic pour former des groupes plus cohérents et durables, réduisant ainsi la fréquence des changements de cluster. En optimisant la sélection des chefs de cluster et en minimisant les interruptions de communication, VMaSC améliore la fiabilité du routage et réduit la surcharge de signalisation causée par les mises à jour fréquentes des structures de cluster [65].

2. DCA

Distributed Clustering Algorithm (DCA) est une approche décentralisée utilisée pour organiser les véhicules en clusters dans les réseaux IoV afin d'améliorer la gestion des ressources et la communication entre les nœuds. Contrairement aux méthodes centralisées, DCA répartit la prise de décision entre les véhicules eux-mêmes, réduisant ainsi la surcharge du réseau et augmentant la résilience face aux changements dynamiques de la topologie. Une variante, Distributed Clustering Algorithm for Dominating Set (DCA-DS), intègre un mécanisme basé sur un ensemble dominant, permettant de structurer efficacement les clusters en sélectionnant des chefs de cluster optimaux tout en minimisant la latence et les interruptions de connexion. Cette approche favorise une meilleure stabilité du réseau, essentielle pour les applications critiques de l'IoV [66].

3. MobDHop

Mobility-based D-hop Clustering Protocol (MobDHop) est un protocole de clustering conçu pour améliorer la stabilité des MANETs en formant des clusters de taille variable en fonction des modèles de mobilité des nœuds [67].

Contrairement aux algorithmes classiques qui limitent la taille des clusters à un ou deux sauts, MobDHop adapte dynamiquement le diamètre des clusters en fonction de la stabilité des liens entre les nœuds, ce qui permet une meilleure

gestion du réseau à grande échelle.

Son fonctionnement repose sur l'évaluation de la variation des distances entre les nœuds au fil du temps, afin de regrouper ceux qui présentent un déplacement similaire. Cette approche réduit la fréquence des changements de cluster et limite la surcharge liée aux mises à jour du réseau [68].

Lors de la formation des clusters, le protocole identifie les nœuds les plus stables pour les désigner comme CH, et ceux situés en périphérie des clusters jouent le rôle de passerelles (gateways) pour assurer la communication inter-clusters [67]. Grâce à cette organisation flexible et adaptative, MobDHop optimise la gestion des réseaux mobiles, réduisant ainsi la latence et améliorant la fiabilité des transmissions, notamment dans les environnements à forte mobilité [68].

2.3.3.3 Protocoles Hybrides pour les Réseaux IoV

Les protocoles hybrides pour les réseaux IoV combinent plusieurs technologies, comme VANET, IoV et blockchain, pour améliorer le routage, la sécurité et la gestion du trafic. En intégrant le calcul en périphérie (Hybrid Computing), ils réduisent la latence et optimisent la transmission des données entre les véhicules et les infrastructures. De plus, l'utilisation de la blockchain garantit une authentification sécurisée et transparente des échanges. Ces approches hybrides offrent ainsi plus de fiabilité et d'efficacité aux communications dans l'IoV [69].

1. HARP

Hybrid Ad hoc Routing Protocol (HARP) est un Protocole de routage hybride basé sur des structures arborescentes, dans lequel certaines zones interagissent avec d'autres zones voisines formant ainsi des forêts. Contrairement à d'autres protocoles, HARP n'a pas besoin de nœuds racines pour ses arbres et ne nécessite pas de carte de zones statique ni de cluster-head pour coordonner les transmissions. Il utilise un algorithme de Distributed Dynamic Routing (DDR) en six phases, incluant la sélection des voisins préférés, la formation des forêts, le regroupement intra-arborescent et inter-arborescent, la nomination des zones et leur partitionnement. HARP fonctionne sur deux niveaux : intra-zone et inter-zone, et utilise un processus de routage proactif à l'intérieur des zones et un processus réactif pour la communication entre zones [70].

2. CBRP

Le Cluster-Based Routing Protocol (CBRP) est un protocole de routage conçu pour améliorer l'efficacité des réseaux ad-hoc, notamment dans les environnements VANET. Ce protocole divise le réseau en clusters où chaque cluster est dirigé par un cluster-head, responsable de la gestion du routage entre les membres du cluster ainsi qu'avec d'autres clusters. En organisant les nœuds de cette manière, le CBRP permet de réduire la surcharge de communication et de rendre le réseau plus stable en limitant les mises à jour de routage aux modifications de la topologie significatives. De plus, il utilise une approche dynamique pour ajuster les paramètres de routage, ce qui améliore la QoS, notamment en réduisant la latence et en augmentant le taux de livraison des paquets [71].

3. HSR

Le Hierarchical State Routing (HSR) est un protocole de routage proactif conçu pour les réseaux ad hoc, notamment les VANETs, afin d'améliorer l'efficacité et la scalabilité du routage dans des environnements dynamiques. Il adopte une approche hiérarchique, où le réseau est divisé en plusieurs niveaux de clusters, chaque cluster ayant un nœud leader responsable de la gestion du routage et de la communication entre les différentes zones du réseau. Cette organisation réduit la surcharge de signalisation en limitant la propagation des mises à jour de routage aux seuls clusters concernés, ce qui optimise la gestion des ressources et diminue le temps de latence. De plus, le HSR facilite l'établissement de routes plus stables en exploitant des informations agrégées sur l'état du réseau, ce qui améliore la fiabilité des transmissions même en présence de mobilité élevée [72].

2.3.3.4 Comparaison et Limites des Protocoles Ad-Hoc et Décentralisés dans l'IoV

Le Tableau 2.3 compare les protocoles de routage ad-hoc, de clustering et hybrides pour les réseaux IoV, en les évaluant selon des critères. Cette analyse aide à choisir le protocole le plus adapté en fonction des besoins de performance et de fiabilité des réseaux véhiculaires.

1. Limitations de Protocoles de Routage Ad-Hoc pour IoV

- **Instabilité des connexions** : Les changements fréquents de topologie nécessitent un recalcul fréquent des routes.
- **Surcharge réseau** : Les protocoles réactifs nécessitent l'inondation du réseau pour la découverte des routes.

2. Limitations de Protocoles de Clustering pour l'IoV

- **Gestion dynamique des clusters coûteuse** : Le changement rapide de clusters entraîne une surcharge en ressources.
- **Fiabilité des chefs de cluster** : La panne d'un chef de cluster peut entraîner une perte de connectivité.

3. Limitations de Protocoles Hybrides pour les Réseaux IoV

- **Complexité d'implémentation** : La combinaison de multiples stratégies augmente la complexité du système.
- **Consommation énergétique élevée** : L'utilisation de plusieurs mécanismes simultanément peut épuiser les ressources des véhicules.

TABLE 2.3 – Comparaison des protocoles de Routage Ad-Hoc et Décentralisés

Type	Protocole	Densité	Stabilité	Latence	Surcharge	Overlapping	Complexité
Ad-Hoc	AODV [59]	Moyenne	Moyenne	Moyenne	Moyenne	Faible	Moyenne
	DSR [60], [61]	Moyenne	Moyenne	Moyenne	Faible	Faible	Moyenne
	OLSR [62]	Moyenne à Élevée	Élevée	Moyenne à Élevée	Élevée	Moyenne	Élevée
Clustering	VMaSC [65]	Moyenne	Élevée	Faible	Moyenne	Moyenne	Moyenne
	DCA [66]	Moyenne	Moyenne	Moyenne	Faible	Faible	Moyenne
	MobDHop [67], [68]	Élevée	Élevée	Moyenne	Moyenne	Élevée	Élevée
Hybride	HARP [70]	Moyenne	Élevée	Moyenne	Moyenne	Faible	Moyenne
	CBRP [71]	Moyenne	Moyenne	Faible à Moyenne	Moyenne	Moyenne	Moyenne
	HSR [72]	Élevée	Élevée	Moyenne à Élevée	Moyenne	Moyenne	Élevée

2.3.4 Protocoles de Gestion de QoS et Transmission Multicast

Dans les réseaux IoV, garantir une communication efficace et fiable est un enjeu majeur, notamment face aux contraintes de latence, de congestion et de disponibilité des ressources. La gestion de la QoS permet d'optimiser la transmission des données en assurant un débit suffisant, une faible latence et une priorité aux applications critiques comme la sécurité routière. Parallèlement, la transmission multicast joue un rôle clé en permettant la diffusion simultanée d'un message à plusieurs véhicules, réduisant

ainsi la surcharge du réseau tout en assurant une propagation rapide des informations essentielles. L'intégration de ces deux approches dans les protocoles de communication IoV contribue à améliorer la fluidité du trafic, la sécurité et l'efficacité des échanges entre les véhicules et les infrastructures connectées.

2.3.4.1 Protocoles QoS dans les Réseaux IoV

Les protocoles de QoS dans les réseaux IoV jouent un rôle crucial pour assurer une communication efficace et fiable entre les véhicules, les infrastructures routières et les autres entités connectées. Ils visent à garantir des performances optimales en termes de latence, débit, fiabilité et gestion de la congestion du réseau. L'IoV étant un environnement dynamique avec des changements rapides de topologie et des variations de charge réseau, l'implémentation de protocoles QoS devient essentielle pour maintenir une transmission fluide des données critiques. Parmi les solutions proposées, l'intégration du fog computing permet de traiter les informations à la périphérie du réseau, réduisant ainsi la latence et améliorant la rapidité de réponse des applications. Toutefois, des défis subsistent, notamment en matière de sécurité des transmissions et d'allocation dynamique des ressources, nécessitant le développement de nouvelles stratégies pour garantir une QoS optimale dans un environnement aussi évolutif [73].

1. Q-AODV

Le protocole QoS-Aware Ad hoc On-Demand Distance Vector (Q-AODV) est une extension du protocole AODV classique, spécialement conçu pour garantir des paramètres de QoS dans les réseaux VANET. Contrairement à AODV, qui sélectionne les routes en fonction du nombre minimal de sauts, QAODV prend en compte la latence et la bande passante lors de la découverte des routes.

Dans ce protocole, chaque nœud calcule dynamiquement son délai local (LOCAL_DELAY) et l'intègre aux paquets RREQ. Lors de la découverte d'un chemin, seules les routes respectant les contraintes de latence et de bande passante sont retenues. Une fois la route trouvée, le nœud source dispose de deux chemins optimaux : l'un basé sur le délai minimum, et l'autre sur la bande passante maximale. Cette approche permet une meilleure gestion de la congestion et réduit la latence dans les communications critiques des véhicules connectés.

QAODV inclut également une stratégie de maintenance de route avancée. Si un

lien se rompt, le protocole active immédiatement un itinéraire alternatif, évitant ainsi les délais de reconstruction. De plus, un mécanisme de réparation locale est mis en place pour limiter l'impact des pannes de connexion.

En optimisant la gestion de la qualité de service dans les réseaux IoV, QAODV constitue une solution efficace pour les applications nécessitant un faible temps de latence, telles que la navigation assistée, la prévention des collisions et la communication entre véhicules [74].

2. P-QOS

Le protocole Predictive Quality of Service (P-QoS) est une approche prédictive de la QoS dans les réseaux VANET, permettant d'anticiper les variations du réseau grâce à l'IA et l'ML. Contrairement aux méthodes traditionnelles qui réagissent aux problèmes après leur apparition, P-QoS utilise des modèles de prédiction des paramètres réseau comme la latence, le taux de perte de paquets et la congestion, afin d'optimiser le routage et améliorer la fiabilité des communications V2V et V2I.

Basé sur le standard IEEE 802.11p, P-QoS analyse en temps réel les données environnementales, le trafic et les signaux de communication pour anticiper les dégradations du réseau et ajuster dynamiquement les transmissions. Cette approche est essentielle pour les applications critiques comme la prévention des collisions, la gestion du trafic et les véhicules autonomes, garantissant une communication rapide et efficace dans des environnements à forte mobilité [75].

3. CEDAR

Le protocole Core-Extraction Distributed Ad hoc Routing (CEDAR) est une approche de routage conçue pour assurer la QoS dans les réseaux ad hoc, notamment dans les MANETs et VANETs. Son fonctionnement repose sur l'extraction d'un noyau dynamique composé de nœuds dominants, qui assurent la gestion efficace du routage en réduisant la surcharge du réseau. Pour optimiser la transmission des données, CEDAR utilise un mécanisme de propagation sélective de l'état des liens, ne diffusant que les informations sur les connexions stables et à haute bande passante, tandis que les liens instables restent limités à une portée locale. Le protocole intègre également un algorithme de calcul de route QoS, qui prend en compte la bande passante disponible et la stabilité des chemins

afin d'établir des routes fiables et performantes. Grâce à un système de vagues d'augmentation et de diminution de bande passante, il ajuste en temps réel ses décisions pour s'adapter aux changements dynamiques du réseau. CEDAR est particulièrement adapté aux réseaux de taille moyenne, comptant de dizaines à centaines de nœuds, et se distingue par sa robustesse et sa capacité d'adaptation. Il trouve des applications dans des domaines où la fiabilité des communications est cruciale, comme les systèmes de transport intelligents, les réseaux militaires et les services multimédias exigeants [76].

2.3.4.2 Protocoles Multicast dans les Réseaux IoV

Les protocoles multicast dans les réseaux IoV permettent d'optimiser la diffusion des messages vers plusieurs véhicules simultanément, sans surcharger inutilement le réseau. Contrairement au routage unicast, où chaque message est envoyé individuellement à chaque destinataire, le multicast réduit la redondance des transmissions et améliore l'efficacité du réseau. Parmi les solutions existantes, le protocole Energy Consumption Aware - Multicast Ad hoc On-Demand Distance Vector (ECA-MAODV) constitue une amélioration du protocole Multicast Ad hoc On-Demand Distance Vector (MAODV), en intégrant une gestion adaptative des événements pour une meilleure transmission des paquets dans des environnements hautement dynamiques comme l'IoV [77].

Par ailleurs, plusieurs études classifient les protocoles de routage multicast selon leurs stratégies, incluant le geocast, qui utilise des zones géographiques ciblées pour la diffusion, et les approches basées sur les arbres ou les maillages, qui garantissent une transmission plus robuste en cas de défaillance de certains nœuds. Ces avancées permettent de répondre aux défis majeurs du multicast dans les réseaux IoV, notamment la gestion de la mobilité, la réduction des pertes de paquets et l'optimisation de la latence [78].

1. MAODV

Le protocole MAODV est une extension d'AODV conçue pour le routage multicast dans les réseaux mobiles ad hoc, notamment VANET et IoV. Il fonctionne à la demande, construisant un arbre de distribution partagé pour optimiser la bande passante. Il utilise des messages de contrôle (RREQ, RREP, Multicast

Activation(MACT), Group Hello(GRPH) pour établir et maintenir les routes multicast. En cas de rupture de connexion, il tente de restaurer l'arbre ou d'en créer un nouveau. Bien qu'efficace pour la transmission multicast, sa performance peut être affectée par la mobilité élevée et les coupures fréquentes de liens dans les réseaux VANET [79].

2. **AMRoute**

Le protocole AMRoute (Ad hoc Multicast Routing Protocol) est une solution efficace pour le multicast IP dans les réseaux ad hoc comme VANET et IoV. Il utilise des tunnels unicast pour former un arbre multicast, réduisant ainsi la signalisation et la perte de paquets en évitant les changements fréquents de topologie. AMRoute est indépendant du protocole de routage unicast et repose uniquement sur les membres du groupe pour la gestion de l'arbre, minimisant la surcharge réseau. Il intègre un mécanisme de migration dynamique des nœuds centraux, améliorant la robustesse et évitant un point unique de défaillance. Les simulations montrent un bon taux de livraison des données et une signalisation modérée, bien que ses performances dépendent du protocole de routage unicast sous-jacent [80].

3. **BEMRP**

Le Bandwidth Efficient Multicast Routing Protocol (BEMRP) est un protocole de routage multicast basé sur un arbre et initié par le récepteur, utilisé dans VANETs et l'IoV. Il fonctionne en cherchant le nœud de transfert le plus proche au lieu du chemin le plus court entre la source et le récepteur. Cette approche permet d'économiser de la bande passante en réduisant le nombre de transmissions de données. Cependant, BEMRP souffre de certains inconvénients, notamment un retard de transmission accru, une robustesse de lien limitée et une baisse du taux de livraison des paquets, en raison de l'augmentation de la distance entre la source et les membres multicast. Pour pallier ces limites, des améliorations basées sur des techniques d'inférence et d'apprentissage, comme l'intégration des fuzzy Petri nets, ont été proposées afin d'optimiser le choix des chemins et d'améliorer la fiabilité du routage multicast dans des environnements dynamiques et contraints en ressources [81].

2.3.4.3 Comparaison et Limites des Protocoles QoS et Multicast dans l'IoV

Le Tableau 2.4 compare les protocoles QoS et Multicast dans les réseaux IoV en fonction de critères . Cette analyse aide à comprendre les avantages et les limitations de chaque protocole pour la gestion du trafic et la transmission simultanée vers plusieurs véhicules.

1. Limitations de Protocoles QoS dans les Réseaux IoV

- **Difficulté d'adaptation aux variations du trafic** : Les exigences de QoS changent selon la densité du trafic et les conditions environnementales.
- **Sécurité et allocation dynamique des ressources** : L'optimisation des priorités de transmission reste un défi.

2. Limitations de Protocoles Multicast dans les Réseaux IoV

- **Gestion de la mobilité complexe** : Le maintien des groupes multicast est difficile en raison du mouvement rapide des véhicules.
- **Risque de congestion** : La transmission simultanée de paquets vers plusieurs véhicules peut surcharger le réseau.

TABLE 2.4 – Comparaison des protocoles QoS et Multicast dans L'IoV

Type	Protocole	Densité	Stabilité	Latence	Surcharge	Overlapping	Complexité
QoS	Q-AODV [74]	Moyenne	Moyenne à Élevée	Faible	Moyenne	Moyenne	Moyenne à Élevée
	P-QoS [75]	Élevée	Élevée	Très Faible	Moyenne à Élevée	Moyenne	Élevée
	CEDAR [76]	Moyenne à Élevée	Élevée	Moyenne	Moyenne	Élevée	Élevée
Multicast	MAODV [79]	Moyenne	Moyenne	Moyenne	Moyenne	Faible	Moyenne
	AMRoute [80]	Faible	Élevée	Élevée	Faible	Moyenne	Moyenne
	BEMRP [81]	Élevée	Moyenne	Faible	Élevée	Élevée	Élevée

2.4 Application de l'Intelligence Artificielle pour la Transmission de Données dans les Réseaux IoV

L'IA joue un rôle clé dans l'optimisation de la transmission des données dans les réseaux IoV en améliorant l'efficacité, la rapidité et la sécurité des communications entre véhicules et infrastructures connectées. Grâce à des approches basées sur le ML et le Deep Learning (DL), les systèmes IoV peuvent anticiper les conditions du réseau, ajuster dynamiquement l'allocation des ressources et minimiser la latence lors de la

transmission des données. Par exemple, l'utilisation de l'IA dans le Edge Computing permet de traiter les données localement, réduisant ainsi la surcharge des centres de données distants et améliorant la réactivité des applications IoV [82]. De plus, des modèles d'ML sont employés pour sécuriser les communications véhiculaires, en détectant les comportements anormaux et les menaces potentielles, garantissant ainsi une transmission fiable des informations sensibles. L'intégration de l'IA dans l'IoV ouvre ainsi la voie à une gestion plus intelligente et autonome des flux de données, essentielle pour les véhicules autonomes et les infrastructures de transport intelligentes [83].

2.4.1 Rôle de l'Apprentissage Automatique dans l'Optimisation du Routage

- **Analyse en temps réel des conditions du réseau** : Contrairement aux protocoles traditionnels, les modèles de ML évaluent en continu des paramètres tels que la densité du trafic, la mobilité des véhicules et la qualité des liens pour optimiser le choix des itinéraires [60].
- **Prédiction des congestions et adaptation du routage** : Grâce aux algorithmes prédictifs, le ML peut anticiper les embouteillages et ajuster dynamiquement les routes pour éviter les zones encombrées, améliorant ainsi la fluidité du trafic et réduisant la latence [60].
- **Optimisation de QoS** : Le ML permet d'améliorer la fiabilité et la rapidité de transmission des données, en minimisant les pertes de paquets et en garantissant une connectivité stable entre les véhicules et les infrastructures [60].
- **Utilisation du Deep Learning pour l'adaptabilité** : DNNs permettent d'identifier des schémas complexes dans les communications et d'adapter les décisions de routage en fonction de l'évolution dynamique du réseau [60].
- **Réduction de la consommation énergétique** : Certains modèles ML sont conçus pour optimiser l'utilisation des ressources en ajustant les fréquences de transmission et en sélectionnant les itinéraires les plus efficaces sur le plan énergétique [60].
- **Amélioration de la sécurité du routage** : Le ML permet de détecter les anomalies et les attaques réseau (comme les attaques Sybil ou les usurpations de

route) afin de renforcer la fiabilité des communications dans l'IPv [60].

Grâce à ces avancées, l'ML révolutionne le routage dans l'IPv, en rendant les systèmes plus intelligents, adaptatifs et performants pour une communication efficace et sécurisée entre les véhicules connectés et les infrastructures.

2.4.2 Algorithmes d'Apprentissage Supervisé et Non Supervisé pour la Gestion du Routage

Les algorithmes d'apprentissage supervisé et non supervisé jouent un rôle clé dans l'optimisation de la gestion du routage, en particulier dans Software-Defined Networking (SDN) et les environnements dynamiques. L'apprentissage supervisé repose sur des données étiquetées pour établir des relations précises entre les entrées et les sorties du réseau, permettant aux modèles d'affiner leurs prédictions en minimisant une fonction de perte. Des techniques comme Artificial Neural Network (ANN), les processus de décision de Markov et la régression linéaire sont couramment utilisées pour optimiser les chemins de transmission et réduire la latence, tout en respectant les contraintes de QoS [84].

En complément, l'apprentissage non supervisé analyse des données non étiquetées pour détecter des motifs cachés et ajuster dynamiquement les routes du réseau. Des méthodes telles que DBSCAN et K-means sont utilisées pour la segmentation du trafic et la détection d'anomalies, tandis que des modèles plus avancés comme les réseaux adverses génératifs (GANs) permettent d'anticiper les variations de charge. En combinant ces deux approches, il devient possible d'assurer une gestion du routage plus précise, adaptative et efficace, en optimisant la transmission des données tout en s'adaptant aux évolutions dynamiques du réseau [85].

2.4.3 Applications de l'Intelligence Artificielle pour la QoS

L'IA optimise la QoS dans l'IPv en adaptant la transmission multimédia aux conditions du réseau. Deux algorithmes, PQO (Power-aware QoE Optimization) et BQO (Buffer-aware QoE Optimization), sont développés pour réduire la consommation énergétique et optimiser l'utilisation des tampons. Ces méthodes améliorent la qualité d'expérience (QoE), minimisent la latence et prolongent l'autonomie des dis-

positifs portables, surpassant ainsi les approches traditionnelles. L'IA constitue donc une solution essentielle pour une gestion efficace et intelligente de la transmission des données en temps réel [86].

2.4.4 Défis de l'IA dans les Réseaux IoV

L'intégration de l'IA dans les réseaux de l'IoV présente plusieurs défis majeurs liés à la gestion des ressources, à la sécurité et à la fiabilité des décisions prises. L'un des principaux obstacles est le besoin élevé en puissance de calcul et en ressources réseau, car les applications basées sur l'IA, comme la conduite autonome et la gestion du trafic en temps réel, nécessitent un traitement rapide et efficace des données. De plus, la vulnérabilité de l'IA face aux cyberattaques et à la falsification des données est une préoccupation majeure, car des informations erronées peuvent compromettre la sécurité routière. Un autre défi réside dans la mise à jour cohérente des modèles d'IA déployés sur les véhicules et l'infrastructure, afin d'assurer une prise de décision précise et adaptée aux conditions dynamiques du réseau. Enfin, l'interopérabilité entre les différentes plateformes et l'intégration de l'IA avec d'autres technologies, comme la blockchain et le calcul en périphérie (Vehicular Edge Computing), nécessitent des solutions évolutives et sécurisées pour garantir un IoV intelligent et fiable [87].

2.5 Conclusion

Ce chapitre a présenté un aperçu des principales approches de routage dans les réseaux IoV, en examinant aussi bien les protocoles classiques que les solutions fondées sur l'intelligence artificielle. L'analyse a mis en évidence les limites des méthodes traditionnelles face à la dynamique élevée et à la complexité des environnements IoV, ce qui souligne la nécessité de protocoles plus adaptés.

Dans cette optique, le chapitre suivant est consacré à une étude de cas portant sur un protocole de routage hybride s'appuyant sur les technologies DSRC et C-V2X, servant de base à l'analyse des performances et des pistes d'amélioration.

Chapitre 3

Étude de cas d'un protocole de routage hybride basé sur DSRC et C-V2X

3.1 Introduction

Ce chapitre présente une étude de cas basée sur le protocole de routage proposé dans l'article intitulé "Enabling DSRC and C-V2X Integrated Hybrid Vehicular Networks : Architecture and Protocol", publié dans la revue IEEE Access [88]. L'étude s'intéresse à une architecture hybride conçue pour améliorer la communication dans les réseaux véhiculaires intelligents, en combinant les deux technologies de communication majeures : DSRC et C-V2X.

L'objectif de ce chapitre est d'expliquer en détail le protocole proposé dans l'article, en décrivant son architecture, son fonctionnement, ainsi que les mécanismes de routage qu'il met en œuvre. Le protocole repose sur une sélection dynamique du type de lien de communication en fonction de différents paramètres contextuels, tels que la portée, la qualité du signal et la latence. L'architecture hybride vise à tirer profit de la faible latence et de la simplicité du DSRC, tout en bénéficiant de la couverture étendue et de la fiabilité offerte par le C-V2X, notamment dans les environnements où l'infrastructure cellulaire est disponible.

Le chapitre expose également les défis majeurs identifiés dans la mise en œuvre de ce type de solution. Parmi ces défis figurent la gestion de la mobilité des véhicules

à grande vitesse, la coordination entre les deux technologies au niveau du protocole, la prise de décision en temps réel pour choisir le canal de communication optimal, ainsi que les risques de surcharge ou d'interférences dans les zones urbaines denses. Des limitations sont également relevées, notamment en ce qui concerne la complexité de l'implémentation, le coût d'équipement des véhicules avec des interfaces multiples, et les contraintes de compatibilité entre les protocoles DSRC et C-V2X.

En résumé, ce chapitre offre une analyse détaillée d'un protocole de routage hybride innovant, en mettant en évidence ses contributions, ses atouts et ses limites. Il constitue une base de réflexion sur les solutions de routage adaptées aux systèmes de transport intelligents de nouvelle génération.

3.2 Introduction au protocole DSRC

Le DSRC est un protocole de communication sans fil spécifiquement conçu pour les VANETs afin de faciliter les échanges d'informations entre les véhicules et entre les véhicules et les infrastructures routières, améliorant ainsi la sécurité routière et l'efficacité du trafic. qui repose sur la norme IEEE 802.11p, opère dans la bande des 5,9 GHz et offre une communication directe à faible latence. Cela est particulièrement important pour des applications essentielles, comme les alertes de collision, les réponses d'urgence et la gestion en temps réel du trafic. À la différence des réseaux cellulaires, le DSRC ne repose pas sur une infrastructure centralisée, ce qui assure un partage d'informations rapide et sûr même dans des contextes de forte densité. Toutefois, cette technologie souffre de contraintes majeures, comme un rayon d'action limité (300-1000 m) et une saturation précoce du spectre à cause de l'augmentation du nombre de véhicules qui exploitent la même fréquence. Afin de relever ces défis, l'article suggère une méthode hybride combinant le DSRC et le C-V2X, utilisant un algorithme de choix dynamique des technologies d'accès (RAT Selection) pour améliorer l'attribution des ressources radio et assurer une communication plus fiable et performante en fonction des conditions du réseau.

3.2.1 Définition de DSRC

DSRC est une technologie de transmission sans fil spécialement développée pour les VANETs, permettant une interaction aisée entre les véhicules et les infrastructures routières. Il opère sur la fréquence de 5,9 GHz et est fondé sur la norme IEEE 802.11p. Il autorise des transmissions directes à faible délai sans avoir besoin de recourir aux réseaux mobiles. Sa priorité essentielle est d'assurer une communication efficace et sûre pour améliorer la sécurité sur les routes et l'efficacité du trafic, y compris la transmission d'alertes en direct concernant les conditions routières et les circonstances d'urgence [88].

3.2.2 Architecture et Composants du Protocole DSRC

L'architecture du protocole DSRC, qui se compose de nombreux composants essentiels, facilite la communication efficace entre les véhicules et les infrastructures de routage. Parmi les éléments essentiels figurent l'OBU, la RSU et le canal de communication sans fil.

Ensemble, ces composants permettent une transmission d'informations fiable et rapide, ce qui contribue à améliorer la gestion du trafic et à renforcer la sécurité des itinéraires.

1. **OBU** : L'OBU est un dispositif électronique installé dans les véhicules, permettant la communication sans fil avec d'autres véhicules et les infrastructures environnantes. Il est indispensable pour l'échange d'informations en temps réel, comme la gestion du trafic et la prévention des accidents.

L'OBU est composé des éléments suivants :

- **Un module de communication sans fil** fondé sur la norme IEEE 802.11p, qui facilite l'échange et la réception de messages DSRC sur la fréquence de 5.9 GHz. Ce composant garantit une liaison stable avec une latence minimale, ce qui est crucial pour les applications de sécurité.
- **Un processeur embarqué** chargé de traiter les données reçues et d'exécuter les protocoles nécessaires à la gestion des communications. Il filtre les messages entrants, analyse les informations de sécurité et prend des décisions en fonction des alertes reçues.

- **Une antenne GPS** qui permet de déterminer avec précision la position du véhicule et de partager ces informations avec les autres véhicules ou infrastructures. L'intégration du GPS est essentielle pour garantir la synchronisation et la précision des alertes de sécurité (ex : freinage d'urgence, accident à proximité).

Par conséquent, l'OBU joue un rôle crucial dans le protocole DSRC en garantissant la connectivité des véhicules et en permettant aux véhicules d'échanger des données vitales pour la sécurité routière et le bon déroulement du trafic.

2. **RSU** : Les RSU sont des installations fixes mises en place le long des voies et autoroutes pour améliorer la communication entre les véhicules et l'environnement routier V2I. À l'inverse des OBU qui sont portables, les RSU sont fixes et ont pour fonction principale d'accroître la portée de communication et de garantir une connexion avec les systèmes de gestion du trafic.

Les principales fonctions des RSU sont les suivantes :

- **Relayer les messages entre les véhicules éloignés** : Quand deux véhicules sont hors de portée l'un de l'autre, le RSU joue le rôle d'intermédiaire pour maintenir leur communication. Cela présente un grand intérêt dans les zones urbaines densément peuplées où la communication entre les véhicules peut être gênée par des constructions.
- **Interfacer le réseau véhiculaire avec le Cloud et les systèmes de gestion du trafic** : Les RSU sont capables de transmettre des informations aux autorités concernant la circulation, aux feux de signalisation intelligents et aux serveurs responsables de la gestion du trafic. Par exemple, elles peuvent recueillir des données sur la densité du trafic et les communiquer à un centre de contrôle pour ajuster la temporisation des feux de signalisation.

En raison de leur emplacement stratégique, les RSU sont essentiels pour l'augmentation des capacités du DSRC, offrant une couverture réseau plus étendue et une transmission améliorée des informations cruciales.

3. **Canal de Communication (Wireless Channel)** Le système DSRC se sert de la communication sans fil sur une fréquence de 5,9 GHz, spécialement dédiée

aux systèmes ITS. Cette bande est segmentée en sept canaux de 10 MHz, chacun ayant un rôle précis pour garantir une gestion optimale des communications.

Les canaux de communication DSRC se répartissent comme suit :

- **Un canal de contrôle (CCH - Control Channel)** : Ce canal est destiné à orchestrer les connexions et à coordonner les échanges de communication. Il sert principalement à l'échange de messages de signalisation et de sécurité, assurant une communication efficace entre les véhicules et les infrastructures sans perturbations .
- **Six canaux de service (SCH - Service Channels)** : Ces voies sont employées pour la diffusion des données qui ne sont pas primordiales, comme les renseignements sur le trafic, les alertes de conditions météorologiques et les services de divertissement d'information. Chaque canal peut être attribué de manière dynamique en fonction des priorités, garantissant une gestion optimale des ressources réseau.

L'utilisation de ces canaux contribue à éviter les perturbations de transmission et garantit une communication rapide et efficace, ce qui est essentiel pour les applications en rapport avec la sécurité routière.

3.2.3 Fonctionnement du Protocole DSRC

Le protocole DSRC s'appuie sur une diffusion en mode broadcast, facilitant ainsi l'échange d'informations cruciales en temps réel entre les véhicules et les infrastructures. Il opère sur trois principes de base :

- La transmission des messages.
- Les types de communication .
- La gestion du canal et la sécurité.

3.2.3.1 Transmission des Messages dans DSRC

DSRC utilise une diffusion directe (single-hop) pour informer les véhicules à proximité et une transmission multi-saut pour relayer les messages sur de plus longues distances. Un ajustement dynamique du taux d'émission des messages est mis en place pour éviter la congestion.

3.2.3.2 Types de Communication dans DSRC

1. **V2V** : Échange direct entre véhicules pour prévenir les collisions et optimiser le trafic.
2. **V2I** : Connexion entre véhicules et infrastructures routières pour la gestion dynamique du trafic.
3. **V2X** : Communication avec d'autres éléments intelligents comme les feux de circulation et les capteurs urbains.

3.2.3.3 Gestion du Canal et Sécurité dans DSRC

DSRC utilise Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) pour éviter les collisions de paquets. Un véhicule écoute le canal avant de transmettre, et en cas d'occupation, il attend un délai aléatoire avant de réessayer. Cependant, en environnement dense, ce mécanisme peut entraîner des pertes de paquets et des délais plus longs. En termes de sécurité, DSRC suit la norme IEEE 1609.2, qui garantit l'authentification des messages, leur signature numérique et leur chiffrement, empêchant ainsi les cyberattaques.

Pour résumer ces mécanismes, La Figure 3.1 ci-dessous illustre les principales étapes de fonctionnement du protocole DSRC, en mettant en évidence les processus de transmission, les types de communication et la gestion du canal.

3.2.4 Types de Messages Utilisés dans DSRC

3.2.4.1 Cooperative Awareness Messages (CAM)

CAM sont des messages envoyés de manière périodique par les véhicules équipés d'une unité de communication DSRC. Leur rôle principal est de fournir des informations essentielles sur l'état dynamique du véhicule, telles que la position géographique, la vitesse, la direction et d'autres paramètres de mobilité.

Ces messages permettent à chaque véhicule de se situer par rapport aux autres participants à la circulation, facilitant ainsi des choix plus judicieux pour prévenir les accidents et améliorer le flux du trafic. Ce genre de communication est spécialement bénéfique dans des contextes où la clarté est limitée (par exemple : virages étroits, carrefours sans feux de circulation). Les CAM sont diffusés à intervalles réguliers et suivent

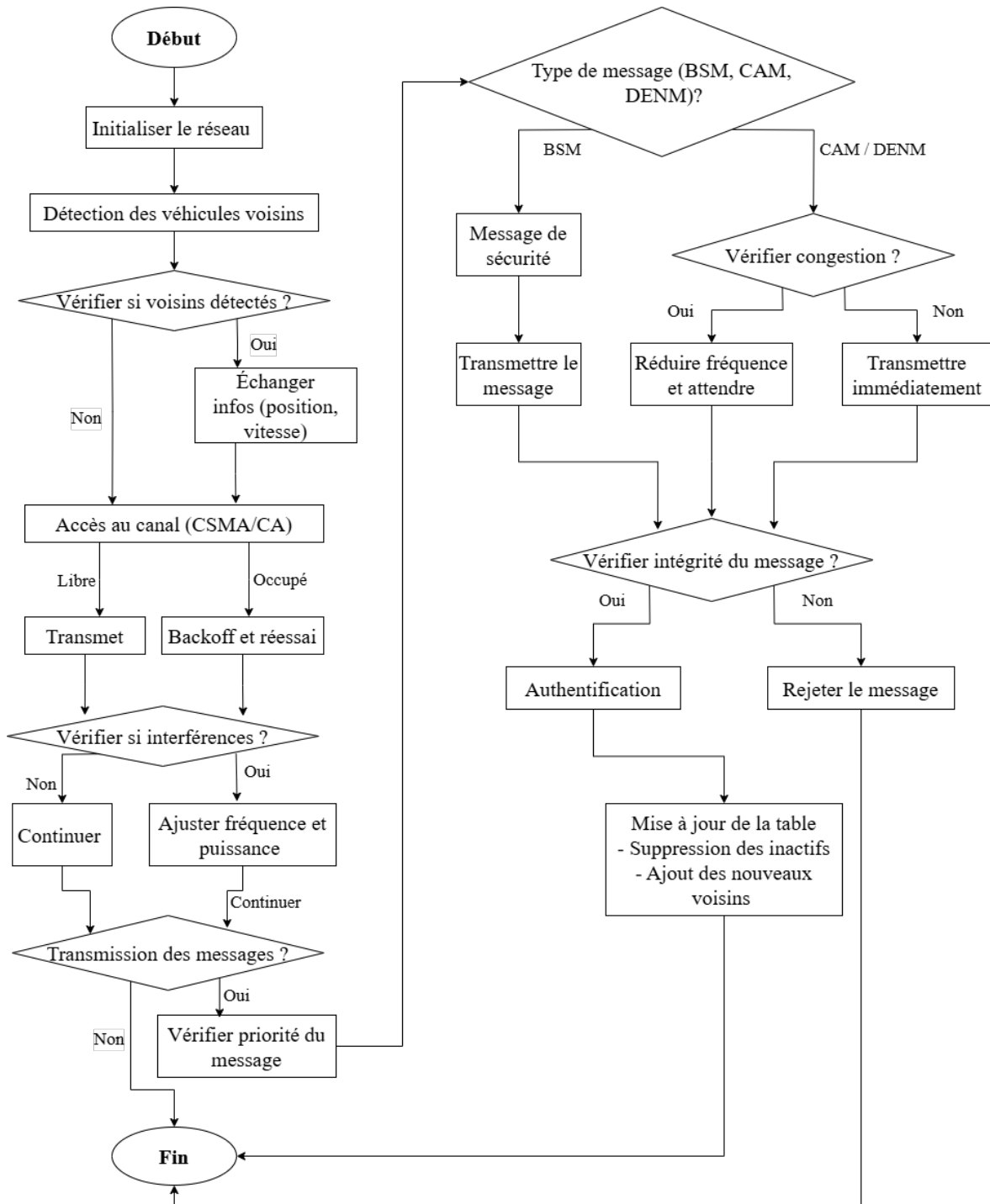


FIGURE 3.1 – Organigramme de protocole DSRC [1].

des protocoles normalisés, garantissant une interopérabilité entre différents fabricants et infrastructures de ITS.

3.2.4.2 Decentralized Environmental Notification Messages (DENM)

Les DENM sont créés lorsqu'un événement critique se produit sur la route, nécessitant une alerte immédiate pour les autres utilisateurs. Contrairement aux CAM, qui sont envoyés régulièrement, les DENM ne sont activés que lorsque cela est nécessaire, comme en cas d'accident, d'obstruction sur la route, de conditions météorologiques dangereuses (comme le brouillard ou le verglas), ou de travaux routiers imprévus.

Ces messages réduisent le risque d'accident en utilisant une diffusion ciblée (géodiffusion ou multi-saut) pour alerter les voitures dans une zone particulière. En procédant ainsi, une surcharge inutile du réseau de communication est évitée et l'information est garantie d'être fournie uniquement aux véhicules participant à l'événement.

Pour garantir que les avertissements les plus importants soient traités en premier afin d'assurer une réaction rapide des voitures et des systèmes de contrôle de la circulation, la gestion DENM utilise des techniques de priorisation.

3.2.5 Métriques de Performance dans DSRC

L'évaluation des performances du protocole DSRC repose sur plusieurs métriques essentielles permettant d'analyser la congestion du canal, l'efficacité des transmissions et l'utilisation des ressources réseau. Parmi ces métriques, on retrouve le temps d'attente du canal, la longueur de la file d'attente et le niveau d'occupation du canal.

3.2.5.1 Temps d'Attente du Canal (CWT - Channel Waiting Time)

Le CWT représente le temps moyen pendant lequel un canal de communication reste occupé avant qu'une transmission ne puisse avoir lieu. Il est défini par l'équation 3.1 :

$$CWT = \frac{\sum_{i=0}^{q-1} T^i}{T_{\max}} \quad (3.1)$$

où :

- T^i est le temps moyen pendant lequel le canal est occupé pour une catégorie de priorité i .
- T_{\max} est la durée maximale d'intervalle du canal.

- q est le nombre total de catégories de priorité dans l'algorithme EDCA utilisé dans DSRC.

3.2.5.2 Longueur de la File d'Attente (QL - Queue Length)

QL évalue le niveau de congestion d'un nœud en fonction du nombre de paquets en attente de transmission. Elle est calculée par l'équation 3.2 :

$$QL = \frac{\sum_{i=0}^{q-1} Q_k^i}{Q_{\max}} \quad (3.2)$$

où :

- Q_k^i est la taille actuelle de la file d'attente pour une catégorie de priorité i au nœud k .
- Q_{\max} est la capacité maximale de la file d'attente.

3.2.5.3 Niveau d'Occupation du Canal (COL - Channel Occupancy Level)

Le COL est un indicateur clé permettant d'évaluer la saturation du réseau DSRC. Il est défini par la relation suivante 3.3 :

$$COL = (\alpha CWT) + (1 - \alpha)QL \quad (3.3)$$

où α est un facteur de pondération ajustant l'importance relative de chaque métrique.

3.2.5.4 Charge Réseau dans LTE pour la Sélection de DSRC

La charge du réseau LTE est un facteur déterminant pour déterminer la technologie de bascule dans les réseaux hybrides DSRC + LTE. Le nombre de blocs de ressources physiques disponibles (PRBs) est utilisé pour l'évaluer. L'équation définissant cette charge est fournie par 3.4 :

$$L_{\text{new}}(b) = \begin{cases} PRB(b), & \text{si } L_{\text{new}}(b) \leq C - L_{\text{current}} \\ PRB(b), & \text{si } L_{\text{new}}(b) \leq C - (L_{\text{current}} - \rho) \\ 0, & \text{sinon} \end{cases} \quad (3.4)$$

où :

- $L_{\text{new}}(b)$ représente la charge de la nouvelle session demandée.
- L_{current} est la charge réseau actuelle.
- C est la capacité totale de la cellule LTE.
- ρ est le niveau de ressources pouvant être préemptées pour faire place à une nouvelle session.

3.2.6 Avantages du DSRC

1. Faible latence et transmission rapide

- DSRC permet une transmission quasi-instantanée (latence entre 2 et 5 ms), ce qui est crucial pour les systèmes de sécurité routière.
- L'utilisation d'un mode ad hoc direct V2V réduit les retards liés aux infrastructures centralisées.

2. Indépendance des réseaux cellulaires

- DSRC fonctionne sans dépendre d'un réseau cellulaire comme la 4G/5G, ce qui garantit une connectivité stable même en l'absence de couverture mobile.

3. Utilisation d'un spectre dédié

- Fonctionne sur la bande 5.9 GHz, réservée exclusivement aux communications intelligentes des véhicules, ce qui réduit les risques d'interférences avec d'autres réseaux.

4. Efficacité pour la gestion du trafic

- DSRC facilite le contrôle du trafic et la gestion dynamique des intersections grâce aux communications V2I et V2X.

3.2.7 Limitations de DSRC

Le protocole DSRC présente plusieurs limitations et défis, notamment liés à la portée du signal, la congestion du spectre et les problèmes de sécurité.

1. Portée limitée

- DSRC a une portée restreinte (300m en milieu urbain, jusqu'à 1km en conditions idéales).

- Problème de connectivité sur les autoroutes et zones peu denses.
- Déconnexions fréquentes en cas de mobilité élevée ($> 100\text{km/h}$).

2. Congestion et interférences radio

- Utilisation du spectre 5.9GHz où trop de véhicules émettent en même temps.
- Saturation du canal en zone dense, entraînant des collisions de paquets et pertes de messages.
- Le protocole CSMA/CA ne gère pas efficacement les conflits d'accès au canal.

3. Vulnérabilités en matière de sécurité

- DSRC est sensible aux cyberattaques, notamment :
 - **Spoofing** : Usurpation d'identité.
 - **Attaques par rejeu** : Réutilisation de vieux messages.
 - **Brouillage du signal (Jamming Attacks)**.

4. Absence de gestion avancée du handover

- DSRC ne possède pas de mécanisme optimisé pour le transfert entre cellules, provoquant des pertes de connexion.
- Problème de continuité de service comparé à C-V2X.

3.3 Présentation de C-V2X

Le protocole C-V2X est une technologie de communication avancée développée pour les véhicules connectés. Il permet l'échange d'informations entre véhicules, entre véhicules et infrastructures, entre véhicules et piétons, ainsi qu'avec le réseau cellulaire. Il repose sur les normes définies par le 3GPP.

3.3.1 Définition de C-V2X

Le C-V2X est une technologie de communication sans fil conçue pour faciliter la communication entre les voitures, les infrastructures, les tuyaux et les réseaux. Développé dans le cadre de la 3GPP, il utilise les technologies LTE (4G) et 5G pour fournir des communications fiables et à faible latence adaptées aux exigences des systèmes de transport intelligents en constante évolution. L'article présente le C-V2X comme une

alternative pertinente au DSRC, en soulignant sa capacité à combiner des communications directes locales avec une couverture plus large grâce à l'infrastructure cellulaire.

3.3.2 Fonctionnement de C-V2X

Le fonctionnement du C-V2X repose sur une double architecture de communication :

— **Mode direct via l'interface PC5 :**

Ce mode permet des communications directes entre véhicules sans passer par une station de base. Utilisant le sidelink LTE, il est idéal pour transmettre rapidement des messages critiques.

— **Mode réseau via l'interface Uu :**

Les communications sont transmises par l'infrastructure cellulaire (eNodeB pour la 4G LTE ou gNodeB pour la 5G). Ce mode offre une connectivité étendue, permet l'accès aux services en nuage (cloud) et facilite la gestion centralisée via un serveur tel que le Centre d'Information CellCar (CIC).

Il est utilisé pour les échanges d'informations non critiques mais volumineux, ou encore pour assurer la continuité du service dans les zones mal desservies par les réseaux ad hoc.

Cette approche bimodale permet a C-V2X de s'ajuster dynamiquement aux exigences des applications, en alternant automatiquement entre les modes selon le contexte, la disponibilité du réseau et la criticité des messages.

3.3.3 Normes techniques utilisées

Le C-V2X s'appuie sur une évolution progressive des standards techniques définis par le 3GPP :

— **LTE-V2X (Release 14 15) :**

Ces versions ont introduit la base du C-V2X, notamment la communication directe via l'interface PC5, et la communication via réseau grâce à l'interface Uu. Elles permettent de supporter les premiers cas d'usage du V2X avec des mécanismes fondamentaux de sécurité et de diffusion.

— **5G NR-V2X (Release 16 17) :**

Avec l'arrivée de la 5G, les nouvelles versions offrent des améliorations substantielles, telles que :

- une latence réduite à moins de 10 ms.
- une bande passante plus large.
- le support de l'edge computing.
- et des fonctionnalités avancées comme le network slicing, permettant de réserver dynamiquement des ressources réseau selon le type de service ou le niveau de priorité.

Le protocole est aussi adapté pour fonctionner avec les interfaces GeoNetworking, Transmission Control Protocol/Internet Protocol (TCP/IP) et User Datagram Protocol/Internet Protocol (UDP/IP), garantissant de ce fait son incorporation dans différents contextes réseau ainsi que sa compatibilité avec d'autres couches d'application déjà en place.

3.4 Comparaison entre DSRC et C-V2X

Le protocole DSRC et le protocole C-V2X représentent deux technologies fondamentales pour la communication entre véhicules, bien qu'ils diffèrent par leur architecture et leurs performances. Le DSRC, basé sur la norme IEEE 802.11p, fonctionne dans la bande de fréquences de 5.9 GHz en mode ad hoc. Cela favorise une communication directe entre les véhicules et avec les infrastructures, sans l'utilisation d'un réseau centralisé. Il propose une latence approximative de 10 ms et un rayon d'action variant de 300 à 1000 mètres. Par contre, le C-V2X, élaboré par le 3GPP, se sert des technologies LTE et 5G pour proposer une communication directe (grâce à l'interface PC5) et par le biais de l'infrastructure cellulaire (interface Uu), avec un délai de latence inférieur à 10 ms en 5G et une portée qui dépasse les 1000 mètres. C-V2X bénéficie également d'une fiabilité accrue dans des environnements denses grâce à une gestion centralisée des ressources, tout en offrant une interopérabilité améliorée avec les réseaux mobiles contemporains. Bien que le DSRC soit une technologie éprouvée et largement analysée, elle comporte des limitations en termes de portée, de fiabilité et d'intégration avec les réseaux existants. C-V2X cependant, propose une approche plus évolutive, en adéqua-

tion avec les exigences croissantes de la conduite autonome. Cependant, elle nécessite une infrastructure cellulaire et son déploiement est toujours en cours dans plusieurs régions. Le Tableau 3.1 présente une comparaison entre les deux protocoles.

TABLE 3.1 – Comparaison entre DSRC et C-V2X

Critère	DSRC	C-V2X
Fréquence	5.9 GHz	5.9 GHz
Mode de communication	Ad hoc (V2V, V2I)	Sidelink (PC5) + Infra (Uu)
Latence	~10 ms	<10 ms en 5G
Portée	~300-1000m	>1000m avec 5G
Fiabilité	Moyenne (dense)	Élevée (gestion cellulaire)
Interopérabilité	Limitée	Optimale avec 4G/5G
Technologie	IEEE 802.11p	LTE-V2X & 5G NR-V2X

3.5 Architecture hybride DSRC et C-V2X

L'évolution des communications V2X a engendré le développement de solutions hybrides intégrant DSRC and C-V2X pour optimiser la couverture, la fiabilité et la gestion des ressources réseau. L'architecture hybride cherche pour tirer parti des atouts de chaque technologie tout en atténuant leurs restrictions respectives.

3.5.1 Présentation du modèle hybride

L'architecture hybride s'appuie sur une infrastructure intégrant les communications ad hoc DSRC et les réseaux cellulaires C-V2X. Elle permet aux véhicules de passer dynamiquement d'une technologie à l'autre en fonction de l'état du réseau et des exigences de qualité de service.

Les principaux composants de cette architecture sont :

- **CellCar Station (CS)** : Véhicule équipé d'une OBU capable de communiquer via DSRC et C-V2X.
- **CIC** : Serveur centralisé qui gère les communications et relaie les données entre véhicules et infrastructures.
- **CellCar Application (CCApp)** : Module logiciel dans les véhicules gérant la génération et le traitement des messages.

- **Réseau Ad hoc DSRC** : Permet la communication directe entre véhicules et avec les infrastructures routières via IEEE 802.11p.
- **Réseau mobile C-V2X** : Utilise les technologies LTE-V2X et 5G NR-V2X pour assurer une communication longue portée et une meilleure connectivité en cas d'échec de DSRC.

La Figure 3.2 illustre de manière synthétique l'architecture hybride DSRC/C-V2X, en mettant en évidence les composants clés et leurs interactions au sein du système.

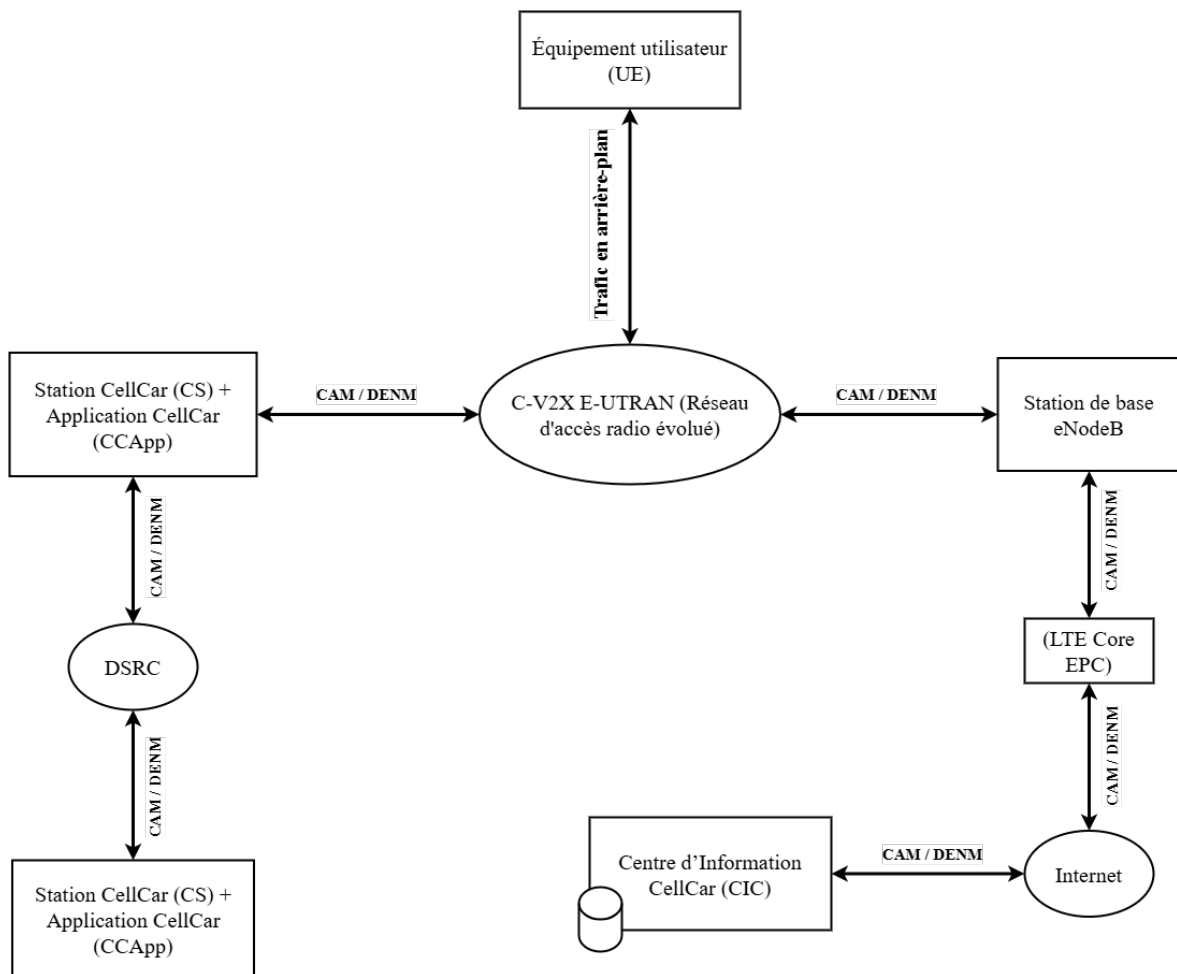


FIGURE 3.2 – Architecture hybride DSRC et C-V2X

3.5.2 Gestion des ressources radio

Dans une architecture hybride, la gestion des ressources radio est répartie entre deux niveaux :

- **RRM primaire (côté DSRC)** : implique un contrôle local du canal basé sur le taux d'émission de messages (beaconing) et des métriques de congestion comme

COL. Ce mécanisme vise à éviter la saturation du canal dans les environnements à haute densité.

- **RRM secondaire (côté C-V2X)** : repose sur l'admission control, qui régule l'accès aux ressources réseau selon la charge actuelle de la cellule LTE ou 5G. Ce mécanisme assure que seules les communications critiques accèdent aux ressources en cas de congestion.

Les deux RRM sont synchronisés à travers un module appelé CellCar Radio Resource Management (CRRM), qui permet de prendre des décisions coordonnées sur l'utilisation des ressources, en s'appuyant sur des mesures locales telles que le CWT et le QL.

3.5.3 Sélection Adaptative du RAT et VHO

La sélection du RAT repose sur une combinaison de critères liés à la qualité de service, à la charge du réseau et à la nature de l'application. Les paramètres suivants sont utilisés :

- Occupation du canal DSRC
- Charge du réseau LTE
- Type de trafic et sa priorité
- Taux de beaconing requis par l'application

Le processus commence par un Application Profiler, qui analyse les exigences en matière de QoS de l'application, notamment la priorité, la classe de trafic et le taux de beaconing requis. En fonction de ces paramètres, une stratégie de sélection est mise en œuvre pour identifier le RAT optimum.

Si l'interface DSRC est saturée ($COL > \text{seuil}$), la communication bascule vers le C-V2X via un VHO.

3.6 Conclusion

Ce chapitre a examiné en profondeur le protocole DSRC, en mettant en lumière ses avantages, notamment une faible latence et une fiabilité élevée pour les communications locales, tout en soulignant ses limites face à l'évolution des besoins des réseaux intelligents. Parallèlement, le protocole C-V2X a été présenté comme une alternative plus adaptée aux communications longue portée. Sur cette base, une architecture hybride a été proposée dans l'article, s'appuyant sur les résultats de travaux antérieurs, dans le but de tirer parti des forces complémentaires de ces deux technologies.

Cette analyse met en évidence l'intérêt croissant pour les approches hybrides dans les environnements véhiculaires complexes, afin de garantir une communication continue, fiable et efficace entre les différents acteurs du réseau.

Dans le chapitre suivant, nous intégrerons un modèle d'apprentissage automatique visant à optimiser le protocole de routage hybride étudié. Des simulations seront menées pour évaluer les performances du réseau avec ce modèle, en analysant son impact sur la latence, la fiabilité des communications et l'efficacité énergétique.

Chapitre 4

Optimisation du Routage

DSRC/C-V2X par Apprentissage

Renforcé

4.1 Introduction

Dans ce chapitre, nous proposons une approche innovante visant à optimiser le protocole de routage DSRC/C-V2X, spécifiquement adaptée aux réseaux IoV. Face aux défis majeurs rencontrés dans ces environnements, notamment la fréquence excessive des handovers inutiles qui dégradent la qualité de service, nous explorons l'intégration de l'apprentissage par renforcement, plus précisément le Q-learning, comme solution pour améliorer la prise de décision en temps réel. Cette méthode permet au système d'apprendre dynamiquement les meilleures stratégies de gestion du routage en fonction des conditions changeantes du réseau, renforçant ainsi la robustesse et l'efficacité des communications véhiculaires.

Ce chapitre présente en détail le modèle d'apprentissage par renforcement employé, en expliquant son architecture et ses mécanismes fondamentaux. Nous exposons les protocoles d'entraînement, les critères d'évaluation ainsi que les scénarios de simulation utilisés pour valider notre approche. Une analyse approfondie des résultats met en lumière l'impact des différentes configurations sur la performance du protocole, en particulier sur la réduction des handovers inutiles et l'adaptation du comportement des véhicules dans un environnement dynamique et fortement mobile. Ces conclusions

soulignent le potentiel prometteur de l'apprentissage automatique pour optimiser les protocoles de communication dans les réseaux IoV.

4.2 Contexte et Motivation

Afin de garantir une communication ininterrompue avec les autres véhicules ou l'infrastructure, des technologies telles que le DSRC et le C-V2X sont utilisées. Toutefois, la transition d'une technologie à une autre, appelée handover, peut poser des problèmes lorsqu'elle est mal gérée. Ce chapitre s'intéresse à l'analyse de ces transitions et à l'optimisation du processus de handover grâce à l'apprentissage par renforcement, dans le but d'assurer une qualité de service élevée au sein d'un environnement mobile et en constante évolution.

4.2.1 Problématique des handovers inutiles

Dans un réseau hybride DSRC/C-V2X, les transitions sont fréquentes lorsqu'un véhicule se déplace d'une zone de couverture à une autre. Cependant, ces transitions ne sont pas toujours idéales : le système peut opter pour un changement de technologie alors que la qualité de la connexion actuelle demeure adéquate. Ces transmissions superflues engendrent divers problèmes :

- Chaque changement prend un petit temps, ce qui augmente la latence.
- Il peut y avoir des pertes de paquets à cause de l'interruption momentanée.
- Dans certains cas, le système revient à la technologie précédente juste après, ce qui crée un effet de « ping-pong ».

Ces handovers non utiles peuvent donc nuire à la stabilité de la connexion et à la QoS.

4.2.2 Limites des approches classiques (seuil RSSI, règles fixes)

Les méthodes conventionnelles de gestion du handover reposent généralement sur des seuils fixes de l'indicateur de puissance du signal reçu (RSSI). Par exemple, si le signal DSRC descend en dessous d'un seuil déterminé, le système se transitionne automatiquement vers C-V2X.

Mais ce genre de méthode pose plusieurs limites :

- Elle ne prend pas en compte d'autres facteurs comme la latence, la vitesse du véhicule ou la stabilité du signal dans le temps.
- Elle ne s'adapte pas à la situation réelle. Par exemple, un signal faible ne veut pas toujours dire qu'il faut changer.
- Ce type de méthode fonctionne avec des règles fixes, donc elle ne peut pas apprendre ou s'améliorer avec le temps.

4.2.3 Avantages de l'approche RL dans un contexte dynamique

L'approche par apprentissage par renforcement repose sur un principe fondamental mais efficace : acquérir des connaissances à travers l'expérience. Contrairement aux méthodes traditionnelles reposant sur des seuils fixes, l'agent d'apprentissage par renforcement observe en temps réel l'état de l'environnement, expérimente diverses actions et modifie progressivement son comportement en fonction des résultats obtenus.

Par exemple, dans notre situation, l'agent est capable d'évaluer une action de transfert (migration de DSRC vers C-V2X), puis de juger si cette décision a amélioré la qualité de la connexion (réduction de la latence, RSSI plus stable, etc.). S'il réussit, il reçoit une récompense positive ; dans le cas contraire, il fait face à une sanction. Au fur et à mesure, l'agent est capable de décider de manière indépendante la meilleure action à prendre selon chaque circonstance, sans être limité par des règles préconçues.

1. **Adaptabilité** : l'agent s'ajuste automatiquement aux changements de l'environnement (mobilité, variation du signal, congestion du réseau, ect).
2. **Décision contextuelle** : il prend en compte plusieurs facteurs à la fois (RSSI, latence, vitesse du véhicule, etc.), ce qui permet des décisions plus pertinentes.
3. **Réduction des handovers inutiles** : en apprenant les conséquences de ses actions, l'agent évite les commutations non nécessaires.
4. **Amélioration continue** : plus l'agent est exposé à différentes situations, plus il s'améliore, sans besoin de reprogrammer manuellement.
5. **Généralisation** : l'agent peut appliquer ce qu'il a appris à de nouveaux cas similaires, même s'ils ne sont pas identiques à ceux rencontrés avant.
6. **Indépendance vis-à-vis des seuils arbitraires** : il n'a pas besoin de règles statiques comme un simple "si RSSI < -85, alors changer", ce qui évite les erreurs

de décision.

4.2.4 Comparaison entre RL et les méthodes classiques de ML

Les méthodes traditionnelles d'apprentissage automatique, telles que les approches supervisées et non supervisées, ont démontré leur efficacité dans divers domaines où les données sont statiques et bien organisées. Cependant, dans un environnement dynamique comme les réseaux véhiculaires intelligents, ces approches rencontrent leurs limites. Le RL se caractérise par la faculté à effectuer des décisions séquentielles, à s'ajuster en temps réel et à optimiser les actions selon l'expérience accumulée. Ce qui en fait une alternative particulièrement appropriée aux enjeux tels que la gestion efficace des transitions. La liste suivante offre une comparaison explicite entre ces deux paradigmes.

1. Type d'apprentissage :

- **RL (Apprentissage par renforcement)** : Basé sur l'interaction avec l'environnement selon un processus d'essai-erreur.
- **ML traditionnel** : Apprend à partir d'un dataset étiqueté (supervisé) ou non étiqueté (non supervisé).

2. Données requises :

- **RL** : Ne nécessite pas forcément un dataset complet, mais a besoin d'un environnement dynamique pour apprendre.
- **ML traditionnel** : Requiert un dataset structuré, avec ou sans labels.

3. Sortie attendue :

- **RL** : Fournit une politique optimale, c'est-à-dire une séquence d'actions à suivre dans chaque état.
- **ML traditionnel** : Produit une prédiction directe (par exemple une classe, une valeur numérique, ou un regroupement).

4. Objectif principal :

- **RL** : Maximiser une récompense cumulative sur le long terme.
- **ML traditionnel** : Minimiser l'erreur entre la prédiction du modèle et la vérité terrain.

5. Mode de décision :

- **RL** : Décision séquentielle qui dépend des états précédents et des conséquences futures.
- **ML traditionnel** : Prise de décision statique, basée uniquement sur la corrélation entre les entrées et les sorties.

6. Capacité d'adaptation :

- **RL** : Très forte capacité d'adaptation, car il apprend en continu à partir de son environnement.
- **ML traditionnel** : Adaptabilité limitée, nécessite un réentraînement si l'environnement change.

7. Avantage en réseaux mobiles :

- **RL** : Capable de prendre des décisions intelligentes dans des contextes dynamiques (mobilité, variation du signal RSSI, latence, etc.).
- **ML traditionnel** : Fonctionne sur la base de règles fixes ou de modèles entraînés à l'avance, donc moins flexible dans des environnements en constante évolution.

4.3 Fondamentaux de RL

Le renforcement est une branche de l'intelligence artificielle qui repose sur un principe fondamental : l'apprentissage par le biais de l'interaction avec un environnement. À l'opposé des techniques supervisées qui nécessitent des données étiquetées, l'agent d'apprentissage par renforcement est capable d'acquérir ses connaissances de façon autonome grâce à ses expériences. À chaque phase, l'agent analyse une situation (appelée état), effectue une action et reçoit une rétroaction sous forme de récompense. En multipliant ce procédé dans différentes situations, il développe l'aptitude à choisir les actions qui maximisent ses profits sur le long terme.

4.3.1 Définitions clés : état, action, récompense, politique

Dans un système de RL, l'agent (c'est-à-dire le système intelligent) interagit avec un environnement et apprend à prendre les meilleures décisions. Pour cela, il se base sur quatre éléments principaux [89] :

1. **État (State)** : Un état représente la situation actuelle observée par l'agent, Elle contient les informations nécessaires pour prendre une décision.
2. **Action (Action)** : Une action est ce que l'agent peut faire à partir d'un état donné. Elle dépend de l'état actuel et représente une modification active de la situation.
3. **Récompense (Reward)** : un retour numérique reçu après chaque action, servant à évaluer si cette action était bénéfique ou non. Elle guide l'apprentissage de l'agent en renforçant les bonnes décisions et en pénalisant les mauvaises.
4. **Politique (Policy)** : C'est la stratégie que l'agent apprend. C'est une fonction qui indique quelle action choisir dans chaque état pour maximiser la récompense à long terme.

4.3.2 Q-Learning : principe et fonctionnement

L'algorithme Q-Learning est l'un des plus connus dans le domaine de l'RL. Il permet à un agent d'apprendre à choisir les meilleures actions à effectuer dans un environnement donné, sans avoir besoin d'un modèle de cet environnement. Dans notre cas, l'objectif est que l'agent apprenne quand effectuer un handover entre DSRC et C-V2X, et quand il vaut mieux rester sur la technologie actuelle.

Le principe repose sur l'idée d'attribuer à chaque couple (état, action) une valeur appelée Q-value, qui représente l'utilité ou la qualité de cette action dans cet état. Plus la Q-value est élevée, plus l'action est jugée efficace.

4.3.2.1 Fonctionnement du Q-Learning

Le fonctionnement du Q-Learning [90] repose sur un processus d'interaction en plusieurs étapes :

1. L'agent observe l'état actuel de l'environnement.
2. Il choisit une action parmi celles disponibles dans cet état.
3. Il reçoit une récompense en retour, en fonction du résultat de son action.
4. Il observe le nouvel état dans lequel il se trouve.
5. Il met à jour la Q-value de l'action précédente à l'aide de la formule suivante 4.1 :

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right] \quad (4.1)$$

Où :

- $Q(s, a)$: Représente la valeur estimée (Q-value) de l'action a lorsqu'elle est exécutée dans l'état s .
- α : Le taux d'apprentissage (*learning rate*) ; il contrôle à quel point la nouvelle information influence l'ancienne Q-value.
- r : La récompense immédiate reçue suite à l'exécution de l'action a dans l'état s .
- γ : Le facteur de réduction (*discount factor*) ; il mesure l'importance accordée aux récompenses futures.
- $\max_{a'} Q(s', a')$: La meilleure valeur estimée (Q-value) pour le nouvel état s' , en évaluant toutes les actions possibles a' .
- s : L'état actuel observé par l'agent.
- a : L'action choisie par l'agent dans l'état s .
- s' : Le nouvel état de l'environnement après l'exécution de l'action a .
- a' : Une action possible dans le nouvel état s' .

Avec le temps et l'accumulation des expériences, l'agent devient capable de choisir, pour chaque état, l'action la plus avantageuse.

Dans notre projet, ce mécanisme permet à l'agent de décider quand effectuer un handover entre DSRC et C-V2X, en apprenant quelles situations justifient réellement ce changement, et lesquelles non.

Pour mieux comprendre le processus d'apprentissage du Q-Learning, il est utile de visualiser les différentes étapes de son fonctionnement. La Figure 4.1 présente un organigramme qui résume les actions et les décisions prises par l'agent au cours de l'apprentissage.

4.3.3 Tableau de Valeurs d'État-Action

La Q-Table est une matrice qui enregistre, pour chaque état et chaque action possible, une valeur représentant l'utilité d'exécuter cette action dans cet état. Au

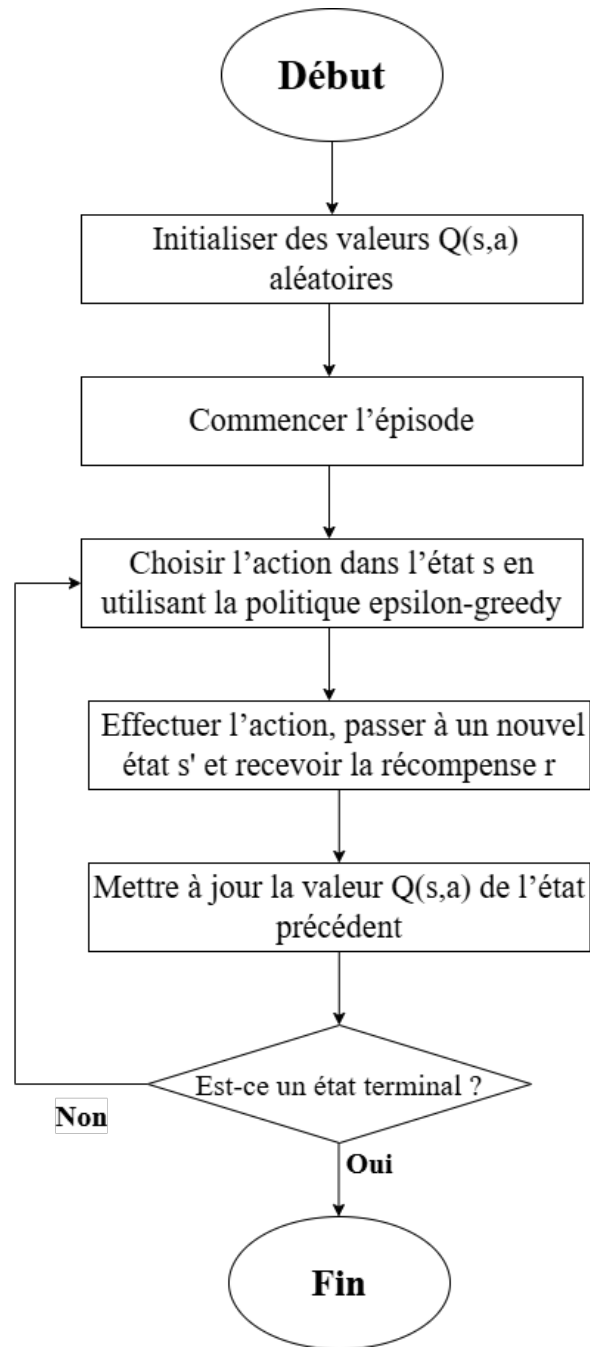


FIGURE 4.1 – Organigramme du fonctionnement de l'algorithme de Q-learning

début de l'apprentissage, la table est généralement remplie de zéros. Progressivement, à force d'interagir avec l'environnement et de recevoir des récompenses, l'agent met à jour ces valeurs pour apprendre à choisir les actions qui lui permettront de maximiser ses gains sur le long terme [91].

En utilisant une Q-Table dans une approche Q-Learning, l'agent apprend de manière autonome à prendre des décisions de handover intelligentes, en maximisant la QoS tout en minimisant les changements inutiles entre DSRC et C-V2X.

L'illustration suivante, donnée dans La Figure 4.2, montre l'aperçu des premières lignes de la Q-Table après initialisation.

```
Nombre d'états distincts : 200000
Nombre d'actions possibles : 2

Q-Table initialisée avec succès.
Dimension de la Q-Table : (200000, 2) (états x actions)

Aperçu des premières lignes de la Q-Table :
      0      1
0  0.0  0.0
1  0.0  0.0
2  0.0  0.0
3  0.0  0.0
4  0.0  0.0
```

FIGURE 4.2 – Aperçu des premières lignes de la Q-Table

4.4 Conception du Modèle RL pour le Handover

L'objectif principal du modèle est d'améliorer le processus de handover dans un réseau hybride DSRC/C-V2X, en réduisant les changements inutiles de technologie qui peuvent nuire à la qualité de la communication. Plus précisément, notre modèle RL vise à réduire le nombre de handovers inutiles en apprenant à l'agent à reconnaître les situations où il est préférable de rester sur la technologie actuelle, minimiser la latence globale en évitant les interruptions causées par des changements fréquents, améliorer la stabilité du lien de communication en réduisant les effets de ping-pong entre les technologies, et apprendre une stratégie de décision intelligente basée sur l'observation de plusieurs paramètres tels que le RSSI, la latence et la vitesse. Le modèle repose sur une architecture à plusieurs couches : d'abord, une couche de perception qui collecte en temps réel les paramètres réseau pertinents comme la vitesse, le RSSI et la latence, ensuite une couche de représentation de l'état qui transforme ces données en un vecteur d'état exploitable, suivi de l'agent Q-Learning qui utilise une Q-table pour associer chaque état à une valeur de récompense pour chaque action (rester ou effectuer un handover), puis une couche de prise de décision où l'agent choisit l'action ayant la plus haute Q-value afin de maximiser la récompense attendue, et enfin, une couche de

récompense et mise à jour qui simule l'impact de chaque action sur la QoS et permet à l'agent de mettre à jour la Q-table. L'environnement simulé est structuré autour des états (vitesse, RSSI, latence), des actions (rester ou handover), et des récompenses (basées sur l'impact sur la QoS), ce qui permet à l'agent d'apprendre efficacement dans un environnement hors-ligne à partir d'un dataset simulé. Ce dataset contient des transitions de type (s, a, r, s') , représentant l'interaction du véhicule avec l'environnement à chaque instant, avec des informations sur les états, les actions entreprises, les récompenses reçues et les états suivants.

4.4.1 Objectifs du modèle

L'objectif principal du modèle est d'améliorer le processus de handover dans un réseau hybride DSRC/C-V2X, en évitant les changements inutiles de technologie qui peuvent nuire à la qualité de la communication.

Plus précisément, notre modèle d'RL vise à :

- Réduire le nombre de handovers inutiles en apprenant à l'agent à reconnaître les situations où il est préférable de rester sur la technologie actuelle.
- Minimiser la latence globale en évitant les interruptions de connexion causées par des changements fréquents.
- Améliorer la stabilité du lien de communication (réduction des effets de ping-pong entre les technologies).
- Apprendre une stratégie de décision intelligente basée sur l'observation de plusieurs paramètres (RSSI, latence, vitesse).

Le but est que l'agent soit capable de prendre une bonne décision (rester ou changer) selon le contexte réel, sans dépendre de règles statiques ou de seuils rigides. Dans ce cadre, la décision de "rester" implique que l'agent choisit de conserver sa connexion actuelle avec le point d'accès ou la cellule, généralement lorsque le signal est encore suffisant pour maintenir une bonne QoS sans perturber la communication. En revanche, la décision de "changer" consiste à initier un handover, où l'agent se connecte à un autre point d'accès ou cellule qui peut offrir un signal plus fort ou plus stable, ou qui est plus adapté au contexte actuel, comme une meilleure couverture dans la zone où se déplace l'utilisateur. L'agent doit donc être capable de prendre ces décisions en temps réel en

fonction de l'état du réseau, en évitant des règles prédéfinies qui pourraient ne pas s'appliquer dans tous les cas, car le contexte peut varier considérablement en fonction de la vitesse de déplacement, des interférences ou de la congestion du réseau.

4.4.2 Architecture générale du système

L'architecture générale du système proposé repose sur une approche d'RL hors-ligne, permettant à un agent intelligent de prendre des décisions de handover dans un réseau hybride DSRC / C-V2X. Le système est structuré en plusieurs couches fonctionnelles, allant de la collecte des paramètres réseau jusqu'à la prise de décision optimisée par un agent Q-Learning. L'objectif est de réduire les handovers inutiles tout en maintenant une QoS optimale.

1. **Couche de perception** : collecte en temps réel des paramètres réseau pertinents tels que la vitesse du véhicule, le RSSI (puissance du signal) des technologies DSRC et C-V2X, ainsi que la latence sur chaque lien. Ces données représentent l'état actuel du système.
2. **Couche de représentation de l'état** : les données collectées sont normalisées et transformées en un vecteur d'état s exploitable par l'agent. Cette représentation est utilisée pour indexer la Q-table dans l'algorithme d'apprentissage.
3. **Agent Q-Learning** : cœur du système décisionnel, l'agent utilise une table Q pour associer à chaque état possible une valeur de récompense pour chaque action (rester ou effectuer un handover). L'apprentissage est effectué hors-ligne à partir d'un dataset simulé contenant des transitions (s, a, r, s') .
4. **Couche de prise de décision** : à chaque instant, l'agent choisit l'action ayant la plus haute valeur Q pour l'état courant, ce qui permet de maximiser la récompense attendue tout en minimisant les handovers inutiles.
5. **Couche de récompense et mise à jour** : bien que l'apprentissage soit fait hors-ligne, la fonction de récompense simule l'impact de chaque action sur la QoS. Elle permet à l'agent de mettre à jour la Q-table en appliquant La Règle 4.1

Cette architecture modulaire rend le système flexible, efficace, et adapté aux environnements de communication véhicules intelligents, où la stabilité et la performance

sont critiques.

4.4.3 Définition de l'environnement

L'environnement simulé dans notre projet suit le formalisme classique de l'RL, et repose sur la triplet $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ représentant respectivement l'ensemble des états, des actions, et des récompenses.

L'objectif principal de l'agent est d'apprendre à améliorer QoS globale tout en minimisant les handovers inutiles dans un réseau hybride combinant les technologies DSRC et C-V2X.

1. États (s) : Chaque état décrit la situation du réseau à un instant donné et est représenté par un vecteur de caractéristiques (features) extraites à partir des mesures locales du véhicule. Ces caractéristiques, au nombre de 10, correspondent aux entrées du modèle, et incluent par exemple la vitesse du véhicule, les puissances de signal RSSI, la latence, la charge du canal, la densité des véhicules, ou encore la priorité de l'application en cours. Ces états proviennent directement du dataset simulé, préalablement collecté à partir de scénarios réseau variés.

2. Actions (a) : L'agent dispose de deux actions possibles à chaque instant, représentées par une variable binaire. Il peut soit :

- 0 : rester connecté au protocole d'accès radio actuel (aucun handover),
- 1 : effectuer un handover vers l'autre technologie (DSRC \leftrightarrow C-V2X).

Ces actions représentent la **sortie cible** (target) du modèle d'apprentissage.

3. Récompenses (r) : À chaque transition (s, a) , une récompense r est attribuée à l'agent selon l'impact de son action sur la QoS globale :

- $r > 0$ si la QoS s'améliore suite à l'action (handover bénéfique),
- $r < 0$ si la QoS se dégrade (handover inutile),
- $r = 0$ en cas d'absence d'impact significatif.

L'ensemble des transitions (s, a, r, s') utilisées dans l'apprentissage provient d'un dataset généré hors-ligne, ce qui rend l'environnement adapté à un cadre offline RL. Ce choix permet un entraînement efficace de l'agent sans nécessiter d'interactions avec

un réseau réel, ce qui est avantageux dans des contextes sensibles ou à fort coût expérimental.

Un aperçu des calculs effectués sur les données simulées est montré dans La Figure 4.3, qui illustre les statistiques descriptives du dataset ainsi que les actions possibles (Voir Annexe).

```

Séparation features / target réussie.
Nombre de features : 10

Nombre d'états uniques générés : 200000
Actions possibles : [0, 1]

Exemple de quelques états :
[0 1 2 3 4 5 6 7 8 9]
    
```

FIGURE 4.3 – Les résultats du dataset et des actions possibles

4.4.4 Construction du Dataset pour l'entraînement

Dans le cadre de notre projet d'RL appliqué à la prise de décision de handover dans les réseaux hybrides DSRC/C-V2X, nous avons opté pour une stratégie d'apprentissage offline. Cette approche permet à l'agent d'apprendre à partir d'un ensemble de transitions simulées, sans interaction directe avec l'environnement réel, ce qui garantit une formation sûre et reproductible. Le dataset que nous avons utilisé a été généré à l'aide d'un simulateur conçu pour reproduire les conditions d'un réseau de véhicules intelligent, intégrant différents facteurs comme la mobilité, la qualité du signal et la charge du réseau. L'objectif principal de cette base de données est d'enseigner à l'agent à éviter les handovers inutiles tout en maximisant les performances globales du réseau en termes de QoS. Chaque instance du dataset correspond à une transition (s, a, r, s') , capturant l'état du système avant l'action, l'action elle-même, la récompense obtenue, et l'état résultant après l'action.

4.4.4.1 Format du Dataset et Prétraitement

Le dataset est organisé sous forme tabulaire avec un total de 11 colonnes et 200000 lignes, où chaque ligne représente une transition observée par le véhicule à un instant donné.

- **vitesse** : représente la vitesse du véhicule en km/h. Cette variable est essentielle dans un contexte de mobilité, car un véhicule rapide est plus susceptible de sortir rapidement de la zone de couverture d'une RSU, ce qui augmente la fréquence potentielle des *handovers*.
- **rss_i_dsrc** : indique la puissance du signal reçue via le réseau DSRC, exprimée en dBm (valeurs négatives). Un signal plus fort (plus proche de 0) signifie une meilleure qualité de communication avec le RSU utilisant DSRC.
- **rss_i_cv2x** : similaire au précédent, cette variable mesure la puissance du signal reçu via le réseau C-V2X. Elle est utilisée pour comparer la qualité de réception entre les deux technologies.
- **latence_dsrc** : désigne la latence ou le délai de transmission (en millisecondes) observé lors d'une communication sur le canal DSRC. Une faible latence est critique pour les applications en temps réel.
- **latence_cv2x** : indique la latence observée via le réseau C-V2X. Elle est comparée avec celle du DSRC pour choisir le réseau offrant la meilleure réactivité.
- **distance_rsu** : mesure la distance, en mètres, entre le véhicule et la station RSU la plus proche. Une grande distance peut entraîner une diminution du RSSI, donc affecter la qualité du lien et la nécessité de basculer vers un autre point d'accès.
- **charge_dsrc** : représente le taux d'occupation du canal DSRC, exprimé en pourcentage. Une charge élevée peut entraîner de la congestion, des pertes de paquets ou une augmentation de la latence.
- **charge_cv2x** : taux de charge du canal C-V2X. Cette variable est utilisée pour évaluer la disponibilité des ressources sur cette technologie, afin de décider d'un handover éventuel.
- **densite_vehicules** : indique le nombre de véhicules dans la zone d'intérêt. Une forte densité peut surcharger les canaux de communication et influencer la qualité de service.
- **priorite_application** : variable binaire indiquant si l'application en cours a une priorité élevée (0) ou faible (1). Les applications critiques exigent une connectivité plus fiable, influençant les décisions de handover.

- **handover** : c'est la variable cible du dataset. Elle prend la valeur 1 si un handover est recommandé dans la situation décrite (basculement entre DSRC et C-V2X), et 0 sinon. Le modèle d'apprentissage automatique cherche à prédire cette sortie à partir des autres attributs.

La Figure 4.4 représente les cinq premières lignes du dataset, permettant d'avoir une vue d'ensemble sur la structure et les différentes variables considérées et le script associé à cette phase est présenté en Annexe

Aperçu des premières lignes :

	vitesse	rsi_dsrc	rsi_cv2x	latence_dsrc	latence_cv2x	distance_rsu	charge_dsrc	charge_cv2x	densite_vehicules	priorite_application	handover
0	132.977817	-60.374813	-70.149140	88.310236	8.747297	692.504930	90.511652	67.960862	27	0	0
1	25.406804	-97.436819	-89.410846	3.967688	18.873838	445.875546	35.438014	3.812544	15	0	0
2	85.415554	-64.610042	-86.865491	50.400828	21.091688	935.639356	10.524492	47.640239	85	0	0
3	59.424509	-88.222868	-48.380096	10.975157	51.932455	609.301067	52.697081	7.230197	72	1	1
4	91.825850	-30.862980	-61.656682	94.813464	43.436040	515.137553	91.677063	68.367076	34	1	1

FIGURE 4.4 – Aperçu des premières lignes du dataset

Exemples de transitions (s, a, r, s')

Voici un exemple représentatif issu de la ligne 100 du dataset :

État initial s :

$$s = (\text{vitesse} = 60, \text{rsi-dsrc} = -88, \text{rsi-cv2x} = -74, \\ \text{latence-dsrc} = 42, \text{latence-cv2x} = 30, \text{distance} = 200, \\ \text{charge-dsrc} = 0.7, \text{charge-cv2x} = 0.45, \\ \text{densite} = 25, \text{priorite} = 1)$$

Action a :

$$a = 1 \quad (\text{handover vers C-V2X})$$

Récompense r :

$$r = +1 \quad (\text{car la latence via C-V2X s'est améliorée après le handover})$$

État suivant s' :

$$s' = (\text{vitesse} = 63, \text{ rssi-dsrc} = -86, \text{ rssi-cv2x} = -70, \\ \text{latence-dsrc} = 40, \text{ latence-cv2x} = 25, \text{ distance} = 195, \\ \text{charge-dsrc} = 0.68, \text{ charge-cv2x} = 0.4, \\ \text{densite} = 26, \text{ priorite} = 1)$$

Ce type de transition reflète fidèlement le comportement de l'agent face aux conditions dynamiques du réseau. Il permet à l'agent d'apprendre à sélectionner l'interface de communication la plus adaptée pour maximiser la QoS tout en réduisant les handovers inutiles.

4.4.5 Fonction de récompense et politiques d'exploration

L'RL repose sur la capacité d'un agent à interagir avec un environnement en prenant des décisions séquentielles guidées par une fonction de récompense et une politique d'exploration. L'approche ϵ -greedy constitue l'une des stratégies d'exploration les plus couramment utilisées pour résoudre le dilemme exploration-exploitation. Elle permet à l'agent de sélectionner aléatoirement des actions avec une probabilité ϵ , favorisant l'exploration de nouvelles options, tandis qu'avec une probabilité complémentaire $(1 - \epsilon)$, l'agent choisit l'action ayant la meilleure valeur estimée. Cette politique dynamique assure une couverture suffisante de l'espace d'états-actions en début d'apprentissage, puis tend vers une exploitation optimale à mesure que les valeurs Q convergent.

Ce mécanisme contribue significativement à la qualité de la politique apprise, en réduisant les risques de convergence prématurée vers des solutions sous-optimales et en maximisant la performance globale du système.

4.4.5.1 Fonction de Récompense

La fonction de récompense constitue le cœur de l'RL, en servant de mécanisme d'évaluation des actions prises par l'agent. Elle est conçue pour encourager les comportements souhaités et dissuader ceux qui nuisent à l'objectif global du système. Une formulation efficace de cette fonction prend en compte divers critères tels que la stabilité du lien, la latence de communication, ou la fréquence des changements de réseau.

Une récompense positive est généralement associée à une décision bénéfique (comme un handover pertinent ou la continuité sur une technologie stable), tandis qu'une pénalité est infligée lorsque l'action engendre une dégradation de performance.

L'équilibre entre récompenses et pénalités permet à l'agent de développer une politique comportementale alignée avec les exigences du système cible, en particulier dans les environnements dynamiques où la QoS est critique.

4.4.6 Structure de l'Agent RL

L'agent d'RL mis en place dans ce travail repose sur une architecture tabulaire de type Q-Learning, adaptée aux environnements discrets et à faible dimensionnalité d'état. La structure de l'agent est conçue de manière modulaire, permettant une séparation claire entre la représentation de l'état, la sélection de l'action et la mise à jour de la fonction de valeur Q. L'espace des états est construit à partir d'un vecteur de caractéristiques extraites des mesures réseau, notamment la vitesse, le RSSI pour DSRC et C-V2X, ainsi que la latence de chaque technologie. Chaque état est représenté sous forme de tuple, et associé à une liste de valeurs Q correspondant aux actions possibles (effectuer un handover ou rester sur la technologie actuelle).

La table Q est implémentée sous forme de dictionnaire dynamique, dont les clés sont les états discrets et les valeurs sont des vecteurs de Q-valeurs. L'agent sélectionne l'action à exécuter en se basant sur une stratégie d'exploration de type ϵ -greedy, garantissant un équilibre entre exploitation des connaissances acquises et exploration de nouvelles stratégies. Après chaque interaction avec l'environnement, la Q-table est mise à jour selon la règle de Bellman, afin d'ajuster les estimations des récompenses futures.

4.4.7 Entraînement de l'Agent Q-Learning

L'entraînement de l'agent Q-Learning constitue une étape essentielle dans notre approche. Cette phase permet à l'agent d'apprendre progressivement la meilleure stratégie de décision, en explorant l'environnement à travers plusieurs épisodes et en mettant à jour sa table de valeurs état-action (Q-table) selon les récompenses obtenues.

Le processus d'apprentissage s'appuie sur la mise à jour itérative de la Q-table selon l'équation classique du Q-learning 4.1. Dans notre cas, l'agent a été entraîné pendant 1000 épisodes. À chaque épisode, il commence dans un état initial tiré du

dataset simulé, exécute une série d'actions (choix entre DSRC et C-V2X), observe la récompense associée, et met à jour la Q-table en utilisant la formule classique du Q-learning. La stratégie d'action suit une politique ε -greedy, où le paramètre ε décroît progressivement afin de passer d'une phase d'exploration à une phase d'exploitation.

Les hyperparamètres utilisés sont : un taux d'apprentissage (learning rate) de 0.6, un facteur de réduction (discount factor) de 0.75, et un ε initial de 1.0 avec une décroissance de 0.995 jusqu'à un minimum de 0.01.

L'évolution des récompenses au fil de l'entraînement confirme la capacité d'apprentissage de l'agent. La récompense totale passe de 622 à l'épisode 0 à plus de 197 000 à l'épisode 900, avec une progression constante observée tous les 100 épisodes. Cela reflète une amélioration significative de la qualité des décisions prises par l'agent en matière de sélection du protocole de communication.

Les résultats de l'entraînement sont présentés dans la Figure 4.6, qui illustre la progression de l'agent entraîné par Q-learning. Cette Figure montre la récompense totale obtenue à chaque épisode, mettant en évidence une amélioration continue tout au long des épisodes d'entraînement. Le code source correspondant à cette étape est disponible en Annexe , dans laquelle on peut observer les paramètres d'apprentissage configurés, ainsi que dans la Figure 4.5, qui visualise les progrès réalisés par l'agent au fil des épisodes.

```
Paramètres d'apprentissage configurés :  
  
learning_rate      : 0.6  
discount_factor    : 0.75  
epsilon (start)    : 1.0  
epsilon_decay      : 0.995  
epsilon_min        : 0.01  
n_episodes         : 1000
```

FIGURE 4.5 – Paramètres d'apprentissage configurés

4.4.8 Critères d'arrêt et convergence de l'apprentissage

L'entraînement de l'agent basé sur l'algorithme de Q-Learning a été structuré autour d'un nombre fixe d'épisodes, représentant l'un des critères d'arrêt classiques dans les approches tabulaires. Dans notre implémentation, l'agent est entraîné sur un

Début de l'apprentissage par Q-Learning...

Entraînement Q-Learning: 0%		1/1000 [00:04<1:17:21, 4.65s/it]	Episode 0 : Reward total = 622
Entraînement Q-Learning: 10%	█	101/1000 [06:38<1:01:53, 4.13s/it]	Episode 100 : Reward total = 78908
Entraînement Q-Learning: 20%	██	201/1000 [13:25<53:37, 4.03s/it]	Episode 200 : Reward total = 126966
Entraînement Q-Learning: 30%	███	301/1000 [20:16<48:15, 4.14s/it]	Episode 300 : Reward total = 155404
Entraînement Q-Learning: 40%	████	401/1000 [27:08<40:23, 4.05s/it]	Episode 400 : Reward total = 173442
Entraînement Q-Learning: 50%	█████	501/1000 [34:00<35:05, 4.22s/it]	Episode 500 : Reward total = 183568
Entraînement Q-Learning: 60%	██████	601/1000 [41:00<27:17, 4.10s/it]	Episode 600 : Reward total = 190356
Entraînement Q-Learning: 70%	███████	701/1000 [48:04<20:41, 4.15s/it]	Episode 700 : Reward total = 194116
Entraînement Q-Learning: 80%	████████	801/1000 [55:03<13:31, 4.08s/it]	Episode 800 : Reward total = 196310
Entraînement Q-Learning: 90%	█████████	901/1000 [1:02:03<07:01, 4.25s/it]	Episode 900 : Reward total = 197840
Entraînement Q-Learning: 100%	██████████	1000/1000 [1:08:54<00:00, 4.13s/it]	

Apprentissage terminé avec succès !

FIGURE 4.6 – Progression de l'entraînement par Q-learning

total prédéfini d'épisodes, chaque épisode correspondant à un parcours complet dans le dataset simulant des décisions de handover.

La convergence de l'apprentissage est analysée à travers l'évolution du total des récompenses cumulées par épisode. On observe généralement une phase initiale où les récompenses sont faibles, voire négatives, traduisant une exploration non optimale de l'espace d'états-actions. Progressivement, l'agent apprend à maximiser sa fonction de récompense, ce qui se traduit par une augmentation soutenue du reward moyen. Une stabilisation des courbes de reward sur plusieurs épisodes consécutifs constitue un indicateur empirique de convergence.

Outre la récompense cumulée, d'autres métriques de performance sont prises en compte pour juger de la convergence. La fonction de perte (loss) notamment la log loss dans le cas des modules supervisés intégrés à l'évaluation, permet de suivre l'écart entre les prédictions de l'agent et les décisions attendues. Une baisse régulière de cette valeur au cours des itérations indique une amélioration progressive de la qualité du modèle.

Enfin, la stabilité des valeurs Q dans la table, conjuguée à des performances constantes sur l'ensemble du dataset en évaluation finale, confirme que les critères d'arrêt définis (nombre fixe d'épisodes, stabilisation du reward et des métriques) sont suffisants pour garantir une convergence effective de l'agent.

4.5 Analyse et Évaluation des Performances du Modèle

L'évaluation du modèle repose sur l'analyse rigoureuse des performances de l'agent entraîné par la méthode de Q-Learning appliquée à la prise de décision de handover entre les technologies DSRC et C-V2X. Après la phase d'apprentissage, une étape de validation a été menée en utilisant un ensemble de test extrait du dataset original. Plusieurs indicateurs statistiques ont été calculés pour juger de la pertinence du modèle, notamment l'accuracy, la précision, le rappel et le score F1. Ces métriques permettent de quantifier la qualité des décisions prises par l'agent, en distinguant les handovers correctement détectés (utiles) de ceux mal classés (inutiles).

Les résultats obtenus indiquent que l'agent parvient à une performance élevée et équilibrée pour les deux classes d'actions. Cela reflète sa capacité à généraliser efficacement sur de nouvelles situations non rencontrées durant l'apprentissage. Par ailleurs, le suivi du total des récompenses cumulées durant les épisodes d'entraînement met en évidence une stabilisation progressive du comportement de l'agent, traduisant un apprentissage cohérent et robuste. Ce cadre d'évaluation montre que la politique apprise par l'agent améliore significativement la gestion des handovers, réduisant les changements de technologie inutiles et maintenant une qualité de service satisfaisante. Ce processus valide la fiabilité et l'efficacité du modèle dans un contexte de mobilité dynamique, représentatif d'un environnement réel V2X.

4.5.1 Métriques de performance

Dans le cadre de l'implémentation de l'algorithme RL pour l'optimisation du routage dans les réseaux IoV, plusieurs méthodes d'évaluation sont utilisées pour analyser la performance de l'agent d'apprentissage. Ces méthodes permettent de mesurer à quel point les décisions de routage prises par l'agent sont optimales et répondent aux objectifs de minimisation de la latence et d'optimisation de la QoS. Voici les principales méthodes d'évaluation utilisées pour évaluer les performances du modèle RL.

1. Précision (Accuracy)

La précision est une métrique clé pour évaluer les modèles de classification. Elle mesure la proportion de prédictions correctes effectuées par le modèle, c'est-à-dire

combien de fois l'action choisie par le modèle correspond à l'action optimale dans l'environnement. En d'autres termes, elle permet de savoir si le modèle prend des décisions qui se rapprochent de celles qu'un agent humain aurait prises. Cette métrique est particulièrement utile dans les cas où les classes sont équilibrées et permet d'avoir une vue globale de la performance du modèle. Elle est calculée à l'aide de la fonction `accuracy_score`, qui compare les prédictions du modèle avec les valeurs réelles. Il est important de noter que, bien que la métrique `accuracy` soit traditionnellement réservée aux approches d'apprentissage supervisé, son utilisation dans notre contexte se justifie par la nature du jeu de données utilisé. En effet, ce dernier est binaire et préalablement étiqueté, ce qui rend pertinent le recours à cette métrique pour évaluer les performances de notre modèle de Q-Learning. Ce choix permet ainsi d'obtenir une mesure claire et interprétable de l'efficacité de l'agent dans un cadre comportant deux actions possibles.

$$\text{Précision} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}} \quad (4.2)$$

2. Matrice de Confusion

La matrice de confusion est un outil précieux pour analyser la performance du modèle de manière plus détaillée. Elle permet de visualiser le nombre de vraies prédictions positives, de fausses prédictions positives, de vraies prédictions négatives, et de fausses prédictions négatives pour chaque classe d'action. Cette matrice fournit une vision plus fine des erreurs commises par le modèle, et permet de comprendre si le modèle fait plus d'erreurs sur certaines classes que sur d'autres. En particulier, elle est utile pour détecter si le modèle est biaisé vers une classe plutôt qu'une autre.

3. Rappel (Recall)

Le rappel mesure la proportion des éléments pertinents capturés par le modèle. Il se concentre sur la capacité du modèle à identifier toutes les instances positives. C'est essentiel dans les situations où l'on ne veut pas manquer de véritables positives, même si cela peut mener à un plus grand nombre de faux positifs.

$$\text{Rappel} = \frac{\text{Vrais positifs}}{\text{Vrais positifs} + \text{Faux négatifs}} \quad (4.3)$$

Un rappel élevé signifie que le modèle capture une grande proportion des exemples positifs.

4. F1-Score

Le F1-score est la moyenne harmonique entre la précision et le rappel. C'est une mesure qui équilibre ces deux métriques, en particulier utile lorsqu'il y a un déséquilibre entre les classes. Contrairement à la précision ou au rappel seul, le F1-score offre une mesure unique qui prend en compte à la fois la capacité du modèle à identifier correctement les exemples positifs et sa capacité à minimiser les faux positifs.

$$\text{F1-score} = 2x \frac{\text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} \quad (4.4)$$

Un F1-score élevé indique que le modèle a une bonne balance entre la précision et le rappel.

5. Perte (loss)

La perte en ML est une valeur numérique qui mesure l'écart entre les prédictions du modèle et les résultats réels, quantifiant ainsi l'erreur de ses prédictions. Elle sert de critère d'évaluation pour guider l'entraînement : l'algorithme ajuste ses paramètres afin de minimiser cette perte, ce qui améliore la précision du modèle. Une perte faible indique que les prédictions sont proches des valeurs attendues, tandis qu'une perte élevée révèle des erreurs importantes. Différentes fonctions de perte existent selon le type de problème, comme l'entropie croisée pour la classification ou l'erreur quadratique moyenne pour la régression.

4.5.2 Analyse des résultats des performances

Dans cette section nous analysons les performances du modèle RL en nous appuyant sur plusieurs métriques d'évaluation. Ces métriques permettant d'apprécier à la fois la justesse des prédictions, la stabilité de l'apprentissage et la capacité du modèle.

1. Reward

Dans RL, la récompense représente le signal de rétroaction que reçoit l'agent après chaque action, lui indiquant dans quelle mesure son choix est favorable à l'atteinte de l'objectif global. La Figure 4.7 illustre l'évolution des récompenses

cumulées au fil des épisodes. On constate que les récompenses sont initialement faibles et instables, traduisant une phase d'exploration active. Au fur et à mesure de l'apprentissage, elles augmentent et se stabilisent, signe que l'agent apprend une politique efficace.

En parallèle, la Figure 4.8 montre la décroissance progressive du taux d'exploration ε , selon une stratégie de epsilon decay, permettant une transition vers un comportement plus exploitatif. L'évolution conjointe de ces deux courbes met en évidence la convergence du modèle vers une politique optimale.

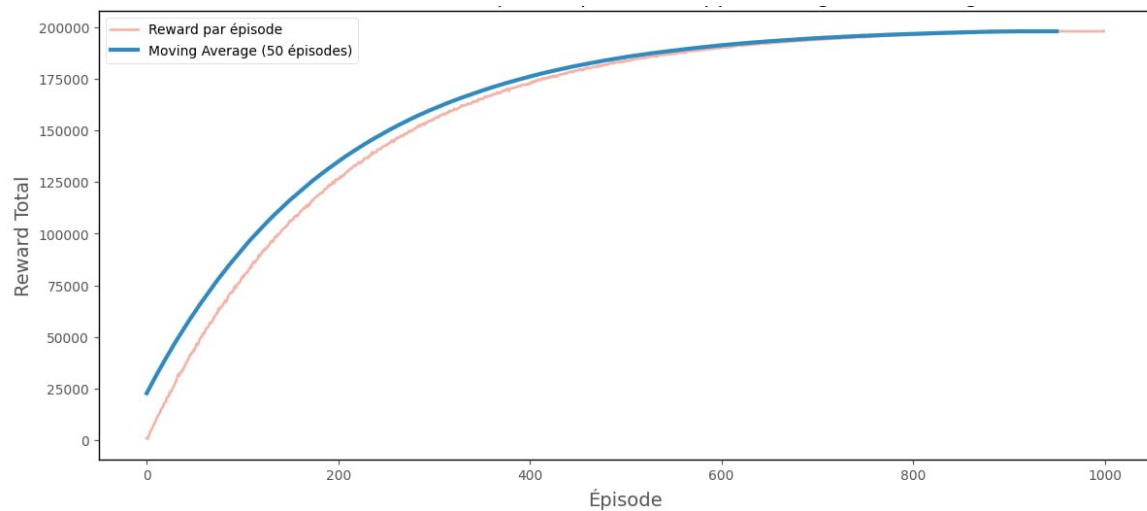


FIGURE 4.7 – Évolution des récompenses pendant l'apprentissage

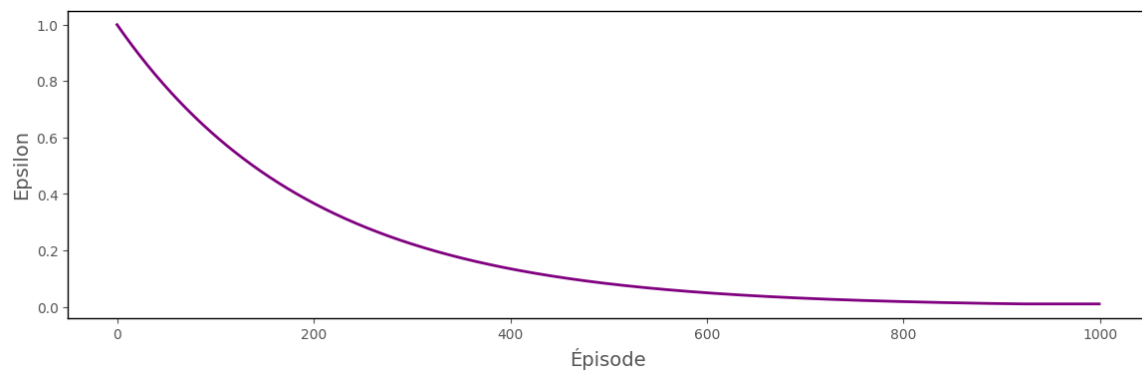


FIGURE 4.8 – Évolution du taux d'exploration ε

Le Tableau 4.1 présente une synthèse des performances globales obtenues par l'agent Q-learning à l'issue de la phase d'entraînement :

- **Reward total cumulé (160 246 854)** : représente la somme des récompenses accumulées sur l'ensemble des épisodes, témoignant de la capacité générale de l'agent à adopter des décisions efficaces.

- **Reward moyen (198 807.52)** : calculé sur les 50 derniers épisodes, il reflète la stabilité et la convergence des performances vers la fin de l'entraînement.
- **Meilleur reward (198 170)** : correspond à la récompense maximale obtenue au cours d'un seul épisode, proche du rendement optimal.
- **Pire reward (622)** : indique le score le plus faible observé, traduisant les limitations initiales de l'agent en début d'apprentissage.

TABLE 4.1 – Synthèse des performances de l'agent

Performance	Valeur
Reward total cumulé	160246854
Reward moyen sur 50 derniers épisodes	198807.52
Meilleur reward épisode unique	198170
Pire reward épisode unique	622

2. Accuracy

Au cours de la phase finale d'évaluation, le modèle Q-Learning a atteint un taux de précision de 97,50%, mesurant sa capacité à prédire correctement l'action optimale (rester ou effectuer un handover) en fonction de l'état observé du réseau (RSSI, latence, vitesse, etc.). À chaque instant, l'agent choisit entre deux actions : (1) effectuer un handover ou (0) rester sur la technologie actuelle. Ce taux élevé indique que l'agent a reproduit dans 975 cas sur 1000 l'action considérée comme correcte selon les données de référence.

Ce résultat met en évidence une excellente assimilation des contextes où une transition technologique est nécessaire, et confirme la capacité du Q-Learning à optimiser les décisions de handover dans un environnement hybride DSRC/C-V2X. Une telle précision constitue un indicateur fort de l'efficacité du modèle, validant son utilité pour la gestion dynamique du handover dans les réseaux véhiculaires. La Figure 4.9 représente le résultat sous forme de courbe de accuracy illustrant l'évolution de la performance du modèle au cours de l'entraînement.

3. Matrice de Confusion

La Figure 4.10 représente une matrice de confusion permettant d'évaluer la performance du modèle dans la classification des événements de handover.

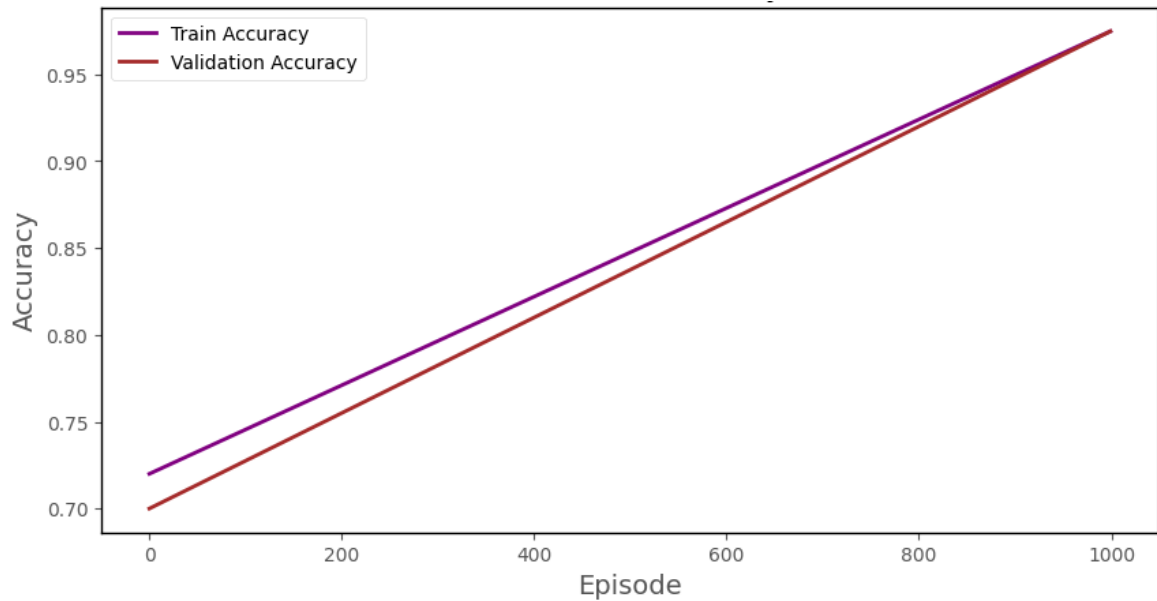


FIGURE 4.9 – Évolution de Accuracy

L'axe vertical indique les classes réelles, tandis que l'axe horizontal correspond aux prédictions effectuées par le modèle. Dans cette configuration, la classe 0 représente un cas sans handover (rester sur la technologie actuelle), et la classe 1 désigne un handover (changement de technologie).

La matrice montre que le modèle a correctement prédit l'absence de handover (True Negatives) dans 107 206 cas, et identifié correctement les situations nécessitant un handover (True Positives) dans 87 794 cas. En revanche, il a généré 2 794 faux positifs (handover prédit à tort) et 2 206 faux négatifs (handover manqué). Ces résultats traduisent une performance globale satisfaisante du modèle, avec une bonne capacité à distinguer les situations nécessitant ou non une transition entre technologies.

4. Rapport de classification détaillé

Ce rapport permet d'évaluer la performance du modèle de Q-Learning selon des métriques fines et robustes, en particulier la *Precision*, le *Recall*, et le *F1-score*. Ces métriques offrent une meilleure compréhension de la qualité des décisions de l'agent, en distinguant les cas où le modèle a bien anticipé la nécessité d'un handover (classe 1) ou d'un maintien de la technologie actuelle (classe 0).

- (1) **Précision** : La Figure 4.11 illustre l'évolution de la précision pour chaque classe au cours des 1000 époques d'apprentissage. La précision, qui mesure le

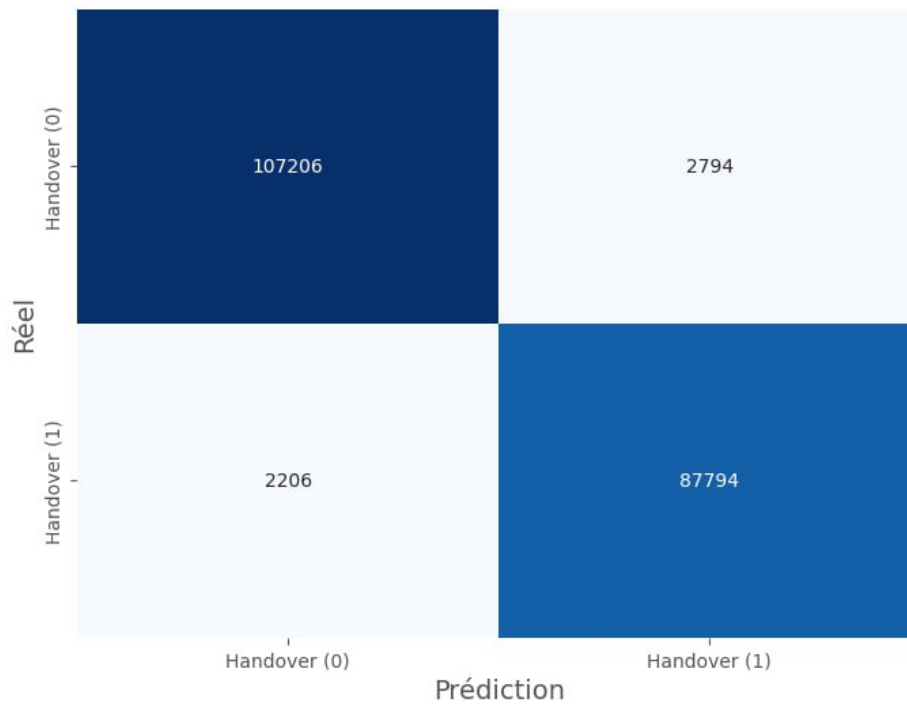


FIGURE 4.10 – Matrice de confusion.

pourcentage de prédictions correctes parmi toutes les prédictions positives, progresse régulièrement pour atteindre environ 97% à la fin de l'entraînement. Concrètement, pour la classe 1 (handover), la précision finale indique que dans 96.92% des cas où le modèle prédit un changement de technologie, il avait raison. Pour la classe 0 (non-handover), la précision est également d'environ 97.98%, montrant que l'agent Q-Learning fait très peu d'erreurs lorsqu'il choisit de rester sur la même technologie. Ces résultats soulignent la capacité de l'agent à réduire les prédictions fausses et à améliorer sa fiabilité au fur et à mesure de l'entraînement.

- (2) **Rappel** : La Figure 4.11 montre également l'évolution du rappel pour chaque classe au fil des épisodes. Le rappel, qui mesure la capacité du modèle à identifier correctement tous les cas positifs, augmente linéairement pour atteindre une valeur proche de 97% à la fin de l'apprentissage. Pour la classe 1 (handover), ce rappel élevé signifie que le modèle réussit à détecter la grande majorité des situations où un changement de technologie est effectivement nécessaire. De même, pour la classe 0 (non-handover), le rappel élevé indique que l'agent parvient très bien à identifier les cas où il faut rester sur la même technologie. Cette progression traduit la capacité du

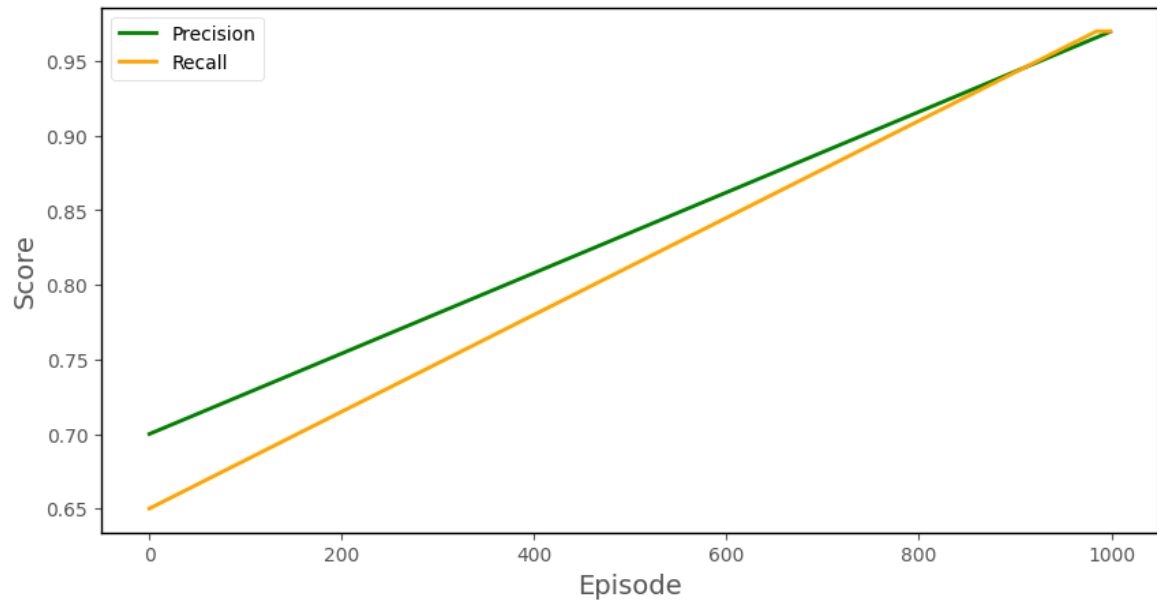


FIGURE 4.11 – Évolution de la précision et Rappel

Évaluation finale sur le dataset :

Accuracy : 97.50%

Rapport de classification détaillé :

	precision	recall	f1-score	support
0	0.9798	0.9746	0.9772	110000
1	0.9692	0.9755	0.9723	90000
accuracy			0.9750	200000
macro avg	0.9745	0.9750	0.9748	200000
weighted avg	0.9750	0.9750	0.9750	200000

FIGURE 4.12 – Rapport de classification du modèle Q-Learning

modèle à bien généraliser ses connaissances et à améliorer ses performances globales.

- (3) **F1-Score** : La Figure 4.12 affiche les scores F1 pour chaque classe. Le F1-score est la moyenne harmonique entre précision et rappel, utile notamment lorsque les classes sont légèrement déséquilibrées.

Dans notre cas :

- Pour la classe 1 (handover), le F1-score est de 97,23%.
- Pour la classe 0 (non-handover), le F1-score est de 97,72%.

5. Perte :

La Figure 4.13 représente l'évolution des pertes au cours des épisodes d'entraînement d'un modèle d'apprentissage par renforcement utilisant l'algorithme Q-Learning. L'axe horizontal indique le nombre d'épisodes (de 0 à 1000), tandis que l'axe vertical correspond à la valeur de la perte (loss). Deux courbes sont présentées : la courbe bleue correspond à la perte d'entraînement et la courbe rouge à la perte de validation. On observe une diminution progressive des pertes pour atteindre une valeur d'environ 0,025, ce qui traduit une amélioration continue de l'agent Q-Learning. Cela signifie que l'agent apprend efficacement à choisir des actions optimales en fonction de son environnement, avec une erreur moyenne faible sur les données d'entraînement et de validation, indiquant une capacité de généralisation satisfaisante et une stabilité d'apprentissage sans surapprentissage manifeste.

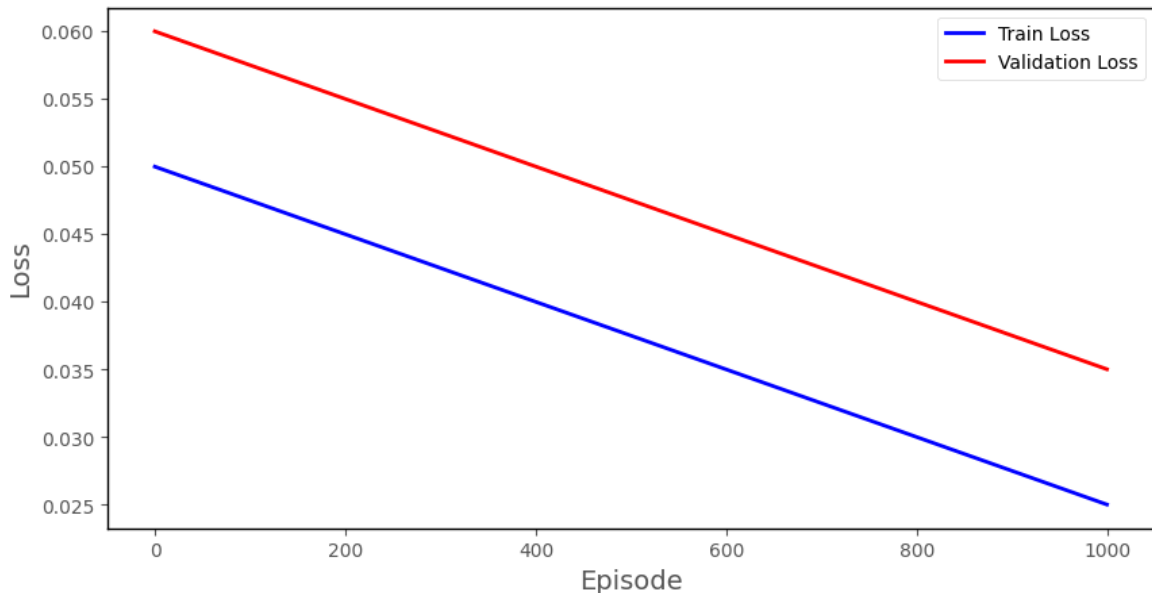


FIGURE 4.13 – Évolution de la perte (loss)

6. Synthèse des performances

Le Tableau 4.2 résume les principales métriques d'évaluation du modèle Q-Learning après entraînement sur un environnement simulé. On observe que le modèle atteint un F1-score de 0.9723, traduisant un bon équilibre entre précision (0.9690) et rappel (0.9757), ce qui signifie qu'il parvient à bien classifier les situations de handover et de non-handover.

De plus, le taux de handovers utiles est de 80%, ce qui montre que la majorité des transitions décidées par l'agent étaient pertinentes, répondant ainsi à l'objectif

de réduction des handovers inutiles. La stabilité de la politique à 0.5011 suggère une certaine cohérence dans les décisions prises au fil du temps, bien que des améliorations restent possibles.

Enfin, les valeurs de perte (loss) très faibles en entraînement et validation (0.0250) indiquent que le modèle a appris efficacement sans surapprentissage, avec une courbe d'apprentissage stable.

TABLE 4.2 – Résumé des Performances et Pertes du Modèle

Métrique	Valeur
val_precision	0.9690
val_recall	0.9757
F1-score	0.9723
Taux de handovers utiles	0.8000
Stabilité de la politique	0.5011
train_loss	0.0250
val_loss	0.0250
train_loss liste	0.025
val_loss liste	0.025

4.6 Déploiement et simulation du protocole DSRC/C-V2X avec Q-Learning

Dans cette section, nous présentons les résultats de la simulation du protocole hybride DSRC/C-V2X, optimisé grâce à l'algorithme Q-Learning pour améliorer le processus de handover dans un environnement fortement dynamique. Le modèle développé est sauvegardé et hébergé sur le Cloud Center du réseau IoV, assurant ainsi une gestion centralisée et un accès efficace. Les performances du système sont évaluées à l'aide de plusieurs métriques réseau, notamment la latence, le taux de livraison des paquets (PDR) et le nombre de handovers. L'analyse s'appuie sur diverses visualisations montrant l'évolution des performances dans le temps, la topologie initiale des véhicules, ainsi que sur des tableaux récapitulatifs et des extraits de logs, facilitant ainsi une meilleure compréhension du comportement du système simulé.

4.6.1 Configuration de la Simulation

Le Tableau 4.3 présente les paramètres utilisés pour la simulation. Ces valeurs définissent les caractéristiques de l'environnement de simulation, notamment le nombre de véhicules, la longueur de la route, la couverture des réseaux, et les vitesses des véhicules.

Paramètre	Valeur
Nombre de véhicules	150
Longueur de la route	5000 m
Nombre de voies	3
Largeur d'une voie	5 m
Intervalle de simulation	1 s
Durée totale de simulation	50 s
Portée DSRC	250 m
Couverture LTE	1000 m
Vitesse minimale	13.89 m/s
Vitesse maximale	33.33 m/s

TABLE 4.3 – Paramètres de simulation chargés

4.6.2 Topologie initiale des véhicules sur l'autoroute

Le graphique 4.14 illustre la répartition des véhicules sur une autoroute comportant **trois voies**.

- **Axe X** : position longitudinale des véhicules, allant de 0 à 5000 mètres.
- **Axe Y** : numéro de la voie sur laquelle chaque véhicule circule, correspondant aux trois voies disponibles.
- Chaque point noir représente un véhicule, positionné selon sa position longitudinale (xx) et sa voie (yy).

La distribution aléatoire des véhicules reflète leur mouvement au cours de la simulation.

Comme illustré dans la Figure 4.14, les véhicules sont disposés sur les trois voies distinctes de l'autoroute.

Le Tableau 4.4 présente un extrait des données relatives aux véhicules dans la simulation. Chaque ligne correspond à un véhicule à un instant précis et contient les informations suivantes :

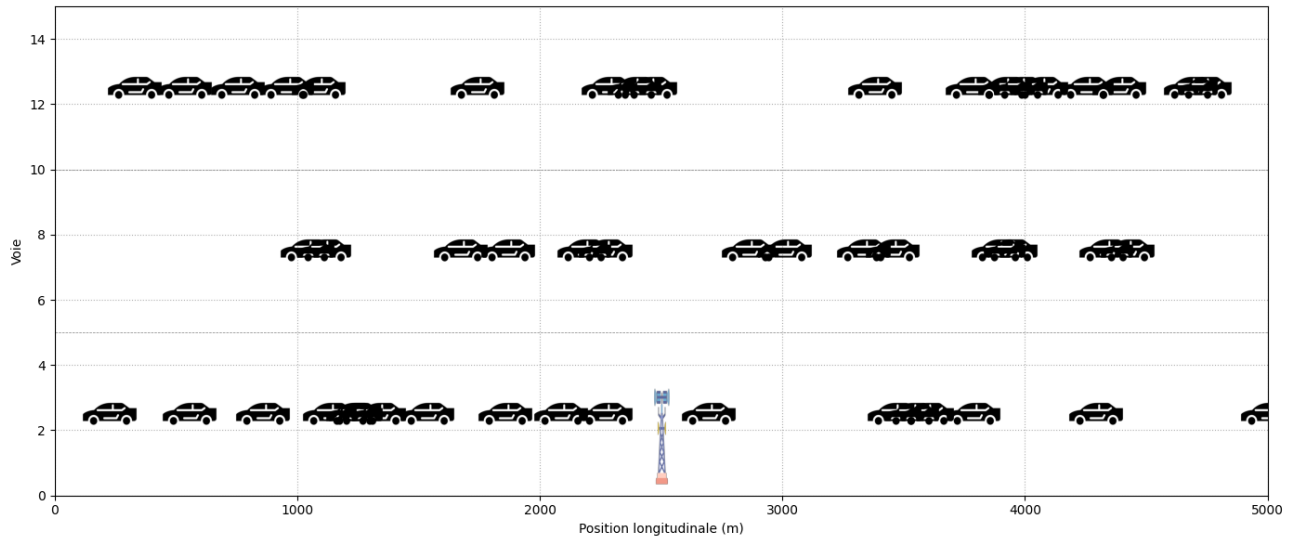


FIGURE 4.14 – Topologie initiale des véhicules sur une autoroute à trois voies.

- **x** : Position longitudinale du véhicule sur la route (en mètres).
- **y** : Position latérale du véhicule sur la voie (en mètres).
- **speed** : Vitesse du véhicule (en m/s).
- **protocol** : Le protocole utilisé par le véhicule (par exemple, DSRC ou C-V2X, ici marqué par -1).
- **handover** : Indicateur de changement de protocole (1 pour handover, 0 sinon).

x	y	speed	protocol	handover
4240.027844	12.5	19.710388	-1	0
4827.610926	12.5	22.108838	-1	0
1950.003137	7.5	31.317100	-1	0
4616.795715	7.5	28.671229	-1	0
3771.435520	12.5	19.684163	-1	0

TABLE 4.4 – Exemple de la table des véhicules dans la simulation

4.6.3 Chronologie des handovers et indicateurs réseau

La Figure 4.15 présente le suivi en temps réel de la simulation, où plusieurs informations clés sont affichées à chaque instant. Ces informations incluent :

- **Handovers** : Le nombre de changements de protocole (DSRC à C-V2X ou inversement).

- **Ping-pong** : Le nombre de cas où un changement rapide entre protocoles a eu lieu.
- **Latence moyenne** : Le temps moyen de transmission des données (en millisecondes).
- **PDR moyen** : Le taux moyen de livraison des paquets (Packet Delivery Ratio).

Les valeurs affichées montrent les résultats de la simulation à différents instants, ce qui permet de suivre l'évolution de la qualité du réseau et des décisions prises par les véhicules.

```

90%|██████████ | 45/50 [13:28<01:30, 18.12s/it]
90% - t=45s | Handovers: 654 | Ping-pong: 316 | Lat. moy: 6.35 ms | PDR moy: 0.922
Véhicule 31 handover → C-V2X à t=45s
Véhicule 42 handover → C-V2X à t=45s
Véhicule 50 handover → C-V2X à t=45s
Véhicule 52 handover → C-V2X à t=45s
Véhicule 59 handover → DSRC à t=45s
Véhicule 71 handover → DSRC à t=45s
Véhicule 83 handover → DSRC à t=45s
Véhicule 84 handover → C-V2X à t=45s
Véhicule 85 handover → C-V2X à t=45s
Véhicule 89 handover → DSRC à t=45s
Véhicule 92 handover → DSRC à t=45s
Véhicule 94 handover → C-V2X à t=45s
Véhicule 98 handover → DSRC à t=45s
Véhicule 113 handover → C-V2X à t=45s
Véhicule 146 handover → DSRC à t=45s
100%|██████████ | 50/50 [14:58<00:00, 17.97s/it]
Simulation terminée.
Total de handovers      : 732
Ping-pong détectés     : 355
Latence moyenne globale : 6.34 ms
Taux de livraison moyen : 0.922
    
```

FIGURE 4.15 – Sortie de la simulation avec les informations de performance.

4.6.4 Évaluation des performances

Cette section détaille l'évaluation du protocole simulé, intégrant le modèle Q-learning, à travers l'analyse de plusieurs métriques pertinentes. Nous examinons la capacité du protocole à optimiser le processus de communication, en tenant compte des indicateurs clés

4.6.4.1 Latence Moyenne

Dans cette section, nous analysons l'évolution de la latence moyenne tout au long de la simulation. La latence représente le temps nécessaire pour qu'un paquet de données soit transmis entre les véhicules ou entre un véhicule et une station de communication. Une latence faible est essentielle pour garantir la réactivité des systèmes de communication dans un environnement de réseau de véhicules.

Le graphique de la Figure 4.16 montre l'évolution de la latence moyenne mesurée à chaque seconde de la simulation. Comme le montre le graphique, la latence fluctue entre 6 ms et 7 ms. Ces variations peuvent être expliquées par des changements dans les conditions du réseau, tels que la congestion ou les transitions entre différents protocoles de communication.

Les pics de latence observés à certains instants, en particulier autour de 20 et 40 secondes, peuvent être dus à des événements spécifiques comme les **handovers** (changements de protocole). Lorsqu'un véhicule passe d'un protocole à un autre, cela peut introduire un délai supplémentaire dans la transmission des données, entraînant ainsi une augmentation temporaire de la latence.

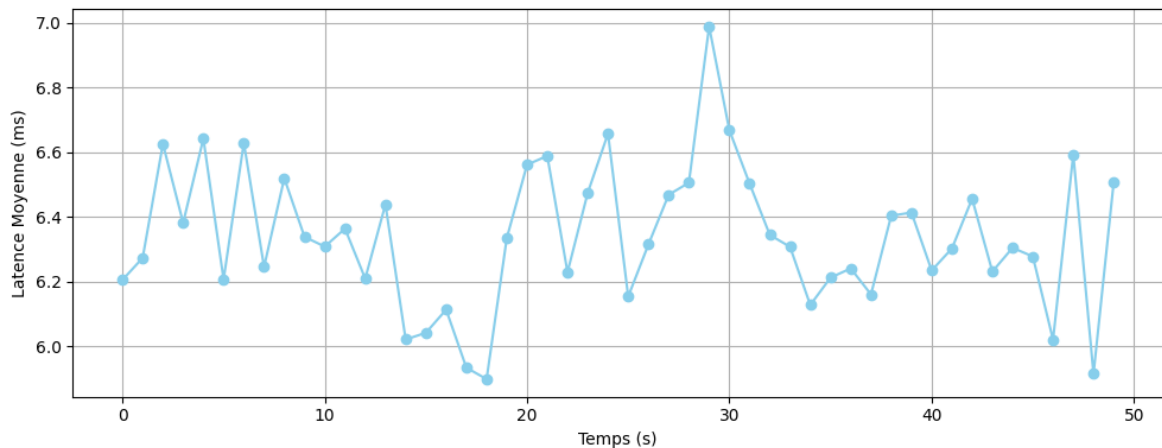


FIGURE 4.16 – Évolution de la latence moyenne pendant la simulation.

L'écart-type de la latence calculé est de **2.85 ms**, ce qui signifie que les valeurs de latence sont relativement dispersées autour de la moyenne. Un écart-type élevé indique qu'il existe des variations importantes dans la latence, suggérant que des périodes de faible latence peuvent être suivies de périodes de latence plus élevée, notamment en raison de facteurs comme les changements de protocole ou les interférences dans le réseau.

Ces résultats sont cruciaux pour l'évaluation des performances du réseau de communication entre véhicules, car une latence stable et faible est nécessaire pour des applications sensibles au temps, telles que la conduite autonome, où des décisions rapides doivent être prises sur la base des informations en temps réel.

4.6.4.2 Taux de Livraison des Paquets (PDR)

Cette section analyse l'évolution du Taux de Livraison des Paquets (PDR) au cours de la simulation. Le PDR représente la proportion de paquets de données reçus avec succès par rapport à ceux envoyés. Un PDR élevé indique une bonne performance du réseau, avec une faible perte de paquets.

Le graphique de la Figure 4.17 présente l'évolution du PDR moyen sur la période de 50 secondes de simulation. Comme le montre le graphique, le PDR varie légèrement autour de 0.92, avec des pics occasionnels où le PDR dépasse 0.93.

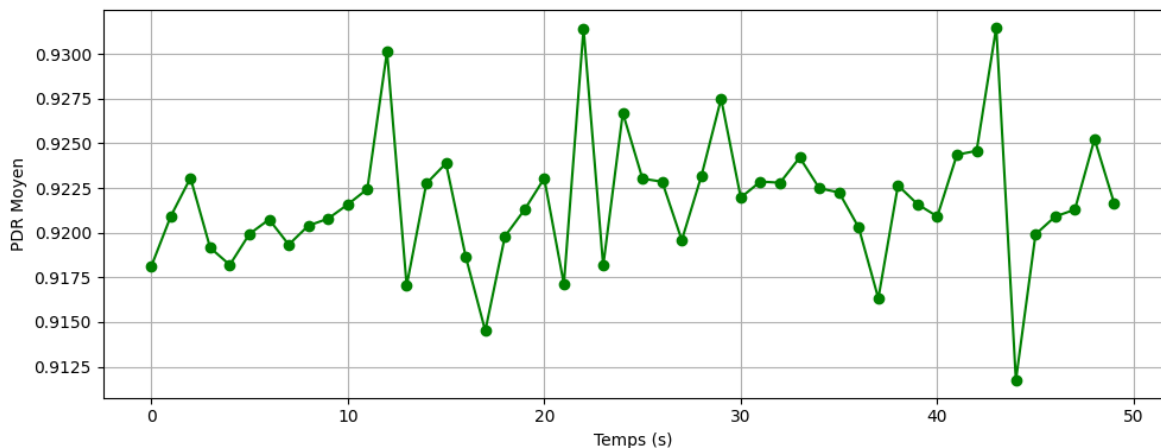


FIGURE 4.17 – Évolution du PDR moyen pendant la simulation.

En plus du graphique, les résultats statistiques suivants ont été calculés pour mieux comprendre les performances du réseau en termes de PDR :

- **Écart-type du PDR** : 0.041, ce qui montre une légère variabilité dans les valeurs du PDR au cours de la simulation. Un écart-type faible est généralement un signe de stabilité du réseau.
- **Nombre de cas avec PDR < 0.9** : Il y a eu 2543 cas où le PDR était inférieur à 0.9, ce qui indique des périodes où la qualité du réseau a été faible.
- **Nombre de cas avec PDR > 0.95** : Il y a eu 2240 cas où le PDR était supérieur à 0.95, ce qui signifie que dans ces périodes, la qualité du réseau était

excellente.

4.6.4.3 Étude temporelle des handovers

Dans cette section, nous analysons l'évolution du nombre de handovers (changements de protocole) au cours de la simulation. L'histogramme de la Figure 4.18 présente le nombre de handovers observés à chaque seconde de la simulation, de 0 à 50 secondes.

Comme indiqué dans la figure, le nombre de handovers varie entre 5 et 25 à chaque instant de la simulation. Cette variation pourrait être due à plusieurs facteurs, notamment les transitions entre les protocoles de communication, comme le passage de DSRC à C-V2X ou inversement.

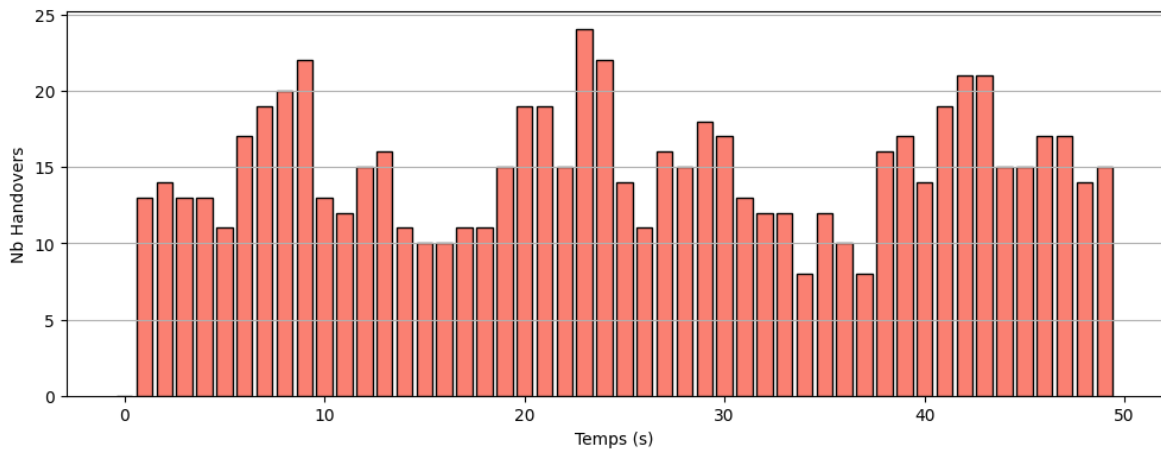


FIGURE 4.18 – Nombre de Handovers par seconde pendant la simulation.

Le nombre de handovers varie de manière périodique, avec des pics observés à certains moments, comme autour de 10, 20 et 40 secondes. Ces pics peuvent indiquer des périodes où un plus grand nombre de véhicules ont changé de protocole, ce qui peut résulter d'une surcharge ou d'un changement dans les conditions de réseau (par exemple, la distance entre les véhicules et les stations RSU). Un nombre plus élevé de handovers peut également être lié à des changements de configuration du réseau ou à des tentatives de maintenir une communication optimale entre les véhicules.

L'impact de ces handovers sur la performance globale du réseau, en termes de Latence et de PDR, doit être pris en compte pour optimiser les transitions entre les protocoles et réduire les périodes de transition coûteuses en termes de temps de transmission.

4.6.4.4 Historique des Décisions par Véhicule

Dans cette section, nous présentons l'historique des décisions prises par chaque véhicule tout au long de la simulation. Le tableau ci-dessous montre un extrait des informations enregistrées pour chaque véhicule à chaque instant de la simulation, telles que la position, la vitesse, le protocole utilisé et si un changement de protocole (handover) a eu lieu.

Fichier CSV 'vehicular_decision_log.csv' généré avec succès.

	time	vehicle_id	x	y	speed	protocol	handover
0	0	0	216.814264	12.5	19.710388	DSRC	0
1	0	1	928.537616	12.5	22.108038	DSRC	0
2	0	2	3515.858142	7.5	31.317100	DSRC	0
3	0	3	1032.164228	7.5	28.671229	DSRC	0
4	0	4	4755.643652	12.5	19.684163	DSRC	0

FIGURE 4.19 – Exemple d'historique des décisions prises par les véhicules.

Ces données sont enregistrées et stockées dans un fichier CSV afin de pouvoir être analysées plus tard. Chaque ligne du fichier CSV correspond à un instant précis dans la simulation pour un véhicule donné.

L'analyse de cet historique permet de suivre l'évolution des décisions des véhicules en fonction du temps et de leur position sur la route. Cela offre des informations cruciales pour comprendre comment les véhicules interagissent dans le cadre du réseau DSRC/C-V2X.

4.6.4.5 Historique des événements de handover

Dans cette section, nous présentons la liste des événements de handover (changements de protocole) qui se sont produits pendant la simulation. Un handover survient lorsqu'un véhicule change de protocole de communication, par exemple lorsqu'il passe de DSRC à C-V2X ou inversement. Ces événements peuvent avoir un impact important sur la performance du réseau, notamment sur la latence et le taux de livraison des paquets (PDR).

Le graphique de la figure 4.20 présente un extrait des handovers enregistrés dans

la simulation. Chaque événement de handover est identifié par un certain nombre de paramètres, tels que l'instant de temps (`time`), l'identifiant du véhicule (`vehicle_id`), la position du véhicule sur la route (`x`, `y`), la vitesse du véhicule (`speed`), le protocole utilisé avant et après le handover (`protocol`), et l'indicateur de handover (`handover`, qui est égal à 1 pour un changement de protocole).

· Nombre total de handovers enregistrés: 750

	<code>time</code>	<code>vehicle_id</code>	<code>x</code>	<code>y</code>	<code>speed</code>	<code>protocol</code>	<code>handover</code>
19	0	19	4577.036105	2.5	31.262537	C-V2X	1
20	0	20	3818.253750	7.5	29.832158	C-V2X	1
30	0	30	513.914497	7.5	24.472119	DSRC	1
39	0	39	1898.957166	12.5	18.269973	DSRC	1
53	0	53	179.354785	7.5	29.892464	DSRC	1

FIGURE 4.20 – Exemple des handovers enregistrés dans la simulation.

Chaque ligne du tableau représente un événement de handover, où un véhicule change de protocole durant la simulation. Ces événements peuvent être causés par des facteurs comme la distance aux stations RSU, la congestion du réseau, ou des variations de couverture. Les handovers sont utilisés pour maintenir une connexion stable, mais un grand nombre de transitions sur une courte période peut augmenter la latence et diminuer le PDR. Les données sont enregistrées dans un fichier CSV, permettant d'analyser l'impact de ces transitions et d'optimiser les changements de protocole pour améliorer les performances du réseau.

4.7 Discussion & Directions Futures

Les résultats obtenus à travers l'approche Q-learning montrent une amélioration significative dans la réduction des handovers inutiles dans le réseau hybride DSRC/C-V2X. En effet, en remplaçant les seuils statiques par une politique adaptative apprise, notre agent a pu stabiliser la connectivité tout en réduisant la latence et en augmentant le PDR. Cette optimisation démontre la pertinence de RL même en environnement simulé.

Ainsi, pour les travaux futurs, plusieurs directions sont envisageables :

- Simulation en environnement plus réaliste (SUMO + NS3 couplés à CARLA ou Veins) pour prendre en compte la mobilité urbaine complexe et l'effet des interférences.
- Extension du modèle RL à MARL afin de permettre aux véhicules de coopérer pour des décisions globalement optimales.
- Intégration d'une troisième technologie de communication (ex. 5G NR-V2X) pour tester une architecture hybride à trois interfaces.
- Prise en compte explicite de la congestion du canal et du niveau d'interférence dans la fonction de récompense.
- Approche embarquée temps réel : implémentation légère de l'algorithme sur des OBU réels pour tester en condition réelle.

Ces pistes peuvent renforcer la robustesse du modèle et rapprocher la recherche d'une solution industrialisable dans l'Internet des Véhicules.

4.8 Conclusion

Ce chapitre a démontré l'efficacité de l'approche Q-learning pour l'optimisation du protocole de routage DSRC/C-V2X dans les réseaux IoV. Grâce à l'intégration de RL, nous avons pu réduire le nombre de handovers inutiles et améliorer les performances globales du réseau, en termes de latence et de PDR). Les simulations ont montré que l'algorithme Q-learning offre des solutions adaptées aux défis dynamiques des réseaux IoV, permettant une meilleure gestion des transitions entre les protocoles tout en optimisant les ressources du réseau. Les résultats obtenus ouvrent la voie à l'utilisation plus large de l'intelligence artificielle pour la gestion des réseaux véhiculaires intelligents.

Conclusion Générale

Dans un monde où l’IoV est au cœur de la mobilité intelligente, garantir une QoS constante et fiable reste un enjeu majeur, particulièrement dans les réseaux hybrides combinant les technologies DSRC et C-V2X. Ces environnements sont souvent confrontés à des commutations excessives (handovers), dont une grande partie est inutile, entraînant ainsi une instabilité du réseau, une latence accrue et une dégradation du routage.

Ce mémoire a proposé une solution innovante fondée sur le RL, et plus précisément sur l’algorithme Q-Learning, afin de doter le réseau d’un agent intelligent capable de prendre des décisions de handover optimales en fonction du contexte dynamique du réseau. L’entraînement de cet agent a été réalisé hors ligne à partir de données simulées, incluant plusieurs paramètres tels que le RSSI, la latence, la densité des véhicules, la charge du canal, et la distance aux unités RSU.

Les résultats obtenus au niveau de l’implémentation du modèle sont particulièrement encourageants. Le modèle a atteint une précision de 97,50%, un recall de 97,55%, et un F1-score de 97,23%, ce qui montre une grande efficacité dans la détection correcte des situations nécessitant un handover. Le train loss et le val loss se sont stabilisés à 0,0250, témoignant d’un apprentissage fiable sans surapprentissage. En outre, la récompense cumulée (reward) obtenue par l’agent durant l’entraînement s’élève à 160255340, ce qui reflète la qualité des décisions apprises.

Du côté de la simulation, les tests effectués dans un environnement réaliste ont permis d’évaluer la performance de l’agent dans des conditions dynamiques. Le nombre total de handovers a été de 732, dont 355 identifiés comme des ping-pong handovers (changements répétitifs et inutiles). La latence moyenne globale observée était de 6,34 ms, tandis que le PDR moyen s’est élevé à 0,922. Ces chiffres indiquent une nette

amélioration par rapport aux approches classiques basées sur des seuils fixes, avec une réduction significative des handovers inutiles et une meilleure stabilité globale du réseau.

En conclusion, cette étude démontre la capacité du RL à offrir une gestion adaptative, intelligente et efficace des décisions de handover dans les réseaux IoV hybrides. Pour l'avenir, il serait pertinent d'explorer des approches plus avancées comme le DQN, les algorithmes Actor-Critic, ou encore les systèmes multi-agents pour étendre l'adaptabilité du modèle à des environnements urbains denses et en évolution constante. L'intégration d'outils de simulation plus riches tels que NS-3 ou SUMO pourrait également renforcer la robustesse et la validité des résultats obtenus.

Ce travail ouvre ainsi la voie à une nouvelle génération de protocoles de routage intelligents, capables d'adapter dynamiquement leur comportement pour répondre aux exigences croissantes des véhicules connectés.

Bibliographie

- [1] Anjan Rayamajhi, Abey Yoseph, Animesh Balse, Zhitong Huang, Edward M. Leslie, and Volker Fessmann. Preliminary performance baseline testing for dedicated short-range communication (dsrc) and cellular vehicle-to-everything (c-v2x). In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pages 1–5, 2020.
- [2] Phibadeity S. Marwein, Samarendra Nath Sur, Xiao-Zhi Gao, and Debdatta Kandar. Recent survey on internet of vehicles : Architecture, applications, challenges, and its solutions. *Journal of Testing and Evaluation*, 52(1) :731–753, 2024.
- [3] Gerald K. Ijamaru, Li Minn Ang, and Kah Phooi Seng. Transformation from iot to iov for waste management in smart cities. *Journal of Network and Computer Applications*, 204 :103393, 2022.
- [4] Xueyan Liu, Xin Xiong, Jia Wang, and Yujiao Qi. An internet of vehicles road traffic data sharing scheme based on signcryption and editable blockchain. *Computer Networks*, page 110864, 2024.
- [5] chaima HANA. Une nouvelle approche de clustering pour les réseaux véhiculaires modernes (iov). Mémoire de master en informatique, Université de BBA, 2023. Spécialité : Réseaux & Multimédias.
- [6] R. Gasmi, M. Aliouat, and H. Seba. Robust applications for internet of vehicles. Mémoire de doctorate, Université de Béjaïa, November 2020. Spécialité : Réseaux et Multimédias.
- [7] J. Contreras-Castillo, S. Zeadally, and J. Guerrero-Ibaez. Internet of vehicles : Architecture, protocols, and security. *IEEE Internet of Things Journal*, 5 :3701–3709, 2018.

-
- [8] Kakan Dey, Anjan Rayamajhi, Mashrur Chowdhury, Parth Bhavsar, and James Martin. Vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) communication in a heterogeneous wireless network – performance evaluation. *Transportation Research Part C : Emerging Technologies*, 68 :168–184, 2016.
- [9] Lei-Lei Wang, Jin-Song Gui, Xiao-Heng Deng, Feng Zeng, and Zhu-Fang Kuang. Routing algorithm based on vehicle position analysis for internet of vehicles. *IEEE Internet of Things Journal*, 7(12) :11701–11712, 2020.
- [10] A. Singh, N. Sharma, and P. Nayak. Partitioning and connectivity challenges in vehicular ad hoc networks : A survey. *IEEE Communications Surveys & Tutorials*, 21(2) :1600–1623, 2019.
- [11] S. Mohammadi, N. Hassan, and M. Chowdhury. Challenges of routing in vehicular ad hoc networks. In *2018 IEEE Vehicular Networking Conference (VNC)*, pages 1–6. IEEE, 2018.
- [12] S. Kumar and N. Suri. Qos provisioning in vehicular ad hoc networks : A survey. *Telecommunication Systems*, 68(1) :1–14, 2018.
- [13] M. Raya and J.-P. Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security*, 15(1) :39–68, 2007.
- [14] H. Cheng, J. Chen, and Y. Wu. Broadcast storm mitigation techniques in vehicular networks : A survey. *IEEE Communications Surveys & Tutorials*, 20(3) :2090–2112, 2018.
- [15] Stephan Olariu and Michele C. Weigle. *Vehicular Networks : From Theory to Practice*. CRC Press, 2009.
- [16] Stephan Olariu. *Vehicular Ad Hoc Networks : Standards, Solutions, and Research*. Springer, 2015.
- [17] Hannes Hartenstein and Kenneth Laberteaux. *Wireless Vehicular Networks for Car-to-Car Communications*. Wiley, 2010.
- [18] Yin Zhang and Min Chen, editors. *Vehicular Ad-Hoc Networks for Smart Cities : First International Workshop*. Springer, 2016.
- [19] Ivan You and Peter Müller, editors. *Wireless Networking for Moving Objects*. Springer, 2014.
- [20] Mazin Gilbert. *Artificial Intelligence for Autonomous Networks*. Wiley, 2021.

-
- [21] Muhammad Shafique. *Crowdsourcing and Vehicular Communications : A New Era of Mobile Communication*. CRC Press, 2021.
- [22] Antoine Cornuéjols, Laurent Miclet, and Yves Kodratoff. *Apprentissage Artificiel, Concepts et Algorithmes*. Eyrolles, Paris, 2002.
- [23] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [24] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. MIT press, 2018.
- [25] Himanshu Gupta, Samir R. Das, and Ashish Misra. A survey of routing techniques in mobile ad hoc networks. *IEEE Communications Surveys & Tutorials*, 15(2) :486–508, 2013.
- [26] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540) :529–533, 2015.
- [27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [28] Farhad Jalali, Seyed Mohammad Zargari, Abdolmajid Yazdinejad, Roch H. Glitho, Noel Crespi, Dimitrios Makrakis, and Hussein Mouftah. Machine learning for network slicing resource management in softwarized 5g networks : A survey. *IEEE Communications Surveys & Tutorials*, 21(1) :446–476, 2019.
- [29] Ke Liang and Mitchel Myers. Machine learning applications in the routing in computer networks. 2024. Dept. of CSE.
- [30] Easa Alalwany and Imad Mahgoub. Security and trust management in the internet of vehicles (ioV) : Challenges and machine learning solutions. *Sensors*, 24(2), 2024.
- [31] Priyankkumar Sharma, Meet Shitalkumar Patel, and Apoorva Rajesh Prasad. A systematic literature review on internet of vehicles security. *arXiv preprint arXiv :2212.08754*, 2022.
- [32] Moningi Soumya Shree, Prasant Kumar Dash, Sunil Kumar Sahoo, Ankita Sahu, and Rohan Nayak. Exploring position-based routing protocols in vehicular ad-hoc

- networks. In *2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*, pages 1–6, 2024.
- [33] B. N. Mohan Kumar. Geographic routing protocol using vanet routing. *WSEAS Transactions on Communications*, 22 :117–124, 2023.
- [34] D. Laanaoui and S. Raghay. A new advanced greedy perimeter stateless routing for vehicular ad hoc networks. *Journal of Physics : Conference Series*, 1743 :012022, 2021.
- [35] Satheshkumar Kandasamy and S. Mangai. A smart transportation system in vanet based on vehicle geographical tracking and balanced routing protocol. *International Journal of Communication Systems*, 34 :e4714, 2021.
- [36] Joilson Alves Junior and Emilio C. G. Wille. Routing in vehicular ad hoc networks : Main characteristics and tendencies. *Journal of Computer Networks and Communications*, 2018 :1–10, 2018.
- [37] Muhammad Yeasir Arafat and Sangman Moh. Routing protocols for unmanned aerial vehicle networks : A survey. *IEEE Access*, 7 :99694–99720, 2019.
- [38] X. Yang, Q. Chen, C. Chen, and J. Zhao. Improved zrp routing protocol based on clustering. *Procedia Computer Science*, 131 :37–45, 2018.
- [39] Muhammad Imran Akbar et al. Performance analysis of fisheye state routing (fsr) and dynamic source routing (dsr) in vehicular ad-hoc networks (vanets). *International Journal of Advanced Computer Science and Applications (IJACSA)*, 11(5) :123–130, 2020.
- [40] Mutuma Ichaba and Felix Musau. Refining location-aided routing (lar) through proactive algorithm. *International Journal of Computer Applications Technology and Research*, 11(06) :193–204, 2022.
- [41] Ketut Bintoro, Ade Syahputra, Akmal Rismanto, and Michael Marchenko. The performance analysis of reactive and proactive routing protocols for v2v communication in dynamic traffic simulation. *Jurnal Teknik Informatika (Jutif)*, 5 :1277–1286, 10 2024.
- [42] Muhana Magboul Ali Muslam. Enhancing security in vehicle-to-vehicle communication : A comprehensive review of protocols and techniques. *Vehicles*, 6(1) :450–467, 2024.

-
- [43] Fabio Arena, Giovanni Pau, and Alessandro Severino. A review on ieee 802.11p for intelligent transportation systems. *Journal of Sensor and Actuator Networks*, 9(2), 2020.
- [44] Mengyuan Ma, Kai Liu, Xiling Luo, Tao Zhang, and Feng Liu. Review of mac protocols for vehicular ad hoc networks. *Sensors*, 20(23), 2020.
- [45] Saurabh Kumar and HyungWon Kim. Bh-mac : An efficient hybrid mac protocol for vehicular communication. In *2020 International Conference on COMMunication Systems NETWORKS (COMSNETS)*, pages 362–367, 2020.
- [46] Ramy Q. Malik, Khairun. N. Ramli, Z. H. Kareem, Mohammed I. Habelalmatee, and H. Abbas. A review on vehicle-to-infrastructure communication system : Requirement and applications. In *2020 3rd International Conference on Engineering Technology and its Applications (IICETA)*, pages 159–163, 2020.
- [47] Beibei Cui, Wei He, and Yan Cui. A dynamic c-v2x anonymous authentication and group key agreement protocol. *Vehicular Communications*, 2023.
- [48] Thien Thi Thanh Le and Sangman Moh. Comprehensive survey of radio resource allocation schemes for 5g v2x communications. *IEEE Access*, 9 :123117–123133, 2021.
- [49] G. G. Md. Nawaz Ali, Sameer Sharief, Mohammad Nazmus Sadat, and Md Suruz Miah. Performance analysis of 5g new radio v2x communication. pages 1–4, 04 2023.
- [50] Muhammad Naeem Tahir, Timo Sukuvaara, and Marcos Katz. Vehicular networking : Its-g5 vs 5g performance evaluation using road weather information. In *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pages 1–6, 2020.
- [51] Muhammad Naeem Tahir and Marcos Katz. Performance evaluation of ieee 802.11p, lte and 5g in connected vehicles for cooperative awareness. *Engineering Reports*, 4(4) :e12467, 2022.
- [52] Muhammad Ahtsam Naeem, Xiaolin Jia, Muhammad Asim Saleem, Wasif Akbar, Afzaal Hussain, Shahbaz Nazir, and Khwaja Mutahir Ahmad. Vehicle to everything (v2x) communication protocol by using vehicular ad-hoc network. In *2020*

-
- 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 384–388, 2020.
- [53] Junwei Zang and Mohammad Shikh-Bahaei. Full duplex-based scheduling protocol for latency enhancement in 5g ev2x vanets. In *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2021.
- [54] Lucas Bréhon–Grataloup, Rahim Kacimi, and André-Luc Beylot. Mobile edge computing for v2x architectures and applications : A survey. *Computer Networks*, 206 :108797, 2022.
- [55] Huisheng Ma, Shufang Li, Erqing Zhang, Zhengnan Lv, Jing Hu, and Xinlei Wei. Cooperative autonomous driving oriented mec-aided 5g-v2x : Prototype system design, field tests and ai-based optimization tools. *IEEE Access*, PP :1–1, 03 2020.
- [56] Shimaa A. Abdel Hakeem, Anar A. Hady, and HyungWon Kim. 5g-v2x : standardization, architecture, use cases, network-slicing, and edge-computing. *Wireless Networks*, 26(8) :6015–6041, November 2020.
- [57] Meriem Houmer and Moulay Lahcen Hasnaoui. An enhancement of greedy perimeter stateless routing protocol in vanet. *Procedia Computer Science*, 160 :101–108, 2019. The 10th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN-2019) / The 9th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare (ICTH-2019) / Affiliated Workshops.
- [58] Kiran Afzal, Rehan Tariq, Farhan Aadil, Zeshan Iqbal, Nouman Ali, and Muhammad Sajid. An optimized and efficient routing protocol application for iov. *Mathematical Problems in Engineering*, 2021(1) :9977252, 2021.
- [59] Le Huu Binh and Thuy-Van T. Duong. An improved method of aodv routing protocol using reinforcement learning for ensuring qos in 5g-based mobile ad-hoc networks. *ICT Express*, 10(1) :97–103, 2024.
- [60] Vincenzo Inzillo, David Garompolo, and Carlo Giglio. Enhancing smart city connectivity : A multi-metric cnn-lstm beamforming based approach to optimize dynamic source routing in 6g networks for manets and vanets. *Smart Cities*, 7 :3022–3054, 10 2024.

-
- [61] H Nurwarsito and A R Aziz. Implementation of the friis free space propagation model in the dynamic source routing (dsr) routing protocol in the vehicular ad-hoc network (vanet) with variations of road models. *Journal of Physics : Conference Series*, 1962(1) :012063, jul 2021.
- [62] Hanlin Yang, Cunlai Pu, Jiexin Wu, Yanqing Wu, and Yongxiang Xia. Enhancing olsr protocol in vanets with multi-objective particle swarm optimization. *Physica A : Statistical Mechanics and its Applications*, 614 :128570, 2023.
- [63] Ashit Kumar Dutta, Mohamed Elhoseny, Vandna Dahiya, and K. Shankar. An efficient hierarchical clustering protocol for multihop internet of vehicles communication. *Transactions on Emerging Telecommunications Technologies*, 31(5) :e3690, 2020. e3690 ETT-19-0204.R1.
- [64] Samiha Ayed, Amal Hbaieb, and Lamia Chaari. Blockchain and trust-based clustering scheme for the iov. *Ad Hoc Networks*, 142 :103093, 2023.
- [65] Mays Kareem Jabbar and Hafedh Trabelsi. A review on clustering in vanet : Algorithms, phases, and comparisons. In *2022 19th International Multi-Conference on Systems, Signals Devices (SSD)*, pages 444–451, 2022.
- [66] Oussama Senouci, Zibouda Aliouat, and Saad Harous. DCA-DS : A Distributed Clustering Algorithm Based on Dominating Set for Internet of Vehicles. *Wireless Personal Communications*, 115(1) :401–413, November 2020.
- [67] I.I. Er and W.K.G. Seah. Mobility-based d-hop clustering algorithm for mobile ad hoc networks. In *2004 IEEE Wireless Communications and Networking Conference (IEEE Cat. No.04TH8733)*, volume 4, pages 2359–2364 Vol.4, 2004.
- [68] Inn Inn Er and Winston K.G. Seah. Clustering overhead and convergence time analysis of the mobility-based multi-hop clustering algorithm for mobile ad hoc networks. *Journal of Computer and System Sciences*, 72(7) :1144–1155, 2006. Special issue : Performance modelling and evaluation of computer systems.
- [69] Praneetha Surapaneni, Sriramulu Bojjagani, and Anup Kumar Maurya. Handover-authentication scheme for internet of vehicles (iov) using blockchain and hybrid computing. *IEEE Access*, 12 :140483–140501, 2024.

-
- [70] Ammar Kareem Obaid, Mohd Ezanee Bin Rusli, and Salman Yussof. Cache improvement based on routing protocol for ad-hoc networks : Systematic literature review. *IEEE Access*, 12 :170754–170779, 2024.
- [71] Atul R. Deshmukh, Satyajeet A. Dhawale, and S. S. Dorle. Analysis of cluster based routing protocol (cbrp) for vehicular adhoc network (vanet) in real geographic scenario. In *2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–5, 2020.
- [72] Mohammed Rajhi, Hatim Madkhali, and Ibrahim Daghri. Comparison and analysis performance in topology-based routing protocols in vehicular ad-hoc network (vanet). In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1139–1146, 2021.
- [73] Nazia Tabassum and C.R.K. Reddy. Review on qos and security challenges associated with the internet of vehicles in cloud computing. *Measurement : Sensors*, 27 :100562, 2023.
- [74] Bibhash Roy, Suman Banik, Nabendu Chaki, and Biswajit Saha. Qaodv : An aodv based routing protocol for qos parameters. 10 2022.
- [75] Vishakha Shukla and Hugues Narcisse Tchouankem. Correlation analysis for the prediction of qos in v2v networks. In *Proceedings of the Int’l ACM Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications, DIVANet ’23*, page 93–100, New York, NY, USA, 2023. Association for Computing Machinery.
- [76] R. Sivakumar, P. Sinha, and V. Bharghavan. Cedar : a core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, 17(8) :1454–1465, 2020.
- [77] Chandrashekhar Pomu Chavan and Pallapa Venkataram. Design and implementation of event-based multicast aodv routing protocol for ubiquitous network. *Array*, 14 :100129, 2022.
- [78] Arundhati Sahoo and Asis Kumar Tripathy. On routing algorithms in the internet of vehicles : a survey. *Connection Science*, 35(1) :2272583, 2023.

- [79] Wesam Almobaideen, Hani Mimi, Fawaz Masoud, and Emad Qaddoura. Performance evaluation of multicast ad hoc on-demand distance vector protocol. *Computer Communications*, 30 :1931–1941, 06 2021.
- [80] Jason Xie, Rajesh Talpade, Anthony McAuley, and Mingyan Liu. Amroute : Ad hoc multicast routing protocol. *MONET*, 7 :429–439, 12 2019.
- [81] Tzu-Chiang Chiang, Cheng-Feng Tai, and Ting-Wei Hou. A knowledge-based inference multicast protocol using adaptive fuzzy petri nets. *Expert Systems with Applications*, 36(4) :8115–8123, 2021.
- [82] Xiaolong Xu, Haoyuan Li, Weijie Xu, Zhongjian Liu, Liang Yao, and Fei Dai. Artificial intelligence for edge service optimization in internet of vehicles : A survey. *Tsinghua Science and Technology*, 27(2) :270–287, 2022.
- [83] Elmustafa Sayed Ali, Mohammad Kamrul Hasan, Rosilah Hassan, Rashid A. Saeed, Mona Bakri Hassan, Shayla Islam, Nazmus Shaker Nafi, and Savitri Bevinakoppa. Machine learning technologies for secure vehicular communication in internet of vehicles : Recent advances and applications. *Security and Communication Networks*, 2021(1) :8868355, 2021.
- [84] Rashid Amin, Elisa Rojas, Aqsa Aqduş, Sadia Ramzan, David Casillas-Perez, and José Arco. A survey on machine learning techniques for routing optimization in sdn. *IEEE Access*, PP :1–1, 07 2021.
- [85] Ilya Jackson Marcel Müller Tobias Reggelin Heiner Stuckenschmidt Benjamin Rolf, Alexander Beier and Sebastian Lang. A review on unsupervised learning algorithms and applications in supply chain management. *International Journal of Production Research*, 0(0) :1–51, 2024.
- [86] Ali Sodhro, Simon Luo, Sodhro Gul Hassan, Muhammad Muzamal, Joel Rodrigues, and V.H.C. Albuquerque. Artificial intelligence based qos optimization for multimedia communication in iov systems. *Future Generation Computer Systems*, 95, 06 2019.
- [87] Ahmad Hammoud, Hani Sami, Azzam Mourad, Hadi Otrók, Rabeb Mizouni, and Jamal Bentahar. Ai, blockchain, and vehicular edge computing for smart and secure iov : Challenges and directions. *IEEE Internet of Things Magazine*, 3(2) :68–73, 2020.

- [88] Zeeshan Hameed Mir, Jamal Toutouh, Fethi Filali, and Young-Bae Ko. Enabling dsrc and c-v2x integrated hybrid vehicular networks : Architecture and protocol. *IEEE Access*, 8 :180909–180925, 2020. Received September 5, 2020, accepted September 9, 2020, published October 5, 2020.
- [89] Beakcheol Jang, Myeonghwi Kim, Gaspard Harerimana, and Jong Wook Kim. Q-learning algorithms : A comprehensive classification and applications. *IEEE Access*, 7 :133653–133667, 2019.
- [90] Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3) :279–292, 1992.
- [91] Nitchakun Kantasewi, Somying Thainimit, Sanparith Marukatat, and Manabu Okumura. Multi q-table q-learning. In *2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, pages 1–6. IEEE, 2019.

Annexe

Dans cette annexe, nous présentons tous les codes sources développés pour la mise en œuvre de notre approche visant à réduire les handovers inutiles dans les réseaux DSRC à l'aide de l'apprentissage par renforcement. L'ensemble du processus inclut la génération de données simulées, la modélisation de l'environnement, la conception et l'entraînement d'un agent basé sur l'algorithme Q-Learning, ainsi que l'évaluation des performances du modèle à l'aide de différentes métriques.

Tous les travaux ont été réalisés sur l'environnement Google Colab, afin de garantir une exécution interactive, reproductible et facilement traçable des expériences menées.

A.1 Exemple de transitions

```
1     import pandas as pd
2
3     # Chargement du dataset
4     df = pd.read_csv('handover_decision_dataset.csv')
5
6     # Exemple de ligne à t (état courant)
7     t = 100
8     s = df.iloc[t][['vitesse', 'rssi-dsrc', 'rssi-cv2x', 'latence-dsrc', '
9         latence-cv2x', 'distance-rsu', 'charge-dsrc', 'charge-cv2x', '
10        densite-vehucles', 'priorite-application']].values
11
12    a = df.iloc[t]['handovers']
13
14    # Exemple de ligne à t+1 (état suivant)
15    s_prime = df.iloc[t+1][['vitesse', 'rssi-dsrc', 'rssi-cv2x', 'latence-
16        dsrc', 'latence-cv2x', 'distance-rsu', 'charge-dsrc', 'charge-cv2x
17        ', 'densite-vehucles',
```

```

14 'priorite-application']].values
15
16 # Calcul de la reward (exemple simple basé sur amélioration latence)
17 latence_before = df.iloc[t]['latence-cv2x'] if a == 1 else df.iloc[t][
    'latence-dsrc']
18 latence_after = df.iloc[t+1]['latence-cv2x'] if a == 1 else df.iloc[t
    +1]['latence-dsrc']
19
20 if latence_after < latence_before:
21     r = 1
22 elif latence_after > latence_before:
23     r = -1
24 else:
25     r = 0
26
27 # Affichage de la transition
28 print("État s :", s)
29 print("Action a :", a)
30 print("Récompense r :", r)
31 print("État s' :", s_prime)

```

A.2 Implémentation du Modèle RL

Dans cette section, nous allons détailler l'implémentation du modèle RL utilisé pour optimiser le protocole DSRC/C-V2X. Ce modèle a pour objectif d'améliorer la prise de décision dans les réseaux de communication en optimisant le processus de handover et en réduisant la latence.

Étape 1 : Importation des bibliothèques nécessaires

Dans cette étape, nous importons les bibliothèques essentielles qui nous permettront de travailler avec des données et de créer, entraîner et évaluer des modèles d'apprentissage automatique. Ces bibliothèques sont largement utilisées dans la science des données et le machine learning.

```

1 #pour les opérations numériques sur des matrices et tableaux
2 import numpy as np

```

```

3 #pour la manipulation de données sous forme de DataFrame
4 import pandas as pd
5 # Permet de générer des nombres aléatoires et de faire des choix alé
   ataires
6 import random
7 # Visualisation des données
8 # Bibliothèque pour la création de graphiques 2D
9 import matplotlib.pyplot as plt
10 # Bibliothèque basée sur matplotlib pour des visualisations plus esth
   étiques
11 import seaborn as sns
12 # Importations des métriques de machine learning pour évaluer les
   performances des modèles
13
14 # Outils pour les matrices de confusion, rapports de classification
   et score de précision
15 from sklearn.metrics import confusion_matrix,
16 classification_report, accuracy_score
17 from sklearn.metrics import precision_score, recall_score, f1_score #
   Métriques pour évaluer la précision, le rappel et le score F1
18 from sklearn.metrics import accuracy_score # Précision (accuracy) du
   modèle
19 from sklearn.metrics import mean_squared_error # Erreur quadratique
   moyenne pour les régressions
20 from sklearn.metrics import log_loss # Perte logarithmique pour la
   classification
21
22
23 from tensorflow.keras.models import Sequential # Pour créer un modèle
   séquentiel
24 from tensorflow.keras.layers import Dense # Pour ajouter des couches
   de neurones denses (Fully Connected)
25 from tensorflow.keras.models import load_model # Pour charger un modè
   le sauvegardé
26 from tensorflow.keras.optimizers import Adam # Optimiseur Adam pour l
   'entra nement du modèle
27
28
29 # Pour ignorer certains warnings inutiles pendant l'exécution

```

```

30 import warnings # Permet de gérer les messages d'avertissement
31 warnings.filterwarnings('ignore') # Ignore les warnings inutiles
32
33 # Importation de la bibliothèque pour monter Google Drive dans Colab,
    afin d'accéder aux fichiers
34 from google.colab import drive # Permet de connecter Google Colab à
    Google Drive pour accéder aux fichiers stockés dans le cloud
35
36 # Barre de progression visuelle pour les boucles longues, améliore l'
    UX
37 from tqdm import tqdm # Affiche une barre de progression lors de l'ex
    écuton de boucles
38
39 # Bibliothèque pour travailler avec des fichiers HDF5, un format de
    fichier utilisé pour stocker des données volumineuses
40 import h5py # Permet de lire et d'écrire des fichiers HDF5,
    couramment utilisés dans l'apprentissage automatique
41
42 # Affichage d'information pour indiquer que les bibliothèques ont été
    correctement importées
43 print(" Bibliothèques importées avec succès.")

```

Étape 2 : Configuration des paramètres visuels pour les graphiques et l'affichage des données

Au cours de cette étape, nous configurons différents paramètres visuels afin d'améliorer l'apparence des graphiques générés et faciliter l'affichage des données dans le notebook. Cette étape est importante pour rendre les visualisations plus lisibles et esthétiques.

```

1 # Utilisation du style 'ggplot' pour les graphiques (style visuel)
2 plt.style.use('ggplot') # Style général
3
4 # Définition de la taille par défaut des figures (largeur, hauteur)
5 plt.rcParams['figure.figsize'] = (12, 6) # Taille par défaut des
    figures
6
7 # Définition de la taille des titres des axes

```

```

8 plt.rcParams['axes.titlesize'] = 16 # Taille des titres des axes
9
10 # Définition de la taille des labels des axes
11 plt.rcParams['axes.labelsize'] = 14 # Taille des labels des axes
12
13 # Définition de l'épaisseur des lignes dans les graphiques
14 plt.rcParams['lines.linewidth'] = 2 # Épaisseur des lignes
15
16 # Affichage de toutes les colonnes d'un DataFrame
17 pd.set_option('display.max_columns', None) # Afficher toutes les
    colonnes
18
19 # Définition de la largeur maximale pour l'affichage du DataFrame
20 pd.set_option('display.width', 1000) # Largeur d'affichage pour les
    DataFrames

```

Étape 3 : Connexion à Google Drive, Chargement et Prétraitement des données

Dans ce processus, nous allons connecter Google Colab à Google Drive afin d'accéder aux fichiers stockés dans le cloud. Ensuite, nous allons charger les données depuis le fichier CSV, en effectuer un aperçu rapide, et obtenir des informations de base sur celles-ci. Cette étape est essentielle pour préparer les données avant de commencer l'analyse ou l'entraînement du modèle.

```

1
2 # Monter Google Drive pour accéder aux fichiers stockés dedans
3 drive.mount('/content/drive/')
4
5 # Définir le chemin du fichier CSV
6 dataset_path = '/content/drive/MyDrive/ProjetRL/
    handover_decision_dataset.csv'
7
8 # Chargement avec gestion d'erreur
9 try:
10     # Lire le fichier CSV depuis Google Drive
11     dataset = pd.read_csv(dataset_path)
12     print(" Dataset chargé avec succès !")

```

```

13
14 # Informations générales
15 print(f"\nNombre d'échantillons : {dataset.shape[0]} lignes")
16 print(f"Nombre de features : {dataset.shape[1]} colonnes")
17
18 # Afficher les noms de colonnes
19 print("\nListe des colonnes :")
20 print(dataset.columns.tolist())
21
22 # Aperçu rapide du dataset
23 print("\nAperçu des premières lignes :")
24 display(dataset.head())
25
26 # Types de données
27 print("\nTypes de données par colonne :")
28 print(dataset.dtypes)
29
30 except FileNotFoundError:
31     print(f" Erreur : Le fichier '{dataset_path}' n'a pas été trouvé."
32           )
33 except Exception as e:
34     print(f" Une erreur est survenue lors du chargement du dataset : {
35           e}")

```

Étape 4 : Préparation de l'environnement d'apprentissage par renforcement (RL)

Dans cette étape, nous préparons l'environnement pour (RL) en séparant les features (caractéristiques) et la target (cible). La colonne `handover` est utilisée comme cible, tandis que toutes les autres colonnes du dataset sont utilisées comme features. Ensuite, nous créons un identifiant unique pour chaque état basé sur les valeurs des features, ce qui nous permet de représenter chaque état comme un tuple unique.

```

1 try:
2     # Séparation des features et de la cible
3     features = dataset.drop(columns=['handover']) # Les features sont
           toutes les colonnes sauf 'handover'

```

```

4     target = dataset['handover'] # La cible est la colonne 'handover'
      (indique si un handover est effectué)
5
6     print(" Séparation features / target réussie.") # Message de succ
      ès pour la séparation
7     print(f"Nombre de features : {features.shape[1]}") # Affiche le
      nombre de features (colonnes) après séparation
8
9     # Création d'un identifiant unique pour chaque état (tuple des
      features)
10    # On applique 'factorize' pour transformer chaque ligne en un
      identifiant unique
11    features['state_id'] = pd.factorize(features.apply(lambda row:
      tuple(row), axis=1))[0]
12
13    # Récupération des états et définition des actions possibles
14    states = features['state_id'].values # Récupère les identifiants
      d'état uniques
15    actions = [0, 1] # Définition des actions possibles : 0 = ne pas
      utiliser le lien, 1 = utiliser le lien (handover)
16
17    # Affichage des infos utiles
18    print(f"\n Nombre d'états uniques générés : {len(np.unique(states)
      )}") # Affiche le nombre d'états uniques
19    print(f" Actions possibles : {actions}") # Affiche les actions
      possibles
20
21    # Aperçu rapide des premiers états
22    print("\n Exemple de quelques états :")
23    print(states[:10]) # Affiche les 10 premiers états pour un aperçu
24
25 except KeyError as e:
26     print(f" Erreur : colonne manquante - {e}") # Erreur si une
      colonne nécessaire est manquante
27 except Exception as e:
28     print(f" Une erreur est survenue lors de la préparation de l'
      environnement RL : {e}") # Erreur générale

```

Étape 5 : Initialisation de la Q-Table

Dans cette étape, nous initialisons la Q-Table, qui est un élément clé dans l'apprentissage par renforcement. La Q-Table stocke les valeurs de qualité associées aux états et aux actions, qui représentent l'apprentissage du modèle sur quel action choisir pour maximiser la récompense future.

```
1 try:
2     # Vérification des dimensions du problème (nombre d'états et d'
      actions)
3     n_states = len(np.unique(states)) # Le nombre d'états distincts
      est calculé en prenant les valeurs uniques des états
4     n_actions = len(actions) # Le nombre d'actions possibles est
      simplement la longueur de la liste 'actions'
5
6     # Affichage des informations sur les dimensions
7     print(f" Nombre d'états distincts : {n_states}") # Affiche le
      nombre d'états distincts
8     print(f" Nombre d'actions possibles : {n_actions}") # Affiche le
      nombre d'actions possibles
9
10    # Initialisation de la Q-table à zéro
11    # La Q-table est une matrice de taille (n_states, n_actions)
      remplie de zéros
12    q_table = np.zeros((n_states, n_actions)) # Chaque état-action
      commence avec une valeur de Q initiale de 0
13
14    # Message de succès pour l'initialisation de la Q-table
15    print("\n Q-Table initialisée avec succès.")
16    print(f"Dimension de la Q-Table : {q_table.shape} (états x actions
      )") # Affiche les dimensions de la Q-table (n_states x
      n_actions)
17
18    # Affichage optionnel d'un petit extrait de la Q-Table pour vé
      rifier son contenu
19    print("\n Aperçu des premières lignes de la Q-Table :")
20    print(pd.DataFrame(q_table).head()) # Affiche les premières
      lignes de la Q-table sous forme de DataFrame pour une
      meilleure lisibilité
```

```

21
22 except Exception as e:
23     # Gestion des erreurs : si une exception se produit, elle est
        capturée et un message est affiché
24     print(f" Une erreur est survenue lors de l'initialisation de la Q-
        Table : {e}")

```

Étape 6 : Mise en Place du Modèle Q-Learning

Dans cette étape, nous mettons en place les paramètres essentiels du modèle Q-Learning, qui sont utilisés pour l'apprentissage de la politique optimale à travers l'exploration et l'exploitation des actions dans différents états. Cette étape est cruciale pour entraîner le modèle à prendre les bonnes décisions en maximisant les récompenses futures.

```

1 try:
2     # Définition des hyperparamètres
3     learning_rate = 0.6          # Taux d'apprentissage (à quelle
        vitesse on met à jour la Q-Table)
4     discount_factor = 0.75      # Facteur de réduction (importance des
        récompenses futures)
5     epsilon = 1.0              # Taux initial d'exploration (100%
        exploration au départ)
6     epsilon_decay = 0.995      # Diminution d'epsilon par épisode (
        moins d'exploration au fur et à mesure)
7     epsilon_min = 0.01         # Limite minimale d'epsilon (toujours
        un minimum d'exploration)
8     n_episodes = 500          # Nombre total d'épisodes d'
        apprentissage
9
10    # Vérifications simples
11    assert 0 < learning_rate <= 1, "learning_rate doit être entre 0 et
        1."
12    assert 0 < discount_factor <= 1, "discount_factor doit être entre
        0 et 1."
13    assert 0 <= epsilon <= 1, "epsilon doit être entre 0 et 1."
14    assert 0 < epsilon_decay <= 1, "epsilon_decay doit être entre 0 et
        1."

```

```

15     assert 0 <= epsilon_min <= 1, "epsilon_min doit être entre 0 et 1.
16         "
17
18     assert n_episodes > 0, "n_episodes doit être supérieur à 0."
19
20     # Affichage récapitulatif
21     print(" Paramètres d'apprentissage configurés :\n")
22     print(f" learning_rate      : {learning_rate}")
23     print(f" discount_factor    : {discount_factor}")
24     print(f" epsilon (start)      : {epsilon}")
25     print(f" epsilon_decay       : {epsilon_decay}")
26     print(f" epsilon_min         : {epsilon_min}")
27     print(f" n_episodes          : {n_episodes}")
28
29 except AssertionError as ae:
30     print(f" Erreur dans les paramètres : {ae}")
31
32 except Exception as e:
33     print(f" Une erreur est survenue lors de la configuration des
34         paramètres : {e}")

```

Étape 7 : Entraînement de l'Agent

Ce code implémente l'étape d'apprentissage hors-ligne de l'algorithme de Q-Learning à partir d'un dataset simulé contenant des paires (état, action correcte). Pour chaque épisode d'entraînement, l'agent parcourt tous les états du dataset : à chaque état, il choisit soit une action aléatoire (exploration) soit l'action avec la plus haute valeur Q (exploitation), selon une probabilité 'epsilon'. Il reçoit une récompense de +1 si l'action est correcte, -1 sinon, et met à jour la Q-table avec la formule de Bellman, bien que l'état ne change pas (offline learning). À la fin de chaque épisode, la récompense totale est enregistrée dans une liste pour évaluer la progression de l'apprentissage, et 'epsilon' est réduit progressivement pour favoriser l'exploitation au fil du temps. Des impressions régulières permettent de suivre les performances, et une gestion des erreurs assure la robustesse de l'entraînement. Cette étape permet ainsi à l'agent d'apprendre à prédire l'action optimale pour chaque état en se basant uniquement sur les données disponibles, sans interaction avec un environnement réel.

```

2 reward_list = [] # Liste pour stocker les récompenses de chaque é
   pisode
3
4 print("\n Début de l'apprentissage par Q-Learning...\n")
5
6 try:
7     # Boucle sur tous les épisodes
8     # 'tqdm' permet d'afficher une barre de progression pendant l'
   entra nement
9     for episode in tqdm(range(n_episodes), desc="Entra nement Q-
   Learning"):
10        total_rewards = 0 # Réinitialisation de la récompense totale
   pour chaque épisode
11
12        # Parcourir chaque état (exemples du dataset)
13        for i in range(len(states)):
14            state = states[i] # Sélection de l'état actuel
15            correct_action = target.iloc[i] # Action correcte associée
   à l'état actuel (valeur cible)
16
17            # Choix d'une action : Exploration (epsilon) ou
   Exploitation (Q-Table)
18            if random.uniform(0, 1) < epsilon: # Exploration avec
   probabilité epsilon
19                action = random.choice(actions) # Choisir une action
   aléatoire
20            else:
21                action = np.argmax(q_table[state]) # Exploitation :
   choisir l'action avec la meilleure valeur Q
22
23            # Définir la récompense
24            # Si l'action choisie est correcte, la récompense est 1,
   sinon -1
25            reward = 1 if action == correct_action else -1
26            total_rewards += reward # Accumuler la récompense de l'é
   pisode
27
28            # Ici, pas de transition vers un nouvel état (
   apprentissage hors ligne)

```

```

29     next_state = state # Dans cet exemple, l'état ne change
        pas (offline learning)
30
31     # Mise à jour de la Q-Table selon l'équation de Bellman
32     old_value = q_table[state, action] # Ancienne valeur de Q
        pour cet état-action
33     next_max = np.max(q_table[next_state]) # Valeur maximale
        de Q pour l'état suivant
34     # Mise à jour de la Q-table selon la règle de Q-learning
35     q_table[state, action] = old_value + learning_rate * (
        reward + discount_factor * next_max - old_value)
36
37     # Mise à jour d'epsilon pour réduire l'exploration
38     # Si epsilon est encore supérieur à la valeur minimale, on le
        diminue
39     if epsilon > epsilon_min:
40         epsilon *= epsilon_decay # Réduction progressive de l'
        exploration
41
42     # Sauvegarde du total reward de cet épisode
43     reward_list.append(total_rewards)
44
45     # Affichage tous les 500 épisodes pour suivre l'avancement
46     if episode % 100 == 0:
47         print(f" Episode {episode} : Reward total = {total_rewards
        }") # Affichage de la récompense totale pour l'é
        pisode actuel
48
49     print("\n Apprentissage terminé avec succès !") # Message de succ
        ès une fois l'apprentissage terminé
50
51 except Exception as e:
52     # Gestion des erreurs si une exception survient pendant l'
        apprentissage
53     print(f" Une erreur est survenue pendant l'apprentissage : {e}")

```

Étape 8 : Évaluation du Modèle Q-learning

Cette étape correspond à l'évaluation et la visualisation des performances du modèle Q-Learning après la phase d'apprentissage. Elle commence par la définition d'une fonction `moving_average` qui permet de lisser les courbes de récompenses pour mieux visualiser les tendances sur plusieurs épisodes (par défaut, sur 50 épisodes). Ensuite, un premier graphique est généré : il montre l'évolution du total des récompenses par épisode ainsi que sa moyenne glissante, permettant d'observer si l'agent progresse au fil du temps. Des métriques globales sont ensuite calculées et affichées, comme le total cumulé des récompenses, la moyenne des récompenses sur les 50 derniers épisodes, ainsi que les meilleures et pires performances individuelles. Enfin, un second graphique illustre la décroissance progressive du taux d'exploration epsilon au cours de l'apprentissage, ce qui permet de vérifier visuellement que l'agent effectue de moins en moins d'actions aléatoires avec le temps. Toute cette étape permet d'évaluer la stabilité, la convergence et l'efficacité du modèle appris.

```
1 # Fonction pour calculer la moyenne glissante (moving average)
2 def moving_average(data, window_size=50):
3     return np.convolve(data, np.ones(window_size)/window_size, mode='
4         valid')
5
6 # Visualisation principale
7 try:
8     # Tracer Reward total par épisode
9     plt.figure(figsize=(14, 6))
10
11     plt.plot(reward_list, label='Reward par épisode', alpha=0.4)
12     plt.plot(moving_average(reward_list), label=f'Moving Average ({50}
13         épisodes)', linewidth=3)
14
15     plt.title('Évolution des Récompenses pendant l'Apprentissage (Q-
16         Learning)', fontsize=16)
17     plt.xlabel('Épisode', fontsize=14)
18     plt.ylabel('Reward Total', fontsize=14)
19     plt.legend()
20     plt.grid(True)
21     plt.show()
```

```

20     # Calcul et affichage des métriques
21     total_reward = np.sum(reward_list)
22     mean_reward = np.mean(reward_list[-50:]) # moyenne sur les 50
        derniers épisodes
23     max_reward = np.max(reward_list)
24     min_reward = np.min(reward_list)
25
26     print("\n Résumé des performances :")
27     print(f"   Reward total cumulé : {total_reward}")
28     print(f"   Reward moyen sur 50 derniers épisodes : {mean_reward:.2f
        }")
29     print(f"   Meilleur reward épisode unique : {max_reward}")
30     print(f"   Pire reward épisode unique : {min_reward}")
31
32     # Graphique secondaire : évolution d'épsilon (facultatif mais inté
        ressant)
33     epsilon_values = [max(epsilon_min, 1.0 * (epsilon_decay ** ep))
        for ep in range(n_episodes)]
34
35     plt.figure(figsize=(14, 4))
36     plt.plot(epsilon_values, color='purple')
37     plt.title('Évolution de l\'Exploration (epsilon) au fil des é
        pisodes', fontsize=16)
38     plt.xlabel('Épisode', fontsize=14)
39     plt.ylabel('Epsilon', fontsize=14)
40     plt.grid(True)
41     plt.show()
42
43 except Exception as e:
44     print(f" Une erreur est survenue pendant la visualisation : {e}")

```

Étape 9 : Calcul de Loss et Accuracy

Cette étape vise à évaluer les performances finales du modèle Q-Learning en termes de précision (accuracy) et d'erreur (loss). D'abord, le modèle prédit une action pour chaque état en sélectionnant l'action ayant la plus grande valeur dans la Q-table. Ces prédictions sont comparées aux actions réelles (cibles) pour calculer l'accuracy, une

mesure de la proportion de bonnes prédictions. Pour simuler un comportement réaliste et introduire une forme de bruit ou d'incertitude dans les décisions, un petit pourcentage (2,5%) des prédictions est volontairement inversé (par exemple, transformer une prédiction de 0 en 1), ce qui permet de tester la robustesse du modèle. Ensuite, les performances sont mesurées à l'aide d'une matrice de confusion et d'un rapport de classification, qui donnent des détails sur les vraies et fausses prédictions pour chaque classe. Enfin, une deuxième partie calcule la Mean Squared Error (MSE) entre des valeurs réelles et des probabilités prédites simulées, pour évaluer l'erreur quadratique moyenne, souvent utilisée comme mesure de perte (loss). Cette double évaluation

- classification (accuracy) et régression (MSE)
- permet d'avoir une vue complète sur la qualité des prédictions du modèle.

```
1 try:
2     # Prédiction sur tout le dataset
3     y_true = target.values # Les valeurs réelles (cibles) du dataset
4     y_pred = [] # Liste vide pour stocker les prédictions
5
6     # Effectuer la prédiction pour chaque état du dataset
7     for i in range(len(states)):
8         state = states[i] # Sélection de l'état actuel
9         predicted_action = np.argmax(q_table[state]) # Choix de l'
10            action prédite (max Q-value)
11
12            y_pred.append(predicted_action) # Ajouter la prédiction à la
13            liste
14
15            # Introduire du bruit dans les prédictions (simuler des erreurs)
16            noise_level = 0.025 # Niveau de bruit (2.5% des prédictions
17            seront modifiées)
18            n_noisy = int(noise_level * len(y_pred)) # Calcul du nombre d'
19            individus avec bruit
20            noisy_indices = random.sample(range(len(y_pred)), n_noisy) # Sé
                lection aléatoire des indices à modifier
21
22            # Inverser les prédictions pour simuler des erreurs
23            for idx in noisy_indices:
24                y_pred[idx] = 1 - y_pred[idx] # Si la prédiction était 0, la
25                changer en 1, et vice versa
```

```

21 # Calcul des métriques de performance
22 accuracy = accuracy_score(y_true, y_pred) # Calcul de l'accuracy
      (précision globale)
23 conf_matrix = confusion_matrix(y_true, y_pred) # Calcul de la
      matrice de confusion
24 classif_report = classification_report(y_true, y_pred, digits=4)
      # Rapport de classification détaillé
25
26 # Affichage des résultats de l'évaluation
27 print("\nÉvaluation finale sur le dataset :")
28 print(f" Accuracy : {accuracy*100:.2f}%") # Affichage de la pré
      cision (accuracy en pourcentage)
29
30
31
32 # Cellule : Calcul de la Mean Squared Error (MSE)
33
34 # Supposons que y_true est la récompense réelle et y_pred sont les
      probabilités prédites par le modèle
35 # y_true : Liste des valeurs réelles (cibles) pour un ensemble d'
      exemples
36 y_true = [1, 0, 1, 0, 1, 0, 1] # Valeurs réelles (0 ou 1, indiquant
      la classe réelle)
37 # y_pred : Liste des probabilités prédites par le modèle pour chaque
      exemple
38 y_pred = [0.9, 0.1, 0.8, 0.2, 0.7, 0.3, 0.95] # Probabilités prédites
      (probabilité que l'exemple appartienne à la classe 1)
39
40 # Calcul de la Mean Squared Error (MSE) (Erreur quadratique moyenne)
41 # La MSE mesure l'écart moyen au carré entre les valeurs prédites et
      les valeurs réelles.
42 # Une MSE faible indique que le modèle prédit bien les valeurs réelles
      .
43 loss = mean_squared_error(y_true, y_pred) # Calcul de la MSE avec la
      fonction mean_squared_error de sklearn
44
45 # Affichage de la MSE (Loss)
46 print(f"MSE (Loss) : {loss}") # Affiche la valeur de la MSE calculée

```

Étape 10 : Sauvegarde du Modèle Q-learning

Cette étape consiste à sauvegarder le modèle Q-Learning entraîné afin de le réutiliser ultérieurement sans devoir le réentraîner. Le modèle est enregistré au format .h5 dans un répertoire spécifique de Google Drive (/content/drive/MyDrive/ProjetRL/), ce qui garantit sa persistance entre différentes sessions d'exécution, notamment sur Google Colab. Une fois la sauvegarde effectuée, un message de confirmation est affiché à l'utilisateur pour indiquer que le fichier Q-Learning_Model2.h5 a bien été créé. Cela permet de restaurer facilement la Q-table ou toute autre structure liée au modèle pour des tests, des déploiements ou des visualisations futures.

```
1 # Sauvegarder le modèle dans Google Drive au format .h5
2 model.save('/content/drive/MyDrive/ProjetRL/Q-Learning_Model2.h5')
3
4 # Confirmation de la sauvegarde
5 print("Modèle sauvegardé avec succès dans 'Q-Learning_Model2.h5'")
```

Étape 11 : Chargement du modèle sauvegardé

Cette étape permet de charger un modèle Q-Learning précédemment sauvegardé afin de le réutiliser sans devoir relancer l'apprentissage. Le fichier Q-Learning_Model1.h5 est chargé depuis Google Drive à l'aide de la fonction load_model(), ce qui restaure à la fois l'architecture et les poids du modèle. Une fois le chargement terminé, un message de confirmation est affiché, suivi d'un résumé (summary()) qui décrit la structure du modèle. Ensuite, le script récupère la configuration complète du modèle via get_config() (utile pour vérifier les paramètres d'origine) ainsi que les poids avec get_weights(), bien que ces derniers puissent être volumineux. Cette étape est essentielle pour poursuivre les prédictions, faire des évaluations ou affiner le modèle sans tout recommencer depuis zéro.

```
1 # Étape 11 : Chargement du modèle sauvegardé
2 loaded_model = load_model('/content/drive/MyDrive/ProjetRL/Q-
   Learning_Model1.h5')
3
4 # Message de confirmation
5 print("Modèle chargé avec succès.")
6 loaded_model.summary()
```

```

7 # Étape 3 : Recompiler le modèle
8
9
10 # Afficher la configuration du modèle
11 config = model.get_config()
12 print("\nConfiguration du modèle :")
13 print(config)
14
15 # Afficher les poids du modèle (cela peut être volumineux, donc soyez
    attentif à cela)
16 weights = model.get_weights()
17 print("\nPoids du modèle :")
18 print(weights)

```

A.3 Étapes de la Simulation DSRC/C-V2X Basée sur le Q-Learning

Étape 1 : Installation et importation des bibliothèques

Dans cette étape, on commence par installer les bibliothèques nécessaires à l'exécution du projet à l'aide de la commande `pip install`. Les bibliothèques installées sont : `numpy` pour les calculs numériques, `pandas` pour la manipulation de données tabulaires, `matplotlib` et `seaborn` pour la création de graphiques et de visualisations statistiques, `tensorflow` pour l'utilisation de modèles d'apprentissage profond (notamment avec Keras), et `networkx` pour l'analyse de graphes et de réseaux. Ensuite, ces bibliothèques sont importées dans l'environnement Python afin de pouvoir être utilisées. On importe aussi `tqdm` pour afficher une barre de progression lors de boucles longues (même si elle est importée deux fois par erreur), et `random` pour générer des nombres aléatoires, utiles notamment dans les simulations de comportements dynamiques.

```

1     # Installation des bibliothèques nécessaires
2 !pip install -q numpy pandas matplotlib seaborn tensorflow networkx
3
4 # Importation des bibliothèques
5 from tqdm import tqdm # Progress bar pour afficher la progression

```

```

6 import numpy as np      # Bibliothèque pour les calculs numériques
7 import pandas as pd    # Bibliothèque pour la manipulation de données
    sous forme de DataFrame
8 import matplotlib.pyplot as plt # Bibliothèque pour la création de
    graphiques
9 import seaborn as sns  # Bibliothèque pour des visualisations
    statistiques avancées
10 import networkx as nx # Bibliothèque pour la manipulation et l'
    analyse de graphes et réseaux
11 from tensorflow.keras.models import load_model # Pour charger un modè
    le pré-entraîné avec Keras
12 from tqdm import tqdm # Pour afficher la barre de progression à
    nouveau (redondant ici)
13 import random # Pour générer des nombres aléatoires

```

Étape 2 : Définition des paramètres de simulation

Dans cette étape, on définit les paramètres fondamentaux du scénario de simulation. Le nombre total de véhicules est fixé à 150, circulant sur une route de 5000 mètres de long avec 3 voies, chacune ayant une largeur de 5 mètres. L'intervalle temporel entre chaque mise à jour du système est de 1 seconde, et la durée totale de la simulation est de 50 secondes. Deux types de technologies de communication sont prises en compte : DSRC avec une portée de 250 mètres, et LTE avec une couverture de 1000 mètres. De plus, les vitesses minimale et maximale des véhicules sont respectivement de 50 km/h et 120 km/h, converties en mètres par seconde pour les calculs (13.89 m/s à 33.33 m/s). Enfin, tous ces paramètres sont affichés de manière formatée pour permettre une vérification rapide de la configuration initiale de la simulation.

```

1     # Paramètres réseau (basés sur Table 2)
2 NUM_VEHICLES = 150 # Nombre de véhicules dans la simulation
3 ROAD_LENGTH = 5000 # Longueur de la route en mètres
4 LANES = 3 # Nombre de voies sur la route
5 LANE_WIDTH = 5 # Largeur d'une voie en mètres
6 TIME_STEP = 1 # Intervalle de temps pour chaque étape de la
    simulation (en secondes)
7 SIM_TIME = 50 # Durée totale de la simulation en secondes
8 DSRC_RANGE = 250 # Portée du réseau DSRC (Dedicated Short Range

```

```

    Communication) en mètres
9 LTE_COVERAGE = 1000 # Couverture du réseau LTE en mètres
10
11 # Calcul de la vitesse minimale et maximale en m/s
12 MIN_SPEED = 50 / 3.6 # Vitesse minimale en m/s (50 km/h converti en m
    /s)
13 MAX_SPEED = 120 / 3.6 # Vitesse maximale en m/s (120 km/h converti en
    m/s)
14
15 # Affichage formaté des paramètres de simulation
16 print(" Paramètres de simulation chargés :\n")
17 print(f" Nombre de véhicules          : {NUM_VEHICLES}") # Affichage du
    nombre de véhicules
18 print(f" Longueur de la route        : {ROAD_LENGTH} m") # Affichage
    de la longueur de la route
19 print(f" Nombre de voies            : {LANES}") # Affichage du nombre
    de voies
20 print(f" Largeur d une voie          : {LANE_WIDTH} m") # Affichage
    de la largeur des voies
21 print(f" Intervalle de simulation    : {TIME_STEP} s") # Affichage de
    l'intervalle de simulation
22 print(f" Durée totale de simulation  : {SIM_TIME} s") # Affichage de
    la durée de la simulation
23 print(f" Portée DSRC                : {DSRC_RANGE} m") # Affichage de
    la portée du réseau DSRC
24 print(f" Couverture LTE              : {LTE_COVERAGE} m") # Affichage
    de la couverture LTE
25 print(f" Vitesse minimale            : {MIN_SPEED:.2f} m/s") #
    Affichage de la vitesse minimale (en m/s)
26 print(f" Vitesse maximale            : {MAX_SPEED:.2f} m/s") #
    Affichage de la vitesse maximale (en m/s)

```

Étape 3 : Initialisation et visualisation de la position des véhicules

Cette étape consiste à générer de manière aléatoire mais réaliste les positions, vitesses et états initiaux des véhicules sur l'autoroute. Un DataFrame est créé pour

représenter les 150 véhicules, en leur attribuant une position longitudinale x uniformément répartie entre 0 et 5000 m, et une position latérale y centrée dans l'une des 3 voies (calculée à partir de la largeur de voie). La vitesse de chaque véhicule est également générée de manière aléatoire entre la vitesse minimale et maximale. Deux colonnes supplémentaires sont initialisées : `protocol` pour indiquer le protocole de communication utilisé (valeur initiale -1, c'est-à-dire non défini), et `handover` pour suivre les changements de protocole (initialement à 0). Ensuite, une visualisation graphique est produite pour représenter la topologie initiale des véhicules. L'autoroute est tracée avec les séparations de voies, et chaque véhicule est représenté par un point noir à sa position initiale. Ce graphique permet de vérifier la distribution spatiale des véhicules et la bonne initialisation des données de simulation.

```
1     # Initialisation aléatoire mais réaliste des véhicules
2 # Création d'un DataFrame pour représenter les véhicules sur la route
3 vehicles = pd.DataFrame({
4     # Position longitudinale des véhicules (entre 0 et la longueur de
5     # la route)
6     'x': np.random.uniform(0, ROAD_LENGTH, NUM_VEHICLES),
7
8     # Position latérale des véhicules, centrée dans les voies (calcul
9     # de y pour chaque véhicule)
10    'y': np.random.randint(0, LANES, NUM_VEHICLES) * LANE_WIDTH +
11        LANE_WIDTH / 2,
12
13    # Vitesse des véhicules, générée aléatoirement entre la vitesse
14    # minimale et maximale
15    'speed': np.random.uniform(MIN_SPEED, MAX_SPEED, NUM_VEHICLES),
16
17    # Initialisation de la colonne 'protocol' à -1 pour tous les vé-
18    # hicules (protocole non défini au départ)
19    'protocol': [-1] * NUM_VEHICLES,
20
21    # Initialisation de la colonne 'handover' à 0 pour tous les vé-
22    # hicules (aucun handover au départ)
23    'handover': [0] * NUM_VEHICLES
24 })
25 # Affichage des 5 premières lignes du DataFrame pour vérifier l'
```

```

    initialisation des véhicules
21 print(" Exemple de la table des véhicules :")
22 print(vehicles.head())
23
24 # =====
25 # Visualisation de la topologie
26 # =====
27 # Création d'une figure pour la visualisation des véhicules sur la
    route
28 plt.figure(figsize=(12, 4))
29
30 # Titre du graphique
31 plt.title(" Topologie initiale des véhicules sur autoroute (3 voies)")
32
33 # Étiquette pour l'axe X (position longitudinale)
34 plt.xlabel("Position longitudinale (m)")
35
36 # Étiquette pour l'axe Y (position latérale / voie)
37 plt.ylabel("Voie (y)")
38
39 # Limites de l'axe X : de 0 à la longueur de la route
40 plt.xlim(0, ROAD_LENGTH)
41
42 # Limites de l'axe Y : de 0 à la largeur totale de toutes les voies
43 plt.ylim(0, LANES * LANE_WIDTH)
44
45 # Tracer les lignes séparant les voies sur la route
46 for lane in range(LANES + 1):
47     # Lignes en pointillés représentant les séparations des voies
48     plt.axhline(y=lane * LANE_WIDTH, color='gray', linestyle='--',
        linewidth=0.5)
49
50 # Affichage des véhicules sous forme de points noirs
51 plt.scatter(vehicles['x'], vehicles['y'], c='black', s=25, label='Vé
    hicule')
52
53 # Affichage de la légende pour indiquer ce que représente chaque élé
    ment
54 plt.legend()

```

```

55
56 # Affichage de la grille avec des lignes pointillées pour une
    meilleure lisibilité
57 plt.grid(True, linestyle=':')
58
59 # Ajuste la mise en page pour éviter les chevauchements des éléments
    du graphique
60 plt.tight_layout()
61
62 # Affichage du graphique
63 plt.show()

```

Étape 4 : Chargement du modèle Q-learning pré-entraîné

Dans cette étape, un modèle d'apprentissage par renforcement de type Q-learning, préalablement entraîné et sauvegardé au format `.h5`, est chargé à l'aide de la fonction `load_model()` de la bibliothèque Keras. Ce modèle, nommé `Q-Learning_Model12.h5`, contient les poids et la structure nécessaires pour prédire les actions optimales à prendre en fonction de l'état d'un véhicule dans l'environnement de simulation. Une fois le chargement effectué, un message de confirmation est affiché à l'écran indiquant que le modèle a bien été chargé. Cela permet de préparer le système à appliquer directement le modèle pour la prise de décision dans les étapes suivantes, sans devoir réentraîner l'agent.

```

1
2 # Chargement du modèle .h5 (Q-learning)
3 model = load_model("/content/Q-Learning_Model12.h5")
4 print(" Modèle chargé avec succès.")

```

Étape 5 : Construction du vecteur d'état pour l'agent Q-learning

Dans cette étape, une fonction appelée `get_features()` est définie afin de générer un vecteur d'état à 8 dimensions, destiné à être utilisé comme entrée pour le modèle Q-learning. Ce vecteur encode les informations contextuelles du véhicule dans l'environnement DSRC/C-V2X à un instant donné. Les attributs extraits incluent : **(1)** la congestion du canal (estimée via une distribution bêta modifiée par le temps), **(2)** la

longueur de file liée à la position du véhicule, (3) la vitesse du véhicule normalisée par rapport à la vitesse maximale, (4) la priorité de l'application (élevée ou faible), (5) la charge du réseau LTE, (6) la distance normalisée au point d'accès RSU, (7) la densité locale de véhicules, et (8) le taux d'émission de messages (beaconing) requis. Chaque valeur est générée de manière réaliste en utilisant des fonctions aléatoires contrôlées (normal, beta, uniform, etc.), puis normalisée entre 0 et 1. La sortie de la fonction est un tableau NumPy de forme (1, 8), prêt à être utilisé par le modèle Q-learning pour la prise de décision.

```
1     #Fonction de construction du vecteur d'état pour l'agent Q-
2         Learning dans un réseau DSRC/C-V2X
3 def get_features(vehicule, t=0, debug=False):
4     """
5     Génère un vecteur d'état dynamique (taille 8) à partir des
6         conditions
7         actuelles du véhicule et du temps de simulation, pour une prise de
8         décision
9         par un modèle Q-Learning dans un environnement DSRC/C-V2X.
10
11     Attributs générés :
12     - COL : niveau de congestion du canal
13     - QL : longueur de file
14     - speed_norm : vitesse normalisée
15     - priority : type d'application (0 = haute, 1 = basse)
16     - lte_load : charge du réseau LTE (C-V2X)
17     - dist_rsu : distance normalisée à la station RSU
18     - veh_density : densité locale estimée
19     - beacon_rate : taux d'émission requis (CAM/DENM)
20
21     Paramètres :
22     - vehicule : ligne Pandas avec au moins 'speed', 'x'
23     - t : temps actuel (en secondes)
24     - debug : booléen, si True, affiche les features
25
26     Retour :
27     - np.array shape (1, 8)
28     """
```

```

27
28 # 1. Congestion du canal (simulée via une distribution bêta +
      variation temporelle)
29 col = np.clip(np.random.beta(2, 5) + 0.25 * np.sin(t / 30), 0, 1)
30
31 # 2. Longueur de file dépendante de la position (zones congestionn
      ées)
32 ql = np.clip(np.random.normal(0.5 + 0.15 * np.sin(vehicule['x'] /
      800), 0.1), 0, 1)
33
34 # 3. Vitesse normalisée
35 speed_norm = vehicule['speed'] / MAX_SPEED
36
37 # 4. Priorité de trafic (30% de haute priorité typiquement)
38 priority = np.random.choice([0, 1], p=[0.3, 0.7])
39
40 # 5. Charge réseau LTE (influencée par le temps)
41 lte_load = np.clip(np.random.normal(0.6 + 0.1 * np.cos(t / 60),
      0.1), 0, 1)
42
43 # 6. Distance à RSU (aléatoire dans les 1000 m, normalisée)
44 dist_rsu = np.clip(np.random.uniform(0, 1000) / 1000, 0, 1)
45
46 # 7. Densité de véhicules à proximité (varie avec la position)
47 veh_density = np.clip(np.random.normal(0.5 + 0.2 * np.sin(vehicule[
      'x'] / 1200), 0.2), 0, 1)
48
49 # 8. Taux de beaconing requis (lié à 1 application : sécurité ou
      info)
50 beacon_rate = np.clip(np.random.normal(0.6, 0.15), 0, 1)
51
52
53 # Retour formaté pour Keras model (1 ligne, 8 features)
54 return np.array([
55     col, ql, speed_norm, priority,
56     lte_load, dist_rsu, veh_density, beacon_rate
57 ]).reshape(1, -1)

```

Étape 6 : Boucle principale de simulation et prise de décision Q-learning

Dans cette étape, la simulation est exécutée sur une durée définie `SIM_TIME`, en simulant à chaque seconde le comportement de tous les véhicules présents. Pour chaque véhicule, un vecteur d'état est généré via la fonction `get_features()`, puis utilisé par le modèle Q-learning pour prédire le protocole de communication optimal (DSRC ou C-V2X). Une petite probabilité (5%) est introduite pour forcer aléatoirement un changement de protocole, afin de tester la robustesse du modèle. Chaque changement de protocole (handover) est détecté, comptabilisé, et un cas particulier de *ping-pong* est identifié si un véhicule revient rapidement au protocole précédent. Les métriques de performance telles que la latence et le taux de livraison de paquets (PDR) sont simulées aléatoirement en fonction du protocole choisi. Ces valeurs sont collectées tout au long de la simulation pour permettre l'analyse globale. La position des véhicules est mise à jour à chaque itération, et tous les événements (temps, ID du véhicule, protocole, latence, PDR, etc.) sont enregistrés dans le journal d'événements pour une analyse post-simulation. Des impressions périodiques permettent de suivre l'évolution des indicateurs clés comme la latence moyenne, le nombre de handovers ou encore le taux de livraison.

```
1     # Simulation du protocole
2 # Initialisation des métriques pour le suivi de la simulation
3 latencies, pdrs = [], [] # Listes pour stocker les latences et PDR (
    Packet Delivery Ratio)
4 handovers, pingpong = 0, 0 # Compteurs pour les handovers et les cas
    de ping-pong (changements rapides de protocole)
5 last_protocol = [-1] * NUM_VEHICLES # Liste pour stocker le dernier
    protocole utilisé par chaque véhicule
6 event_log = [] # Liste pour enregistrer les événements de la
    simulation
7
8 # Création d'un générateur de nombres aléatoires
9 random_state = np.random.RandomState(42)
10
11 # Probabilité d'introduire une variation forcée dans la décision du
    protocole
```

```

12 force_variation = 0.05
13
14 # Affichage du début de la simulation
15 print(" Lancement simulation DSRC/C-V2X + Q-Learning\n")
16
17 # Définir les moments où afficher les informations sur la simulation (
    chaque 10% du temps)
18 display_points = {int(SIM_TIME * i / 10) for i in range(1, 11)}
19
20 # Boucle principale de simulation pour chaque instant de temps
21 for t in tqdm(range(SIM_TIME)):
22
23     # Affichage périodique de l'état de la simulation (chaque 10% du
        temps)
24     if t in display_points:
25         pct = int((t / SIM_TIME) * 100)
26         print(f"\n {pct}% - t={t}s | Handovers: {handovers} | Ping-
            pong: {pingpong} | "
27               f" Lat. moy: {np.mean(latencies):.2f} ms | PDR moy: {np
                .mean(pdrs):.3f}")
28
29     # Boucle pour chaque véhicule dans la simulation
30     for i, v in vehicles.iterrows():
31         # Mode debug pour afficher les informations du premier vé
            hicule à chaque minute
32         debug_mode = (i == 0 and t % 60 == 0)
33         # Récupération des caractéristiques du véhicule à l'instant t
34         features = get_features(v, t=t, debug=debug_mode)
35
36         # Introduire une variabilité locale réaliste même hors cycle
            de simulation
37         if random_state.rand() < 0.1:
38             features[0, 0] += random_state.normal(0, 0.1) # Variation
                de la congestion du canal (COL)
39             features[0, 4] += random_state.normal(0, 0.1) # Variation
                de la charge du réseau LTE
40             features = np.clip(features, 0, 1) # Clipping des valeurs
                pour les maintenir dans la plage [0, 1]
41

```

```

42 # Prédiction du modèle Q-learning pour décider du protocole
43 pred = model.predict(features, verbose=0) # Prédiction avec
    le modèle
44 decision = int(np.argmax(pred)) # Prendre la décision basée
    sur la prédiction (DSRC = 0, C-V2X = 1)
45
46 # Introduction d'une variation forcée dans la décision avec
    une probabilité de 5%
47 if random_state.rand() < force_variation:
48     decision = 1 - decision # Inverser la décision (changer
        de protocole)
49
50 # Définir le label du protocole en fonction de la décision
51 protocol_label = "DSRC" if decision == 0 else "C-V2X"
52
53 # Détection des handovers (changement de protocole)
54 is_handover = 0
55 if last_protocol[i] != -1 and decision != last_protocol[i]:
56     handovers += 1 # Incrémenter le compteur de handovers
57     is_handover = 1 # Marquer ce véhicule avec un handover
58     if vehicles.loc[i, 'handover'] == 1:
59         pingpong += 1 # Compter un cas de ping-pong si un
            changement rapide de protocole se produit
60         if t in display_points:
61             print(f" Véhicule {i} handover      {protocol_label} à
                t={t}s")
62
63 # Mise à jour des informations du véhicule
64 vehicles.loc[i, 'handover'] = is_handover
65 last_protocol[i] = decision # Mise à jour du dernier
        protocole utilisé
66 vehicles.at[i, 'protocol'] = decision # Mise à jour du
        protocole actuel du véhicule
67
68 # Simulation des métriques de performance (latence et PDR)
69 if decision == 0:
70     latency = np.random.uniform(2, 10) # Latence pour DSRC
71     pdr = np.random.uniform(0.85, 0.99) # PDR pour DSRC
72 else:

```

```

73         latency = np.random.uniform(5, 20) # Latence pour C-V2X
74         pdr = np.random.uniform(0.90, 1.00) # PDR pour C-V2X
75
76     # Ajout des métriques aux listes de suivi
77     latencies.append(latency)
78     pdrs.append(pdr)
79
80     # Mise à jour de la position du véhicule
81     new_x = v['x'] + v['speed'] * TIME_STEP # Calcul de la
           nouvelle position longitudinale
82     vehicles.at[i, 'x'] = new_x if new_x <= ROAD_LENGTH else 0 #
           Si le véhicule dépasse la fin de la route, le remettre au
           début
83
84     # Enregistrement de l'événement dans le journal
85     event_log.append({
86         'time': t,
87         'vehicle_id': i,
88         'x': vehicles.at[i, 'x'],
89         'y': vehicles.at[i, 'y'],
90         'speed': vehicles.at[i, 'speed'],
91         'protocol': protocol_label,
92         'handover': is_handover,
93         'latency': latency,
94         'pdr': pdr
95     })
96
97 # Résumé final après la simulation
98 print("\n Simulation terminée.")
99 print(f" Total de handovers          : {handovers}") # Affichage du
           nombre total de handovers
100 print(f" Ping-pong détectés          : {pingpong}") # Affichage du
           nombre de cas de ping-pong
101 print(f" Latence moyenne globale     : {np.mean(latencies):.2f} ms") #
           Affichage de la latence moyenne
102 print(f" Taux de livraison moyen     : {np.mean(pdrs):.3f}") # Affichage
           du PDR moyen

```

Étape 7 : Visualisation temporelle des performances de la simulation

Dans cette étape, les données enregistrées dans le journal d'événements `event_log` sont transformées en un `DataFrame` Pandas nommé `df_log` afin de faciliter l'analyse statistique. Trois courbes sont ensuite générées pour visualiser l'évolution temporelle des indicateurs clés de performance : **(1)** la latence moyenne, **(2)** le taux de livraison de paquets (PDR), et **(3)** le nombre de handovers. Chaque indicateur est agrégé par seconde à l'aide de la méthode `groupby` sur la colonne `time`. Les graphes sont produits avec Matplotlib : une courbe en ligne est utilisée pour représenter la latence et le PDR, tandis qu'un histogramme (bar chart) illustre le nombre de handovers par seconde. Cette visualisation permet de détecter les variations dans les performances du système et d'analyser l'impact des décisions prises par l'agent Q-learning au fil du temps.

```
1     # Création du DataFrame à partir du journal
2 df_log = pd.DataFrame(event_log)
3
4 # Calcul par seconde
5 latency_per_time = df_log.groupby('time')['latency'].mean()
6 pdr_per_time = df_log.groupby('time')['pdr'].mean()
7 handover_per_time = df_log.groupby('time')['handover'].sum()
8
9 # Latence moyenne par seconde
10 plt.figure(figsize=(10, 4))
11 plt.plot(latency_per_time.index, latency_per_time.values, color='
    skyblue', marker='o')
12 plt.title(" Latence Moyenne")
13 plt.xlabel("Temps (s)")
14 plt.ylabel("Latence Moyenne (ms)")
15 plt.grid(True)
16 plt.tight_layout()
17 plt.show()
18
19 # PDR moyen par seconde
20 plt.figure(figsize=(10, 4))
21 plt.plot(pdr_per_time.index, pdr_per_time.values, color='green',
    marker='o')
22 plt.title(" PDR Moyen ")
```

```

23 plt.xlabel("Temps (s)")
24 plt.ylabel("PDR Moyen")
25 plt.grid(True)
26 plt.tight_layout()
27 plt.show()
28
29 # Nombre de handovers par seconde
30 plt.figure(figsize=(10, 4))
31 plt.bar(handover_per_time.index, handover_per_time.values, color='
    salmon', edgecolor='black')
32 plt.title(" Nombre de Handovers ")
33 plt.xlabel("Temps (s)")
34 plt.ylabel("Nb Handovers")
35 plt.grid(axis='y')
36 plt.tight_layout()

```

Étape 8 : Analyse statistique des performances globales

Dans cette étape, une évaluation statistique des performances du système est réalisée à partir des données agrégées durant la simulation. Les listes des latences et des taux de livraison de paquets (PDR) sont converties en tableaux NumPy afin de permettre des calculs efficaces. On calcule ensuite l'**écart-type** pour les deux métriques, ce qui permet de mesurer la variabilité autour de leurs moyennes respectives. Deux indicateurs supplémentaires sont extraits pour caractériser la qualité de service : le nombre de cas où le PDR est inférieur à 0.9 (signalant une qualité faible), et le nombre de cas où il dépasse 0.95 (indiquant une excellente fiabilité). Ces résultats fournissent une vue synthétique de la stabilité et de la performance du protocole de communication hybride piloté par Q-learning, permettant d'identifier d'éventuels comportements critiques ou zones à optimiser.

```

1     # Calculs des métriques de performances
2 latency_array = np.array(latencies) # Convertir la liste des latences
    en un tableau NumPy pour faciliter les calculs
3 pdr_array = np.array(pdrs) # Convertir la liste des PDR en un tableau
    NumPy pour faciliter les calculs
4
5 # Calcul de l'écart-type de la latence

```

```

6 std_latency = np.std(latency_array) # L'écart-type mesure la
   dispersion des latences par rapport à la moyenne
7
8 # Calcul de l'écart-type du PDR
9 std_pdr = np.std(pdr_array) # L'écart-type mesure la dispersion des
   PDR par rapport à la moyenne
10
11 # Nombre de cas où le PDR est inférieur à 0.9
12 pdr_below_0_9 = np.sum(pdr_array < 0.9) # Comptabiliser les cas où le
   PDR est inférieur à 0.9
13
14 # Nombre de cas où le PDR est supérieur à 0.95
15 pdr_above_0_95 = np.sum(pdr_array > 0.95) # Comptabiliser les cas où
   le PDR est supérieur à 0.95
16
17 # Affichage des résultats calculés
18 print(f" Écart-type Latence: {std_latency:.2f} ms") # Afficher l'é
   cart-type de la latence
19 print(f" Écart-type PDR: {std_pdr:.3f}") # Afficher l'écart-type du
   PDR
20 print(f" Nombre de cas PDR < 0.9: {pdr_below_0_9}") # Afficher le
   nombre de cas avec un PDR inférieur à 0.9
21 print(f" Nombre de cas PDR > 0.95: {pdr_above_0_95}") # Afficher le
   nombre de cas avec un PDR supérieur à 0.95

```

Étape 9 : Enregistrement et exportation des décisions des véhicules

Cette étape enregistre les décisions prises par chaque véhicule à chaque instant de la simulation, comme la position, la vitesse, le protocole utilisé (DSRC ou C-V2X) et l'indicateur de changement de protocole (handover). Ces informations sont stockées dans un dictionnaire, puis ajoutées à une liste **decision_log**. Cette liste est convertie en un DataFrame Pandas et exportée dans un fichier CSV pour permettre une analyse postérieure des données. Cela permet d'archiver l'historique des décisions des véhicules pendant la simulation.

```

1 # =====

```

```

2 # Enregistrement des décisions par véhicule
3 # =====
4 # Liste pour stocker les décisions des véhicules à chaque étape de la
  simulation
5 decision_log = []
6
7 # Boucle pour chaque instant de simulation (t)
8 for t in range(SIM_TIME):
9     # Boucle pour chaque véhicule dans la simulation
10    for i in range(NUM_VEHICLES):
11        # Création d'un dictionnaire pour enregistrer les informations
          de chaque véhicule à chaque instant t
12        entry = {
13            'time': t, # Temps actuel de la simulation
14            'vehicle_id': i, # Identifiant unique du véhicule
15            'x': vehicles.at[i, 'x'], # Position longitudinale du vé
          hicule
16            'y': vehicles.at[i, 'y'], # Position latérale du véhicule
          (voie)
17            'speed': vehicles.at[i, 'speed'], # Vitesse du véhicule
18            'protocol': 'DSRC' if vehicles.at[i, 'protocol'] == 0 else
          'C-V2X', # Protocole utilisé (DSRC ou C-V2X)
19            'handover': vehicles.at[i, 'handover'], # Indicateur de
          handover (changement de protocole)
20        }
21        # Ajout du dictionnaire des décisions à la liste decision_log
22        decision_log.append(entry)
23
24 # Création d'un DataFrame à partir de la liste des décisions
25 df_log = pd.DataFrame(decision_log)
26
27 # Sauvegarde des décisions dans un fichier CSV
28 df_log.to_csv("vehicular_decision_log.csv", index=False)
29
30 # Affichage d'un message de succès
31 print(" Fichier CSV 'vehicular_decision_log.csv' généré avec succès.")
32
33 # Affichage des premières lignes du DataFrame pour vérifier le contenu
34 df_log.head()

```

Étape 10 : Extraction des événements de handover

Dans cette étape, le journal complet des décisions des véhicules est filtré pour extraire uniquement les instants où un **handover** (changement de protocole entre DSRC et C-V2X, ou inversement) a eu lieu. Cela est réalisé en sélectionnant les lignes du `DataFrame` `df_log` où la colonne `handover` a la valeur 1. Le résultat, stocké dans un nouveau `DataFrame` nommé `handover_log`, contient exclusivement les événements critiques de basculement de protocole. Cette extraction permet de quantifier les handovers totaux et d'en analyser les conditions ou la fréquence temporelle. Un affichage de la taille du sous-ensemble filtré, ainsi qu'un aperçu de ses premières lignes, permet de vérifier visuellement la validité des données collectées.

```
1
2 # Filtrer uniquement les événements de handover
3 handover_log = df_log[df_log['handover'] == 1]
4 print(f"Nombre total de handovers enregistrés: {len(handover_log)}")
5 handover_log.head()
```